

1 Porting Apache::XS Modules from mod_perl 1.0 to 2.0

1.1 Description

This document talks mainly about porting modules using XS code. It's also helpful to those who start developing mod_perl 2.0 packages.

Also make sure to first read about porting Apache:: Perl modules.

1.2 Porting Makefile.PL

It's only an issue if it was using Apache::src. A new configuration system is in works. So watch this space for updates on this issue.

ModPerl::MM is the new replacement of Apache::src.

1.3 Porting XS Code

If your module's XS code relies on the Apache and mod_perl C APIs, it's very likely that you will have to adjust the XS code to the Apache 2.0 and mod_perl 2.0 C API.

The C API has changed a lot, so chances are that you are much better off not to mix the two APIs in the same XS file. However if you do want to mix the two you will have to use something like the following:

```
#include ap_mmn.h
/* ... */
#if AP_MODULE_MAGIC_AT_LEAST(20020903,3)
    /* 2.0 code */
#else
    /* 1.0 code */
#endif
```

The 20020903, 3 is the value of the magic version number matching Apache 2.0.46, the earliest Apache version supported by mod_perl 2.0.

1.4 Thread Safety

META: to be written

```
#ifdef MP_THREADED
    /* threads specific code goes here */
#endif
```

For now see: http://httpd.apache.org/docs-2.0/developer/thread_safety.html

1.5 PerlIO

PerlIO layer has become usable only in perl 5.8.0, so if you plan on working with PerlIO, you can use the PERLIO_LAYERS constant. e.g.:

```
#ifdef PERLIO_LAYERS
#include "perliol.h"
#else
#include "iperlsys.h"
#endif
```

1.6 'make test' Suite

The `Apache::Test` testing framework that comes together with `mod_perl 2.0` works with 1.0 and 2.0 `mod_perl` versions. Therefore you should consider porting your test suite to use the `Apache::Test` Framework.

1.7 Apache C Code Specific Notes

Most of the documentation covering migration to Apache 2.0 can be found at: <http://httpd.apache.org/docs-2.0/developer/>

The Apache 2.0 API documentation now resides in the C header files, which can be conveniently browsed via <http://docx.webperf.org/>.

The APR API documentation can be found here <http://apr.apache.org/>.

The new Apache and APR APIs include many new functions. Though certain functions have been preserved, either as is or with a changed prototype (for example to work with pools), others have been renamed. So if you are porting your code and the function that you've used doesn't seem to exist in Apache 2.0, first refer to the "compat" header files, such as: *include/ap_compat.h*, *srclib/apr/include/apr_compat.h*, and *srclib/apr-util/include/apu_compat.h*, which list functions whose names have changed but which are otherwise the same. If this fails, proceed to look in other headers files in the following directories:

- *ap_* functions in *include/*
- *apr_* functions in *srclib/apr/include/* and *srclib/apr-util/include/*

1.7.1 *ap_soft_timeout()*, *ap_reset_timeout()*, *ap_hard_timeout()* and *ap_kill_timeout()*

If the C part of the module in 1.0 includes *ap_soft_timeout()*, *ap_reset_timeout()*, *ap_hard_timeout()* and *ap_kill_timeout()* functions simply remove these in 2.0. There is no replacement for these functions because Apache 2.0 uses non-blocking I/O. As a side-effect of this change, Apache 2.0 no longer uses SIGALRM, which has caused conflicts in `mod_perl 1.0`.

1.8 Maintainers

Maintainer is the person(s) you should contact with updates, corrections and patches.

Stas Bekman <stas (at) stason.org>

1.9 Authors

- Stas Bekman <stas (at) stason.org>
- Doug MacEachern <dougm (at) covalent.net>

Only the major authors are listed above. For contributors see the Changes file.

Table of Contents:

1	Porting Apache:: XS Modules from mod_perl 1.0 to 2.0	1
1.1	Description	2
1.2	Porting Makefile.PL	2
1.3	Porting XS Code	2
1.4	Thread Safety	2
1.5	PerlIO	3
1.6	'make test' Suite	3
1.7	Apache C Code Specific Notes	3
1.7.1	ap_soft_timeout(), ap_reset_timeout(), ap_hard_timeout() and ap_kill_timeout()	3
1.8	Maintainers	4
1.9	Authors	4