

 Getting up to speed on  
PHP 5.3

Lukas Kahwe Smith - [lukas@liip.ch](mailto:lukas@liip.ch)  
PHPCon Italia - Roma 18. - 20. Marzo 2009



# Who is Lukas Smith?

First steps into PHP in 1999

Joined PEAR in 2002

Member of the PEAR Group

Author of PEAR::MDB2 and PEAR::LiveUser

Release Manager for PHP 5.3

RDBMS Aficionado

Ultimate Frisbee Lover



# Release Management

For the first time there are **two release managers (RMs)**

Johannes Schlüter is the technical RM

I am the organizational RM

This means that the RMs are more available as the work load is shared across two shoulders (\*)

**Increased transparency** for all developers and more importantly end users

(\*) This includes these slides, a lot of them came from a talk Johannes created



# Feature Overview

**New Syntax:** Namespaces, Closures, LSB, goto ..

**New functions and classes:** SPL, OpenSSL, Date ..

**New Extensions:** PHAR, intl, fileinfo ..

New optional garbage collector

Optional replacement for libmysql aka mysqlnd

Under the hood **performance improvements**

**Better Windows support**

Much more ..

# Namespaces

Namespaces (NS) are mostly resolved at compile-time to ensure that there is little performance impact

May contain classes, functions and constants

When using NS their declaration **has to be** at the beginning of the file

There can be multiple NS per file and a single namespace **may span across multiple files**

New constant with the current NS `__NAMESPACE__` but “namespace” may also be used accessing elements of the same NS: “`new namespace\some_class();`”

# Namespaces

**Alias long NS names** with the “use” statement for the rest of the current file

There is no way to “use” several namespaces at once (aka there no “use foo\\*“)

**Function** resolution automatically **falls back to the global NS** if the function isn’t defined in the current NS

**Class resolution** calls the **autoload** mechanism if the class is not defined in the current NS

Fully qualified NS names start with a backslash and make the resolution entirely explicit



# Namespaces

MyClass.php

```
1. <?php
2.
3. namespace my \foo;
4.
5. class MyClass {}
6.
7. function strlen($string) { return 42; }
8. const MY_CONST = 'foo';
```

Other.php

```
1. <?php
2.
3. require_once( './MyClass.php' );
4.
5. use my \foo;
6. $foo = new foo \MyClass();
7.
8. use my \foo \MyClass;
9. $foo = new MyClass();
10.
11. use my \foo \MyClass AS somethingElse;
12. $foo = new somethingElse();
```



# Late Static Binding

Static references use to be resolved at compile time and static constants to where they were defined

“**static::**” is now resolved at runtime

```
1. <?php
2. class A {
3.     public static function who() {
4.         echo __CLASS__;
5.     }
6.     public static function test() {
7.         static::who();
8.     }
9. }
10. class B extends A {
11.     public static function who() {
12.         echo __CLASS__;
13.     }
14. }
15. B::test(); // B
```



# Other new syntax

New shortcut for “\$foo ? \$foo : \$bar;” as “\$foo ?: \$bar;”

New **NOWDOC** syntax which is similar to HEREDOC but does no expanding of variables

Support for **dynamic static calls** (like \$a::\$b)

New goto syntax limited to only support jumps out of a block but staying in the current scope

New **\_\_callStatic()** as an identical solution to \_\_call() but for static methods

Lambda functions and **closures** enable defining



# New extension features

**New classes in SPL** cover various data structures:  
SplDoublyLinkedList, SplStack, SplFixedArray,  
SplPriorityQueue, SplQueue,  
SplHeap, SplMinHeap, SplMaxHeap

Improved OpenSSL to enable easy implementation of  
**OpenID in PHP**

New classes and functions for Date:  
date\_create\_from\_format(), date\_get\_last\_errors(),  
DateInterval, DatePeriod

# New Extensions

New **fileinfo extension** replaces mimetype extension, which was moved to PECL for PHP 5.2

New **intl extension** provides forwards compatibility to PHP 6 and provides I18N formatters and collators

Other new extensions: enchant (replaces pspell), sqlite3

# PHAR

Similar to Java's JAR

Possibly the future default way for **distributing applications**

PHAR files can use a custom file format or be based on tar or zip archives

PHAR includes a flexible front controller system to do the mapping from request to a file inside the phar

PHAR's can be manipulated inside PHP and on the CLI

# mysqlnd

PHP native replacement for the libmysql

Developed by Sun/MySQL and contributed under the PHP license

**Deeply integrated into PHP** (memory management, user land stream API) with no external dependencies

Less bugs, faster turn around time when fixing bugs, lower memory overhead

May optionally be used with ext/mysql, mysqli and pdo\_mysql



# Garbage Collector

PHP by default has trouble freeing memory for cyclic references

New **optional garbage collector** developed during GSoC 2007 can free such memory

Enable via `php.ini` or with `gc_enable()`, `gc_disable()`

May be triggered automatically or explicitly with `gc_collect_cycles()`

For complex or **long running scripts** this can reduce memory usage by the cost of CPU time and slightly higher default memory use

# E\_DEPRECATED

Splits off deprecation notices from E\_STRICT

E\_DEPRECATED is part of E\_ALL

Notify user about features that will be removed in the next major version

E\_STRICT is now only used to warn about “bad” coding practices



# Under the hood tweaks

**New scanner** based on re2c

md5() now faster

**require\_once / include\_once** faster on second usage

Constants marked read only reduces memory usage

Improved engine stack implementation

Removed direct executor recursion

Use fastcall calling convention on x86

Various opcode optimisations



# Under the hood tweaks

PHP 5.3 is faster than PHP 5.2



# Better windows support

Added implementation for several functions previously only available on \*nix: `symlink()`, `getopt()`, `dns_*`

**Build both with VC6 and VC9** for faster binaries and improved installer

New website in the works that will provide a replacement for the now defunct pecl4win site

Its now a team effort supported with developer and hardware/software resources from Microsoft

There is now **documentation on how to compile PHP for windows** on <http://wiki.php.net/internals/windows>



# New php.ini features

Change PHP settings similarly to .htaccess with FCGI

New security feature via “request\_order” to define what parameters are included in \$\_REQUEST

Per-directory ini settings and INI constants / variables

```
1. ;Name for user-defined php.ini (.htaccess) files. Default is ".user".
2. user_ini.filename = ".user.ini"
3. ;To disable this feature set this option to empty value
4. user_ini.filename =
5. ;TTL for user-defined php.ini files (time-to-live) in seconds.
6. ;Default is 300 seconds (5 minutes)
7. user_ini.cache_ttl = 300
8. [PATH=/var/www/domain.com]
9. variables_order = GPC
10. safe_mode = 1
11. [my variables]
12. somevar = "1234"
13. anothervar = ${somevar} ; anothervar == somevar
14. [ini arrays]
15. foo[bar] = 1
16. foo[123] = 2
17. foo[] = 3
```



# BC breaks

## **Changes to the parameter parsing API (!!)**

New Keywords: namespace, closure

New global identifiers

New extensions

New Closure class

Removed Extensions (most were simply moved to PECL):  
ncurses, fpdf, dbase, fbsql, ming, msql

Dropped `zend.ze1_compatibility_mode`



# Release date .. ?

First Release Candidate (RC) planned for next Tuesday (March 24th)

Expect about 2-4 RC's in total released in 2-3 week intervals

Most minor versions get really stable within 2-3 patch releases

# Get ready for TestFest 09



<http://qa.php.net/testfest.php>

[http://gcov.php.net/viewer.php?version=PHP\\_5\\_3](http://gcov.php.net/viewer.php?version=PHP_5_3)

# Links

<http://downloads.php.net/johannes/> (Source)

<http://windows.php.net/> (Windows Binaries)

<http://snaps.php.net/> (Latest snapshots)

<http://cvs.php.net/php-src/UPGRADING>

<http://php.net/manual/>

<http://wiki.php.net/doc/scratchpad/upgrade/53>

<http://wiki.php.net/todo/php53>



Grazie!

Questions? Comments?

Lukas Kahwe Smith

[lukas@liip.ch](mailto:lukas@liip.ch) - <http://pooteweet.org>