1   Apache::compat -- 1.0 backward compatibility functions deprecated in 2.0

1.1 SYNOPSIS

```
# either add at the very beginning of startup.pl
use Apache2
use Apache::compat;
# or httpd.conf
PerlModule Apache2
PerlModule Apache::compat
```

1.2 DESCRIPTION

`Apache::compat` provides mod_perl 1.0 compatibility layer and can be used to smooth the transition process to mod_perl 2.0.

It includes functions that have changed their API or were removed in mod_perl 2.0. If your code uses any of those functions, you should load this module at the server startup, and everything should work as it did in 1.0. If it doesn't please report the bug, but before you do that please make sure that your code does work properly under mod_perl 1.0.

However, remember, that it's implemented in pure Perl and not C, therefore its functionality is not optimized and it's the best to try to port your code not to use deprecated functions and stop using the compatibility layer.

1.3 Use in CPAN Modules

The short answer: **Do not use** `Apache::compat` in CPAN modules.

The long answer:

`Apache::compat` is useful during the mod_perl 1.0 code porting. Though remember that it's implemented in pure Perl. In certain cases it overrides mod_perl 2.0 methods, because their API is very different and doesn't map 1:1 to mod_perl 1.0. So if anything, not under user's control, loads `Apache::compat` user's code is forced to use the potentially slower method. Which is quite bad.

Some users may choose to keep using `Apache::compat` in production and it may perform just fine. Other users will choose not to use that module, by porting their code to use mod_perl 2.0 API. However it should be users' choice whether to load this module or not and not to be enforced by CPAN modules.

If you port your CPAN modules to work with mod_perl 2.0, you should follow the porting Perl and XS module guidelines.

Users that are stuck with CPAN modules preloading `Apache::compat`, can prevent this from happening by adding

```
$INC{'Apache/compat.pm'} = __FILE__;
```

at the very beginning of their *startup.pl*. But this will most certainly break the module that needed this module.

1.4 API

You should be reading the mod_perl 1.0 API docs for usage of the methods and functions in this package, since what this module is doing is providing a backwards compatibility and it makes no sense to duplicate documentation.

Another important document to read is: Migrating from mod_perl 1.0 to mod_perl 2.0 which covers all mod_perl 1.0 constants, functions and methods that have changed in mod_perl 2.0.

Table of Contents: