Contents

Preface

The creation of this book has provided me, Mike McGrath, a welcome opportunity to demonstrate the latest server-side scripting techniques with PHP and MySQL databases. All examples I have given in this book demonstrate modern features of the PHP scripting language using the current MySQL Relational Database Management System that is supported on both Windows and Linux operating systems. I sincerely hope you enjoy discovering the exciting possibilities of PHP and MySQL, and have as much fun with it as I did in writing this book.

In order to clarify the code listed in the steps given in each example, I have adopted certain colorization conventions. Components of the PHP language are colored blue; programmer-specified names are red; numeric and string data values are black; and comments are green:

```
<?php
        # Write the traditional greeting.
        $string = '<p>Hello World!</p>' ;
        echo $string ;
?>
```

Similarly, components of the SQL query language are colored blue; programmer-specified names are red; numeric and string data values are black; and comments are green:

```
# Insert 5 records into the "top_5_films" table.
INSERT INTO top_5_films ( position , title , year ) VALUES ( 1 , "Citizen Kane" , 1941 ) ;
```

Additionally, in order to identify each source code file described in the steps, a colored icon and file name appears in the margin alongside the steps:

script.php

query.sql

index.html

style.css

For convenience I have placed source code files from the examples featured in this book into a single ZIP archive. You can obtain the complete archive by following these easy steps:

1. Browse to **http://www.ineasysteps.com** then navigate to Free Resources and choose the Downloads section

2. Find PHP & MySQL in easy steps, 2nd edition in the list, then click on the hyperlink entitled All Code Examples to download the archive

3. Now, extract the archive contents to any convenient location on your computer

1 Getting started

Welcome to the exciting world of the data-driven web with PHP & MySQL. This chapter demonstrates how to create a dynamic development environment incorporating the Abyss Web Server, the PHP engine, and the MySQL database server.

Introducing PHP & MySQL

The most appealing modern websites provide a customized user experience by dynamically responding to some current conditions – user name, time of day, latest blog, shopping cart contents, etc. Many of these dynamic websites are created with PHP and MySQL.

What is PHP?

PHP is a widely-used general purpose scripting language that is especially suited for web development, and can be embedded into HTML. It was created by programmer Rasmus Lerdorf as a set of scripts to maintain his website that he released as "Personal Home Page Tools (PHP Tools) version 1.0" on June 8, 1995. These were extended in the version 2 release of 1997, and the name changed to become a recursive acronym "PHP: Hypertext Preprocessor" in version 3 the following year. Performance, reliability, and extensibility were improved in 2000 with the release of PHP4, which was powered by the new Zend engine – a virtual machine. The current version, PHP5, is powered by the Zend II engine and produced as free software by the PHP group. Today, PHP is installed on over 20 million websites and 1 million web servers.

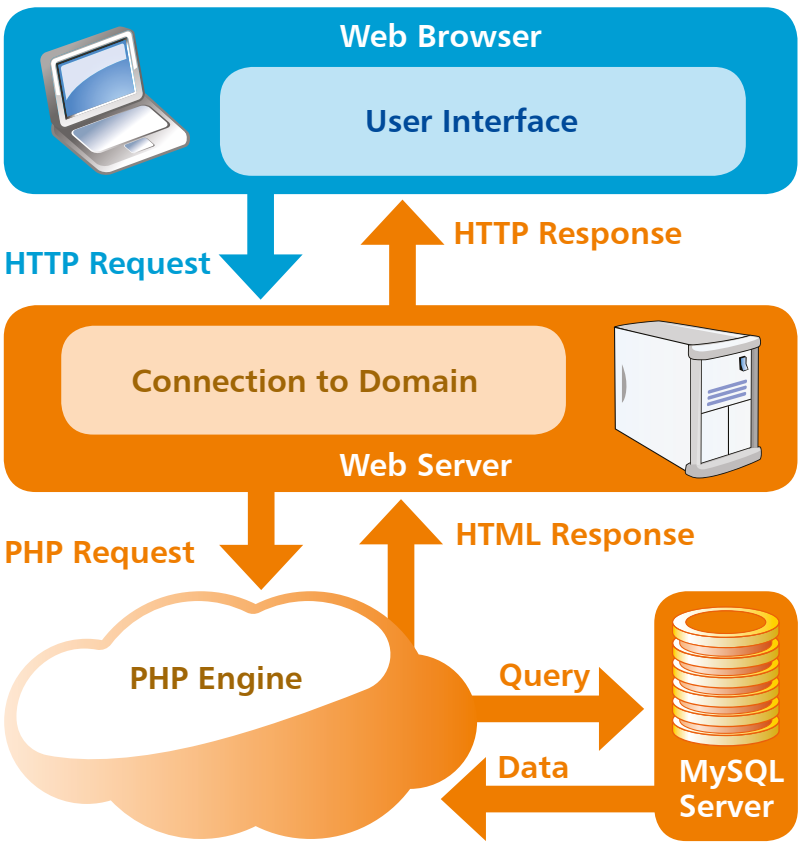
What is MySQL?

MySQL is the world's most popular database software. It is used to manage stored data and is described as DataBase Management Software (DBMS) or Relational DataBase Management Software (RDBMS). MySQL was created by Michael Wildenius and David Axmark back in 1995. Its name ("My-S-Q-L" officially, but often pronounced "My Sequel") is a combination of Michael's daughter's name "My" and the term "SQL" (Structured Query Language). MySQL was originally produced by the company MySQL AB, founded by its creators, which was acquired by Sun Microsystems in 2008, and subsequently by Oracle in 2010. The current version, MySQL 8.0, is powered by the InnoDB storage engine, and the MySQL Community Server edition is available as free software. Today, MySQL is used on some of the most frequently visited websites, including Google, Wikipedia, Facebook and Twitter.

It is important to recognize that PHP and MySQL are both "server-side" technologies – that is to say they reside on the web server. They are not "client-side" technologies resident on the user's computer. So their magic takes place in "The Cloud".

Understanding The Cloud

Whenever a user asks to view a web page in their browser it requests the page from the web server, and receives the page in response, via the HTTP protocol. Where a web page contains PHP script, the web server may first call upon the PHP engine to process the code and, if required, request data from a MySQL database before sending the response to the browser.



HTTP (HyperText Transfer Protocol) is the common communication standard that allows any computer connected to any web server to access files across the web.

The examples in this book are created and tested with the listed software versions but may require modification for other versions.

The ensuing pages describe how to create a development environment for data-driven websites by installing the following server-side technologies on your own computer:

- Web Server – Abyss Web Server X1 Free Personal Edition
- PHP Engine – PHP 7.2.4
- MySQL Server – MySQL Community Server 8.0.11

Installing Abyss Web Server

Abyss X1 is a free compact web server available for Windows, macOS/Mac OS X, and Linux operating systems available for download at **aprelium.com**. Despite its small footprint, Abyss supports many powerful features, including dynamic content generation with server-side scripts – so is an ideal companion for PHP & MySQL.

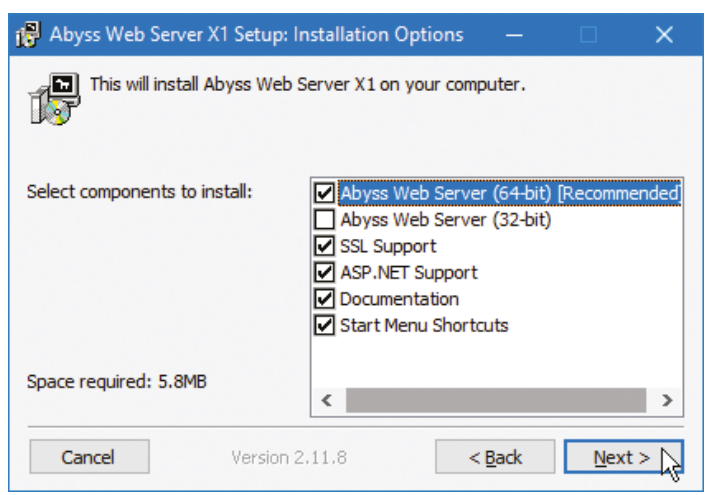
The Abyss Web Server can be installed on your own computer to provide an environment for PHP & MySQL website development.

Further guidance on installation of the Abyss Web Server is available at aprelium.com/abyssws/start.html

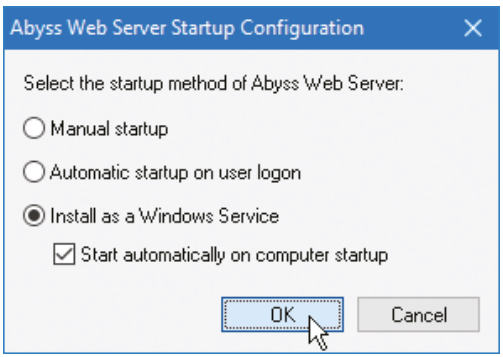
1. Download the Abyss X1 Web Server setup package for your system from **aprelium.com/abyssws/download.php**

2. Run the setup installer and accept the License terms, then choose to install with the recommended options

The Abyss setup package for Windows is an executable file named **abwsx1.exe** that you run to install the web server.



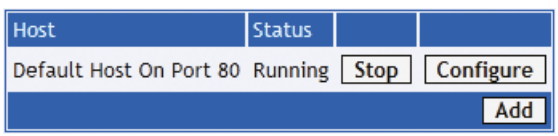
3 Accept the suggested location of **C:\Abyss Web Server**, then choose to install Abyss as a Windows Service



If you choose the Manual startup option, the Abyss logo will not appear in your system tray for easy start/stop control and access to the server console. Instead, the console can be found with your browser at **http://localhost:9999**.

4 Next, select your preferred language, then enter a name and password for future access to the Abyss server console

5 Now, log in using your chosen name and password to see the Abyss console confirm the server status as Running



6 Finally, enter **http://localhost** into your browser address field, then hit Return to see the default Abyss Welcome page get served up by the Abyss Web Server

In the Abyss console, click the Configure button then the General icon to see the default HTTP Port is 80 and the default Documents Path (where your web pages will reside) is **/htdocs**.

Installing the PHP engine

The PHP interpreter "engine", which implements PHP scripts within web pages, is available for Windows, macOS/Mac OS X, and Linux operating systems as a free download at **php.net**

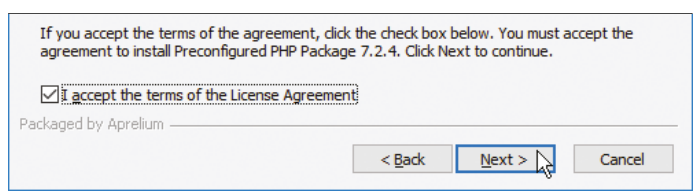
Additionally, a pre-configured package for the Abyss Web Server on Windows is available from **aprelium.com** and is recommended for a simple, fast installation.

Further guidance on installation of PHP is available at **php.net/manual/en/install.php**

1. Download the PHP setup package for your system from **aprelium.com/downloads**

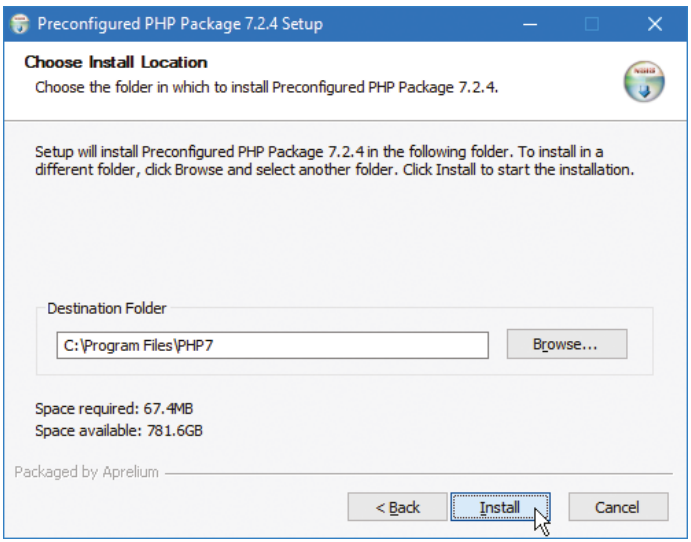
2. Run the downloaded executable file to launch the Setup Wizard, then click on the Next button to begin

3. Next, accept the License terms, then click on the Next button to proceed with the installation



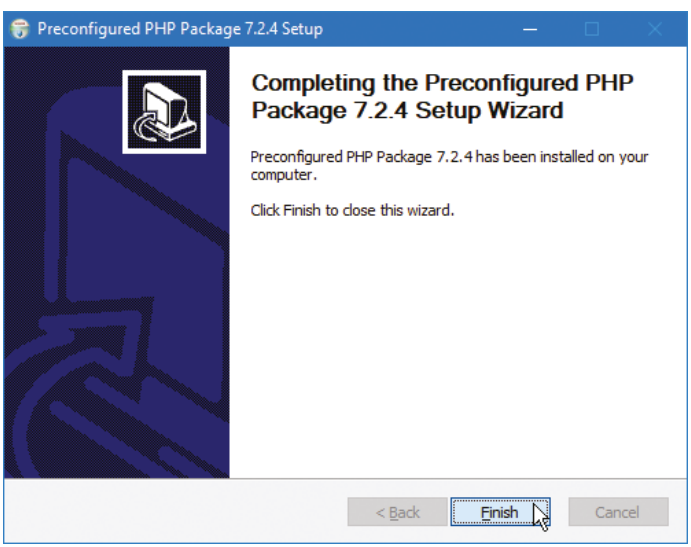
If installing PHP for Abyss on Windows from **php.net**, be sure to choose the VC6 Thread Safe version – as it requires fewer Windows dependencies.

4 Accept the suggested location of **C:\Program Files\PHP7**, then click the Install button to continue



The PHP installation location will be required when configuring the Abyss Web Server for PHP – make a note of the Destination Folder.

5 Finally, after the installation completes, click on the Finish button to close the Setup Wizard



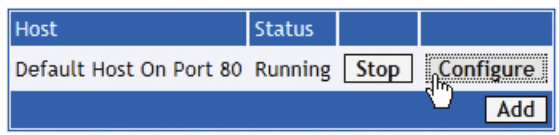
Following installation of PHP, the web server cannot yet execute PHP scripts until it is configured to recognize them and to find the PHP interpreter engine – all as described on pages 14-15.

Configuring Abyss for PHP

The Abyss Web Server must be configured to recognize PHP scripts and employ the PHP interpreter when it encounters them. This is achieved in the Abyss server console by associating the file extension ".php" as being PHP scripts, and by specifying the location of the PHP engine on your system to interpret them.

Further guidance on configuration of the Abyss Web Server is available online at **aprelium.com/abyssws/start.html**

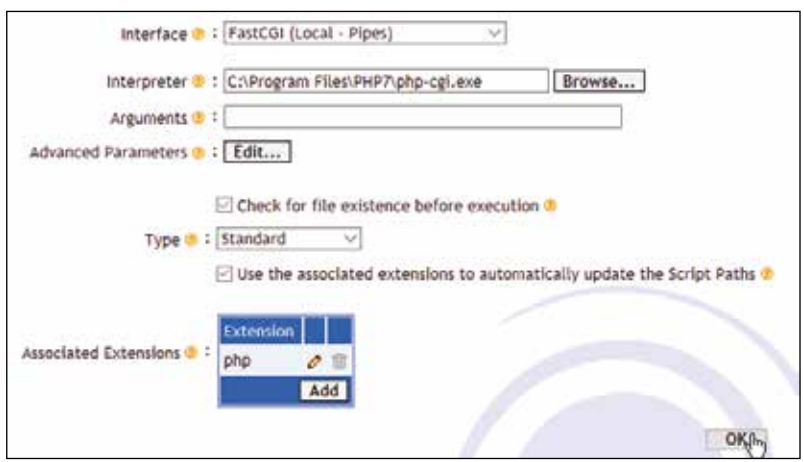
1. Enter **http://localhost:9999** into your browser address field (or click on the Abyss logo in your system tray) to launch the Abyss server console, then click the **Configure** button to open its Configuration page



2. Next, click on the  Scripting Parameters icon to open the Scripting Parameters page for editing

3. Ensure that the **Enable Scripts Execution** box is checked, then click the **Add** button in the Interpreters table to open the Interpreters-Add page

4. Now, set the **Interface** to "FastCGI", the **Interpreter** to the location of the PHP interpreter (**php-cgi.exe**) on your system, and **Add** the file Extension "php" then click **OK**



The **localhost** domain name is an alias for the IP address **127.0.0.1** – so the Abyss Web Server console can alternatively be addressed as **http://127.0.0.1:9999**.

5 Click **OK** in the Console window, then click the **Restart** button to apply the changes to the Abyss configuration

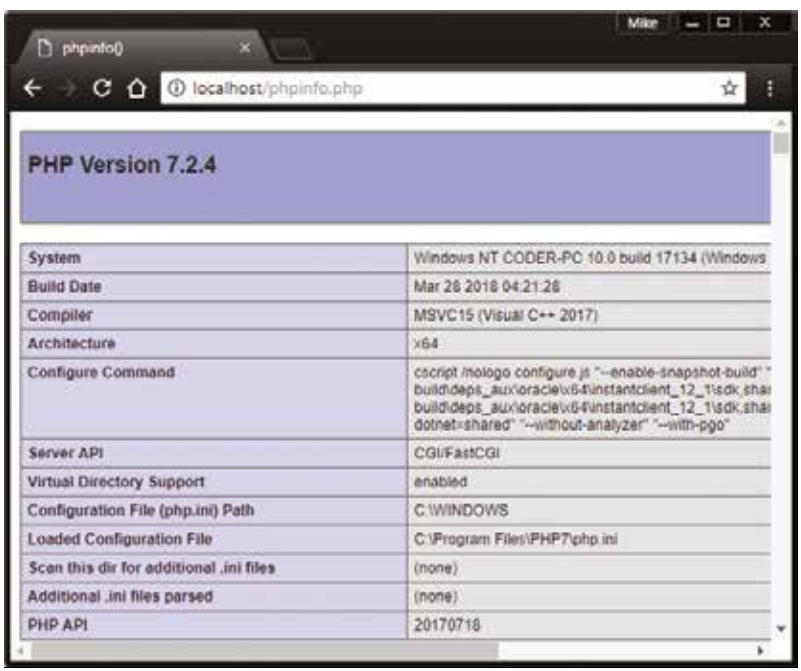
The Abyss Web Server should now be running on your system, correctly configured to recognize that documents having the **.php** file extension should be interpreted by the PHP engine. Configuration can now be tested by creating a simple PHP script for service to your web browser by Abyss.

6 Open a plain text editor and exactly type the script below

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

7 Save the script as **phpinfo.php** in the Abyss document path directory, typically at **C:\Abyss Web Server\htdocs**

8 Exactly enter the location **http://localhost/phpinfo.php** into your web browser's address field to see Abyss serve up a web page containing your PHP version information



PHP scripts are case-sensitive so you must copy the listed script using lowercase characters only.

phpinfo.php

Documents can only be interpreted by the PHP engine if served up by the web server using the HTTP protocol. You cannot simply open a PHP file in your browser directly. Always use the location **http://localhost/**.

Embedding PHP script

PHP script may be embedded within HTML documents – meaning PHP and HTML code can both happily co-exist in the same file. All embedded PHP code must be contained within **<?php** and **?>** tags so it can be readily recognized by the PHP engine for interpretation. Typically, the PHP code will write content into the body section of the HTML document, which is then sent to the web browser.

hello.php

1 Launch a plain text editor and create this valid barebones HTML5 document with an empty body section

```
<!DOCTYPE HTML>
<html lang="en">
<head><meta charset="UTF-8">
<title>Getting Started With PHP</title>
</head>
<body>

</body>
</html>
```

2 Insert tags into the body section to contain PHP code

```
<?php

?>
```

3 Now, insert between the PHP tags a descriptive comment and a line of code to write content into the body section

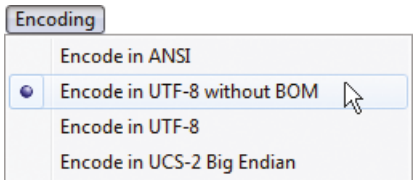
```
# Write the traditional greeting.
echo '<h1>Hello World!</h1>' ;
```

All whitespace and PHP comments are ignored by the interpreter. Single-line comments may begin with # or // and multi-line comments contained between /* and */ – as with the C programming language.

hello.php

```
1   <!DOCTYPE HTML>
2   <html lang="en">
3   <head>
4   <meta charset="UTF-8">
5   <title>Getting Started With PHP</title>
6   </head>
7   <body>
8   <?php
9   # Write the traditional greeting.
10  echo '<h1>Hello World!</h1>' ;
11  ?>
12  </body>
13  </html>
```

4 Set the document encoding to UTF-8 format then save it as **hello.php** in the Abyss server's **/htdocs** folder



Windows' Notepad automatically adds a hidden "Byte Order Mark" (BOM) to the file, while other editors (such as Notepad++ shown here) allow this to be omitted. Notepad++ can be freely downloaded from **notepad-plus-plus.org**

5 Next, enter the location **http://localhost/hello.php** into your web browser's address field to see Abyss serve up a web page containing content written by embedded PHP code

6 Now, use your web browser's View Source facility to see that PHP has written the content into the body section, including the HTML **<h1></h1>** heading tags

The PHP **echo** instruction statement literally writes the entire content contained within the pair of ' single quote marks. Like all other PHP statements it must be terminated by a ; semi-colon character.

PHP script can be embedded in earlier versions of HTML in just the same way. Other examples in this book demonstrate embedded PHP script but do not repeatedly list the HTML code.

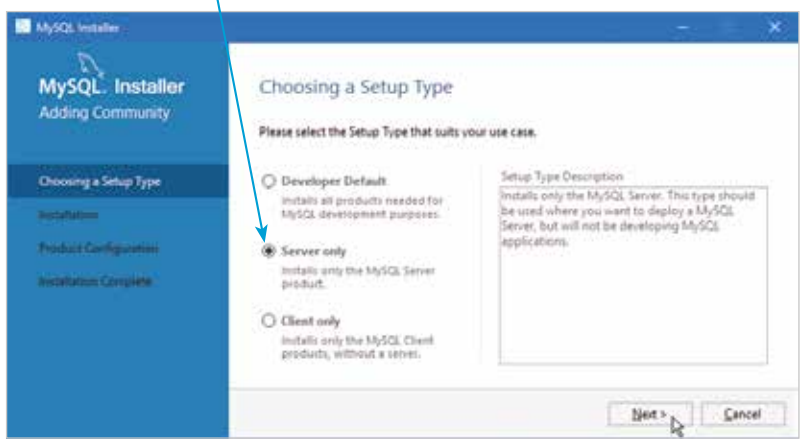
Installing the MySQL Server

The MySQL database server, which provides "back-end" storage for data-driven websites, is available for Windows, macOS/Mac OS X, and Linux operating systems as a free download at **mysql.com**

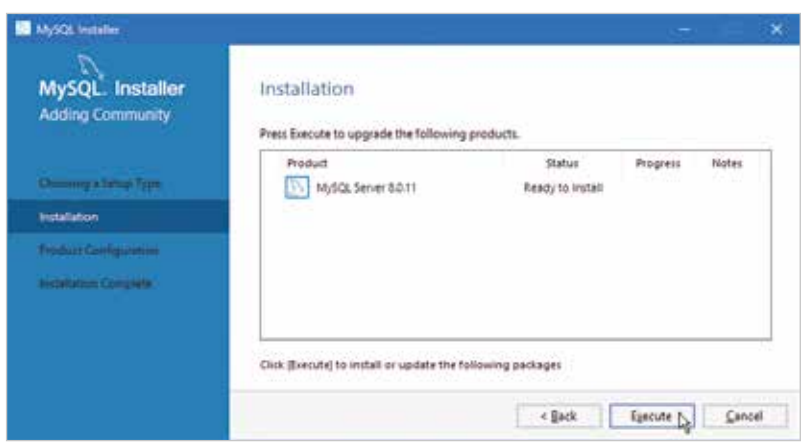
Further guidance on installation of the MySQL Server is available at http://dev.mysql.com/doc/refman/8.0/en/installing.html

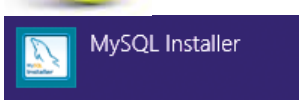
1 Download the MySQL Community Edition Server installer for your system from **mysql.com/downloads/mysql**

2 Run the installer and accept the License terms, then select "Server only" and click **Next** to continue



3 Click the **Execute** button to install the MySQL server

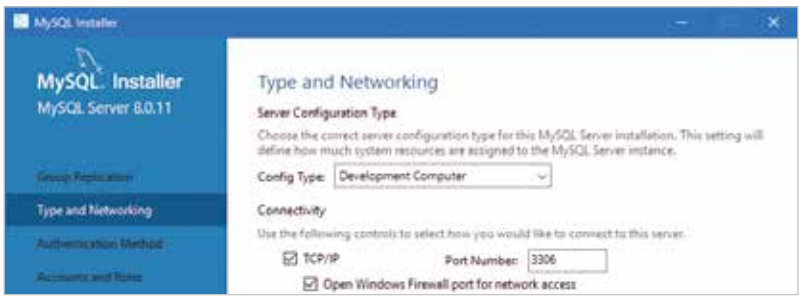




The MySQL Installer can be launched at any time from the Windows Start menu – to change the configuration or to install updates.

4 Next, choose the "Standalone MySQL Server" option, then click **Next** to continue

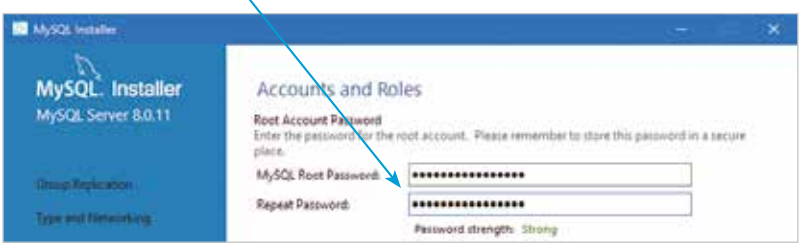
5 Select "Development Computer" as the server configuration type, then click **Next** to continue



By default, the MySQL Server uses port 3306. If you are running a firewall you may need to specifically allow the MySQL Server connections via this port. Refer to your firewall documentation for further guidance.

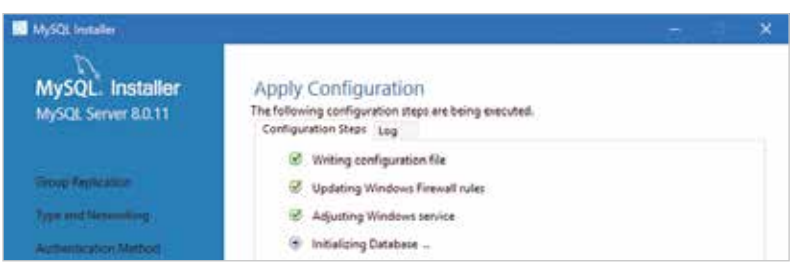
6 Select "Use Strong Password Encryption for Authentication", then click **Next** to continue

7 Enter a root user password of your choice twice into the **Password** fields, then click **Next** to continue



Write down your chosen root user password, username, and user password – you will need them often.

8 Click **Next** to run as a Windows Service, and **Next** to ignore Plugins, then click the **Execute** button to install the MySQL Server with your selected configuration



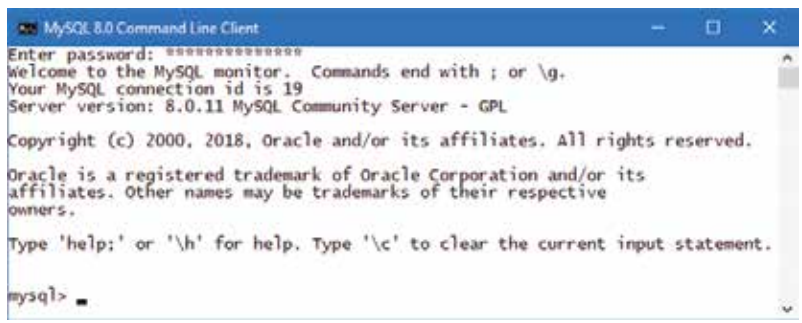
Ensure that your MySQL Server Configuration completes successfully before continuing. If necessary, repeat the installation process.

Using the MySQL Client

After installation of the MySQL Server as a Windows service, described on pages 18-19, you can communicate with databases via the MySQL Command Line Client that gets installed with the server package. Upon its launch it will first request the root user password you chose during installation. Once the password has been verified, the MySQL Command Line Client then presents a **mysql>** command prompt from which you can create and manipulate databases.

The MySQL Command Line Client can also be launched from a regular Command prompt by issuing the command **mysql -u root -p** where **mysql** is added to your system Path, or from within its **/bin** directory.

1. Launch the MySQL Command Line Client from the MySQL group that has been added to the Start menu, then enter the root password you chose during installation

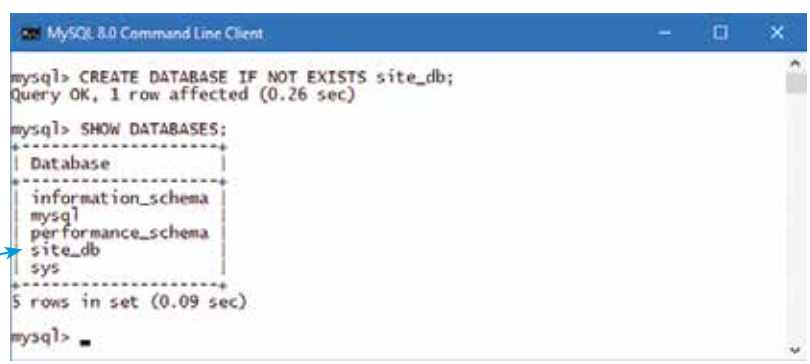


All MySQL commands end with a **;** semi-colon.

2. At the **mysql>** command prompt, precisely issue this command to create a new database named "site_db"
CREATE DATABASE IF NOT EXISTS site_db ;

3. Now, precisely issue a further command to display all databases that now exist on your MySQL Server
SHOW DATABASES ;

Installation creates some default databases, such as "sys", but your "site_db" database is the one that will be used throughout this book.



Creating MySQL users

While the MySQL root user is allowed complete control over the databases on the MySQL Server, it is obviously inadvisable to allow other users such freedom for best security of the databases. The root user can therefore create users with specific "privileges" controlling what actions they may perform on the MySQL Server.

The root user can create a user in the MySQL Command Line Client by issuing a clause to identify a unique user, like this:

CREATE USER IF NOT EXISTS '*username*'@'*hostname*'
IDENTIFIED WITH mysql_native_password BY '*password*' ;

The root user can then issue a clause to specify privileges allowed for a particular database to a particular user, like this:

GRANT *privileges* ON *database*.* TO '*username*'@'*hostname*' ;

The privileges are specified as a comma-separated list of keywords which that user may use when accessing the specified database. For instance, basic privileges to **SELECT**, **INSERT**, and **UPDATE**.

1. Log into the MySQL Command Line Client as the root user, then precisely issue this clause to create a user
   CREATE USER IF NOT EXISTS 'mike'@'locahost'
   IDENTIFIED WITH mysql_native_password BY 'easysteps' ;

2. Next, allow privileges to access the "site_db" database
   GRANT SELECT, INSERT, UPDATE ON site_db.*
   TO 'mike'@'localhost' ;

3. Now, issue a clause to confirm this user's privileges
   SHOW GRANTS FOR 'mike'@'localhost' ;

MySQL is case-sensitive and requires precise syntax – you must be sure to use correct spacing and letter case as it appears listed. For example, do not leave spaces around the @ character that defines the user name and host.

MySQL 8 introduces **caching_sha2_password** authentication, but the **mysql_native_password** earlier authentication can still be used to specify unencrypted passwords.

Select MySQL 8.0 Command Line Client

```
mysql> CREATE USER IF NOT EXISTS 'mike'@'localhost' IDENTIFIED WITH mysql_native_password BY 'easysteps';
Query OK, 0 rows affected (0.12 sec)

mysql> GRANT SELECT, INSERT, UPDATE ON site_db.* TO 'mike'@'localhost';
Query OK, 0 rows affected (0.56 sec)

mysql> SHOW GRANTS FOR 'mike'@'localhost';
+-----------------------------------------------------------------+
| Grants for mike@localhost                                       |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO `mike`@`localhost`                        |
| GRANT SELECT, INSERT, UPDATE ON `site_db`.* TO `mike`@`localhost` |
+-----------------------------------------------------------------+
2 rows in set (0.00 sec)

mysql> _
```

Connecting PHP & MySQL

Connection to a MySQL database can be attempted in PHP with a standard piece of script that describes four connection parameters of Host, Username, Password, and Database name. Upon failure, the script provides a descriptive message, whereas on success it specifies the character set to be used when sending data to and from the database server:

```
$dbc = mysqli_connect ( 'host', 'user', 'password', 'database' )
  OR die ( mysqli_connect_error() ) ;
mysqli_set_charset( $dbc, 'charset' ) ;
```

You need not understand in detail how the script works at this stage, but recognize that it contains sensitive information. For this reason it should not be placed in the web server's **/htdocs** directory like all other PHP scripts, where its contents may be accessible, but placed instead safely in **/htdocs** parent directory – for example, in the **"C:/Abyss Web Server"** directory rather than in the **"C:/Abyss Web Server/htdocs"** directory.

Any PHP script can incorporate another PHP script by using a "require" statement to specify the other script's path. This feature can be used to good effect to incorporate the connection script without revealing its sensitive information.

connect_db.php

1. Launch a plain text editor and create a connection script describing the parameters from the previous pages

```
<?php

# Connect  on 'localhost' for user 'mike'
# with password 'easysteps' to database 'site_db'.
$dbc = mysqli_connect
  ( 'localhost' , 'mike' , 'easysteps' , 'site_db' )
OR die
  ( mysqli_connect_error() ) ;

# Set encoding to match PHP script encoding.
mysqli_set_charset( $dbc , 'utf8' ) ;
```

2. Save the file in the parent directory of your web server's **/htdocs** directory folder as **connect_db.php**

3 Now, begin a second script that incorporates the connection script, stating its path in the parent directory

```
<?php
# Incorporate the MySQL connection script.
require ( '../connect_db.php' ) ;
```

require.php

4 Next, complete this script by adding instructions to display connection information if the attempt has succeeded

```
# Display MySQL version and host.
if( mysqli_ping( $dbc ) )
{ echo 'MySQL Server ' . mysqli_get_server_info( $dbc ).
        'connected on ' . mysqli_get_host_info( $dbc ) ; }
```

5 Save the script in your web server's **/htdocs** directory as **require.php**, then open it in your browser (via HTTP) to see the connection details on successful connection

6 Temporarily edit **connect_db.php** by changing the database name to one that does not exist, then save the file and reload the page in your browser to see the error

7 Finally, correct the **connect_db.php** script by changing the database name back to **site_db**, then save the file and reload the page to see the connection succeed once more

You do not need to understand how these scripts work just now – they merely ensure you can connect to MySQL with PHP. But you can usefully refer back to them later to see how your knowledge of PHP has progressed.

Congratulations, you have now successfully configured the Abyss Web Server, PHP engine, and MySQL Server for development.

Summary

- PHP is a scripting language that is especially suited for web development, as it can be embedded in HTML.
- MySQL is Relational DataBase Management Software (RDBMS) that can provide back-end data storage for websites.
- PHP and MySQL are both server-side technologies that deliver data-driven websites to the browser from The Cloud.
- A local development environment can be created by installing a Web Server, the PHP engine, and the MySQL Server.
- The Web Server must be configured to recognize scripts so it will direct them to the PHP engine for interpretation.
- All embedded PHP code must be contained within **`<?php`** and **`?>`** tags so it can be readily recognized by the PHP engine.
- Documents containing PHP script can best be encoded using the popular UTF-8 character format.
- The MySQL Server can be installed on Windows systems as a background service so it is always readily available.
- The MySQL Command Line Client gets installed with the Server package and can create and manipulate databases.
- The root user password chosen during installation of MySQL can be used to launch the MySQL Command Line Client.
- Connection to a MySQL database can be attempted in PHP with a standard piece of script that describes connection parameters of Host, Username, Password, and Database name.
- Any PHP script can incorporate another PHP script by using a require statement stating the other script's path.