NAME

IO::AtomicFile – write a file which is updated atomically

SYNOPSIS

```
use IO::AtomicFile;

### Write a temp file, and have it install itself when closed:
my $FH = IO::AtomicFile->open("bar.dat", "w");
print $FH "Hello!\n";
$FH->close || die "couldn't install atomic file: $!";

### Write a temp file, but delete it before it gets installed:
my $FH = IO::AtomicFile->open("bar.dat", "w");
print $FH "Hello!\n";
$FH->delete;

### Write a temp file, but neither install it nor delete it:
my $FH = IO::AtomicFile->open("bar.dat", "w");
print $FH "Hello!\n";
$FH->detach;
```

DESCRIPTION

This module is intended for people who need to update files reliably in the face of unexpected program termination.

For example, you generally don't want to be halfway in the middle of writing */etc/passwd* and have your program terminate! Even the act of writing a single scalar to a filehandle is *not* atomic.

But this module gives you true atomic updates, via **rename**(). When you open a file */foo/bar.dat* via this module, you are *actually* opening a temporary file */foo/bar.dat..TMP*, and writing your output there. The act of closing this file (either explicitly via **close**(), or implicitly via the destruction of the object) will cause **rename**() to be called... therefore, from the point of view of the outside world, the file's contents are updated in a single time quantum.

To ensure that problems do not go undetected, the ''close'' method done by the destructor will raise a fatal exception if the **rename**() fails. The explicit **close**() just returns undef.

You can also decide at any point to trash the file you've been building.

AUTHOR

Primary Maintainer

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

Original Author

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

REVISION

$Revision: 1.2 $