# Apache MPM prefork

| | |
|---|---|
| **Description:** | Implements a non-threaded, pre-forking web server |
| **Status:** | MPM |
| **Module Identifier:** | mpm_prefork_module |
| **Source File:** | prefork.c |

## Summary

This Multi-Processing Module (MPM) implements a non-threaded, pre-forking web server that handles requests in a manner similar to Apache 1.3. It is appropriate for sites that need to avoid threading for compatibility with non-thread-safe libraries. It is also the best MPM for isolating each request, so that a problem with a single request will not affect any other.

This MPM is very self-regulating, so it is rarely necessary to adjust its configuration directives. Most important is that `MaxClients` be big enough to handle as many simultaneous requests as you expect to receive, but small enough to assure that there is enough physical RAM for all processes.

## Topics

## Directives

( ➠ This directive is defined elsewhere. See: `mpm_common` )

## See also

- Setting which addresses and ports Apache uses[1]

## How it Works

A single control process is responsible for launching child processes which listen for connections and serve them when they arrive. Apache always tries to maintain several *spare* or idle server processes, which stand ready to serve incoming requests. In this way, clients do not need to wait for a new child processes to be forked before their requests can be served.

The `StartServers`, `MinSpareServers`, `MaxSpareServers`, and `MaxClients` regulate how the parent process creates children to serve requests. In general, Apache is very self-regulating, so most sites do not need to adjust these directives from their default values. Sites which need to serve more than 256 simultaneous requests may need to increase `MaxClients`, while sites with limited memory may need to decrease `MaxClients` to keep the server from thrashing (swapping memory to disk and back). More information about tuning process creation is provided in the performance hints [2] documentation.

While the parent process is usually started as `root` under Unix in order to bind to port 80, the child processes are launched by Apache as a less-privileged user. The `User` and `Group` directives are used to set the privileges of the Apache child processes. The child processes must be able to read all the content that will be served, but should have as few privileges beyond that as possible.

`MaxRequestsPerChild` controls how frequently the server recycles processes by killing old ones and launching new ones.

## MaxSpareServers Directive

| | |
|---|---|
| **Description:** | Maximum number of idle child server processes |
| **Syntax:** | `MaxSpareServers` *`number`* |
| **Default:** | `MaxSpareServers 10` |
| **Context:** | server config |
| **Status:** | MPM |
| **Module:** | prefork |

The `MaxSpareServers` directive sets the desired maximum number of *idle* child server processes. An idle process is one which is not handling a request. If there are more than `MaxSpareServers` idle, then the parent process will kill off the excess processes.

Tuning of this parameter should only be necessary on very busy sites. Setting this parameter to a large number is almost always a bad idea. If you are trying to set the value lower than `MinSpareServers`, Apache will automatically adjust it to `MinSpareServers + 1`.

### See also

- `MinSpareServers`
- `StartServers`

## MinSpareServers Directive

| | |
|---|---|
| **Description:** | Minimum number of idle child server processes |
| **Syntax:** | `MinSpareServers` *`number`* |
| **Default:** | `MinSpareServers 5` |
| **Context:** | server config |
| **Status:** | MPM |
| **Module:** | prefork |

The `MinSpareServers` directive sets the desired minimum number of *idle* child server processes. An idle process is one which is not handling a request. If there are fewer than `MinSpareServers` idle, then the parent process creates new children at a maximum rate of 1 per second.

Tuning of this parameter should only be necessary on very busy sites. Setting this parameter to a large number is almost always a bad idea.

### See also

- `MaxSpareServers`
- `StartServers`

## URI References

[1] http://httpd.apache.org/docs-2.1/bind.html
[2] http://httpd.apache.org/docs-2.1/misc/perf-tuning.html