NAME

IO::File - supply object methods for filehandles

SYNOPSIS

```
use IO::File;

$fh = new IO::File;
if ($fh->open("< file")) {
    print <$fh>;
    $fh->close;
}

$fh = new IO::File "> file";
if (defined $fh) {
    print $fh "bar\n";
    $fh->close;
}

$fh = new IO::File "file", "r";
if (defined $fh) {
    print <$fh>;
    undef $fh;       # automatically closes the file
}

$fh = new IO::File "file", O_WRONLY|O_APPEND;
if (defined $fh) {
    print $fh "corge\n";

    $pos = $fh->getpos;
    $fh->setpos($pos);

    undef $fh;       # automatically closes the file
}

autoflush STDOUT 1;
```

DESCRIPTION

`IO::File` inherits from `IO::Handle` and `IO::Seekable`. It extends these classes with methods that are specific to file handles.

CONSTRUCTOR

new (FILENAME [,MODE [,PERMS]])

Creates an `IO::File`. If it receives any parameters, they are passed to the method `open`; if the open fails, the object is destroyed. Otherwise, it is returned to the caller.

new_tmpfile

Creates an `IO::File` opened for read/write on a newly created temporary file. On systems where this is possible, the temporary file is anonymous (i.e. it is unlinked after creation, but held open). If the temporary file cannot be created or opened, the `IO::File` object is destroyed. Otherwise, it is returned to the caller.

METHODS

open(FILENAME [,MODE [,PERMS]])

open(FILENAME, IOLAYERS)

`open` accepts one, two or three parameters. With one parameter, it is just a front end for the built-in `open` function. With two or three parameters, the first parameter is a filename that may include whitespace or other special characters, and the second parameter is the open mode, optionally followed by a file permission value.

If `IO::File::open` receives a Perl mode string (">", "+<", etc.) or an ANSI C fopen() mode string ("w", "r+", etc.), it uses the basic Perl `open` operator (but protects any special characters).

If `IO::File::open` is given a numeric mode, it passes that mode and the optional permissions value to the Perl `sysopen` operator. The permissions default to 0666.

If `IO::File::open` is given a mode that includes the `:` character, it passes all the three arguments to the three-argument `open` operator.

For convenience, `IO::File` exports the O_XXX constants from the Fcntl module, if this module is available.

binmode([LAYER])

`binmode` sets `binmode` on the underlying `IO` object, as documented in `perldoc -f binmode`.

`binmode` accepts one optional parameter, which is the layer to be passed on to the `binmode` call.

NOTE

Some operating systems may perform `IO::File::new()` or `IO::File::open()` on a directory without errors. This behavior is not portable and not suggested for use. Using `opendir()` and `readdir()` or `IO::Dir` are suggested instead.

SEE ALSO

perlfunc, *"I/O Operators" in perlop*, *IO::Handle*, *IO::Seekable*, *IO::Dir*

HISTORY

Derived from FileHandle.pm by Graham Barr <*gbarr@pobox.com*>.