

# USING MYSQL WITH YOUR APPS

## HOW TO USE THE SAMPLE

### 1 Introduction

---

This sample aims to show you how to use and ship MySQL with your applications compiled with ExeOutput for PHP. For legal reasons, ExeOutput for PHP does not include MySQL itself; however, your applications compiled with ExeOutput for PHP may use MySQL.

One of the problems you can meet is deployment: installing MySQL on each end user's machine is not really easy to achieve. Therefore, in attempt to avoid MySQL installation, we use the portable webserver named Server2Go to run a local and portable copy of MySQL. Thus, for easier deployment of your applications, MySQL does not have to be installed on the end user's machine.

Server2Go is a freeware WAMP stack. It is a software package that bundles Windows versions of PHP, MySQL, Apache and Perl. Server2Go is designed to run on removable and write-protected media such as USB Sticks or CD-ROMs.

Thanks to Server2Go, your applications made with ExeOutput for PHP can run MySQL locally and will be easily deployed, without requiring complex installations.

This guide shows you how the shipped sample works, and how you can use it for your own applications.

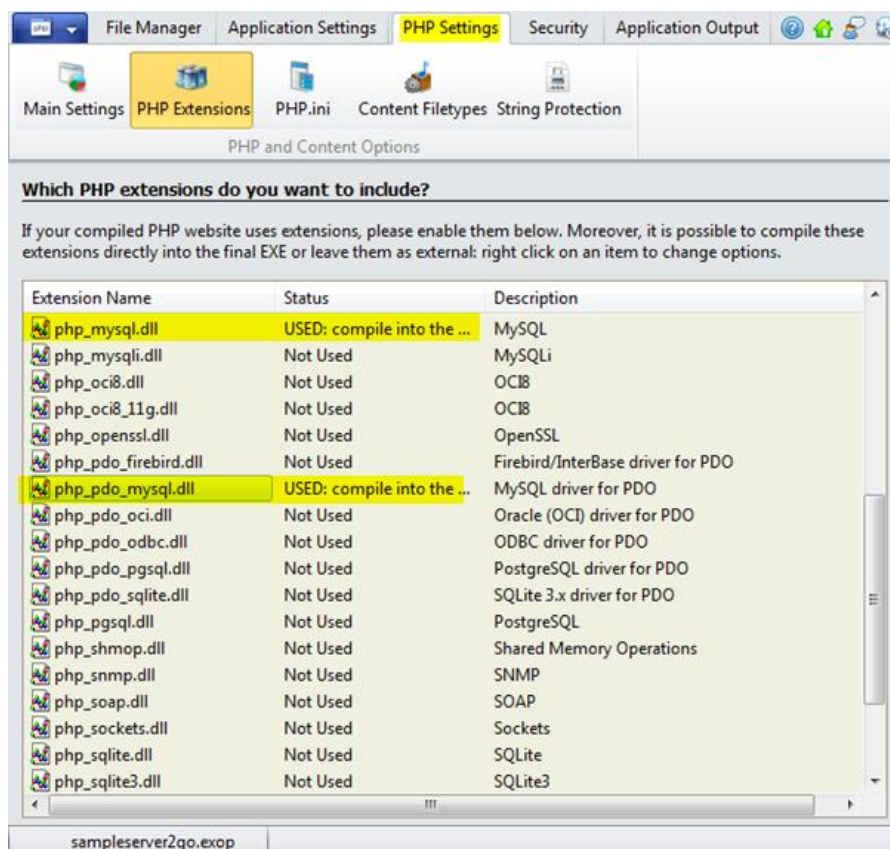
## 2 Testing the shipped sample

The sample comes with two folders: “**source**” and “**server2go**”, and an ExeOutput for PHP project file named sampleserver2go.exop.

- “Source” contains the PHP files of the application that we compile with ExeOutput for PHP.
- “server2go” contains the original Server2Go distribution, with some minor modifications as described below.

Open the ExeOutput for PHP project file and let’s find out some of the important changes we have made:

1. First, we have added MySQL support to our application: we have enabled the two PHP extensions for MySQL named php\_mysql.dll and php\_pdo\_mysql.dll, as shown below:

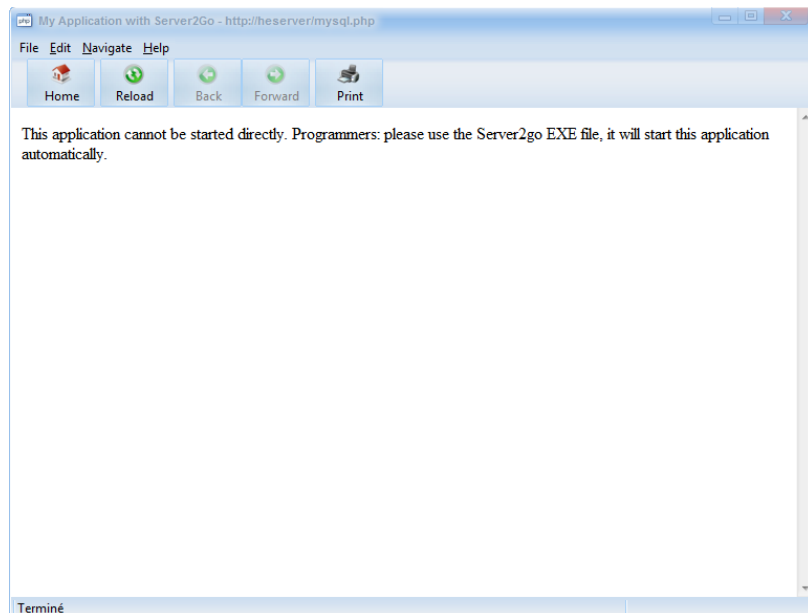


2. This is important: In “Application Output / Output Settings”, we have chosen to place the EXE file compiled by ExeOutput for PHP in the server2go distribution, exactly at the following relative path:  
“server2go\myapp\myapp.exe”

In fact, this EXE file is our application's program file and it will be run by Server2Go when the latter is started.

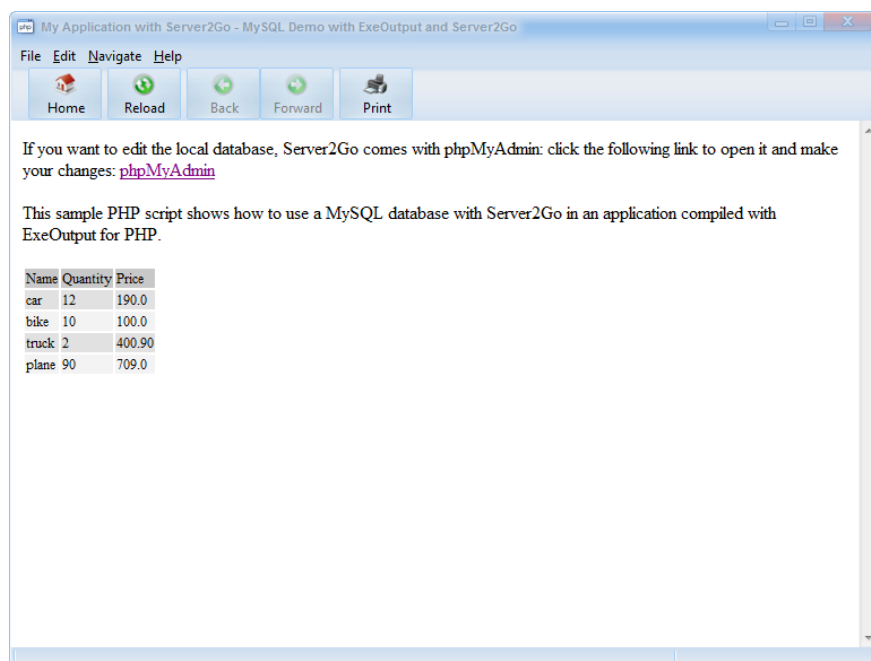
If you open the "server2go" folder, navigate to the "myapp" subfolder and you will find "myapp.exe".

Beware: you should not start "myapp.exe" directly; otherwise, it will return such an error:



In fact, to run your application, you should actually start "Server2Go.exe" (in the "server2go" folder).

After a splash screen, the application is started and shows the output:



Conclusion: this simple application was compiled with ExeOutput for PHP and it works with a local MySQL database. MySQL was not installed on the end user's computer.

### 3 How it works

- Server2go starts: a splash screen (that you can customize) is displayed while Server2go is loading its different components such as MySQL, Apache...
- After that, it runs the EXE file located at "server2go\myapp\myapp.exe", which should be your own application made with ExeOutput for PHP.
- When the application is closed, it shuts down MySQL and the other components.

Server2go's behavior can be customized thanks to its settings file, named "pms\_config.ini". We have made some modifications to the default settings, especially to make Server2go run the application EXE file made with ExeOutput for PHP, and not the default web browser.

If you are familiar with Server2go, you can take a look at the pms\_config.ini file and find out:

BrowserType=EXTERNAL

BrowserPath=myapp/myapp.exe

That's why the EXE created with ExeOutput for PHP should be placed at "server2go\myapp\myapp.exe".

If you want more information and documentation about configuring Server2go, please refer to its documentation at: <http://www.server2go-web.de/>

## 4 Using the local MySQL server from your PHP application

To establish a connection to your local databases, use the following PHP code:

```
<?php
// By default, Server2go comes with a sample database. Root admin is not
password-protected.
$mysqlusername = "root";
$mysqlpass = "";
$mysqlport = getenv('S2G_MYSQL_PORT'); // Returns the port of MySQL.
$mysqlhost = "localhost: ".$mysqlport; // The MySQL server is accessed
locally.
// We verify that our ExeOutput application was started by Server2go,
otherwise, the MySQL server may not have started.
if (empty($mysqlport)) die("This application cannot be started directly.
Programmers: please use the Server2go EXE file, it will start this
application automatically.");
$conn = mysql_connect($mysqlhost, $mysqlusername, $mysqlpass);
// etc.
?>
```

Note that the port used for MySQL may be random, that's why this is important to use `getenv('S2G_MYSQL_PORT')` to retrieve it.

## 5 Importing your data in the local MySQL server

Server2go ships with a ready-to-use MySQL distribution. The root user has simple login info: "root" and no password.

You have got several ways to import your data: SQL, PHP, phpMyAdmin...

If you want to edit the local database visually, the easier way is phpMyAdmin. In fact, Server2Go comes with phpMyAdmin by default.

For instance, in the shipped sample application, you can access phpMyAdmin directly from the application itself.

In your own applications, just add this PHP/HTML code to your index page:

```
<p><a href="<?php $apacheport = getenv('S2G_APACHE_PORT');
echo("http://localhost: $apacheport/phpmyadmin"); ?>">phpMyAdmin</a></p>
```

Then run your application and click the link to open phpMyAdmin and make your changes. The login info is: "root" and no password (leave the field empty).

It is possible to use phpMyAdmin to change the root password, create new databases, users...

Another way is to use PHP to create your databases, tables, import your SQL data... or the local MySQL tools available in "server2go\server\mysql\bin".

## 6 Steps for using the code sample in your own applications

1. You should already have a source folder (with all PHP files that will be compiled with ExeOutput for PHP), and a destination folder where the application EXE file will be created.
2. Open the destination folder, and copy the entire contents of the **server2go\** subfolder to a subfolder with the same name in your destination folder.
3. Start a new ExeOutput for PHP project, configure your project and do not forget to enable the two PHP extensions for MySQL named `php_mysql.dll` and `php_pdo_mysql.dll`, as explained above (see paragraph 2).
4. Modify your PHP files that contain the code to connect to the MySQL database, as explained above (see paragraph 4).
5. Optional: if you want to use phpMyAdmin to import data in MySQL, add the previously-mentioned HTML code (see paragraph 5) to your index page, so you can access phpMyAdmin while designing your application.
6. In ExeOutput for PHP, go to "Application Output / Output Settings", and choose the path to the EXE file so that it is created in the "server2go" subfolder, exactly at the following relative path: "server2go\myapp\myapp.exe" (see paragraph 2).
7. Compile your application and remember not to run it from ExeOutput for PHP, but instead, use "Server2Go.exe" in the "server2go\ folder" to start your application.

---

## 7 Deploying your application

---

You have to ship the entire contents of the “server2go\” folder. Nothing else is required.

If you want to deploy the application on a USB stick or CD, burn the contents of the “server2go” folder (at root). Moreover, an AUTORUN.INF sample is already available in “server2go\” if you wish your application to run when the disk is inserted.

Remember that “Server2Go.exe” must be used to start your application.

If you want to have a single EXE instead of a large bunch of files, you can pack all of these files into a Self-Extracting .EXE file with our setup tool Paquet Builder, available at <http://www.gdgsoft.com/pb>

---

## 8 Support

---

We do not provide free technical support by email for this sample. If you have questions or problems, feel free to post them in our forum at <http://www.gdgsoft.info>

---

## 9 Licenses

---

This sample is provided as free and open-source software.

THIS SAMPLE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THIS PRODUCT IS WITH YOU. SHOULD THIS PRODUCT PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION.

IN NO EVENT WILL THE AUTHOR OF THIS PRODUCT, OR ANY APPLICABLE COPYRIGHT/TRADEMARK OWNER BE LIABLE TO YOU OR TO ANY OTHER PARTY FOR ANY DAMAGES, INCLUDING, BUT NOT LIMITED TO, ANY DIRECT, INDIRECT, GENERAL, SPECIAL, INCIDENTAL, PUNITIVE, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, CORRUPTION OR LOSS OF DATA, ANY LOSSES SUSTAINED BY YOU OR THIRD PARTIES, A FAILURE OF THIS PRODUCT TO OPERATE WITH ANY OTHER PRODUCT, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR BUSINESS INTERRUPTION), WHETHER IN CONTRACT, STRICT LIABILITY, TORT (INCLUDING NEGLIGENCE) OR OTHERWISE, ARISING OUT OF THE USE, COPYING, MODIFICATION, (RE)DISTRIBUTION OF YOUR PRODUCT (OR A PORTION THEREOF), OR INABILITY TO USE THIS PRODUCT (OR A PORTION THEREOF), EVEN IF THE AUTHOR, COPYRIGHT/TRADEMARK OWNER, OR ANY OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

License terms of Server2Go:

Server2Go is Donationware. That means you can download and use it for free and you don't have to pay any royalty charges when distribute an application on CD-ROM that uses Server2Go. But if you use it commercially or like to keep it free available you should donate to the project. How to donate? Just take a look at the donation page. There are a lot of inexpensive possibilities to help.

Every donator that is donating more than 10 Euros is getting a user id and password with that he can download additional software that is not public available. See the download page for details.

Server2Go: <http://www.server2go-web.de/>

ExeOutput for PHP: <http://www.exeoutput.com>

MySQL: <http://www.mysql.com/>

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.