A
Preparing Your Working Environment

In order to avoid any headaches while going through the case studies in this book, it's best to install the necessary software and configure your environment the right way from the start. Although we assume you already have some experience developing PHP applications, we'll quickly go through the steps to install your machine with the necessary software.

The good news is that all the required software is free, powerful, and (finally!) comes with installers that make the programs easy for anyone to set up and configure. The bad news is that there are many possible configurations, so the instructions written might not apply 100% to you (for example, if you are using Windows, you may prefer to use IIS instead of Apache, and so on).

We'll cover the installation instructions separately for Windows and *nix based machines. We'll also cover preparing the database that is used in many examples throughout the book; these instructions apply to both Windows and *nix users, so be sure not to miss this section at the end of the appendix.

To build websites with AJAX and PHP you will need (quite unsurprisingly) to install **PHP**. The preferred version is PHP 5, because we use some of its features in Chapter 11. You also need a web server. We will cover installing **Apache**, which is the web server preferred by most PHP developers and web hosting companies. Because we've tried to make the examples in this book as relevant as possible for real-world scenarios, many of them need a database. We cover installing **MySQL**, which is the most popular database server in the PHP world. Because we used simple SQL code, you can easily use another database server without major code changes, or older versions of MySQL.

At the end of this chapter, we'll cover installing **phpMyAdmin**, which is a very useful web tool for administering your databases. You'll then learn how to use this tool to create a new database, and then a database user with full privileges to this database.

In the following pages, you'll learn how to:

- Install Apache 2, PHP 5, and MySQL 5 on your Windows machine
- Install Apache 2, PHP 5, and MySQL 5 on your *nix machine
- Install phpMyAdmin
- Create a new database and then a database user using phpMyAdmin

TIP

Programmers who don't want to install the required software manually have the option of using a software package such as XAMPP, which bundles all of them (and many more) in a single installer file. XAMPP is packaged for Linux, Windows, Mac OS X, and Solaris, and is free of charge. You can get XAMPP from `http://www.apachefriends.org/en/xampp.html`.

If you decide to use XAMPP, you can skip directly to setting up the `ajax` database, as shown at the end of this appendix.

Preparing Your Windows Playground

Here we cover installing these software products in your Windows machine:

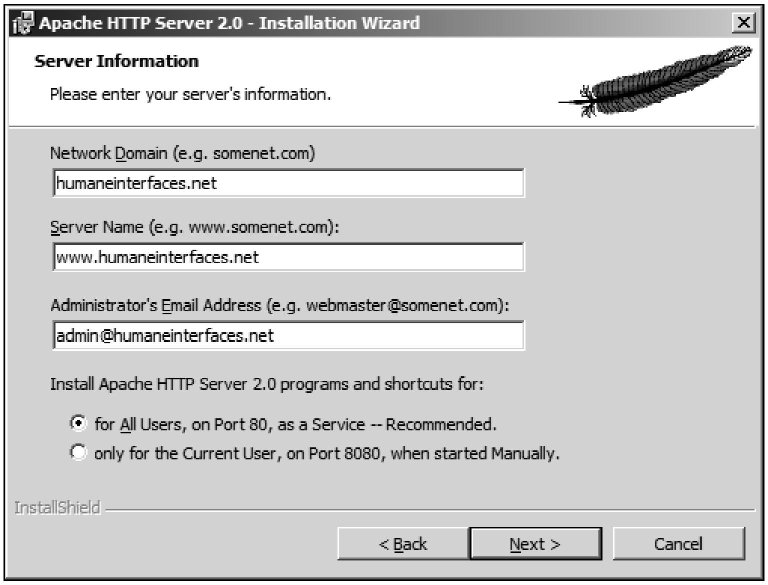
- Apache 2
- PHP 5
- MySQL 5

Installing Apache

You can download the latest MSI Installer version of the Apache HTTP Server from `http://httpd.apache.org/download.cgi`. Download the latest Win32 Binary (MSI Installer), which should have a name like `apache_2.x.y-win32-x86-no_ssl.msi`, then execute it.

The default installation location of Apache 2 is `C:\Program Files\Apache Group\Apache2\`, but the installer allows you to specify a different path. You can choose a more convenient location (such as `C:\Apache`), which can make your life working with Apache easier.

During installation you'll be asked to enter your server's information:



Apache HTTP Server 2.0 - Installation Wizard

Server Information

Please enter your server's information.

Network Domain (e.g. somenet.com)
humaneinterfaces.net

Server Name (e.g. www.somenet.com):
www.humaneinterfaces.net

Administrator's Email Address (e.g. webmaster@somenet.com):
admin@humaneinterfaces.net

Install Apache HTTP Server 2.0 programs and shortcuts for:

● for All Users, on Port 80, as a Service -- Recommended.
○ only for the Current User, on Port 8080, when started Manually.

Figure A.1: Installing Apache 2.0

If you're not sure about how to complete the form, just type **localhost** for the first two fields, and write your email address for the last. You can change this information later by editing the Apache configuration file. The default location of this file is `C:\Program Files\Apache Group\Apache2\conf\httpd.conf`.

You can also choose to install Apache on Port 80, or on Port 8080. The default port is 80, but if you already have a web server (such as IIS) on your machine, you'll need to install Apache on a different port. If you choose to run Apache on Port 8080, you will need to start the Apache service manually by going to the Apache bin folder (by default `C:\Program Files\Apache Group\Apache2\bin`), and typing

```
apache -k install
```

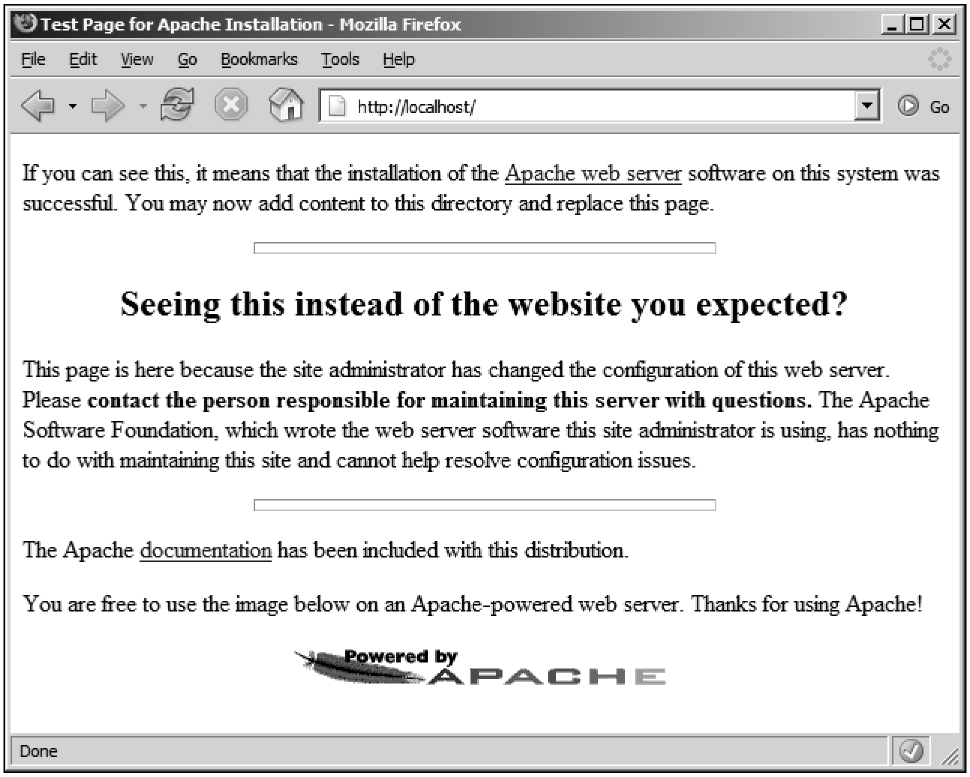
When the web server runs on a port different than 80, you need to specify the port manually when making HTTP requests, such as in `http://localhost:8080/ajax/suggest`.

In the next setup screens, you can safely use the default options.

Along with the Apache server the installer will also start the **Apache Service Monitor** program, which is available from the taskbar. The taskbar icon reflects the current state of the web server (stopped, running, etc.), and also allows you to start, stop, or restart the Apache service.

Keep in mind that you'll need to restart (or stop and then start) the Apache service after making any changes to the `httpd.conf` configuration file, in order for the changes to become effective.

After installing Apache 2, make sure it works OK. If you installed it on port 80, browse to `http://localhost/`. If you installed it on 8080, go to `http://localhost:8080/`. You should see an Apache welcome message similar to this:



If you can see this, it means that the installation of the Apache web server software on this system was successful. You may now add content to this directory and replace this page.

Seeing this instead of the website you expected?

This page is here because the site administrator has changed the configuration of this web server. Please **contact the person responsible for maintaining this server with questions.** The Apache Software Foundation, which wrote the web server software this site administrator is using, has nothing to do with maintaining this site and cannot help resolve configuration issues.

The Apache documentation has been included with this distribution.

You are free to use the image below on an Apache-powered web server. Thanks for using Apache!

Figure A.2: Apache Installed Successfully

Installing MySQL

The official website of MySQL is `http://www.mysql.com`. At the time of this writing the latest stable version is MySQL 5.0, and you can download it from `http://dev.mysql.com/downloads/mysql/5.0.html`. However, it's good to know that we made our SQL queries compliant with the SQL 92 standard, so you should be able to reuse them with other database systems with minimum of translation effort.

In the Downloads page, scroll down to the Windows downloads section, and download the Windows Essentials file. You'll be asked to choose a mirror site, and the file will be named something like `mysql-essential-5.0.xx-win32.msi`. After downloading the file, simply execute it to install your MySQL Server.

After installation you'll be given the chance to configure your server. Do so. It's safe to use the default options all the way through. At some point you'll need to set the root password, which will correspond to the `root@localhost` user. Choose a password that's complicated enough for others not to guess and simple enough for you to remember.

Before going through any case studies in this book, remember to see the *Preparing the AJAX Database* section at the end of this appendix.

Installing PHP

The official website of PHP is `http://www.php.net`. Start by downloading from the Windows Binaries section the latest PHP 5 zip package (*not the installer!*) from `http://www.php.net/downloads.php`. We prefer not to use the installer because it doesn't include extensions that you may need for your projects, and it doesn't do much configuration work for you anyway.

After you download the Windows binaries, follow these steps to install PHP:

1. Unzip the zip package (it should be a file with a name like `php-5.x.y-win32.zip`) into a folder named `C:\PHP\`. You can choose another name or location for this folder if you want.
2. Copy `php.ini-recommended` from `C:\PHP\` to your Windows folder (`C:\Windows`), renaming it as `php.ini`.
3. Open `php.ini` for editing with the text editor of your choice (such as Notepad) and uncomment the `php_gd2.dll`, `php_mysql.dll`, and `php_xsl.dll` extension lines (by removing the leading semicolons), and add a similar line for `php_mysqli`:

   ```
   extension=php_gd2.dll
   extension=php_mysql.dll
   extension=php_mysqli.dll
   extension=php_xsl.dll
   ```

4. We recommend enabling full error reporting for PHP on the development machine, but this is optional (this option is the default). Be warned that this change can alter the functionality of other scripts on your server. Find the `error_reporting` line in `php.ini` and change it to:

   ```
   error_reporting = E_ALL
   ```

5. Copy `php5ts.dll` and `libmysql.dll` located in `C:\PHP\`, to the Windows `System32` folder (by default `\Windows\System32`).
6. Copy `php_mysql.dll`, `php_mysqli.dll`, `php_xsl.dll`, and `php_gd2.dll` from `C:\PHP\ext`, to the Windows `System32` folder.
7. Open the Apache configuration file for editing. By default, the location of this file is `C:\Program Files\Apache Group\Apache2\conf\httpd.conf`.
8. In `httpd.conf`, find the portion with many `LoadModule` entries, and add the following lines:

   ```
   LoadModule php5_module c:/php/php5apache2.dll
   AddType application/x-httpd-php .php
   ```

9. Also in `httpd.conf`, find the `DirectoryIndex` entry, and add `index.php` at the end of the line, like this:

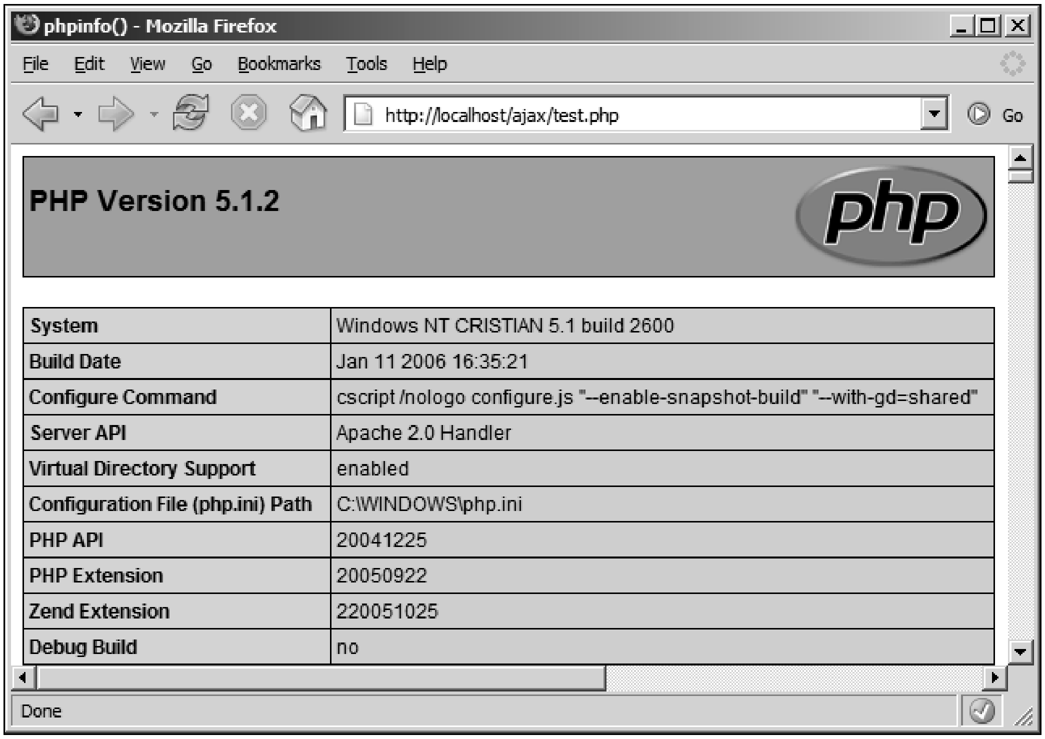
   ```
   DirectoryIndex index.html index.html.var index.php
   ```

10. Save the `httpd.conf` file, and then restart the Apache 2 service, using the Apache Service Monitor by clicking its icon in the Notification Area of the taskbar. (If you get any error at this point, make sure that you followed correctly all the previous steps of the exercise.) If Apache restarts without generating any errors, that is a good sign.

11. Create a folder called `ajax` under the `htdocs` folder (by default `C:\Program Files\Apache Group\Apache2\htdocs`).
12. To make sure that your Apache instance can also correctly parse PHP code, create a file named `test.php` in the `ajax` folder, and then add the following code to it:

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

13. Point your web browser to `http://localhost/ajax/test.php` (or `http://localhost:8080/ajax/test.php` if you installed Apache to work on port 8080) to test if everything went OK with the installation. You should get a page like this:



PHP Version 5.1.2

System	Windows NT CRISTIAN 5.1 build 2600
Build Date	Jan 11 2006 16:35:21
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS\php.ini
PHP API	20041225
PHP Extension	20050922
Zend Extension	220051025
Debug Build	no

Figure A.3: PHP Installation Working

Congratulations, you just finished installing Apache, MySQL, and PHP!

The configuration set up isn't yet finished. If you're running Windows (and you probably are, since you're reading this), please skip the Preparing Your *nix Playground section, and go through the *Installing phpMyAdmin* and *Preparing the AJAX Database* sections at the end of this appendix.

Preparing Your *nix Playground

Almost all the UNIX and Linux distributions include Apache, PHP, and MySQL; however, you should check the versions of these programs. It would be good to have MySQL 4.1 or newer, and it's very important to have at least PHP 5. The code in this book will not work with older versions of PHP.

Installing Apache

To install Apache on your Unix-based server, follow these simple steps:

1. First, download the latest Apache Unix Source code for your system from `http://httpd.apache.org/download.cgi` and decompress it with a command such as:

   ```
   tar -zxvf httpd-2.0.55.tar.gz
   ```

2. To compile and install the Apache Web Server on your system, go to the folder containing the sources and execute the following commands, while logged in as root:

   ```
   ./configure --prefix=/usr/local/apache2 --enable-so --enable-ssl --with-
   ssl --enable-auth-digest

   make

   make install
   ```

Installing MySQL

The official website of MySQL is `http://www.mysql.com`. At the time of this writing the latest stable version is MySQL 5.0, and you can download it from `http://dev.mysql.com/downloads/mysql/5.0.html`. However, it's good to know that we made our SQL queries compliant with the SQL 92 standard, so you should be able to reuse them with other database systems with minimum of translation effort. Chapter 2 of the MySQL 5 manual covers installation procedures for all supported platforms, and you can read it here: `http://dev.mysql.com/doc/refman/5.0/en/installing.html`.

If your Linux distribution supports RPMs, you'll need to download the RPMs for Server, Client programs, and Libraries and header files. Install MySQL as explained in the manual at `http://dev.mysql.com/doc/refman/5.0/en/linux-rpm.html`. If your platform doesn't support RPMs, install MySQL as explained at `http://dev.mysql.com/doc/refman/5.0/en/installing-binary.html`.

After installing MySQL, you should change the MySQL administrator's password (the `root@localhost` user), which is blank by default. Read more about MySQL passwords at `http://dev.mysql.com/doc/mysql/en/Passwords.html`. One way to change root's password is to execute:

```
mysqladmin -u root password 'your_new_password.'
```

Alternatively, you can access MySQL through a console program or by using a database administration tool such as phpMyAdmin, and execute this command:

```
SET PASSWORD FOR root@localhost=PASSWORD('your_new_password');
```

You can now test your MySQL server by executing the following command in your console:

```
#mysql -u root -p
```

Installing PHP

Every time you want to get a new PHP library working on Linux, you need to recompile the PHP module. That's why it's recommended to make a *good* compilation, with all the needed libraries, from the start.

1. Go to http://www.php.net/downloads.php and get the complete source code archive of PHP 5.x and extract the contents into a directory. At the time of writing, the latest PHP version was 5.1.2.
2. Go to the folder where you extracted the PHP source and execute the following commands:

   ```
   ./configure --with-config-file-path=/etc --with-mysql=/usr/include/mysql
   --with-apxs2=/usr/local/apache2/bin/apxs --with-zlib --with-gd --with-xsl

   make

   make install
   ```

If you are compiling PHP for XAMPP, you need to use the following `configure` command instead:

```
./configure --with-config-file-path=/opt/lampp/etc --with-mysql=/opt/lampp
--with-apxs2=/opt/lampp/bin/apxs --with-zlib --with-gd
```

After executing `make` and `make install`, you need to copy the newly created `php_src/libs/libphp5.so` file to `/opt/lampp/modules/libphp5.so`.

3. Copy `php.ini-recommended` to `/etc/php.ini` by executing the following command:

   ```
   cp php.ini-recommended /etc/php.ini.
   ```

4. Open the Apache configuration file (`httpd.conf`), find the `DirectoryIndex` entry, and make sure you have `index.php` at the end of the line:

   ```
   DirectoryIndex index.html index.html.var index.php
   ```

5. Restart your Apache Web Server using the following command:

   ```
   /usr/local/apache2/bin/apachectl restart
   ```

6. Create a folder called `ajax` under the `htdocs` folder (by default `/usr/local/apache2/htdocs/`).
7. To make sure your PHP installation works, create a file named `test.php` in the `ajax` folder you've just created, with the following contents in it:

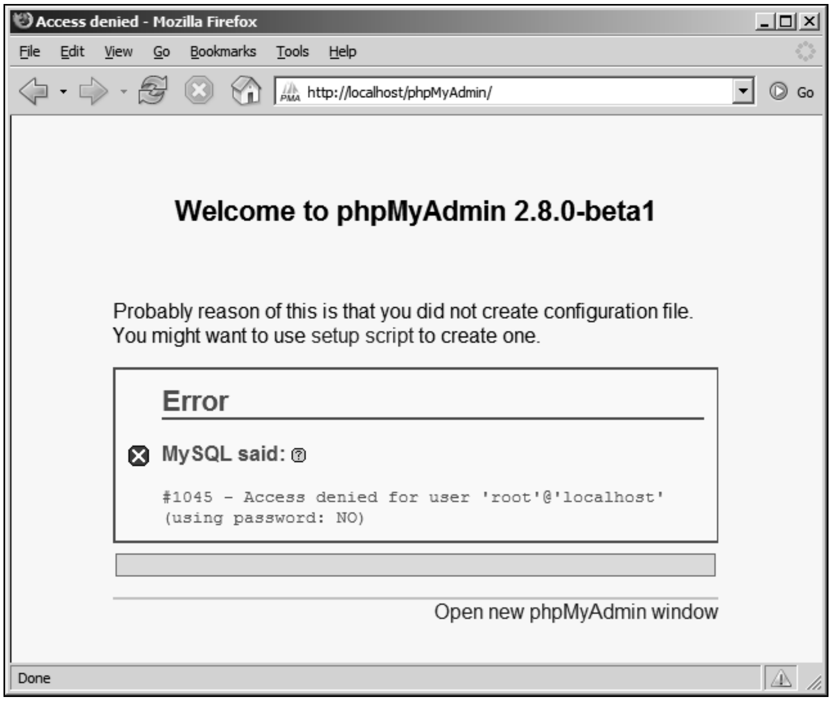
   ```
   <?php
   phpinfo();
   ?>
   ```

8. Finally, point your web browser to `http://localhost/test.php`, to ensure PHP was correctly installed under Apache (you should get a page similar to Figure A.3).

Installing phpMyAdmin

phpMyAdmin is a very popular MySQL administration tool written in PHP. It allows you to manage your MySQL databases using a simple-to-use web interface. The official web page is `http://www.phpmyadmin.net`. Follow these steps to install and configure this program:

1. Start by downloading the latest version of phpMyAdmin from `http://www.phpmyadmin.net/home_page/downloads.php`. If you aren't sure what file to download, the safest bet is to go with the zip archive.
2. Unzip the archive somewhere on your disk. The archive contains a folder named with the complete phpMyAdmin version (for example, at the time of this writing, the folder for the beta version of phpMyAdmin is called `phpMyAdmin-2.8.0-beta1`).
3. To make your life easier, rename this folder to simply `phpMyAdmin`.
4. Move the phpMyAdmin folder to the `htdocs` folder of Apache 2 (by default `C:\Program Files\Apache Group\Apache2\htdocs`).
5. To make sure your phpMyAdmin installation is accessible by Apache, load `http://localhost/phpMyAdmin` in your favorite web browser. If everything worked OK, you should get a message such as this:


Access denied - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/phpMyAdmin/
Go
Done

Welcome to phpMyAdmin 2.8.0-beta1

Probably reason of this is that you did not create configuration file. You might want to use setup script to create one.

Error

MySQL said:

```
#1045 - Access denied for user 'root'@'localhost'
(using password: NO)
```

Open new phpMyAdmin window

Figure A.4: phpMyAdmin Doesn't Have Access to MySQL

6. The error message is suggestive enough—you need to instruct phpMyAdmin how to access your MySQL server. Under the `phpMyAdmin` folder search for a file named `config.inc.php`. If you find this file, change its options as shown in the following code snippet. If you don't find this file, create it with the following contents:

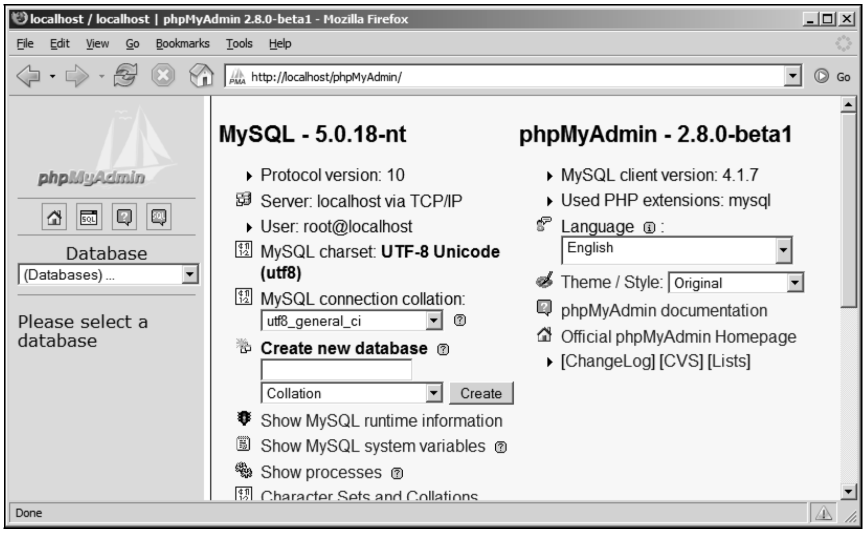
```
<?php
$cfg['PmaAbsoluteUri'] = "http://localhost/phpMyAdmin/";
$cfg['Servers'][1]['host'] = "localhost";
$cfg['Servers'][1]['auth_type'] = 'config';
$cfg['Servers'][1]['user'] = "root";
$cfg['Servers'][1]['password'] = "password";
?>
```

> For more details on installing and using phpMyAdmin, see its documentation at `http://www.phpmyadmin.net/home_page/docs.php`. Packt Publishing has a separate book for those of you who want to learn more about phpMyAdmin—*Mastering phpMyAdmin for Effective MySQL Management* (ISBN: 1-904811-03-5). In case you're not a native English speaker, it's good to know that the book is also available in Czech, German, French, and Italian.

Preparing the AJAX Database

As an exercise for both using phpMyAdmin and working with MySQL, let's create a database called `ajax`, and create a MySQL user with full privileges to this database. You'll use this database and this user for all the exercises in this book. Follow these steps:

1. Load `http://localhost/phpMyAdmin` in your web browser. If the configuration data you wrote in `config.inc.php` was correct, you should see something like this:



localhost / localhost | phpMyAdmin 2.8.0-beta1 - Mozilla Firefox

http://localhost/phpMyAdmin/

MySQL - 5.0.18-nt

- Protocol version: 10
- Server: localhost via TCP/IP
- User: root@localhost
- MySQL charset: **UTF-8 Unicode (utf8)**
- MySQL connection collation: utf8_general_ci

phpMyAdmin - 2.8.0-beta1

- MySQL client version: 4.1.7
- Used PHP extensions: mysql
- Language: English
- Theme / Style: Original

Please select a database

Done

File
Edit
View
Go
Bookmarks
Tools
Help
Go
phpMyAdmin
Database
(Databases) ...
Create new database
Collation
Create
Show MySQL runtime information
Show MySQL system variables
Show processes
Character Sets and Collations
phpMyAdmin documentation
Official phpMyAdmin Homepage
[ChangeLog] [CVS] [Lists]

Figure A.5: phpMyAdmin in Action

2. Write ajax in the Create a new database box, and then click the Create button.
3. phpMyAdmin doesn't have the visual tools to create new users, so you'll need to write some SQL code now. You need to create a user with full access to the ajax database, which will be used in all the case studies throughout the book. This user will be called (surprise!) ajaxuser, and its password will be practical. To add this user, click the SQL tab at the top of the page, and write this code in it:

```
GRANT ALL PRIVILEGES ON ajax.*
TO ajaxuser@localhost IDENTIFIED BY "practical"
```

SQL does sound a bit like plain English, but a few things need to be mentioned. The * in ajax.* means *all objects in the ajax database*. So this command tells MySQL "give all possible privileges to the ajax database to a user of this local machine called ajaxuser, whose password is practical".

4. Click the Go button.

Congratulations, you're all set for your journey through this book. Have fun learning AJAX!