

1 Apache::SIG - Override apache signal handlers with Perl's

1.1 Synopsis

```
PerlFixupHandler Apache::SIG
```

1.2 Description

When a client drops a connection and apache is in the middle of a write, a timeout will occur and httpd sends a SIGPIPE. When apache's SIGPIPE handler is used, Perl may be left in the middle of it's eval context, causing bizarre errors during subsequent requests are handled by that child. When `Apache::SIG` is used, it installs a different SIGPIPE handler which rewinds the context to make sure Perl is back to normal state, preventing these bizarre errors.

If you would like to log when a request was cancelled by a SIGPIPE in your Apache *access_log*, you can declare `Apache::SIG` as a handler (any `Perl*Handler` will do, as long as it is run before `PerlHandler`, e.g. `PerlFixupHandler`), and you must also define a custom `LogFormat` in your `httpd.conf`, like so:

```
PerlFixupHandler Apache::SIG
LogFormat "%h %l %u %t \"%r\" %s %b %{SIGPIPE}e"
```

If the server has noticed that the request was cancelled via a SIGPIPE, then the log line will end with 1, otherwise it will just be a dash.

1.3 Caveats

The signal handler in this package uses the `subprocess_env` table of the main request object to supply the SIGPIPE "environment variable" to the log handler. If you already use the key `SIGPIPE` in your `subprocess_env` table, then you can redefine the key like this:

```
$Apache::SIG::PipeKey = 'my_SIGPIPE';
```

and log it like this:

```
LogFormat "%h %l %u %t \"%r\" %s %b %{my_SIGPIPE}e"
```

1.4 Maintainers

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

- **The documentation mailing list**

1.5 Authors

- **Doug MacEachern**
- **Doug Bagley**

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

1.6 See Also

perlvar(1)

Table of Contents:

1	Apache::SIG - Override apache signal handlers with Perl's	1
1.1	Synopsis	2
1.2	Description	2
1.3	Caveats	2
1.4	Maintainers	2
1.5	Authors	2
1.6	See Also	3