mod_perl 2.0 Documentation

A collection of the documents specific to the mod_perl 2.0
generation.

Last modified Sun Feb 16 01:36:39 2014 GMT

FYI: Porting CPAN modules to mod_perl 2.0 Status

- 1. mod_perl 2.0 renaming
    This document discusses the effects of the decision to rename parts of the mod_perl API.

- 2. User's guide
    All you need to know about using mod_perl 2.0

- 3. OS-specific Info
    Documents concerning OS-specific installation, configuration, pitfalls, etc.

- 4. mod_perl 2.0 API
    The Apache::, APR:: and ModPerl:: namespaces APIs for mod_perl 2.0

- 5. Developer's guide
    This guide is aimed for mod_perl 2.0 core and 3rd party modules developers.

1   mod_perl 2.0 renaming

1.1 Description

This document discusses the effects of the decision to rename parts of the mod_perl API.

1.2 Synopsis

During a sequence of release candidates leading up to the official release of mod_perl 2, in late December, 2004 there was considerable discussion regarding the choice of namespaces for the 2.0 API. The main concern expressed was that some of the `Apache::*` modules used in mod_perl 2 had the same names as those in mod_perl 1. After a lively and, at times, heated debate, a proposal was made to rename parts of the mod_perl 2 API - see the thread at http://marc.theaimsgroup.com/?t=111135037100002&r=1&w=2. This proposal was subsequently voted on and then adopted.

1.3 Changes

The main changes involved in this renaming are as follows.

- Rename all `Apache::*` modules to `Apache2::*`, save for the following exceptions:

  - The `Apache::Test*` modules in the `Apache-Test` distribution are unaffected by this rename.

  - Constants under the `Apache::*` namespace, such as `Apache::OK`, are now called as `Apache2::Const::*` (for example, `Apache2::Const::OK`).

  - Constants under the `APR::*` namespace, such as `APR::SUCCESS`, are now called as `APR::Const::*` (for example, `APR::Const::SUCCESS`).

- `Apache->request()` has been renamed to `Apache2::RequestUtil->request()`.

- `Apache->server()` has been renamed to `Apache2::ServerUtil->server()`.

- Rename `mod_perl.pm` to `mod_perl2.pm`

- Discontinue the practice of installing mod_perl in a relative *Apache2* subdirectory, so that the use of `Apache2.pm` to adjust `@INC` is no longer needed.

Concurrent with these changes, a decision was also made to refuse to install mod_perl 2 in the same Perl tree as any installed mod_perl 2 package of version prior to 1.999_22.

1.4 Effects

The mod_perl developers did not take lightly the effects of changing the API at such a late stage in the development cycle. For most users the change to the new API should be relatively straightforward:

- Before installing mod_perl 2, either remove an earlier mod_perl 2 installation (prior to 1.999022), or else choose to install the new version in a different area. To remove enough of a prior installation of mod_perl 2 in order to satisfy this requirement, one can do, on Unix,

  ```
  % find /usr/lib/perl5 -name 'Apache2*' -exec rm -rf {} \;
  ```

  where */usr/lib/perl5* is the top-level root of your Perl tree. Note that this will also remove any 3rd party Apache modules installed under an *Apache2* subdirectory. In many situations you would want to do this, as such modules will need porting to the new API. However, if there are modules under this location that you want to keep, you can use

  ```
  % find /usr/lib/perl5 -name 'Apache2*' -exec ls -lR {} \;
  ```

  to list the files under this tree.

  For Win32 ppm users, the command

  ```
  ppm> remove mod_perl
  ```

  within the ppm shell will remove an installed `mod_perl` ppm package.

- If building mod_perl 2 from source, the `Makefile.PL` argument of `MP_INST_APACHE2` is no longer supported.

- Replace the use of

  ```
  use Apache::SomeCoreModule;
  ```

  in a script or module by

  ```
  use Apache2::SomeCoreModule;
  ```

  Similarly, an Apache directive

  ```
  PerlModule Apache::SomeCoreModule
  ```

  in an *httpd.conf* file should be replaced by

  ```
  PerlModule Apache2::SomeCoreModule
  ```

  and `mod_perl` should be replaced by `mod_perl2` in the same places.

  Note that "core modules" are defined as those that ship with the mod_perl 2 distribution. Other CPAN modules in the `Apache::` namespace may not follow the same conventions.

- Replace the use of constants `Apache::*` by `Apache2::Const::*` and `APR::*` by `APR::Const::*`.

- Remove any occurrences of

```
use Apache2;
```

in a script or module, as well as

```
PerlModule Apache2
```

in an Apache configuration file.

- In a script or module, replace `Apache->request()` by `Apache2::RequestUtil->request()` and `Apache->server()` by `Apache2::ServerUtil->server()`.

- If using `CGI.pm` you will need to upgrade to version 3.11 - versions 3.10 and older contain bugs and versions 3.07 and older do not support support the new mod_perl API. you can get `CGI.pm` from http://search.cpan.org/dist/CGI.pm/.

Beginning with mod_perl-1.999_22, an environment variable $ENV{MOD_PERL_API_VERSION} has been added to specify which mod_perl version is being used.

1.5 See Also

The directions for installing mod_perl 2.0 . Help is also available through the archives of and subscribing to the mod_perl mailing list.

1.6 Authors

The mod_perl development team and numerous contributors.

Table of Contents: