

# 1 User Help

## 1.1 Description

This chapter is for those needing help using mod\_perl and related software.

There is a parallel Getting Help document written mainly for mod\_perl core developers, but may be found useful to non-core problems as well.

## 1.2 Reporting Problems

Whenever you want to report a bug or a problem remember that in order to help you, you need to provide us the information about the software that you are using and other relevant details. Please follow the instructions in the following sections when reporting problems.

**The most important thing** to understand is that you should try hard to provide **all** the information that may assist to understand and reproduce the problem. When you prepare a bug report, put yourself in the position of a person who is going to try to help you, realizing that a guess-work on behalf of that helpful person, more often doesn't work than it does. Unfortunately most people don't realize that, and it takes several emails to squeeze the needed details from the person reporting the bug, a process which may drag for days.

### *1.2.1 Wrong Apache/mod\_perl combination*

First of all:

```
Apache 2.0 doesn't work with mod_perl 1.0.  
Apache 1.0 doesn't work with mod_perl 2.0.
```

So if you aren't using Apache 2.x with mod\_perl 2.0 please do not send any bug reports.

META: mod\_perl-1.99\_xx is mod\_perl 2.0 to-be.

### *1.2.2 Before Posting a Report*

Before you post the report, make sure that you've checked the *error\_log* file (t/logs/error\_log in case of the failing test suite). Usually the errors are self-descriptive and if you remember to always check this file whenever you have a problem, chances are that you won't need to ask for help.

### *1.2.3 Test with the Latest mod\_perl 2.0 Version*

If you are using an older version than the most recently released one, chances are that a bug that you are about to report has already been fixed. If possible, save us and yourself time and try first to upgrade to the latest version, and only if the bug persists report it.

Reviewing the Changes file may help as well. Here is the Changes file of the most recently released version: [http://perl.apache.org/dist/mod\\_perl-2.0-current/Changes](http://perl.apache.org/dist/mod_perl-2.0-current/Changes) .

If the problem persists with the latest version, you may also want to try to reproduce the problem with the latest development version. It's possible that the problem was resolved since the last release has been made. Of course if this version solves the problem, don't rush to put it in production unless you know what you are doing. Instead ask the developers when the new version will be released.

### ***1.2.4 Use a Proper Subject***

Make sure to include a good subject like explaining the problem in a few words. Also please mention that this a problem with mod\_perl 2.0 and not mod\_perl 1.0. Here is an example of a good subject:

```
Subject: [mp2] protocol module doesn't work with filters
```

This is especially important now that we support mod\_perl versions 1.0 and 2.0 on the same list.

### ***1.2.5 Send the Report Inlined***

When sending the bug report, please inline it and don't attach it to the email. It's hard following up on the attachments.

### ***1.2.6 Important Information***

Whenever you send a bug report make sure to include the information about your system by doing the following:

```
% cd modperl-2.0
% t/REPORT > mybugreport
```

where modperl-2.0 is the source directory where mod\_perl was built. The t/REPORT utility is auto-generated when perl Makefile.PL is run, so you should have it already after building mod\_perl.

META: soon we will have mp2bug report script which will be installed system-wide. For now, if you don't have the source, you can create the report by running the following:

```
% perl -MApache2 -MApache::TestReportPerl \
-le 'Apache::TestReportPerl->new->run'
```

Now add the problem description to the report and send it to the list.

### ***1.2.7 Problem Description***

If the problem incurs with your own code, please try to reduce the code to the very minimum and include it in the bug report. Remember that if you include a long code, chances that somebody will look at it are low. If the problem is with some CPAN module, just provide its name.

Also remember to include the relevant part of *httpd.conf* and of *startup.pl* if applicable. Don't include whole files, only the parts that should aid to understand and reproduce the problem.

Finally don't forget to copy-n-paste (not type!) the **relevant** part of the *error\_log* file (not the whole file!).

To further increase the chances that bugs your code exposes will be investigated, try using *Apache-Test* to create a self-contained test that core developers can easily run. To get you started, an *Apache-Test* bug skeleton has been created:

<http://perl.apache.org/~geoff/bug-reporting-skeleton-mp2.tar.gz>

Detailed instructions are contained within the README.

## 1.2.8 'make test' Failures

If when running 'make test' some of the tests fail, please re-run them in the verbose mode and post the output of the run and the contents of the *error\_log* file to the list.

For example if 'make test' reports:

```
Failed Test          Stat Wstat Total Fail  Failed  List of Failed
-----
compat/apache_util.t          15     1    6.67%  13
modperl/pnotes.t             5     1    20%    2
```

Do the following:

```
% cd modperl-1.99_xx
% make test TEST_VERBOSE=1 \
  TEST_FILES="compat/apache_util.t modperl/pnotes.t"
```

or use an alternative way:

```
% cd modperl-1.99_xx
% rm t/logs/error_log
% t/TEST -verbose compat/apache_util.t modperl/pnotes.t
```

If you are using the latter, remember to remove the *error\_log* file before running tests, so you won't have clutter from the previous run. `make test` always removes the old *error\_log* file for you.

Also please notice that there is more than one `make test` run. The first one is running at the top directory, the second inside a sub-directory *ModPerl-Registry/*. The first logs errors to *t/logs/error\_log*, the second to *ModPerl-Registry/t/logs/error\_log*. Therefore if you get failures in the second run, make sure to `chdir()` to that directory before you look at the *t/logs/error\_log* file and re-run tests in the verbose mode. For example:

```
% cd modperl-1.99_xx/ModPerl-Registry
% rm t/logs/error_log
% t/TEST -verbose closure.t
```

## 1.2.9 Resolving Segmentation Faults

If during `make test` or the use of `mod_perl` you get a segmentation fault you should send to the list a stack backtrace. This section explains how to extract this backtrace.

Of course to generate a useful backtrace you need to have `mod_perl` with debugging symbols in it (and probably `perl` and/or `httpd` too) and also to be able to see the arguments in the calls trace. To accomplish that do:

- **mod\_perl**

rebuild `mod_perl` with `MP_DEBUG=1`.

```
% perl Makefile.PL MP_DEBUG=1 ...
% make && make test && make install
```

- **httpd**

If the segfault happens inside `ap_` or `apr_` calls, rebuild `httpd` with `--enable-maintainer-mode`:

```
% ./configure --enable-maintainer-mode ...
% make && make install
```

- **perl**

If the segfault happens inside `Perl_` calls, rebuild `perl` with `-Doptimize='-g'`:

```
% ./Configure -Doptimize='-g' ...
% make && make test && make install
```

- **3rd party perl modules**

if the trace happens in one of the 3rd party perl modules, make sure to rebuild them, now that you've perl re-built with debugging flags. They will automatically pick the right compile flags from perl.

Once a proper stack backtrace is obtained append it to the bug report as explained in the previous section.

## 1.2.10 Please Ask Only Questions Related to mod\_perl

If you have general Apache questions, please refer to: <http://httpd.apache.org/lists.html>.

If you have general Perl questions, please refer to: <http://lists.perl.org/>.

For other remotely related to `mod_perl` questions see the references to other documentation.

Finally, if you are not familiar with the `modperl` list etiquette, please refer to the `mod_perl` mailing lists' Guidelines before posting.

## 1.3 Help on Related Topics

When developing with mod\_perl, you often find yourself having questions regarding other projects and topics like Apache, Perl, SQL, etc. This document will help you find the right resource where you can find the answers to your questions.

## 1.4 Maintainers

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

- **Stas Bekman**

## 1.5 Authors

- **Stas Bekman**

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

## Table of Contents:

1	User Help . . . . .	1
1.1	Description . . . . .	2
1.2	Reporting Problems . . . . .	2
1.2.1	Wrong Apache/mod_perl combination . . . . .	2
1.2.2	Before Posting a Report . . . . .	2
1.2.3	Test with the Latest mod_perl 2.0 Version . . . . .	2
1.2.4	Use a Proper Subject . . . . .	3
1.2.5	Send the Report Inlined . . . . .	3
1.2.6	Important Information . . . . .	3
1.2.7	Problem Description . . . . .	3
1.2.8	'make test' Failures . . . . .	4
1.2.9	Resolving Segmentation Faults . . . . .	5
1.2.10	Please Ask Only Questions Related to mod_perl . . . . .	5
1.3	Help on Related Topics . . . . .	6
1.4	Maintainers . . . . .	6
1.5	Authors . . . . .	6