

NAME

Archive::Zip::MemberRead – A wrapper that lets you read Zip archive members as if they were files.

SYNOPSIS

```
use Archive::Zip;
use Archive::Zip::MemberRead;
$zip = Archive::Zip->new("file.zip");
$fzh = Archive::Zip::MemberRead->new($zip, "subdir/abc.txt");
while (defined($line = $fzh->getline()))
{
    print $fzh->input_line_number . " #: $line\n";
}

$read = $fzh->read($buffer, 32*1024);
print "Read $read bytes as :$buffer:\n";
```

DESCRIPTION

The Archive::Zip::MemberRead module lets you read Zip archive member data just like you read data from files.

METHODS**Archive::Zip::Member::readFileHandle()**

You can get a Archive::Zip::MemberRead from an archive member by calling readFileHandle():

```
my $member = $zip->memberNamed('abc/def.c');
my $fzh = $member->readFileHandle();
while (defined($line = $fzh->getline()))
{
    # ...
}
$fzh->close();
```

Archive::Zip::MemberRead->new(\$zip, \$fileName)

Archive::Zip::MemberRead->new(\$zip, \$member)

Archive::Zip::MemberRead->new(\$member)

Construct a new Archive::Zip::MemberRead on the specified member.

```
my $fzh = Archive::Zip::MemberRead->new($zip, 'fred.c')
```

setLineEnd(expr)

Set the line end character to use. This is set to \n by default except on Windows systems where it is set to \r\n. You will only need to set this on systems which are not Windows or Unix based and require a line end different from \n. This is a class method so call as Archive::Zip::MemberRead->setLineEnd(\$nl)

rewind()

Rewinds an Archive::Zip::MemberRead so that you can read from it again starting at the beginning.

input_record_separator(expr)

If the argument is given, input_record_separator for this instance is set to it. The current setting (which may be the global \$/) is always returned.

input_line_number()

Returns the current line number, but only if you're using getline(). Using read() will not update the line number.

close()

Closes the given file handle.

`buffer_size([$size])`

Gets or sets the buffer size used for reads. Default is the chunk size used by Archive::Zip.

getline()

Returns the next line from the currently open member. Makes sense only for text files. A read error is considered fatal enough to die. Returns undef on eof. All subsequent calls would return undef, unless a **rewind()** is called. Note: The line returned has the `input_record_separator` (default: newline) removed.

`getline({ preserve_line_ending => 1 })`

Returns the next line including the line ending.

`read($buffer, $num_bytes_to_read)`

Simulates a normal `read()` system call. Returns the no. of bytes read. undef on error, 0 on eof, *e.g.:*

```
$fh = Archive::Zip::MemberRead->new($zip, "sreeji/secrets.bin");
while (1)
{
    $read = $fh->read($buffer, 1024);
    die "FATAL ERROR reading my secrets !\n" if (!defined($read));
    last if (!$read);
    # Do processing.
    ....
}
```

AUTHOR

Sreeji K. Das <sreeji_k@yahoo.com>

See Archive::Zip by Ned Konz without which this module does not make any sense!

Minor mods by Ned Konz.

COPYRIGHT

Copyright 2002 Sreeji K. Das.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.