NAME

IO−stringy − I/O on in−core objects like strings and arrays

SYNOPSIS

```
IO::
::AtomicFile    adpO   Write a file which is updated atomically    ERYQ
::Lines         bdpO   I/O handle to read/write to array of lines  ERYQ
::Scalar        RdpO   I/O handle to read/write to a string        ERYQ
::ScalarArray   RdpO   I/O handle to read/write to array of scalars ERYQ
::Wrap          RdpO   Wrap old−style FHs in standard OO interface  ERYQ
::WrapTie       adpO   Tie your handles & retain full OO interface  ERYQ
```

DESCRIPTION

This toolkit primarily provides modules for performing both traditional and object-oriented i/o) on things *other* than normal filehandles; in particular, IO::Scalar, IO::ScalarArray, and IO::Lines.

In the more-traditional IO::Handle front, we have IO::AtomicFile which may be used to painlessly create files which are updated atomically.

And in the ''this-may-prove-useful'' corner, we have IO::Wrap, whose exported **wraphandle()** function will clothe anything that's not a blessed object in an IO::Handle−like wrapper... so you can just use OO syntax and stop worrying about whether your function's caller handed you a string, a globref, or a FileHandle.

WARNINGS

Perl's TIEHANDLE spec was incomplete prior to 5.005_57; it was missing support for `seek()`, `tell()`, and `eof()`. Attempting to use these functions with an IO::Scalar, IO::ScalarArray, IO::Lines, etc. **will not work** prior to 5.005_57. None of the relevant methods will be invoked by Perl; and even worse, this kind of bug can lie dormant for a while. If you turn warnings on (via $ˆW or `perl -w`), and you see something like this...

```
seek() on unopened file
```

...then you are probably trying to use one of these functions on one of our IO:: classes with an old Perl. The remedy is to simply use the OO version; e.g.:

```
$SH−>seek(0,0);      ### GOOD: will work on any 5.005
seek($SH,0,0);       ### WARNING: will only work on 5.005_57 and beyond
```

INSTALLATION

Requirements

As of version 2.x, this toolkit requires Perl 5.005 for the IO::Handle subclasses, and 5.005_57 or better is **strongly** recommended. See ''WARNINGS'' for details.

Directions

Most of you already know the drill...

```
perl Makefile.PL
make
make test
make install
```

For everyone else out there... if you've never installed Perl code before, or you're trying to use this in an environment where your sysadmin or ISP won't let you do interesting things, **relax:** since this module contains no binary extensions, you can cheat. That means copying the directory tree under my ''./lib'' directory into someplace where your script can ''see'' it. For example, under Linux:

```
cp −r IO-stringy-1.234/lib/* /path/to/my/perl/
```

Now, in your Perl code, do this:

```
use lib "/path/to/my/perl";
use IO::Scalar;                    ### or whatever
```

Ok, now you've been told. At this point, anyone who whines about not being given enough information gets an unflattering haiku written about them in the next change log. I'll do it. Don't think I won't.

VERSION

`$Id: Stringy.pm,v 1.3 2005/02/10 21:24:05 dfs Exp $`

TO DO

(2000/08/02) Finalize $/ support

Graham Barr submitted this patch half a *year* ago; Like a moron, I lost his message under a ton of others, and only now have the experimental implementation done.

Will the sudden sensitivity to $/ hose anyone out there? I'm worried, so you have to enable it explicitly in 1.x. It will be on by default in 2.x, though only IO::Scalar has been implemented.

(2001/08/08) Remove IO::WrapTie from new IO:: classes

It's not needed. Backwards compatibility could be maintained by having **new_tie()** be identical to **new()**. Heck, I'll bet that IO::WrapTie should be reimplemented so the returned object is just like an IO::Scalar in its use of globrefs.

CHANGE LOG

Version 2.110   (2005/02/10)

Maintainership taken over by DSKOLL <dfs@roaringpenguin.com>

Closed the following bugs at https://rt.cpan.org/NoAuth/Bugs.html?Dist=IO−stringy:

```
2208 IO::ScalarArray->getline does not return undef for EOF if undef($/)
7132 IO-stringy/Makefile.PL bug - name should be module name
11249 IO::Scalar flush shouldn't return undef
2172 $\ (output record separator) not respected
8605 IO::InnerFile::seek() should return 1 on success
4798 *.html in lib/
4369 Improvement: handling of fixed-size reads in IO::Scalar
     (Actually, bug 4369 was closed in Version 2.109)
```

Version 2.109   (2003/12/21)

IO::Scalar::getline now works with ref to int. *Thanks to Dominique Quatravaux for this patch.*

Version 2.108   (2001/08/20)

The terms-of-use have been placed in the distribution file "COPYING". Also, small documentation tweaks were made.

Version 2.105   (2001/08/09)

Added support for various **seek()** whences to IO::ScalarArray.

Added support for consulting $/ in IO::Scalar and IO::ScalarArray. The old `use_RS()` is not even an option. Unsupported record separators will cause a **croak()**.

Added a lot of regression tests to supoprt the above.

Better on-line docs (hyperlinks to individual functions).

Version 2.103   (2001/08/08)

After sober consideration I have reimplemented **IO::Scalar::print**() so that it once again always seeks to the end of the string. Benchmarks show the new implementation to be just as fast as Juergen's contributed patch; until someone can convince me otherwise, the current, safer implementation stays.

I thought more about giving IO::Scalar two separate handles, one for reading and one for writing, as suggested by Binkley. His points about what **tell()** and **eof()** return are, I think, show-stoppers for this feature. Even the manpages for stdio's **fseek()** seem to imply a *single* file position indicator, not two. So I think I will take this off the TO DO list. **Remedy:** you can always have two handles open on the same scalar, one which you only write to, and one which you only read from. That should give the same effect.

Version 2.101   (2001/08/07)

Alpha release. This is the initial release of the "IO::Scalar and friends are now subclasses of IO::Handle". I'm flinging it against the wall. Please tell me if the banana sticks. When it does, the banana will be called 2.2x.

First off, *many many thanks to Doug Wilson*, who has provided an *invaluable* service by patching IO::Scalar and friends so that they (1) inherit from IO::Handle, (2) automatically tie themselves so that the `new()` objects can be used in native i/o constructs, and (3) doing it so that the whole damn thing passes its regression tests. As Doug knows, my globref Kung-Fu was not up to the task; he graciously provided the patches. This has earned him a seat at the Co-Authors table, and the right to have me address him as *sensei*.

Performance of **IO::Scalar::print()** has been improved by as much as 2x for lots of little prints, with the cost of forcing those who print-then-seek-then-print to explicitly seek to end-of-string before printing again. *Thanks to Juergen Zeller for this patch.*

Added the COPYING file, which had been missing from prior versions. *Thanks to Albert Chin-A-Young for pointing this out.*

IO::Scalar consults $/ by default (1.x ignored it by default). Yes, I still need to support IO::ScalarArray.

Version 1.221   (2001/08/07)

I threatened in "INSTALLATION" to write an unflattering haiku about anyone who whined that I gave them insufficient information... but it turns out that I left out a crucial direction. D'OH! *Thanks to David Beroff for the "patch" and the haiku...*

```
        Enough info there?
          Here's unflattering haiku:
        Forgot the line, "make"!  ;-)
```

Version 1.220   (2001/04/03)

Added untested SEEK, TELL, and EOF methods to IO::Scalar and IO::ScalarArray to support corresponding functions for tied filehandles: untested, because I'm still running 5.00556 and Perl is complaining about "**tell()** on unopened file". *Thanks to Graham Barr for the suggestion.*

Removed not-fully-blank lines from modules; these were causing lots of POD-related warnings. *Thanks to Nicolas Joly for the suggestion.*

Version 1.219   (2001/02/23)

IO::Scalar objects can now be made sensitive to $/ . Pains were taken to keep the fast code fast while adding this feature. *Cheers to Graham Barr for submitting his patch; jeers to me for losing his email for 6 months.*

Version 1.218   (2001/02/23)

IO::Scalar has a new **sysseek()** method. *Thanks again to Richard Jones.*

New "TO DO" section, because people who submit patches/ideas should at least know that they're in the system... and that I won't lose their stuff. Please read it.

New entries in "AUTHOR". Please read those too.

Version 1.216   (2000/09/28)

IO::Scalar and IO::ScalarArray now inherit from IO::Handle. I thought I'd remembered a problem with this ages ago, related to the fact that these IO:: modules don't have "real" filehandles, but the problem apparently isn't surfacing now. If you suddenly encounter Perl warnings during global destruction (especially if you're using tied filehandles), then please let me know! *Thanks to B. K. Oxley (binkley) for this.*

Nasty bug fixed in IO::Scalar::write(). Apparently, the offset and the number-of-bytes arguments were, for all practical purposes, *reversed.* You were okay if you did all your writing with **print()**, but boy was *this* a stupid bug! *Thanks to Richard Jones for finding this one. For you, Rich, a double-*

length haiku:

```
        Newspaper headline
           typeset by dyslexic man
        loses urgency

        BABY EATS FISH is
           simply not equivalent
        to FISH EATS BABY
```

New sysread and syswrite methods for IO::Scalar. *Thanks again to Richard Jones for this.*

Version 1.215　(2000/09/05)
Added 'bool' overload to '""' overload, so object always evaluates to true. (Whew. Glad I caught this before it went to CPAN.)

Version 1.214　(2000/09/03)
Evaluating an IO::Scalar in a string context now yields the underlying string. *Thanks to B. K. Oxley (binkley) for this.*

Version 1.213　(2000/08/16)
Minor documentation fixes.

Version 1.212　(2000/06/02)
Fixed IO::InnerFile incompatibility with Perl5.004. *Thanks to many folks for reporting this.*

Version 1.210　(2000/04/17)
Added **flush()** and other no-op methods. *Thanks to Doru Petrescu for suggesting this.*

Version 1.209　(2000/03/17)
Small bug fixes.

Version 1.208　(2000/03/14)
Incorporated a number of contributed patches and extensions, mostly related to speed hacks, support for "offset", and WRITE/CLOSE methods. *Thanks to Richard Jones, Doru Petrescu, and many others.*

Version 1.206　(1999/04/18)
Added creation of ./testout when Makefile.PL is run.

Version 1.205　(1999/01/15)
Verified for Perl5.005.

Version 1.202　(1998/04/18)
New IO::WrapTie and IO::AtomicFile added.

Version 1.110
Added IO::WrapTie.

Version 1.107
Added IO::Lines, and made some bug fixes to IO::ScalarArray. Also, added **getc()**.

Version 1.105
No real changes; just upgraded IO::Wrap to have a `$VERSION` string.

AUTHOR

Primary Maintainer
Dianne Skoll (*dfs@roaringpenguin.com*).

Original Author
Eryq (*eryq@zeegee.com*). President, ZeeGee Software Inc (*http://www.zeegee.com*).

Co-Authors
For all their bug reports and patch submissions, the following are officially recognized:

```
        Richard Jones
        B. K. Oxley (binkley)
        Doru Petrescu
        Doug Wilson (for picking up the ball I dropped, and doing tie() right)
```

Go to *http://www.zeegee.com* for the latest downloads and on-line documentation for this module.

Enjoy. Yell if it breaks.