NAME

PerlIO::gzip – Perl extension to provide a PerlIO layer to gzip/gunzip

SYNOPSIS

```
use PerlIO::gzip;
open FOO, "<:gzip", "file.gz" or die $!;
print while <FOO>; # And it will be uncompressed...

binmode FOO, ":gzip(none)" # Starts reading deflate stream from here on
```

DESCRIPTION

PerlIO::gzip provides a PerlIO layer that manipulates files in the format used by the `gzip` program. Compression and Decompression are implemented, but not together. If you attempt to open a file for reading and writing the open will fail.

EXPORT

PerlIO::gzip exports no subroutines or symbols, just a perl layer `gzip`

LAYER ARGUMENTS

The `gzip` layer takes a comma separated list of arguments. 4 exclusive options choose the header checking mode:

gzip
: The default. Expects a standard gzip file header for reading, writes a standard gzip file header.

none
: Expects or writes no file header; assumes the file handle is immediately a deflate stream (eg as would be found inside a `zip` file)

auto
: Potentially dangerous. If the first two bytes match the `gzip` header "\x1f\x8b" then a gzip header is assumed (and checked) else a deflate stream is assumed. No different from gzip on writing.

autopop
: Potentially dangerous. If the first two bytes match the `gzip` header "\x1f\x8b" then a gzip header is assumed (and checked) else the layer is silently popped. This results in gzip files being transparently decompressed, other files being treated normally. Of course, this has sides effects such as File::Copy becoming gunzip, and File::Compare comparing the uncompressed contents of files.

  In autopop mode Opening a handle for writing (or reading and writing) will cause the gzip layer to automatically be popped.

Optionally you can add this flag:

lazy
: For reading, defer header checking until the first read. For writing, don't write a header until the first buffer empty of compressed data to disk. (and don't write anything at all if no data was written to the handle)

  By default, gzip header checking is done before the `open` (or `binmode`) returns, so if an error is detected in the gzip header the `open` or `binmode` will fail. However, this will require reading some data, or writing a header. With lazy set on a file opened for reading the check is deferred until the first read so the `open` should always succeed, but any problems with the header will cause an error on read.

  ```
  open FOO, "<:gzip(lazy)", "file.gz" or die $!; # Dangerous.
  while (<FOO>) {
    print;
  } # Whoa. Bad. You're not distinguishing between errors and EOF.
  ```

  If you're not careful you won't spot the errors – like the example above you'll think you got end of file.

lazy is ignored if you are in autopop mode.

AUTHOR

Nicholas Clark, <nwc10+perlio−gzip@colon.colondot.net>

SEE ALSO

perl, gzip, rfc 1952 <http://www.ietf.org/rfc/rfc1952.txt> (the gzip file format specification), rfc 1951 <http://www.ietf.org/rfc/rfc1951.txt> (DEFLATE compressed data format specification)