Apache Module mod_cache

Description:	Content cache keyed to URIs.
Status:	Experimental
Module Identifier:	cache_module
Source File:	mod_cache.c

Summary

This module is experimental. Documentation is still under development...

mod_cache implements an RFC 2616[1] compliant HTTP content cache that can be used to cache either local or proxied content. mod_cache requires the services of one or more storage management modules. Two storage management modules are included in the base Apache distribution:

mod_disk_cache
implements a disk based storage manager.

mod_mem_cache
implements a memory based storage manager. mod_mem_cache can be configured to operate in two modes: caching open file descriptors or caching objects in heap storage. mod_mem_cache can be used to cache locally generated content or to cache backend server content for mod_proxy when configured using ProxyPass (aka *reverse proxy*)

Content is stored in and retrieved from the cache using URI based keys. Content with access protection is not cached.

Topics

Directives

Related Modules and Directives

Related Modules	Related Directives
mod_disk_cache	CacheRoot
mod_mem_cache	CacheSize
	CacheGcInterval
	CacheDirLevels
	CacheDirLength
	CacheExpiryCheck
	CacheMinFileSize
	CacheMaxFileSize
	CacheTimeMargin
	CacheGcDaily
	CacheGcUnused
	CacheGcClean
	CacheGcMemUsage

Related Modules	Related Directives
	MCacheSize MCacheMaxObjectCount MCacheMinObjectSize MCacheMaxObjectSize MCacheRemovalAlgorithm MCacheMaxStreamingBuffer

Sample Configuration

Sample httpd.conf

```
#
# Sample Cache Configuration
#
LoadModule cache_module modules/mod_cache.so

<IfModule mod_cache.c>
  #LoadModule disk_cache_module modules/mod_disk_cache.so
  <IfModule mod_disk_cache.c>
    CacheRoot c:/cacheroot
    CacheSize 256
    CacheEnable disk /
    CacheDirLevels 5
    CacheDirLength 3
  </IfModule>

  LoadModule mem_cache_module modules/mod_mem_cache.so
  <IfModule mod_mem_cache.c>
    CacheEnable mem /
    MCacheSize 4096
    MCacheMaxObjectCount 100
    MCacheMinObjectSize 1
    MCacheMaxObjectSize 2048
  </IfModule>
</IfModule>
```

CacheDefaultExpire Directive

Description:	The default duration to cache a document when no expiry date is specified.
Syntax:	`CacheDefaultExpire` *seconds*
Default:	`CacheDefaultExpire 3600 (one hour)`
Context:	server config, virtual host
Status:	Experimental
Module:	mod_cache

The `CacheDefaultExpire` directive specifies a default time, in seconds, to cache a document if neither an expiry date nor last-modified date are provided with the document. The value specified with the `CacheMaxExpire` directive does *not* override this setting.

```
CacheDefaultExpire 86400
```

CacheDisable Directive

Description:	Disable caching of specified URLs
Syntax:	`CacheDisable` *`url-string`*
Context:	server config, virtual host
Status:	Experimental
Module:	mod_cache

The `CacheDisable` directive instructs `mod_cache` to *not* cache urls at or below *url-string*.

Example

```
CacheDisable /local_files
```

CacheEnable Directive

Description:	Enable caching of specified URLs using a specified storage manager
Syntax:	`CacheEnable` *`cache_type url-string`*
Context:	server config, virtual host
Status:	Experimental
Module:	mod_cache

The `CacheEnable` directive instructs `mod_cache` to cache urls at or below *url-string*. The cache storage manager is specified with the *cache_type* argument. *cache_type* `mem` instructs `mod_cache` to use the memory based storage manager implemented by `mod_mem_cache`. *cache_type* `disk` instructs `mod_cache` to use the disk based storage manager implemented by `mod_disk_cache`. *cache_type* `fd` instructs `mod_cache` to use the file descriptor cache implemented by `mod_mem_cache`.

In the event that the URL space overlaps between different `CacheEnable` directives (as in the example below), each possible storage manager will be run until the first one that actually processes the request. The order in which the storage managers are run is determined by the order of the `CacheEnable` directives in the configuration file.

```
CacheEnable mem /manual
CacheEnable fd /images
CacheEnable disk /
```

CacheForceCompletion Directive

Description:	Percentage of document served, after which the server will complete caching the file even if the request is cancelled.
Syntax:	`CacheForceCompletion` *`Percentage`*
Default:	`CacheForceCompletion 60`
Context:	server config, virtual host
Status:	Experimental
Module:	mod_cache

Ordinarily, if a request is cancelled while the response is being cached and delivered to the client the processing of the response will stop and the cache entry will be removed. The `CacheForceCompletion` directive specifies a threshold beyond which the document will continue to be cached to completion, even if the request is cancelled.

The threshold is a percentage specified as a value between `1` and `100`. A value of `0` specifies that the default be used. A value of `100` will only cache documents that are served in their entirety. A value between 60 and 90 is recommended.

```
CacheForceCompletion 80
```

Note:

This feature is currently *not* implemented.

CacheIgnoreCacheControl Directive

Description:	Ignore the fact that the client requested the content not be cached.
Syntax:	`CacheIgnoreCacheControl On|Off`
Default:	`CacheIgnoreCacheControl Off`
Context:	server config, virtual host
Status:	Experimental
Module:	mod_cache

Ordinarily, documents with no-cache or no-store header values will not be stored in the cache. The `CacheIgnoreCacheControl` directive allows this behavior to be overridden. `CacheIgnoreCacheControl` On tells the server to attempt to cache the document even if it contains no-cache or no-store header values. Documents requiring authorization will *never* be cached.

```
CacheIgnoreCacheControl On
```

CacheIgnoreNoLastMod Directive

Description:	Ignore the fact that a response has no Last Modified header.
Syntax:	`CacheIgnoreNoLastMod On|Off`
Default:	`CacheIgnoreNoLastMod Off`
Context:	server config, virtual host
Status:	Experimental
Module:	mod_cache

Ordinarily, documents without a last-modified date are not cached. Under some circumstances the last-modified date is removed (during `mod_include` processing for example) or not provided at all. The `CacheIgnoreNoLastMod` directive provides a way to specify that documents without last-modified dates should be considered for caching, even without a last-modified date. If neither a last-modified date nor an expiry date are provided with the document then the value specified by the `CacheDefaultExpire` directive will be used to generate an expiration date.

```
CacheIgnoreNoLastMod On
```

CacheLastModifiedFactor Directive

Description:	The factor used to compute an expiry date based on the LastModified date.
Syntax:	`CacheLastModifiedFactor` *`float`*
Default:	`CacheLastModifiedFactor 0.1`
Context:	server config, virtual host
Status:	Experimental
Module:	mod_cache

In the event that a document does not provide an expiry date but does provide a last-modified date, an expiry date can be calculated based on the time since the document was last modified. The `CacheLastModifiedFactor` directive specifies a *factor* to be used in the generation of this expiry date according to the following formula: `expiry-period = time-since-last-modified-date * `*`factor`*` expiry-date = current-date + expiry-period` For example, if the document was last modified 10 hours ago, and *factor* is 0.1 then the expiry-period will be set to 10*0.1 = 1 hour. If the current time was 3:00pm then the computed expiry-date would be 3:00pm + 1hour = 4:00pm. If the expiry-period would be longer than that set by `CacheMaxExpire`, then the latter takes precedence.

```
CacheLastModifiedFactor 0.5
```

CacheMaxExpire Directive

Description:	The maximum time in seconds to cache a document
Syntax:	`CacheMaxExpire` *`seconds`*
Default:	`CacheMaxExpire 86400 (one day)`
Context:	server config, virtual host
Status:	Experimental
Module:	mod_cache

The `CacheMaxExpire` directive specifies the maximum number of seconds for which cachable HTTP documents will be retained without checking the origin server. Thus, documents will be out of date at most this number of seconds. This maximum value is enforced even if an expiry date was supplied with the document.

```
CacheMaxExpire 604800
```

URI References

[1] http://www.ietf.org/rfc/rfc2616.txt