

# 1 Debugging mod\_perl Perl Internals

## 1.1 Description

This document explains how to debug Perl code under `mod_perl`.

Most of the `mod_perl` 1.0 debug documentation applies to `mod_perl` 2.0:

### 1.1.1 *Detecting Hanging Processes*

See *Hanging Processes: Detection and Diagnostics* for the explanation, but under `mp2` to use signals to detect where the process is spinning, you can't use `$SIG{USR2}`, you have to use POSIX signals. i.e. the code becomes:

```
use Carp ();
use POSIX qw(SIGUSR2);
my $mask      = POSIX::SigSet->new( SIGUSR2 );
my $action    = POSIX::SigAction->new(\&tell_where_spinning, $mask);
my $oldaction = POSIX::SigAction->new();
POSIX::sigaction(SIGUSR2, $action, $oldaction );

sub tell_where_spinning {
    Carp::confess("caught SIGUSR2!");
};
```

and then:

```
% kill USR2 <pid_of_the_spinning_process>
```

and watch for the trace in *error\_log*.

## 1.2 Maintainers

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

- Stas Bekman [<http://stason.org/>]

## 1.3 Authors

- Stas Bekman [<http://stason.org/>]

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

## Table of Contents:

1	Debugging mod_perl Perl Internals . . . . .	1
1.1	Description . . . . .	2
1.1.1	Detecting Hanging Processes . . . . .	2
1.2	Maintainers . . . . .	2
1.3	Authors . . . . .	2