

# Apache MPM threadpool

<b>Description:</b>	Yet another experimental variant of the standard <code>worker</code> MPM
<b>Status:</b>	MPM
<b>Module Identifier:</b>	<code>mpm_threadpool_module</code>
<b>Source File:</b>	<code>threadpool.c</code>

## Summary

### Warning

This MPM is a developer playground and highly experimental, so it may or may not work as expected.

This is an experimental variant of the standard `worker` MPM. Rather than queuing connections like the `worker` MPM, the `threadpool` MPM queues idle worker threads and hands each accepted connection to the next available worker.

The `threadpool` MPM can't match the performance of the `worker` MPM in benchmark testing. As of 2.0.39, some of the key load-throttling concepts from the `threadpool` MPM have been incorporated into the `worker` MPM. The `threadpool` code is useful primarily as a research platform. For general-purpose use and for any production environments, use `worker` instead.

## Directives

<code>AcceptMutex</code> .....	➡	<code>MinSpareThreads</code> .....	➡
<code>CoreDumpDirectory</code> .....	➡	<code>PidFile</code> .....	➡
<code>Group</code> .....	➡	<code>ScoreBoardFile</code> .....	➡
<code>Listen</code> .....	➡	<code>SendBufferSize</code> .....	➡
<code>ListenBacklog</code> .....	➡	<code>ServerLimit</code> .....	➡
<code>LockFile</code> .....	➡	<code>StartServers</code> .....	➡
<code>MaxClients</code> .....	➡	<code>ThreadLimit</code> .....	➡
<code>MaxMemFree</code> .....	➡	<code>ThreadsPerChild</code> .....	➡
<code>MaxRequestsPerChild</code> .....	➡	<code>User</code> .....	➡
<code>MaxSpareThreads</code> .....	➡		

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