# Multi-Processing Modules (MPMs)



This document describes what a Multi-Processing Module is and how they are used by the Apache HTTP Server.



## Introduction

The Apache HTTP Server is designed to be a powerful and flexible web server that can work on a very wide variety of platforms in a range of different environments. Different platforms and different environments often require different features, or may have different ways of implementing the same feature most efficiently. Apache has always accommodated a wide variety of environments through its modular design. This design allows the webmaster to choose which features will be included in the server by selecting which modules to load either at compile-time or at run-time.

Apache 2.0 extends this modular design to the most basic functions of a web server. The server ships with a selection of Multi-Processing Modules (MPMs) which are responsible for binding to network ports on the machine, accepting requests, and dispatching children to handle the requests.

Extending the modular design to this level of the server allows two important benefits:

- Apache can more cleanly and efficiently support a wide variety of operating systems. In particular, the Windows version of Apache is now much more efficient, since `mpm_winnt` can use native networking features in place of the POSIX layer used in Apache 1.3. This benefit also extends to other operating systems that implement specialized MPMs.
- The server can be better customized for the needs of the particular site. For example, sites that need a great deal of scalability can choose to use a threaded MPM like `worker` or `event`, while sites requiring stability or compatibility with older software can use a `prefork`.

At the user level, MPMs appear much like other Apache modules. The main difference is that one and only one MPM must be loaded into the server at any time. The list of available MPMs appears on the module index page.

## Choosing an MPM

MPMs must be chosen during configuration, and compiled into the server. Compilers are capable of optimizing a lot of functions if threads are used, but only if they know that threads are being used.

To actually choose the desired MPM, use the argument `--with-mpm=`*NAME* with the `configure` script. *NAME* is the name of the desired MPM.

Once the server has been compiled, it is possible to determine which MPM was chosen by using `./httpd -l`. This command will list every module that is compiled into the server, including the MPM.

## MPM Defaults

The following table lists the default MPMs for various operating systems. This will be the MPM selected if you do not make another choice at compile-time.

| | |
|---|---|
| BeOS | `beos` |
| Netware | `mpm_netware` |
| OS/2 | `mpmt_os2` |
| Unix | `prefork` |
| Windows | `mpm_winnt` |