

ADVERTISEMENT

# SSIS Package Monitoring

Real-Time Monitoring with BI xPress 14 Day Free Trial



## 🔒 PHP - PDO or MySQLi? - Which is better/more versatile?

**Zapato33****Dec '14 #1**

I'm currently doing the PHP & MySQL Web Development for Beginners course (MySQLi), however, last night I decided to check out the Jump Start PHP book by Callum Hopkins and noticed that the code examples provided are a bit different. I found out about PDO and got me thinking whether it would be a better career choice to learn it instead of MySQLi.

I was trying to find out about the main differences between the two while reading people's opinions about them but I'm still unsure which one to choose. I know that PDO allows you to code for multiple database drivers and reusing code is easier, however, people say that MySQLi is easier to learn and understand (?), more efficient and as secure as PDO (?). In regards to PDO's compatibility with multiple database drivers, what kind of situation would require a programmer to need to switch to a different database? Could those situations present themselves multiple times in a programmer's career path?

I was hoping someone could shed some light about the two and what would make PDO/MySQLi a better career choice over the other.

Thanks a lot in advance.

**Mittineague** Former Community Award Winner**Dec '14 #2**

Hi Zapato33 welcome to the forum

Zapato33:

MySQLi is easier to learn and understand

IMHO only if you're coming from experience using deprecated mysql\_ and want to avoid OOP.

Zapato33:

more efficient and as secure as PDO

That's debatable. I think it's always possible to write inefficient / insecure code. Nothing it itself will magically take care of all problems for you. But some let you tackle certain problems easier I guess.

Zapato33:

Could those situations present themselves multiple times in a programmer's career path?

How long are you planning to work with databases? For example, the currently available drivers are <http://php.net/manual/en/pdo.drivers.php>

Driver name	Supported databases
PDO_CUBRID	Cubrid
PDO_DBLIB	FreeTDS / Microsoft SQL Server / Sybase
PDO_FIREBIRD	Firebird
PDO_IBM	IBM DB2
PDO_INFORMIX	IBM Informix Dynamic Server
PDO_MYSQL	MySQL 3.x/4.x/5.x
PDO_OCI	Oracle <b>Call Interface</b>
PDO_ODBC	ODBC v3 (IBM DB2, unixODBC <b>and</b> win32 ODBC)
PDO_PGSQL	PostgreSQL
PDO_SQLITE	SQLite <b>3</b> <b>and</b> SQLite <b>2</b>
PDO_SQLSRV	Microsoft <b>SQL Server</b> / <b>SQL</b> Azure
PDO_4D	<b>4D</b>

Besides MySQL, I've worked with 3 others of those in about 20 years time. But almost exclusively MySQL

---

Zapato33

Dec '14 #3

Mittineague, thanks a lot for your response.

I'm actually still a computer science student but I wanted to learn PHP because I want to do web development. For now, just focus on small personal projects, stuff like that. However, I want to make sure I learn what is best, must appropriate and future-proved. I don't want to spend my time learning something that will eventually loose popularity/momentum. Someone told me that PDO is the more modern way of dealing with databases, is Object oriented and will eventually become the only way of working with databases.

Also, is there a reason for you to stick with MySQL or was it just a matter of MySQL being the first database driver you learned? Are there any advantages/disadvantages of using any one in particular?

---

**Mittineague** Former Community Award Winner

Dec '14 #4

Actually the first database I worked with was MS Access. with VB code.

But when I decided to have a "real" website (not AOL Hometown or Geocities) it was a LAMP host so I learned PHP with MySQL. At the time, mysqli\_ was about all that I could deal with as a newbie.

I've had a long road getting away from thinking procedural after many years of writing it, so moving to mysqli\_ that also has procedural style was easier. But over the years I've slowly got a bit of OOP-think and when I needed to work with SQLite I began tackling PDO.

I can't say if one db is better than another in any particular way AFAIK with my limited experience. The only reason I mostly worked (and still do) with MySQL is because that's what my host has.

There are syntax and datatype differences but AFAIK they all perform about the same at the scale of use I've put them through.

---

**Pepster**

Dec '14 #5

I'll give my .05 cents (it used to be .02 cents, but due to inflation 😊)

I started off using mysqli, but notice a few books I that I bought on PHP were using PDO and various people posting that PDO was what they were using and recommend PDO. So I decided to give PDO a whirl and when I starting learning OOP that is really when things started to click. For me now PDO is second nature and easier to remember than a bunch of ? marks that mysqli lends itself to. As being more efficient I think any code a person writes can be efficient or inefficient, it depends on what type of programmer you are.

---

**lorenw** SitePoint Wizard

Dec '14 #6

I was a last holdout with mysql functions, I made 4 attempts with mysqli and/or PDO but it seemed too complicated.

I installed the latest PHP distro in localhost and got the dreaded mysql functions deprecated message. I made one final go of it and the information was overwhelming.

I saw many sites, <http://code.tutsplus.com/tutorials/pdo-vs-mysqli-which-should-you-use--net-24059> That prefer PDO over mysqli.

I learned PDO and never looked back.

PDO is also compatible with many databases, mysqli is only for mysql.

**Michael\_Morris** Former Member of the Month

Dec '14 #7

PDO is a wrapping library for multiple database drivers, including mysqli. So when you are using PDO you are using mysqli and so the security vulnerabilities are more or less the same. Injection can occur with either if you forgo the prepare statement/execute paradigm. PDO is ever so slightly slower than mysqli, but if the difference in speed between the two is important to you then you're using the wrong programming language to begin with and would be better off using C.

**Zapato33**

Dec '14 #8

Everyone, thanks so much for your comments. They have been very helpful and I will definitely give PDO a go.

Michael\_Morris, you said

Michael\_Morris:

PDO is ever so slightly slower than mysqli, but if the difference in speed between the two is important to you then you're using the wrong programming language to begin with and would be better off using C.

What are the reasons for you to say that in such case you are better off with C?

**Michael\_Morris** Former Member of the Month

Dec '14 #9

C is harder to develop with, but it's "closer to the metal" and therefore more efficient than PHP in almost all use cases if the code is written efficiently. The tradeoff is it takes more time to develop a C program. As a rule of thumb developer time is more valuable than computer processor time. If how fast the program can execute is a real concern for any reason, C is the way to go.

**xiris**

Dec '14 #10

Codifying in C for web development I really think is a madness. If you are evaluating others languages, take a look at **python** or **ruby** for example (What about HHVM 😊)

This two points bellow I think it was amazing:

- **PHP >= 5.5**
  - Native opcode implemented, increase performance a lot;
- **Zephir**
  - Concepted by Phalcon developers as an hybrid language;
  - You can use to put some part of your project as a PHP library and improve your project load time, for

example;

PHP is growing up 😊

---

**Mittineague** Former Community Award Winner

Dec '14 #11

Zapato33:

What are the reasons for you to say that in such case you are better off with C?

From

<http://php.net/get-involved.php>

You will also need experience in C programming as PHP is written entirely in C.

So PHP is kind of a "middle-man".

Not to worry much though. For most sites the size, resource use, and amount of traffic are more than adequately handled, Only when you get to the extreme that it starts to be the bottle-neck.

---

**oddz** SitePoint Wizard

Dec '14 #12

Typically the underlying culprit of noticeable performance issues tends to be interacting with a persistent storage system like MySQL. On high traffic php based sites there are most likely going to be several caching layers implemented like memcache, and varnish. Those will certainly circumvent any short comings of interpreted code when implemented properly. Especially if you can get away with running a static cache for high volume pages.

All and all I wouldn't worry to much about performance in the beginning of learning php. Performance tuning is a very wide and vast topic that requires multiple technologies to get it right for supporting high traffic web presences. Not something you really need to know when just starting out.

I might go as far to say you're falling victim to over engineering unless outages and performance are a common problem with the site being ran. There is something to be said for anticipating scale problems in the beginning but again that is a more advanced topic not really appropriate for someone just starting out imo.

That being said if the book is teaching mysqli use mysqli. Once you understand the basic concepts of interacting with mysql through mysqli\_\* functions transitioning to PDO isn't a huge leap.