# PSR-4 附加文档

## 1. 摘要

本文档的目的是制定一个可互操作的PHP加载器的规范， 令其可以将命名空间与文件路径相对应，且可以与其他 SPL 注册加载器共存。 这是一个 PSR-0 的补充文档，而不是替代。

## 2. 为什么要制定此文档？

### PSR-0 的历史

The PSR-0 class naming and autoloading standard rose out of the broad acceptance of the Horde/PEAR convention under the constraints of PHP 5.2 and previous. With that convention, the tendency was to put all PHP source classes in a single main directory, using underscores in the class name to indicate pseudo-namespaces, like so:

/path/to/src/  
 VendorFoo/  
 Bar/  
 Baz.php # VendorFoo\_Bar\_Baz  
 VendorDib/  
 Zim/  
 Gir.php # Vendor\_Dib\_Zim\_Gir

With the release of PHP 5.3 and the availability of namespaces proper, PSR-0 was introduced to allow both the old Horde/PEAR underscore mode *and* the use of the new namespace notation. Underscores were still allowed in the class name to ease transition from the older namespace naming to the newer naming, and thereby to encourage wider adoption.

/path/to/src/  
 VendorFoo/  
 Bar/  
 Baz.php # VendorFoo\_Bar\_Baz  
 VendorDib/  
 Zim/  
 Gir.php # VendorDib\_Zim\_Gir  
 Irk\_Operation/  
 Impending\_Doom/  
 V1.php  
 V2.php # Irk\_Operation\Impending\_Doom\V2

This structure is informed very much by the fact that the PEAR installer moved source files from PEAR packages into a single central directory.

### Along Comes Composer

With Composer, package sources are no longer copied to a single global location. They are used from their installed location and are not moved around. This means that with Composer there is no "single main directory" for PHP sources as with PEAR. Instead, there are multiple directories; each package is in a separate directory for each project.

To meet the requirements of PSR-0, this leads to Composer packages looking like this:

vendor/  
 vendor\_name/  
 package\_name/  
 src/  
 Vendor\_Name/  
 Package\_Name/  
 ClassName.php # Vendor\_Name\Package\_Name\ClassName  
 tests/  
 Vendor\_Name/  
 Package\_Name/  
 ClassNameTest.php # Vendor\_Name\Package\_Name\ClassNameTest

The "src" and "tests" directories have to include vendor and package directory names. This is an artifact of PSR-0 compliance.

Many find this structure to be deeper and more-repetitive than necessary. This proposal suggests that an additional or superseding PSR would be useful so that we can have packages that look more like the following:

vendor/  
 vendor\_name/  
 package\_name/  
 src/  
 ClassName.php # Vendor\_Name\Package\_Name\ClassName  
 tests/  
 ClassNameTest.php # Vendor\_Name\Package\_Name\ClassNameTest

This would require an implementation of what was initially called "package-oriented autoloading" (as vs the traditional "direct class-to-file autoloading").

### 基于包的自动载入

It's difficult to implement package-oriented autoloading via an extension or amendment to PSR-0, because PSR-0 does not allow for an intercessory path between any portions of the class name. This means the implementation of a package-oriented autoloader would be more complicated than PSR-0. However, it would allow for cleaner packages.

Initially, the following rules were suggested:

1. Implementors MUST use at least two namespace levels: a vendor name, and package name within that vendor. (This top-level two-name combination is hereinafter referred to as the vendor-package name or the vendor-package namespace.)
2. Implementors MUST allow a path infix between the vendor-package namespace and the remainder of the fully qualified class name.
3. The vendor-package namespace MAY map to any directory. The remaining portion of the fully-qualified class name MUST map the namespace names to identically-named directories, and MUST map the class name to an identically-named file ending in .php.

Note that this means the end of underscore-as-directory-separator in the class name. One might think underscores should be honored as they are under PSR-0, but seeing as their presence in that document is in reference to transitioning away from PHP 5.2 and previous pseudo-namespacing, it is acceptable to remove them here as well.

## 3. Scope

### 3.1 Goals

* Retain the PSR-0 rule that implementors MUST use at least two namespace levels: a vendor name, and package name within that vendor.
* Allow a path infix between the vendor-package namespace and the remainder of the fully qualified class name.
* Allow the vendor-package namespace MAY map to any directory, perhaps multiple directories.
* End the honoring of underscores in class names as directory separators

### 3.2 Non-Goals

* Provide a general transformation algorithm for non-class resources

## 4. Approaches

### 4.1 Chosen Approach

This approach retains key characteristics of PSR-0 while eliminating the deeper directory structures it requires. In addition, it specifies certain additional rules that make implementations explicitly more interoperable.

Although not related to directory mapping, the final draft also specifies how autoloaders should handle errors. Specifically, it forbids throwing exceptions or raising errors. The reason is two-fold.

1. Autoloaders in PHP are explicitly designed to be stackable so that if one autoloader cannot load a class another has a chance to do so. Having an autoloader trigger a breaking error condition violates that compatibility.
2. class\_exists() and interface\_exists() allow "not found, even after trying to autoload" as a legitimate, normal use case. An autoloader that throws exceptions renders class\_exists() unusable, which is entirely unacceptable from an interoperability standpoint. Autoloaders that wish to provide additional debugging information in a class-not-found case should do so via logging instead, either to a PSR-3 compatible logger or otherwise.

Pros:

* Shallower directory structures
* More flexible file locations
* Stops underscore in class name from being honored as directory separator
* Makes implementations more explicitly interoperable

Cons:

* It is no longer possible, as under PSR-0, to merely examine a class name to determine where it is in the file system (the "class-to-file" convention inherited from Horde/PEAR).

### 4.2 Alternative: Stay With PSR-0 Only

Staying with PSR-0 only, although reasonable, does leave us with relatively deeper directory structures.

Pros:

* No need to change anyone's habits or implementations

Cons:

* Leaves us with deeper directory structures
* Leaves us with underscores in the class name being honored as directory separators

### 4.3 Alternative: Split Up Autoloading And Transformation

Beau Simensen and others suggested that the transformation algorithm might be split out from the autoloading proposal, so that the transformation rules could be referenced by other proposals. After doing the work to separate them, followed by a poll and some discussion, the combined version (i.e., transformation rules embedded in the autoloader proposal) was revealed as the preference.

Pros:

* Transformation rules could be referenced separately by other proposal

Cons:

* Not in line with the wishes of poll respondents and some collaborators

### 4.4 Alternative: Use More Imperative And Narrative Language

After the second vote was pulled by a Sponsor after hearing from multiple +1 voters that they supported the idea but did not agree with (or understand) the wording of the proposal, there was a period during which the voted-on proposal was expanded with greater narrative and somewhat more imperative language. This approach was decried by a vocal minority of participants. After some time, Beau Simensen started an experimental revision with an eye to PSR-0; the Editor and Sponsors favored this more terse approach and shepherded the version now under consideration, written by Paul M. Jones and contributed to by many.

### Compatability Note with PHP 5.3.2 and below

PHP versions before 5.3.3 do not strip the leading namespace separator, so the responsibility to look out for this falls on the implementation. Failing to strip the leading namespace seperator could lead to unexpected behavior.

## 5. 相关人员

### 5.1 编辑

* Paul M. Jones, Solar/Aura

### 5.2 发起人

* Phil Sturgeon, PyroCMS (Coordinator)
* Larry Garfield, Drupal

### 5.3 贡献者

* Andreas Hennings
* Bernhard Schussek
* Beau Simensen
* Donald Gilbert
* Mike van Riel
* Paul Dragoonis
* 其余人员不一一细数

## 6. 投票

* **投票入口：** <https://groups.google.com/d/msg/php-fig/_LYBgfcEoFE/ZwFTvVTIl4AJ>
* **接纳的投票：**
  + 第一轮: <https://groups.google.com/forum/#!topic/php-fig/Ua46E344_Ls>, presented prior to new workflow; aborted due to accidental proposal modification
  + 第二轮: <https://groups.google.com/forum/#!topic/php-fig/NWfyAeF7Psk>, cancelled at the discretion of the sponsor <https://groups.google.com/forum/#!topic/php-fig/t4mW2TQF7iE>
  + 第三轮: 待定

## 7. 相关链接

* [Autoloader, round 4](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/lpmJcmkNYjM)
* [POLL: Autoloader: Split or Combined?](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/fGwA6XHlYhI)
* [PSR-X autoloader spec: Loopholes, ambiguities](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/kUbzJAbHxmg)
* [Autoloader: Combine Proposals?](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/422dFBGs1Yc)
* [Package-Oriented Autoloader, Round 2](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/Y4xc71Q3YEQ)
* [Autoloader: looking again at namespace](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/bnoiTxE8L28)
* [DISCUSSION: Package-Oriented Autoloader - vote against](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/SJTL1ec46II)
* [VOTE: Package-Oriented Autoloader](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/Ua46E344_Ls)
* [Proposal: Package-Oriented Autoloader](https://groups.google.com/forum/#!topicsearchin/php-fig/autoload/php-fig/qT7mEy0RIuI)
* [Towards a Package Oriented Autoloader](https://groups.google.com/forum/#!searchin/php-fig/package$20oriented$20autoloader/php-fig/JdR-g8ZxKa8/jJr80ard-ekJ)
* [List of Alternative PSR-4 Proposals](https://groups.google.com/forum/#!topic/php-fig/oXr-2TU1lQY)
* [Summary of [post-Acceptance Vote pull] PSR-4 discussions](https://groups.google.com/forum/#!searchin/php-fig/psr-4$20summary/php-fig/bSTwUX58NhE/YPcFgBjwvpEJ)