

# PHP-FIG

[Home](#)[Blog](#)[PSRs](#)[Personnel](#)[Bylaws](#)[FAQs](#)[Get Involved](#)[PSR-0: Autoloading Standard](#)

**Deprecated** - As of 2014-10-21 PSR-0 has been marked as deprecated. **PSR-4** is now recommended as an alternative.

The following describes the mandatory requirements that must be adhered to for autoloader interoperability.

## Mandatory

- A fully-qualified namespace and class must have the following structure `\<Vendor Name>\(<Namespace>\)*<Class Name>`
- Each namespace must have a top-level namespace ("Vendor Name").
- Each namespace can have as many sub-namespaces as it wishes.
- Each namespace separator is converted to a `DIRECTORY_SEPARATOR` when loading from the file system.
- Each `_` character in the CLASS NAME is converted to a `DIRECTORY_SEPARATOR`. The `_` character has no special meaning in the namespace.
- The fully-qualified namespace and class are suffixed with `.php` when loading from the file system.

- Alphabetic characters in vendor names, namespaces, and class names may be of any combination of lower case and upper case.

## Examples

- `\Doctrine\Common\IsolatedClassLoader` => `/path/to/project/lib/vendor/Doctrine/Common/IsolatedClassLoader.php`
- `\Symfony\Core\Request` => `/path/to/project/lib/vendor/Symfony/Core/Request.php`
- `\Zend\Acl` => `/path/to/project/lib/vendor/Zend/Acl.php`
- `\Zend\Mail\Message` => `/path/to/project/lib/vendor/Zend/Mail/Message.php`

## Underscores in Namespaces and Class Names

- `\namespace\package\Class_Name` => `/path/to/project/lib/vendor/namespace/package/Class/Name.php`
- `\namespace\package_name\Class_Name` => `/path/to/project/lib/vendor/namespace/package_name/Class/Name.php`

The standards we set here should be the lowest common denominator for painless autoloader interoperability. You can test that you are following these standards by utilizing this sample `SplClassLoader` implementation which is able to load PHP 5.3 classes.

## Example Implementation

Below is an example function to simply demonstrate how the above proposed standards are autoloader.

```
<?php

function autoload($className)
{
    $className = ltrim($className, '\\');
    $fileName  = '';
    $namespace = '';
    if ($lastNsPos = strrpos($className, '\\')) {
        $namespace = substr($className, 0, $lastNsPos);
        $className = substr($className, $lastNsPos + 1);
        $fileName  = str_replace('\\', DIRECTORY_SEPARATOR, $namespace) . DIRECTORY_SEPARATOR;
    }
    $fileName .= str_replace('_', DIRECTORY_SEPARATOR, $className) . '.php';

    require $fileName;
}

spl_autoload_register('autoload');
```

## SplClassLoader Implementation

The following gist is a sample SplClassLoader implementation that can load your classes if you follow the autoloader interoperability standards proposed above. It is the current recommended way to load PHP 5.3 classes that follow these standards.

- <http://gist.github.com/221634>

## Additional Info:

---

PSR-0: Autoloading Standard

[Follow us on Twitter](#)

[Chat on IRC Channel](#)

[Contribute via GitHub](#)

[Join our mailing list](#)

© 2020 PHP Framework Interop Group. Site design by Jonathan Reinink. Hosting sponsored by the [Platform.sh](#) PHP PaaS