# Overview of new features in Apache HTTP Server 2.4

Available Languages: **en** | **es** | **fr** | **tr**

This document describes some of the major changes between the 2.2 and 2.4 versions of the Apache HTTP Server. For new features since version 2.0, see the 2.2 new features document.

### See also

- Comments

## Core Enhancements

**Run-time Loadable MPMs**
: Multiple MPMs can now be built as loadable modules at compile time. The MPM of choice can be configured at run time via `LoadModule` directive.

**Event MPM**
: The Event MPM is no longer experimental but is now fully supported.

**Asynchronous support**
: Better support for asynchronous read/write for supporting MPMs and platforms.

**Per-module and per-directory LogLevel configuration**
: The `LogLevel` can now be configured per module and per directory. New levels `trace1` to `trace8` have been added above the `debug` log level.

**Per-request configuration sections**
: `<If>`, `<ElseIf>`, and `<Else>` sections can be used to set the configuration based on per-request criteria.

**General-purpose expression parser**
: A new expression parser allows to specify complex conditions using a common syntax in directives like `SetEnvIfExpr`, `RewriteCond`, `Header`, `<If>`, and others.

**KeepAliveTimeout in milliseconds**
: It is now possible to specify `KeepAliveTimeout` in milliseconds.

**NameVirtualHost directive**
: No longer needed and is now deprecated.

**Override Configuration**
: The new `AllowOverrideList` directive allows more fine grained control which directives are allowed in `.htaccess` files.

**Config file variables**
: It is now possible to `Define` variables in the configuration, allowing a clearer representation if the same value is used at many places in the configuration.

**Reduced memory usage**
: Despite many new features, 2.4.x tends to use less memory than 2.2.x.

## New Modules

**`mod_proxy_fcgi`**
: FastCGI Protocol backend for `mod_proxy`

**`mod_proxy_scgi`**
: SCGI Protocol backend for `mod_proxy`

**`mod_proxy_express`**
: Provides dynamically configured mass reverse proxies for `mod_proxy`

**`mod_remoteip`**
: Replaces the apparent client remote IP address and hostname for the request with the IP address list presented by a proxies or a load balancer

via the request headers.

**mod_heartmonitor**, **mod_lbmethod_heartbeat**
: Allow mod_proxy_balancer to base loadbalancing decisions on the number of active connections on the backend servers.

**mod_proxy_html**
: Formerly a third-party module, this supports fixing of HTML links in a reverse proxy situation, where the backend generates URLs that are not valid for the proxy's clients.

**mod_sed**
: An advanced replacement of mod_substitute, allows to edit the response body with the full power of sed.

**mod_auth_form**
: Enables form-based authentication.

**mod_session**
: Enables the use of session state for clients, using cookie or database storage.

**mod_allowmethods**
: New module to restrict certain HTTP methods without interfering with authentication or authorization.

**mod_lua**
: Embeds the Lua language into httpd, for configuration and small business logic functions. (Experimental)

**mod_log_debug**
: Allows the addition of customizable debug logging at different phases of the request processing.

**mod_buffer**
: Provides for buffering the input and output filter stacks

**mod_data**
: Convert response body into an RFC2397 data URL

**mod_ratelimit**
: Provides Bandwidth Rate Limiting for Clients

**mod_request**
: Provides Filters to handle and make available HTTP request bodies

**mod_reflector**
: Provides Reflection of a request body as a response via the output filter stack.

**mod_slotmem_shm**
: Provides a Slot-based shared memory provider (ala the scoreboard).

**mod_xml2enc**
: Formerly a third-party module, this supports internationalisation in libxml2-based (markup-aware) filter modules.

**mod_macro (available since 2.4.5)**
: Provide macros within configuration files.

**mod_proxy_wstunnel (available since 2.4.5)**
: Support web-socket tunnels.

**mod_authnz_fcgi (available since 2.4.10)**
: Enable FastCGI authorizer applications to authenticate and/or authorize clients.

**mod_http2 (available since 2.4.17)**
: Support for the HTTP/2 transport layer.

**mod_proxy_http2 (available since 2.4.19)**
: HTTP/2 Protocol backend for mod_proxy

**mod_proxy_hcheck (available since 2.4.21)**
: Support independent dynamic health checks for remote proxiy backend servers.

**mod_brotli (available since 2.4.26)**
: Support the Brotli compression algorithm.

**mod_md (available since 2.4.30)**
: Support the ACME protocol to automate certificate provisionning.

**mod_socache_redis (available since 2.4.39)**
: Support Redis based shared object cache provider.

**mod_systemd (available since 2.4.42)**
: systemd integration. It allows httpd to be used in a service with the systemd `Type=notify`.

## Module Enhancements

**mod_ssl**

mod_ssl can now be configured to use an OCSP server to check the validation status of a client certificate. The default responder is configurable, along with the decision on whether to prefer the responder designated in the client certificate itself.
mod_ssl now also supports OCSP stapling, where the server pro-actively obtains an OCSP verification of its certificate and transmits that to the client during the handshake.
mod_ssl can now be configured to share SSL Session data between servers through memcached
EC keys are now supported in addition to RSA and DSA.
Support for TLS-SRP (available in 2.4.4 and later).

**mod_proxy**

The `ProxyPass` directive is now most optimally configured within a `Location` or `LocationMatch` block, and offers a significant performance advantage over the traditional two-parameter syntax when present in large numbers.
The source address used for proxy requests is now configurable.
Support for Unix domain sockets to the backend (available in 2.4.7 and later).

**mod_proxy_balancer**

More runtime configuration changes for BalancerMembers via balancer-manager
Additional BalancerMembers can be added at runtime via balancer-manager
Runtime configuration of a subset of Balancer parameters
BalancerMembers can be set to 'Drain' so that they only respond to existing sticky sessions, allowing them to be taken gracefully offline.
Balancer settings can be persistent after restarts.

**mod_cache**

The mod_cache CACHE filter can be optionally inserted at a given point in the filter chain to provide fine control over caching.
mod_cache can now cache HEAD requests.
Where possible, mod_cache directives can now be set per directory, instead of per server.
The base URL of cached URLs can be customised, so that a cluster of caches can share the same endpoint URL prefix.
mod_cache is now capable of serving stale cached data when a backend is unavailable (error 5xx).
mod_cache can now insert HIT/MISS/REVALIDATE into an X-Cache header.

**mod_include**

Support for the 'onerror' attribute within an 'include' element, allowing an error document to be served on error instead of the default error string.

**mod_cgi, mod_include, mod_isapi, ...**

Translation of headers to environment variables is more strict than before to mitigate some possible cross-site-scripting attacks via header injection. Header names containing invalid characters (including underscores) are no longer converted to environment variables. Environment Variables in Apache has some pointers on how to work around broken legacy clients which require such headers. (This affects all modules which use these environment variables.)

**mod_authz_core Authorization Logic Containers**

Advanced authorization logic may now be specified using the `Require` directive and the related container directives, such as `<RequireAll>`.

**mod_rewrite**

`mod_rewrite` adds the `[QSD]` (Query String Discard) and `[END]` flags for `RewriteRule` to simplify common rewriting scenarios.
Adds the possibility to use complex boolean expressions in `RewriteCond`.
Allows the use of SQL queries as `RewriteMap` functions.

**mod_ldap, mod_authnz_ldap**

`mod_authnz_ldap` adds support for nested groups.
`mod_ldap` adds `LDAPConnectionPoolTTL`, `LDAPTimeout`, and other improvements in the handling of timeouts. This is especially useful for setups where a stateful firewall drops idle connections to the LDAP server.
`mod_ldap` adds `LDAPLibraryDebug` to log debug information provided by the used LDAP toolkit.

**mod_info**

`mod_info` can now dump the pre-parsed configuration to stdout during server startup.

**mod_auth_basic**

New generic mechanism to fake basic authentication (available in 2.4.5 and later).

## Program Enhancements

**fcgistarter**

New FastCGI daemon starter utility

**htcacheclean**

Current cached URLs can now be listed, with optional metadata included.
Allow explicit deletion of individual cached URLs from the cache.
File sizes can now be rounded up to the given block size, making the size limits map more closely to the real size on disk.
Cache size can now be limited by the number of inodes, instead of or in addition to being limited by the size of the files on disk.

**rotatelogs**

May now create a link to the current log file.
May now invoke a custom post-rotate script.

**htpasswd, htdbm**

Support for the bcrypt algorithm (available in 2.4.4 and later).

## Documentation

**mod_rewrite**

The `mod_rewrite` documentation has been rearranged and almost completely rewritten, with a focus on examples and common usage, as well as on showing you when other solutions are more appropriate. The Rewrite Guide is now a top-level section with much more detail and better organization.

**mod_ssl**

The `mod_ssl` documentation has been greatly enhanced, with more examples at the getting started level, in addition to the previous focus on technical details.

**Caching Guide**

The Caching Guide has been rewritten to properly distinguish between the RFC2616 HTTP/1.1 caching features provided by `mod_cache`, and the generic key/value caching provided by the socache interface, as well as to cover specialised caching provided by mechanisms such as `mod_file_cache`.

## Module Developer Changes

**Check Configuration Hook Added**

A new hook, `check_config`, has been added which runs between the `pre_config` and `open_logs` hooks. It also runs before the `test_config` hook when the `-t` option is passed to `httpd`. The `check_config` hook allows modules to review interdependent configuration directive values and adjust them while messages can still be logged to the console. The user can thus be alerted to misconfiguration problems before the core `open_logs` hook function redirects console output to the error log.

**Expression Parser Added**

We now have a general-purpose expression parser, whose API is exposed in *ap_expr.h*. This is adapted from the expression parser previously implemented in `mod_ssl`.

**Authorization Logic Containers**

Authorization modules now register as a provider, via ap_register_auth_provider(), to support advanced authorization logic, such as `<RequireAll>`.

**Small-Object Caching Interface**

The *ap_socache.h* header exposes a provider-based interface for caching small data objects, based on the previous implementation of the `mod_ssl` session cache. Providers using a shared-memory cyclic buffer, disk-based dbm files, and a memcache distributed cache are currently supported.

**Cache Status Hook Added**

The `mod_cache` module now includes a new `cache_status` hook, which is called when the caching decision becomes known. A default implementation is provided which adds an optional `X-Cache` and `X-Cache-Detail` header to the response.

The developer documentation contains a detailed list of API changes.