

# **1 Apache2::Status - Embedded interpreter status information**

## 1.1 Synopsis

```
<Location /perl-status>

    # disallow public access
    Order Deny, Allow
    Deny from all
    Allow from 127.0.0.1

    SetHandler modperl
    PerlOptions +GlobalRequest
    PerlResponseHandler Apache2::Status
</Location>
```

or

```
<Location /perl-status>

    # disallow public access
    Order Deny, Allow
    Deny from all
    Allow from 127.0.0.1

    SetHandler perl-script
    PerlResponseHandler Apache2::Status
</Location>
```

## 1.2 Description

The `Apache2::Status` module provides some information about the status of the Perl interpreter embedded in the server.

Configure like so:

```
<Location /perl-status>

    # disallow public access
    Order Deny, Allow
    Deny from all
    Allow from 127.0.0.1

    SetHandler modperl
    PerlOptions +GlobalRequest
    PerlResponseHandler Apache2::Status
</Location>
```

Notice that under the "modperl" core handler the *Environment* menu option will show only the environment under that handler. To see the environment seen by handlers running under the "perl-script" core handler, configure `Apache2::Status` as:

```
<Location /perl-status>

    # disallow public access
    Order Deny, Allow
    Deny from all
    Allow from 127.0.0.1

    SetHandler perl-script
    PerlResponseHandler Apache2::Status
</Location>
```

Other modules can "plugin" a menu item like so:

```
require Apache2::Module;
Apache2::Status->menu_item(
    'DBI' => "DBI connections", #item for Apache::DBI module
    sub {
        my ($r, $q) = @_; #request and CGI objects
        my (@strings);
        push @strings, "blobs of html";
        return \@strings;      #return an array ref
    }
) if Apache2::Module::loaded('Apache2::Status');
```

**WARNING:** Apache2::Status must be loaded before these modules via the PerlModule or PerlRequire directives (or from *startup.pl*).

A very common setup might be: Perl Module B::TerseSize

```
<Location /perl-status>
    SetHandler perl-script
    PerlResponseHandler Apache2::Status
    PerlSetVar StatusOptionsAll On
    PerlSetVar StatusDeparseOptions "-p -sC"
</Location>
```

due to the implementation of Apache2::Status::noh\_fileline in B::TerseSize, you must load B::TerseSize first.

## 1.3 Options

### 1.3.1 StatusOptionsAll

This single directive will enable all of the options described below.

```
PerlSetVar StatusOptionsAll On
```

## ***1.3.2 StatusDumper***

When browsing symbol tables, the values of arrays, hashes and scalars can be viewed via `Data::Dumper` if this configuration variable is set to On:

```
PerlSetVar StatusDumper On
```

## ***1.3.3 StatusPeek***

With this option On and the `Apache::Peek` module installed, functions and variables can be viewed ala `Devel::Peek` style:

```
PerlSetVar StatusPeek On
```

## ***1.3.4 StatusLexInfo***

With this option On and the `B::LexInfo` module installed, subroutine lexical variable information can be viewed.

```
PerlSetVar StatusLexInfo On
```

## ***1.3.5 StatusDeparse***

With this option On subroutines can be "deparsed".

```
PerlSetVar StatusDeparse On
```

Options can be passed to `B::Deparse::new` like so:

```
PerlSetVar StatusDeparseOptions "-p -sC"
```

See the `B::Deparse` manpage for details.

## ***1.3.6 StatusTerse***

With this option On, text-based op tree graphs of subroutines can be displayed, thanks to `B::Terse`.

```
PerlSetVar StatusTerse On
```

## ***1.3.7 StatusTerseSize***

With this option On and the `B::TerseSize` module installed, text-based op tree graphs of subroutines and their size can be displayed. See the `B::TerseSize` docs for more info.

```
PerlSetVar StatusTerseSize On
```

### 1.3.8 *StatusTerseSizeMainSummary*

With this option On and the `B::TerseSize` module installed, a *"Memory Usage"* will be added to the `Apache2::Status` main menu. This option is disabled by default, as it can be rather cpu intensive to summarize memory usage for the entire server. It is strongly suggested that this option only be used with a development server running in `-X` mode, as the results will be cached.

```
PerlSetVar StatusTerseSizeMainSummary On
```

### 1.3.9 *StatusGraph*

When `StatusDumper` is enabled, another link *"OP Tree Graph"* will be present with the dump if this configuration variable is set to On:

```
PerlSetVar StatusGraph
```

This requires the `B` module (part of the Perl compiler kit) and `B::Graph` (version 0.03 or higher) module to be installed along with the `dot` program.

`Dot` is part of the graph visualization toolkit from AT&T: <http://www.graphviz.org/>.

**WARNING:** Some graphs may produce very large images, some graphs may produce no image if `B::Graph`'s output is incorrect.

### 1.3.10 *Dot*

Location of the `dot` program for `StatusGraph`, if other than `/usr/bin` or `/usr/local/bin`

### 1.3.11 *GraphDir*

Directory where `StatusGraph` should write it's temporary image files. Default is `$Server-Root/logs/b_graphs`.

## 1.4 Prerequisites

The `Devel::Symdump` module, version 2.00 or higher.

Other optional functionality requirements: `B::Deparse` - 0.59, `B::Fathom` - 0.05, `B::Graph` - 0.03.

## 1.5 Copyright

`mod_perl` 2.0 and its core modules are copyrighted under The Apache Software License, Version 2.0.

## 1.6 See Also

perl(1), Apache(3), Devel::Symdump(3), Data::Dumper(3), B(3), B::Graph(3), mod\_perl 2.0 documentation.

## 1.7 Authors

Doug MacEachern with contributions from Stas Bekman

# Table of Contents:

1	Apache2::Status - Embedded interpreter status information	1
1.1	Synopsis	2
1.2	Description	2
1.3	Options	3
1.3.1	StatusOptionsAll	3
1.3.2	StatusDumper	4
1.3.3	StatusPeek	4
1.3.4	StatusLexInfo	4
1.3.5	StatusDeparse	4
1.3.6	StatusTerse	4
1.3.7	StatusTerseSize	4
1.3.8	StatusTerseSizeMainSummary	5
1.3.9	StatusGraph	5
1.3.10	Dot	5
1.3.11	GraphDir	5
1.4	Prerequisites	5
1.5	Copyright	5
1.6	See Also	6
1.7	Authors	6