Apache Module mod_deflate

Description:	Compress content before it is delivered to the client
Status:	Extension
Module Identifier:	deflate_module
Source File:	mod_deflate.c

Summary

The `mod_deflate` module provides the `DEFLATE` output filter that allows output from your server to be compressed before being sent to the client over the network.

Topics

Directives

See also

- The filter documentation[1]

Recommended Configuration

This is a sample configuration for the impatient. But please take the time and read the sections below for a detailed description!

Compress only a few types

```
AddOutputFilterByType DEFLATE text/html text/plain text/xml
```

Compress everything except images

```
<Location />
  # Insert filter
  SetOutputFilter DEFLATE

  # Netscape 4.x has some problems...
  BrowserMatch ^Mozilla/4 gzip-only-text/html

  # Netscape 4.06-4.08 have some more problems
  BrowserMatch ^Mozilla/4\.0[678] no-gzip

  # MSIE masquerades as Netscape, but it is fine
  BrowserMatch \bMSIE !no-gzip !gzip-only-text/html

  # Don't compress images
  SetEnvIfNoCase Request_URI \
    \.(?:gif|jpe?g|png)$ no-gzip dont-vary

  # Make sure proxies don't deliver the wrong content
  Header append Vary User-Agent env=!dont-vary
</Location>
```

Enabling Compression

Output Compression

Compression is implemented by the `DEFLATE` filter[1]. The following directive will enable compression for documents in the container where it is placed:

```
SetOutputFilter DEFLATE
```

Some popular browsers cannot handle compression of all content so you may want to set the `gzip-only-text/html` note to `1` to only allow html files to be compressed (see below). If you set this to *anything but* `1` it will be ignored.

If you want to restrict the compression to particular MIME types in general, you may use the `AddOutputFilterByType` directive. Here is an example of enabling compression only for the html files of the Apache documentation:

```
<Directory "/your-server-root/manual">
  AddOutputFilterByType DEFLATE text/html
</Directory>
```

For browsers that have problems even with compression of all file types, use the `BrowserMatch` directive to set the `no-gzip` note for that particular browser so that no compression will be performed. You may combine `no-gzip` with `gzip-only-text/html` to get the best results. In that case the former overrides the latter. Take a look at the following excerpt from the configuration example defined in the section above:

```
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4\.0[678] no-gzip
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
```

At first we probe for a `User-Agent` string that indicates a Netscape Navigator version of 4.x. These versions cannot handle compression of types other than `text/html`. The versions 4.06, 4.07 and 4.08 also have problems with decompressing html files. Thus, we completely turn off the deflate filter for them.

The third `BrowserMatch` directive fixes the guessed identity of the user agent, because the Microsoft Internet Explorer identifies itself also as "Mozilla/4" but is actually able to handle requested compression. Therefore we match against the additional string "MSIE" (`\b` means "word boundary") in the `User-Agent` Header and turn off the restrictions defined before.

> **Note**
>
> The `DEFLATE` filter is always inserted after RESOURCE filters like PHP or SSI. It never touches internal subrequests.

Input Decompression

The `mod_deflate` module also provides a filter for decompressing a gzip compressed request body . In order to activate this feature you have to insert the `DEFLATE` filter into the input filter chain using

SetInputFilter or AddInputFilter, for example:

```
<Location /dav-area>
  SetInputFilter DEFLATE
</Location>
```

Now if a request contains a `Content-Encoding: gzip` header, the body will be automatically decompressed. Few browsers have the ability to gzip request bodies. However, some special applications actually do support request compression, for instance some WebDAV[2] clients.

Note on Content-Length

If you evaluate the request body yourself, *don't trust the `Content-Length` header!* The Content-Length header reflects the length of the incoming data from the client and *not* the byte count of the decompressed data stream.

Dealing with proxy servers

The mod_deflate module sends a `Vary: Accept-Encoding` HTTP response header to alert proxies that a cached response should be sent only to clients that send the appropriate `Accept-Encoding` request header. This prevents compressed content from being sent to a client that will not understand it.

If you use some special exclusions dependent on, for example, the `User-Agent` header, you must manually configure an addition to the `Vary` header to alert proxies of the additional restrictions. For example, in a typical configuration where the addition of the `DEFLATE` filter depends on the `User-Agent`, you should add:

```
Header append Vary User-Agent
```

If your decision about compression depends on other information than request headers (*e.g.* HTTP version), you have to set the `Vary` header to the value `*`. This prevents compliant proxies from caching entirely.

Example

```
Header set Vary *
```

DeflateBufferSize Directive

Description:	Fragment size to be compressed at one time by zlib
Syntax:	`DeflateBufferSize` *`value`*
Default:	`DeflateBufferSize 8096`
Context:	server config, virtual host
Status:	Extension
Module:	mod_deflate

The DeflateBufferSize directive specifies the size in bytes of the fragments that zlib should compress at one time.

DeflateFilterNote Directive

Description:	Places the compression ratio in a note for logging
Syntax:	`DeflateFilterNote` *`notename`*
Context:	server config, virtual host
Status:	Extension
Module:	mod_deflate

The `DeflateFilterNote` directive specifies that a note about compression ratios should be attached to the request. The name of the note is the value specified for the directive. You can use that note for statistical purposes by adding the value to your access log[3].

Example

```
DeflateFilterNote ratio

LogFormat '"%r" %b (%{ratio}n) "%{User-agent}i"' deflate
CustomLog logs/deflate_log deflate
```

See also

- `mod_log_config`

DeflateMemLevel Directive

Description:	How much memory should be used by zlib for compression
Syntax:	`DeflateMemLevel` *`value`*
Default:	`DeflateMemLevel 9`
Context:	server config, virtual host
Status:	Extension
Module:	mod_deflate

The `DeflateMemLevel` directive specifies how much memory should be used by zlib for compression (a value between 1 and 9).

DeflateWindowSize Directive

Description:	Zlib compression window size
Syntax:	`DeflateWindowSize` *`value`*
Default:	`DeflateWindowSize 15`
Context:	server config, virtual host
Status:	Extension
Module:	mod_deflate

The `DeflateWindowSize` directive specifies the zlib compression window size (a value between 1 and 15). Generally, the higher the window size, the higher can the compression ratio be expected.

URI References

[1] http://httpd.apache.org/docs-2.1/filter.html
[2] http://www.webdav.org
[3] http://httpd.apache.org/docs-2.1/logs.html#accesslog