

# **1 Apache::porting -- a helper module for mod\_perl 1.0 to mod\_perl 2.0 porting**

## 1.1 Synopsis

```
# either add at the very beginning of startup.pl
use Apache2;
use Apache::porting;

# or httpd.conf
PerlModule Apache2
PerlModule Apache::porting

# now issue requests and look at the error_log file for hints
```

## 1.2 Description

`Apache::porting` helps to port `mod_perl 1.0` code to run under `mod_perl 2.0`. It doesn't provide any back-compatibility functionality, however it knows trap calls to methods that are no longer in the `mod_perl 2.0` API and tell what should be used instead if at all. If you attempts to use `mod_perl 2.0` methods without first loading the modules that contain them, it will tell you which modules you need to load. Finally if your code tries to load modules that no longer exist in `mod_perl 2.0` it'll also tell you what are the modules that should be used instead.

`Apache::porting` communicates with users via the *error\_log* file. Everytime it traps a problem, it logs the solution (if it finds one) to the error log file. If you use this module coupled with `Apache::Reload` you will be able to port your applications quickly without needing to restart the server on every modification.

It starts to work only when child process start and doesn't work for the code that gets loaded at the server startup. This limitation is explained in the Culprits section.

It relies heavily on `ModPerl::MethodLookup`, which can also be used manually to lookup things.

## 1.3 Culprits

`Apache::porting` uses the `UNIVERSAL::AUTOLOAD` function to provide its functionality. However it seems to be impossible to create `UNIVERSAL::AUTOLOAD` at the server startup, Apache segfaults on restart. Therefore it performs the setting of `UNIVERSAL::AUTOLOAD` only during the *child\_init* phase, when child processes start. As a result it can't help you with things that get preloaded at the server startup.

If you know how to resolve this problem, please let us know. To reproduce the problem try to use an earlier phase, e.g. `PerlPostConfigHandler`:

```
Apache->server->push_handlers(PerlPostConfigHandler => \&porting_autoload);
```

## Table of Contents:

1	Apache::porting -- a helper module for mod_perl 1.0 to mod_perl 2.0 porting	1
1.1	Synopsis	2
1.2	Description	2
1.3	Culprits	2