

GzipPipeline

Gzip pipeline

Overview

In this wiki there are other alternatives for generating gzip compressed output, a feature of HTTP 1.1. The advantage of this implementation is that it doesn't require any specific serializer or reader implementation. Unlike implementations that use servlet filters, gzip compression can be controlled from the sitemap, and content-length calculation is determined by serializers and readers.

Usage

First declare the new components in the sitemap:

```
<map:actions>
  <map:action name="gzip" logger="pipe.gzip" src="fcc.ima.cocoon.GzipAction" />
</map:actions>
<map:pipes default="gzip-no-cache">
  <map:pipe name="gzip-no-cache" src="fcc.ima.cocoon.GzipNonCachingProcessingPipeline"
    logger="pipe.gzip" />
</map:pipes>
```

Create a pipe that is gzip aware:

```
<map:pipelines>
  <map:pipeline type="gzip-no-cache">
    <map:match src="my-url">
      <map:act type='gzip' />
      <map:read src='my-file' />
    </map:match>
  </map:pipeline>
</map:pipelines>
```

In order to activate gzip compression use:

```
<map:act type='gzip' />
```

and for deactivating it:

```
<map:act type='gzip'>
  <map:parameter name='use' value='0' />
</map:act>
```

You can (de)activate compression as many times as needed in a pipeline.

Source code

See attachments.

Notes

- Mozilla 1.5 seems not to like compressed files.
- jpg and gif files are already compressed, so compress them twice isn't very useful.

To do

- Make a caching implementation

See also

- [GzipXMLSerializer](#)
- [Cocoon%and%Apache%mod%gzip](#)

Comments and improvements are welcome!.

My email address is dperezcar at fcc dot es (rename at and dot).

Attachment: [GzipNonCachingProcessingPipeline.java](#)

Attachment: [GzipAction.java](#)