Upgrading to 2.4 from 2.2

In order to assist folks upgrading, we maintain a document describing information critical to existing Apache HTTP Server users. These are intended to be brief notes, and you should be able to find more information in either the New Features document, or in the `src/CHANGES` file. Application and module developers can find a summary of API changes in the API updates overview.

This document describes changes in server behavior that might require you to change your configuration or how you use the server in order to continue using 2.4 as you are currently using 2.2. To take advantage of new features in 2.4, see the New Features document.

This document describes only the changes from 2.2 to 2.4. If you are upgrading from version 2.0, you should also consult the 2.0 to 2.2 upgrading document.

Compile-Time Configuration Changes

The compilation process is very similar to the one used in version 2.2. Your old `configure` command line (as found in `build/config.nice` in the installed server directory) can be used in most cases. There are some changes in the default settings. Some details of changes:

- These modules have been removed: mod_authn_default, mod_authz_default, mod_mem_cache. If you were using mod_mem_cache in 2.2, look at `mod_cache_disk` in 2.4.
- All load balancing implementations have been moved to individual, self-contained mod_proxy submodules, e.g. `mod_lbmethod_bybusyness`. You might need to build and load any of these that your configuration uses.
- Platform support has been removed for BeOS, TPF, and even older platforms such as A/UX, Next, and Tandem. These were believed to be broken anyway.
- configure: dynamic modules (DSO) are built by default
- configure: By default, only a basic set of modules is loaded. The other `LoadModule` directives are commented out in the configuration file.
- configure: the "most" module set gets built by default
- configure: the "reallyall" module set adds developer modules to the "all" set

Run-Time Configuration Changes

There have been significant changes in authorization configuration, and other minor configuration changes, that could require changes to your 2.2 configuration files before using them for 2.4.

Authorization

Any configuration file that uses authorization will likely need changes.

You should review the Authentication, Authorization and Access Control Howto, especially the section Beyond just authorization which explains the new mechanisms for controlling the order in which the authorization directives are applied.

Directives that control how authorization modules respond when they don't match the authenticated user have been removed: This includes AuthzLDAPAuthoritative, AuthzDBDAuthoritative, AuthzDBMAuthoritative, AuthzGroupFileAuthoritative, AuthzUserAuthoritative, and AuthzOwnerAuthoritative. These directives have been replaced by the more expressive `RequireAny`, `RequireNone`, and `RequireAll`.

If you use mod_authz_dbm, you must port your configuration to use `Require dbm-group ...` in place of `Require group ...`.

Access control

In 2.2, access control based on client hostname, IP address, and other characteristics of client requests was done using the directives Order, Allow, Deny, and Satisfy.

In 2.4, such access control is done in the same way as other authorization checks, using the new module mod_authz_host. The old access control idioms should be replaced by the new authentication mechanisms, although for compatibility with old configurations, the new module mod_access_compat is provided.

> **Mixing old and new directives**
>
> Mixing old directives like Order, Allow or Deny with new ones like Require is technically possible but discouraged. mod_access_compat was created to support configurations containing only old directives to facilitate the 2.4 upgrade. Please check the examples below to get a better idea about issues that might arise.

Here are some examples of old and new ways to do the same access control.

In this example, there is no authentication and all requests are denied.

2.2 configuration:

```
Order deny,allow
Deny from all
```

2.4 configuration:

```
Require all denied
```

In this example, there is no authentication and all requests are allowed.

2.2 configuration:

```
Order allow,deny
Allow from all
```

2.4 configuration:

```
Require all granted
```

In the following example, there is no authentication and all hosts in the example.org domain are allowed access; all other hosts are denied access.

2.2 configuration:

```
Order Deny,Allow
Deny from all
Allow from example.org
```

2.4 configuration:

```
Require host example.org
```

In the following example, mixing old and new directives leads to unexpected results.

Mixing old and new directives: NOT WORKING AS EXPECTED

```
DocumentRoot "/var/www/html"

<Directory "/">
```

```
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

<Location "/server-status">
    SetHandler server-status
    Require local
</Location>

access.log - GET /server-status 403 127.0.0.1
error.log - AH01797: client denied by server configuration
```

Why httpd denies access to servers-status even if the configuration seems to allow it? Because `mod_access_compat` directives take precedence over the `mod_authz_host` one in this configuration merge scenario.

This example conversely works as expected:

Mixing old and new directives: WORKING AS EXPECTED

```
DocumentRoot "/var/www/html"

<Directory "/">
    AllowOverride None
    Require all denied
</Directory>

<Location "/server-status">
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow From 127.0.0.1
</Location>

access.log - GET /server-status 200 127.0.0.1
```

So even if mixing configuration is still possible, please try to avoid it when upgrading: either keep old directives and then migrate to the new ones on a later stage or just migrate everything in bulk.

In many configurations with authentication, where the value of the `Satisfy` was the default of *ALL*, snippets that simply disabled host-based access control are omitted:

2.2 configuration:

```
# 2.2 config that disables host-based access control and u
Order Deny,Allow
Allow from all
AuthType Basic
AuthBasicProvider file
AuthUserFile /example.com/conf/users.passwd
AuthName secure
Require valid-user
```

2.4 configuration:

```
# No replacement of disabling host-based access control ne
AuthType Basic
AuthBasicProvider file
AuthUserFile /example.com/conf/users.passwd
AuthName secure
Require valid-user
```

In configurations where both authentication and access control were meaningfully combined, the access control directives should be migrated. This example allows requests meeting *both* criteria:

2.2 configuration:

```
Order allow,deny
Deny from all
# Satisfy ALL is the default
Satisfy ALL
Allow from 127.0.0.1
AuthType Basic
AuthBasicProvider file
AuthUserFile /example.com/conf/users.passwd
AuthName secure
Require valid-user
```

2.4 configuration:

```
AuthType Basic
AuthBasicProvider file
AuthUserFile /example.com/conf/users.passwd
AuthName secure
<RequireAll>
  Require valid-user
  Require ip 127.0.0.1
</RequireAll>
```

In configurations where both authentication and access control were meaningfully combined, the access control directives should be migrated. This example allows requests meeting *either* criteria:

2.2 configuration:

```
Order allow,deny
Deny from all
Satisfy any
Allow from 127.0.0.1
AuthType Basic
AuthBasicProvider file
AuthUserFile /example.com/conf/users.passwd
AuthName secure
Require valid-user
```

2.4 configuration:

```
AuthType Basic
AuthBasicProvider file
AuthUserFile /example.com/conf/users.passwd
AuthName secure
# Implicitly <RequireAny>
Require valid-user
Require ip 127.0.0.1
```

Other configuration changes

Some other small adjustments may be necessary for particular configurations as discussed below.

- `MaxRequestsPerChild` has been renamed to `MaxConnectionsPerChild`, describes more accurately what it does. The old name is still supported.
- `MaxClients` has been renamed to `MaxRequestWorkers`, which describes more accurately what it does. For async MPMs, like `event`, the maximum number of clients is not equivalent than the number of worker threads. The old name is still supported.
- The `DefaultType` directive no longer has any effect, other than to emit a warning if it's used with any value other than `none`. You need to use other configuration settings to replace it in 2.4.
- `AllowOverride` now defaults to `None`.
- `EnableSendfile` now defaults to Off.
- `FileETag` now defaults to "MTime Size" (without INode).
- `mod_dav_fs`: The format of the `DavLockDB` file has changed for systems with inodes. The old `DavLockDB` file must be deleted on upgrade.

- KeepAlive only accepts values of On or Off. Previously, any value other than "Off" or "0" was treated as "On".
- Directives AcceptMutex, LockFile, RewriteLock, SSLMutex, SSLStaplingMutex, and WatchdogMutexPath have been replaced with a single Mutex directive. You will need to evaluate any use of these removed directives in your 2.2 configuration to determine if they can just be deleted or will need to be replaced using Mutex.
- mod_cache: CacheIgnoreURLSessionIdentifiers now does an exact match against the query string instead of a partial match. If your configuration was using partial strings, e.g. using sessionid to match /someapplication/image.gif;jsessionid=123456789, then you will need to change to the full string jsessionid.
- mod_cache: The second parameter to CacheEnable only matches forward proxy content if it begins with the correct protocol. In 2.2 and earlier, a parameter of '/' matched all content.
- mod_ldap: LDAPTrustedClientCert is now consistently a per-directory setting only. If you use this directive, review your configuration to make sure it is present in all the necessary directory contexts.
- mod_filter: FilterProvider syntax has changed and now uses a boolean expression to determine if a filter is applied.
- mod_include:
  - The #if expr element now uses the new expression parser. The old syntax can be restored with the new directive SSILegacyExprParser.
  - An SSI* config directive in directory scope no longer causes all other per-directory SSI* directives to be reset to their default values.
- mod_charset_lite: The DebugLevel option has been removed in favour of per-module LogLevel configuration.
- mod_ext_filter: The DebugLevel option has been removed in favour of per-module LogLevel configuration.
- mod_proxy_scgi: The default setting for PATH_INFO has changed from httpd 2.2, and some web applications will no longer operate properly with the new PATH_INFO setting. The previous setting can be restored by configuring the proxy-scgi-pathinfo variable.
- mod_ssl: CRL based revocation checking now needs to be explicitly configured through SSLCARevocationCheck.
- mod_substitute: The maximum line length is now limited to 1MB.
- mod_reqtimeout: If the module is loaded, it will now set some default timeouts.
- mod_dumpio: DumpIOLogLevel is no longer supported. Data is always logged at LogLevel trace7.
- On Unix platforms, piped logging commands configured using either ErrorLog or CustomLog were invoked using /bin/sh -c in 2.2 and earlier. In 2.4 and later, piped logging commands are executed directly. To restore the old behaviour, see the piped logging documentation.

Misc Changes

- mod_autoindex: will now extract titles and display descriptions for .xhtml files, which were previously ignored.
- mod_ssl: The default format of the *_DN variables has changed. The old format can still be used with the new LegacyDNStringFormat argument to SSLOptions. The SSLv2 protocol is no longer supported. SSLProxyCheckPeerCN and SSLProxyCheckPeerExpire now default to On, causing proxy requests to HTTPS hosts with bad or outdated certificates to fail with a 502 status code (Bad gateway)
- htpasswd now uses MD5 hash by default on all platforms.
- The NameVirtualHost directive no longer has any effect, other than to emit a warning. Any address/port combination appearing in multiple virtual

hosts is implicitly treated as a name-based virtual host.
- mod_deflate will now skip compression if it knows that the size overhead added by the compression is larger than the data to be compressed.
- Multi-language error documents from 2.2.x may not work unless they are adjusted to the new syntax of mod_include's `#if expr=` element or the directive SSILegacyExprParser is enabled for the directory containing the error documents.
- The functionality provided by `mod_authn_alias` in previous versions (i.e., the AuthnProviderAlias directive) has been moved into mod_authn_core.
- mod_cgid uses the servers Timeout to limit the length of time to wait for CGI output. This timeout can be overridden with CGIDScriptTImeout.

Third Party Modules

All modules must be recompiled for 2.4 before being loaded.

Many third-party modules designed for version 2.2 will otherwise work unchanged with the Apache HTTP Server version 2.4. Some will require changes; see the API update overview.

Common problems when upgrading

- Startup errors:
  - `Invalid command 'User', perhaps misspelled or defined by a module not included in the server configuration` - load module mod_unixd
  - `Invalid command 'Require', perhaps misspelled or defined by a module not included in the server configuration`, or `Invalid command 'Order', perhaps misspelled or defined by a module not included in the server configuration` - load module mod_access_compat, or update configuration to 2.4 authorization directives.
  - `Ignoring deprecated use of DefaultType in line NN of /path/to/httpd.conf` - remove DefaultType and replace with other configuration settings.
  - `Invalid command 'AddOutputFilterByType', perhaps misspelled or defined by a module not included in the server configuration` - AddOutputFilterByType has moved from the core to mod_filter, which must be loaded.
- Errors serving requests:
  - `configuration error: couldn't check user: /path` - load module mod_authn_core.
  - `.htaccess` files aren't being processed - Check for an appropriate AllowOverride directive; the default changed to `None` in 2.4.