1 APR::PerlIO -- Perl IO layer for APR

1.1 Synopsis

```
# under mod_perl
use APR::PerlIO ();

sub handler {
    my $r = shift;

    die "This Perl build doesn't support PerlIO layers"
        unless APR::PerlIO::PERLIO_LAYERS_ARE_ENABLED;

    open my $fh, ">:APR", $filename, $r->pool or die $!;
    # work with $fh as normal $fh
    close $fh;

    return Apache2::Const::OK;
}

# outside mod_perl
% perl -MAPR -MAPR::PerlIO -MAPR::Pool -le \
'open my $fh, ">:APR", "/tmp/apr", APR::Pool->new or die "$!"; \
 print $fh "whoah!"; \
 close $fh;'
```

1.2 Description

`APR::PerlIO` implements a Perl IO layer using APR's file manipulation API internally.

Why do you want to use this? Normally you shouldn't, probably it won't be faster than Perl's default layer. It's only useful when you need to manipulate a filehandle opened at the APR side, while using Perl.

Normally you won't call open() with APR layer attribute, but some mod_perl functions will return a filehandle which is internally hooked to APR. But you can use APR Perl IO directly if you want.

1.3 Prerequisites

Not every Perl will have full `APR::PerlIO` functionality available.

Before using the Perl IO APR layer one has to check whether it's supported by the used APR/Perl build. Perl 5.8.x or higher with perlio enabled is required. You can check whether your Perl fits the bill by running:

```
% perl -V:useperlio
useperlio='define';
```

It should say *define*.

If you need to do the checking in the code, there is a special constant provided by `APR::PerlIO`, which can be used as follows:

```
use APR::PerlIO ();
die "This Perl build doesn't support PerlIO layers"
    unless APR::PerlIO::PERLIO_LAYERS_ARE_ENABLED;
```

Notice that loading `APR::PerlIO` won't fail when Perl IO layers aren't available since `APR::PerlIO` provides functionality for Perl builds not supporting Perl IO layers.

1.4 Constants

1.4.1 ***`APR::PerlIO::PERLIO_LAYERS_ARE_ENABLED`***

See Prerequisites.

1.5 API

Most of the API is as in normal perl IO with a few nuances listed in the following sections.

META: need to rework the exception mechanism here. Current success in using errno ($!) being set (e.g. on open()) is purely accidental and not guaranteed across all platforms and functions. So don't rely on $!. Will use `APR::Error` for that purpose.

1.5.1 ***`open`***

Open a file via APR Perl IO layer.

```
open my $fh, ">:APR", $filename, $r->pool or die $!;
```

- **arg1: `$fh` (GLOB filehandle)**

  The filehandle.

- **arg2: `$mode` (string)**

  The mode to open the file, constructed from two sections separated by the `:` character: the first section is the mode to open the file under (>, <, etc) and the second section must be a string *APR*. For more information refer to the *open* entry in the *perlfunc* manpage.

- **arg3: `$filename` (string)**

  The path to the filename to open

- **arg4: `$p` (`APR::Pool`)**

  The pool object to use to allocate APR::PerlIO layer.

- **ret: (integer)**

success or failure value (boolean).

- **since: 2.0.00**

1.5.2 `seek`

Sets `$fh`'s position, just like the `seek()` Perl call:

```
  seek($fh, $offset, $whence);
```

If `$offset` is zero, `seek()` works normally.

However if `$offset` is non-zero and Perl has been compiled with with large files support (`-Duselargefiles`), whereas APR wasn't, this function will croak. This is because largefile size `Off_t` simply cannot fit into a non-largefile size `apr_off_t`.

To solve the problem, rebuild Perl with `-Uuselargefiles`. Currently there is no way to force APR to build with large files support.

- **since: 2.0.00**

1.6 C API

The C API provides functions to convert between Perl IO and APR Perl IO filehandles.

META: document these

1.7 See Also

mod_perl 2.0 documentation. The *perliol(1)*, *perlapio(1)* and *perl(1)* manpages.

1.8 Copyright

mod_perl 2.0 and its core modules are copyrighted under The Apache Software License, Version 2.0.

1.9 Authors

The mod_perl development team and numerous contributors.

Table of Contents: