

1 Getting Your Feet Wet with mod_perl

1.1 Description

This chapter gives you the bare minimum information to get you started with mod_perl 2.0. For most people it's sufficient to get going.

1.2 Installation

If you are a Win32 user, please refer to the Win32 installation document.

First, download the mod_perl 2.0 source.

Before installing mod_perl, you need check that you have the mod_perl 2.0 prerequisites **installed**. Apache and the right Perl version have to be built and installed **before** you can proceed with building mod_perl.

In this chapter we assume that httpd and all helper files were installed under *\$HOME/httpd/prefork*, if your distribution doesn't install all the files under the same tree, please refer to the complete installation instructions.

Now, configure mod_perl:

```
% tar -xvzf mod_perl-2.x.xx.tar.gz
% cd modperl-2.0
% perl Makefile.PL MP_APXS=$HOME/httpd/prefork/bin/apxs MP_INST_APACHE2=1
```

where MP_APXS is the full path to the apxs executable, normally found in the same directory as the httpd executable, but could be put in a different path as well.

Finally, build, test and install mod_perl:

```
% make && make test && make install
```

Become *root* before doing `make install` if installing system-wide.

If something goes wrong or you need to enable optional features please refer to the complete installation instructions.

1.3 Configuration

If you are a Win32 user, please refer to the Win32 configuration document.

Enable mod_perl built as DSO, by adding to *httpd.conf*:

```
LoadModule perl_module modules/mod_perl.so
```

Next, tell Perl where to find mod_perl2 libraries:

```
PerlModule Apache2
```

There are many other configuration options which you can find in the configuration manual.

If you want to run mod_perl 1.0 code on mod_perl 2.0 server enable the compatibility layer:

```
PerlModule Apache::compat
```

For more information see: Migrating from mod_perl 1.0 to mod_perl 2.0.

1.4 Server Launch and Shutdown

Apache is normally launched with `apachectl`:

```
% $HOME/httpd/prefork/bin/apachectl start
```

and shut down with:

```
% $HOME/httpd/prefork/bin/apachectl stop
```

Check `$HOME/httpd/prefork/logs/error_log` to see that the server has started and it's a right one. It should say something similar to:

```
[Thu May 29 12:22:12 2003] [notice] Apache/2.0.46-dev (Unix)
mod_perl/1.99_10-dev Perl/v5.9.0 mod_ssl/2.0.46-dev OpenSSL/0.9.7
DAV/2 configured -- resuming normal operations
```

1.5 Registry Scripts

To enable registry scripts add to `httpd.conf`:

```
Alias /perl/ /home/httpd/httpd-2.0/perl/
<Location /perl/>
    SetHandler perl-script
    PerlResponseHandler ModPerl::Registry
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Location>
```

and now assuming that we have the following script:

```
#!/usr/bin/perl
print "Content-type: text/plain\n\n";
print "mod_perl 2.0 rocks!\n";
```

saved in `/home/httpd/httpd-2.0/perl/rock.pl`. Make the script executable and readable by everybody:

```
% chmod a+rx /home/httpd/httpd-2.0/perl/rock.pl
```

Of course the path to the script should be readable by the server too. In the real world you probably want to have a tighter permissions, but for the purpose of testing that things are working this is just fine.

Now restart the server and issue a request to *http://localhost/perl/rock.pl* and you should get the response:

```
mod_perl 2.0 rocks!
```

If that didn't work check the *error_log* file.

For more information on the registry scripts refer to the `ModPerl::Registry` manpage. (XXX: on day there will a tutorial on registry, should port it from 1.0's docs).

1.6 Handler Modules

Finally check that you can run `mod_perl` handlers. Let's write a response handler similar to the registry script from the previous section:

```
#file:MyApache/Rocks.pm
#-----
package MyApache::Rocks;

use strict;
use warnings;

use Apache::RequestRec ();
use Apache::RequestIO ();

use Apache::Const -compile => qw(OK);

sub handler {
    my $r = shift;

    $r->content_type('text/plain');
    print "mod_perl 2.0 rocks!\n";

    return Apache::OK;
}
1;
```

Save the code in the file *MyApache/Rocks.pm*, somewhere where `mod_perl` can find it. For example let's put it under */home/httpd/httpd-2.0/perl/MyApache/Rocks.pm*, and we tell `mod_perl` that */home/httpd/httpd-2.0/perl/* is in `@INC`, via a startup file which includes just:

```
use lib qw(/home/httpd/httpd-2.0/perl);
1;
```

and loaded from *httpd.conf*:

```
PerlRequire /home/httpd/httpd-2.0/perl/startup.pl
```

Now we can configure our module in *httpd.conf*:

```
<Location /rocks>
    SetHandler perl-script
    PerlResponseHandler MyApache::Rocks
</Location>
```

Now restart the server and issue a request to *http://localhost/rocks* and you should get the response:

```
mod_perl 2.0 rocks!
```

If that didn't work check the *error_log* file.

1.7 Troubleshooting

If after reading the complete installation and configuration chapters you are still having problems, take a look at the troubleshooting sections. If the problem persists, please report them using the following guidelines.

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>

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

Table of Contents:

1	Getting Your Feet Wet with mod_perl	1
1.1	Description	2
1.2	Installation	2
1.3	Configuration	2
1.4	Server Launch and Shutdown	3
1.5	Registry Scripts	3
1.6	Handler Modules	4
1.7	Troubleshooting	5
1.8	Maintainers	5
1.9	Authors	5