

## Apache Module mod\_expires

<b>Description:</b>	Generation of Expires HTTP headers according to user-specified criteria
<b>Status:</b>	Extension
<b>Module Identifier:</b>	expires_module
<b>Source File:</b>	mod_expires.c

### Summary

This module controls the setting of the Expires HTTP header in server responses. The expiration date can set to be relative to either the time the source file was last modified, or to the time of the client access.

The Expires HTTP header is an instruction to the client about the document's validity and persistence. If cached, the document may be fetched from the cache rather than from the source until this time has passed. After that, the cache copy is considered "expired" and invalid, and a new copy must be obtained from the source.

### Topics

Alternate Interval Syntax ..... 1

### Directives

ExpiresActive ..... 2  
 ExpiresByType ..... 2  
 ExpiresDefault ..... 3

## Alternate Interval Syntax

---

The ExpiresDefault and ExpiresByType directives can also be defined in a more readable syntax of the form:

```
ExpiresDefault "<base> [plus] {<num> <type>}"
ExpiresByType type/encoding "<base> [plus] {<num> <type>}"
```

where <base> is one of:

- access
- now (equivalent to 'access')
- modification

The plus keyword is optional. <num> should be an integer value [acceptable to atoi()], and <type> is one of:

- years
- months
- weeks
- days
- hours
- minutes

---

## Apache Module mod\_expires

---

- seconds

For example, any of the following directives can be used to make documents expire 1 month after being accessed, by default:

```
ExpiresDefault "access plus 1 month"  
ExpiresDefault "access plus 4 weeks"  
ExpiresDefault "access plus 30 days"
```

The expiry time can be fine-tuned by adding several '<num> <type>' clauses:

```
ExpiresByType text/html "access plus 1 month 15 days 2 hours"  
ExpiresByType image/gif "modification plus 5 hours 3 minutes"
```

Note that if you use a modification date based setting, the Expires header will **not** be added to content that does not come from a file on disk. This is due to the fact that there is no modification time for such content.

### ExpiresActive Directive

---

<b>Description:</b>	Enables generation of Expires headers
<b>Syntax:</b>	ExpiresActive On Off
<b>Context:</b>	server config, virtual host, directory, .htaccess
<b>Override:</b>	Indexes
<b>Status:</b>	Extension
<b>Module:</b>	mod_expires

This directive enables or disables the generation of the Expires header for the document realm in question. (That is, if found in an .htaccess file, for instance, it applies only to documents generated from that directory.) If set to Off, no Expires header will be generated for any document in the realm (unless overridden at a lower level, such as an .htaccess file overriding a server config file). If set to On, the header will be added to served documents according to the criteria defined by the ExpiresByType and ExpiresDefault directives (*q.v.*).

Note that this directive does not guarantee that an Expires header will be generated. If the criteria aren't met, no header will be sent, and the effect will be as though this directive wasn't even specified.

### ExpiresByType Directive

---

<b>Description:</b>	Value of the Expires header configured by MIME type
<b>Syntax:</b>	ExpiresByType <i>MIME-type</i> <code>seconds
<b>Context:</b>	server config, virtual host, directory, .htaccess
<b>Override:</b>	Indexes
<b>Status:</b>	Extension
<b>Module:</b>	mod_expires

This directive defines the value of the Expires header generated for documents of the specified type (*e.g.*, text/html). The second argument sets the number of seconds that will be added to a base time to construct the expiration date.

---

## Apache Module mod\_expires

---

The base time is either the last modification time of the file, or the time of the client's access to the document. Which should be used is specified by the `<code>` field; `M` means that the file's last modification time should be used as the base time, and `A` means the client's access time should be used.

The difference in effect is subtle. If `M` is used, all current copies of the document in all caches will expire at the same time, which can be good for something like a weekly notice that's always found at the same URL. If `A` is used, the date of expiration is different for each client; this can be good for image files that don't change very often, particularly for a set of related documents that all refer to the same images (*i.e.*, the images will be accessed repeatedly within a relatively short timespan).

**Example:**

```
# enable expirations
ExpiresActive On
# expire GIF images after a month in the client's cache
ExpiresByType image/gif A2592000
# HTML documents are good for a week from the
# time they were changed
ExpiresByType text/html M604800
```

Note that this directive only has effect if `ExpiresActive On` has been specified. It overrides, for the specified MIME type *only*, any expiration date set by the `ExpiresDefault` directive.

You can also specify the expiration time calculation using an alternate syntax, described earlier in this document.

## ExpiresDefault Directive

---

<b>Description:</b>	Default algorithm for calculating expiration time
<b>Syntax:</b>	<code>ExpiresDefault &lt;code&gt;seconds</code>
<b>Context:</b>	server config, virtual host, directory, <code>.htaccess</code>
<b>Override:</b>	Indexes
<b>Status:</b>	Extension
<b>Module:</b>	<code>mod_expires</code>

This directive sets the default algorithm for calculating the expiration time for all documents in the affected realm. It can be overridden on a type-by-type basis by the `ExpiresByType` directive. See the description of that directive for details about the syntax of the argument, and the alternate syntax description as well.