

# **1 Apache::Module - Perl API for creating and working with Apache modules**

## 1.1 Synopsis

```
use Apache::Module ();
```

META: to be completed

## 1.2 Description

META: to be completed

See Apache Server Configuration Customization in Perl.

## 1.3 API

Apache::Module provides the following functions and/or methods:

### *1.3.1 find\_linked\_module*

META: Autogenerated - needs to be reviewed/completed

Find a module based on the name of the module

```
$ret = find_linked_module($name);
```

- **arg1: \$name (string)**

the name of the module

- **ret: \$ret (Apache::Module)**

the module structure if found, NULL otherwise

### *1.3.2 find\_module\_name*

META: Autogenerated - needs to be reviewed/completed

Find the name of the specified module

```
$ret = $module->find_module_name();
```

- **arg1: \$module (Apache::Module)**

The module to get the name for

- **ret: \$ret (string)**

the name of the module

### ***1.3.3 remove\_loaded\_module***

META: Autogenerated - needs to be reviewed/completed

Remove a module from the chained modules list and the list of loaded modules

```
$module->remove_loaded_module();
```

- **arg1: \$module (Apache::Module)**
- **ret: no return value**

### ***1.3.4 remove\_module***

META: Autogenerated - needs to be reviewed/completed

Remove a module from the server. There are some caveats: when the module is removed, its slot is lost so all the current per-dir and per-server configurations are invalid. So we should only ever call this function when you are invalidating almost all our current data. I.e. when doing a restart.

```
$module->remove_module();
```

- **arg1: \$module (Apache::Module)**  
the module structure of the module to remove
- **ret: no return value**

### ***1.3.5 top\_module***

META: Autogenerated - needs to be reviewed/completed

```
$ret = Apache::Module->top_module();
```

- **arg1: Apache::Module (class name)**
- **ret: \$ret (Apache::Module)**

### ***1.3.6 version***

META: Autogenerated - needs to be reviewed/completed

API version, \*not\* module version; check that module is compatible with this version of the server.

```
$version = $module->version();
```

### 1.3.7 *minor\_version*

- **arg1:** `$module (Apache::Module)`
- **ret:** `$version (integer)`

### ***1.3.7 minor\_version***

META: Autogenerated - needs to be reviewed/completed

API minor version. Provides API feature milestones. Not checked during module init

```
$minor_version = $module->minor_version();
```

- **arg1:** `$module (Apache::Module)`
- **ret:** `$minor_version (integer)`

### ***1.3.8 module\_index***

META: Autogenerated - needs to be reviewed/completed

Index to this modules structures in config vectors.

```
$index = $module->module_index();
```

- **arg1:** `$module (Apache::Module)`
- **ret:** `$index (integer)`

### ***1.3.9 name***

META: Autogenerated - needs to be reviewed/completed

The name of the module's C file

```
$name = $module->name();
```

- **arg1:** `$module (Apache::Module)`
- **ret:** `$name (string)`

### ***1.3.10 dynamic\_load\_handle***

META: Autogenerated - needs to be reviewed/completed

The handle for the DSO. Internal use only

```
$dl_handle = $module->dynamic_load_handle();
```

- **arg1:** `$module (Apache::Module)`
- **ret:** `$dl_handle (SCALAR)`

### ***1.3.11 next***

META: Autogenerated - needs to be reviewed/completed

A pointer to the next module in the list

```
$next_module = $module->next();
```

- **arg1:** `$module` (`Apache::Module`)
- **ret:** `$next_module` (`Apache::Module`)

### ***1.3.12 cmds***

META: Autogenerated - needs to be reviewed/completed

A `command_rec` table that describes all of the directives this module defines.

```
$cmd_rec = $module->cmds();
```

- **arg1:** `$module` (`Apache::Module`)
- **ret:** `$cmd_rec` (`Apache::Command`)

## **1.4 See Also**

`mod_perl` 2.0 documentation.

## **1.5 Copyright**

`mod_perl` 2.0 and its core modules are copyrighted under The Apache Software License, Version 1.1.

## **1.6 Authors**

The `mod_perl` development team and numerous contributors.



## Table of Contents:

1	Apache::Module - Perl API for creating and working with Apache modules	1
1.1	Synopsis	2
1.2	Description	2
1.3	API	2
1.3.1	find_linked_module	2
1.3.2	find_module_name	2
1.3.3	remove_loaded_module	3
1.3.4	remove_module	3
1.3.5	top_module	3
1.3.6	version	3
1.3.7	minor_version	4
1.3.8	module_index	4
1.3.9	name	4
1.3.10	dynamic_load_handle	4
1.3.11	next	5
1.3.12	cmds	5
1.4	See Also	5
1.5	Copyright	5
1.6	Authors	5