



100,000 SQLite installations – lessons learned

International PHP Conference 2005, 11/09/2005

Björn Schotte



Björn Schotte - „PHP Pioneer“

- 1998/1999: PHP-Center.de
- 2000: world's first PHP Conference
- 2001: editor-in-chief PHP Magazin
- 2001: International PHP Conference
- 2001: co-founder of ThinkPHP
- 2004: refounding Mayflower GmbH



ThinkPHP – The LAMP Experts



- high-level software development since 2001
- PHP trainings for more than 300 people from Top50 companies
- PHP Support – <http://www.thinkphp.de/support/>
 - Enterprise PHP Support
 - Guaranteed reaction times up to 2 hours
 - Basic, Professional, Gold – 24/7 available on request
- Fostering the PHP community and PHP itself in Germany
- „ThinkPHP by Mayflower“ – a division of Mayflower GmbH

„100,000 SQLite installs“

- ~ 35 employees
- „ThinkPHP – by Mayflower“
- Customers include:
 - Siemens COM
 - Vaillant Group
 - Telefónica Germany
 - Landeshauptstadt München
 - HypoVereinsbank
 - Sixt AG
 - Stadtwerke München (SWM)
 - Bundeskriminalamt
 - ...
- 4 managing partners (founders of ThinkPHP)
- 2 offices (Munich and Würzburg)
- well-known in the PHP community
- Know how is more than PHP-only (High Performance web architecture, .NET, ...)
- Big projects in the range of man-years with PHP (HVB, Vaillant, Sixt, Telefónica)
- OpenSource Software vendor (i.e. PHProjekt, Outlook Sync, ...)

The customer

- Vaillant Group
- a leading european heating device manufacturer
- about 9,000 employees
- revenues 2004: 1,8 billion €
- ten brands: Vaillant, SaunierDuval, Bulex, Glowworm, Bongioanni, ...



The project: spare parts shop

- If parts of the boiler or the heating system break down, the installer needs a spare part
- Vaillant provides a web shop for buying those spare parts
- Searchable for EAN numbers, device groups etc.
- The shop provides exploded images of the devices for the installer to easily recognize the spare part he is searching

Spare parts: let's look at it



The situation before

- we displaced an already existing version
- that was:
 - ASP web application
 - access database for delivering the shop on CD
 - Not very flexible
 - exploded images had to be „hotpointed“ (for HTML image map) by a 3rd party

The new system, Web

- web shop based on PHP and MySQL
- importer for SAP based catalog data (CSV files)
- OCR technology (using gocr) to automagically try to „hotpoint“ exploded images when uploaded via WebDAV
- a Flash-based editor to edit the hotpoints for each image
- integrating more than 30,000 spare parts data from SAP
- based on the Chairman framework of Mayflower
 - DB abstraction layer (PHPLIB)
 - easily implemented a SQLite driver

SQLite2: SELECTs

```
$db->query(„select a.foo, b.bar from tblA a,  
tblB b where a.id=b.id“);  
while ($db->next_record()) {  
    print $db->f(„foo“); // error!  
}
```

Error! Because `sqlite_fetch_array()` delivers in this case
„a.foo“ as the array key

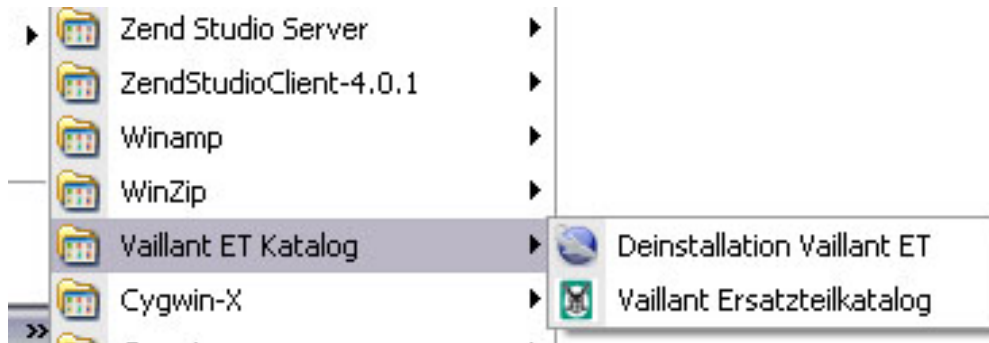
You have to strip out the „a.“ to work transparently.

The new system, CD version

- MySQL2SQLite converter
- Microweb as the webserver component
- PHP as the scripting language: 1:1 copy of the web shop
 - of course without the admin tools ☺
- MySQL was no choice for the CD version
 - license cost would be too high

The architecture of the Web2CD process

- Goal: to provide the customer the possibility to create the country CD versions standalone
 - formerly, a 3rd party had to generate it at high cost
- Solution: InstallAnywhere packager on the web!



The architecture of the Web2CD process

- the country admins of the spare parts shop have their own interface
- they can generate a packaged CD version on the fly
- The generation should be available over the web independantly of a 3rd party



The architecture of the Web2CD process

- in the background via cron-/batchjob:
 - convert relevant MySQL tables to SQLite
 - package all the scripts and exploded images (> 1,000 images)
 - run InstallAnywhere on Linux command line (XML configuration file) to package a .EXE file
 - mailto country admin that his CD is ready!
- Vaillant reduces the time to package a CD version from about several months to several days!

Using SQLite in this project, I

- SQLite has no licensing cost, MySQL licensing cost were too high
- it was available not only for PHP5, but also for PHP4
- the problem domain is a classical one-user-application
 - we don't need concurrency as in a normal web application
 - it mainly is a read-only application (cart data will be saved in a client cookie for later usage)
- BUT: it's pretty fast, even from CD!
- currently, there's SQLite3, but we stay using SQLite2

Using SQLite in this project, II

- you can't port your existing table structure 1:1 to SQLite2
- consult the homepage of SQLite, <http://www.sqlite.org/>
- we ended up in writing a converter tool that converts the table structure in a batch
- don't try to import the data on a slow server/desktop system
 - as SQLite is just one database file, I/O speed is important
 - on a slow desktop system, more than 30 minutes needed
 - on a normal server, 2 minutes needed

The rollout phase of the application

- first deadline for the German market, other countries followed
- biggest market with about 80,000 CDs to be shipped
- What does that mean?

We can't control the end user's system configuration!

Problems during rollout

- users had problems with the Microweb server
- because they used Internet providers like AOL, T-Online etc.
- within older windows versions, there was the TCP/IP socket missing
 - the call center had to guide the users through the installation
- there were some problems with php.ini settings
 - We had to give them an update of php.ini
- after two very strong weeks, we solved nearly every problem
- SQLite was like a rock!

„100,000 SQLite installs“

Other possible implementations?

- in our case:
 - be aware of not being able to control the user's environment
 - if that's okay, then Microweb+PHP+SQLite is a good choice
- XUL/Flexx a choice for implementation?
 - yes, but at a higher cost: you have to write the application two times
 - with XUL, you are bound to the Mozilla platform

Other possible implementations?

- improvements: PHP-in-a-box.exe
- use Delphi:
 - Integrate Webserver and PHP itself into .exe
 - Point it to the document root
 - Use SQLite as database
- or:
 - Integrate Webserver, PHP and PHP scripts into .exe
 - Use SQLite as database
- use case: „PHPProjekt in a box“ (project PHPProjekt.exe)

PHPProjekt in a box: let's have a look!



■ Download at <http://www.thinkforge.org/>



„100,000 SQLite installs“



Thank you!

Björn Schotte
Mayflower GmbH
Pleichertorstr. 2
97070 Würzburg
+49 (931) 35 9 65 - 0
schotte@mayflower.de

