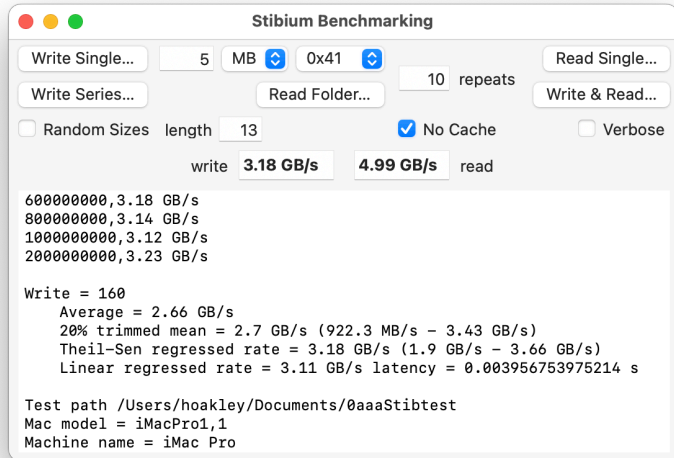


→ [Details](#)

# Start



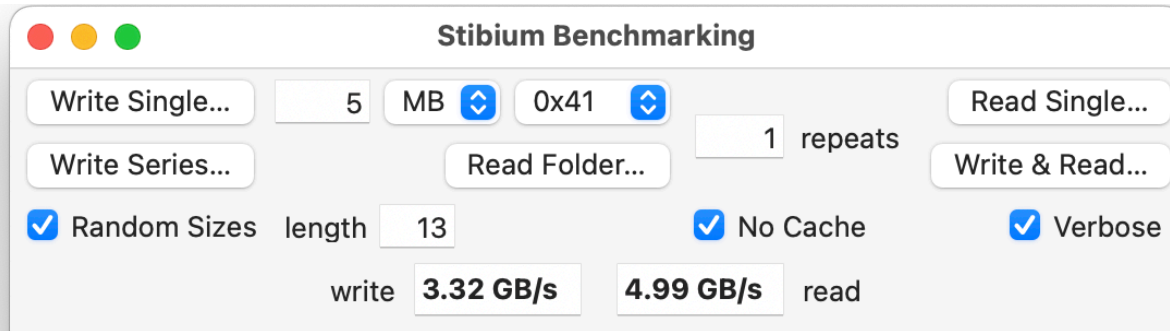
Stibium is the start of an app intended to produce ‘real world’ benchmarks for storage (SSDs in particular) across the full range of Intel and Apple Silicon Macs. Rather than use low-level access to produce an abstract benchmark for storage, it works at a similar level to the apps that we use daily. It’s therefore hoped that the results which it delivers reflect more accurately the performance you’ll experience when using regular apps in real life.

→ [Details](#)

→ [Rationale](#)

→ [Technical information](#)

# Details



Stibium's single window consists of a set of controls above a scrolling text view in which its results are displayed. A future version will add graphical summaries of results in separate windows.

Change the size of the results text using ⌘+ to enlarge, or ⌘- to reduce. Results text (and the values in the write and read boxes) can be copied using ⌘C for pasting into other apps. Use ⌘A to select all the text available. Use the **Save Report...** command in the **File** menu, or ⌘S, to export all the results text into a text file. When you close its window, Stibium quits, saving its settings first. The last results in the read and write boxes are also saved, so they will be loaded when you next run Stibium.

## Run different tests:

→ ['Gold Standard'](#)

Series tests: → [Write](#) → [Read](#)

→ [Write & Read](#)

Random tests: → [Write](#) → [Read](#)

Single sizes: → [Write](#) → [Read](#)

→ [Benchmarking](#)

→ [Controls & Settings](#)

→ [Test Methods](#)

→ [Data Analysis](#)

→ [Technical Information](#)

→ [Change List](#)

# Rationale

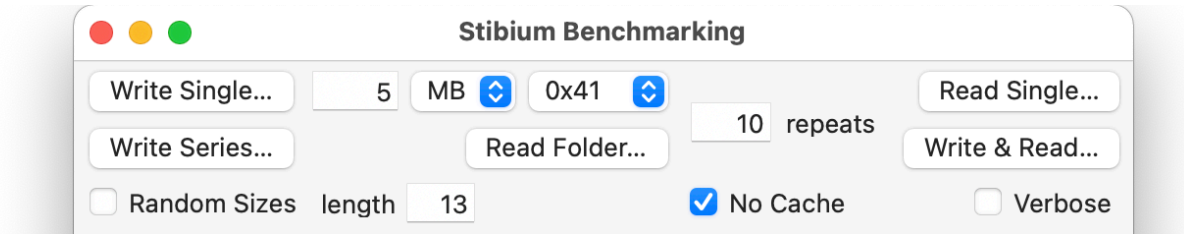
I once cherished occasional rides in the second-fastest E-type Jaguar in the UK. My friend Ken, who took it to drag races each weekend, worked hard to maximise performance from its huge petrol engine. In the dead of night, when no one was around, he used to time its acceleration from 80 to 120 miles per hour (130-190 km/hour). On public roads.

His benchmark was based on what was, to him, the critical point of his drag start, where he won or lost. When we discuss performance of Macs, that's the sort of measurement that we need. When it comes to assessing Apple's internal SSDs, this is a serious business too: Apple currently charges the user around £/\$/€ 400 per TB of SSD, and £/\$/€ 200 for additional memory to take an M1 from 8 to 16 GB. Even if cost is no obstacle, before you commit any money you need to know whether the current maximum of 16 GB of RAM is sufficient.

Car analogies with computers fail in the end, but one common essential in both these benchmark tests is that we know exactly what we're talking about. Ken used a long, flat and straight section of country road for his testing, which was very similar to a dragstrip. No matter where we look for SSD performance benchmarks, details of the tests performed seem rather ill-defined, and none appears to bear much resemblance to the real world. This Help book explains exactly what Sibium does and doesn't do.

→ [Benchmarking](#)

# Run the ‘Gold Standard’ tests



Choose which disk you're going to benchmark, and create on it a new folder to contain the test files, in a folder to which you have read and write permissions. For example, if you want to test the internal SSD holding your Documents folder, create a new folder in that named OStibiumTest or similar. This disk requires around 54 GB free space for those files.

Then restart your Mac, and leave it a minute or two for everything startup to settle. Ensure that it's not due to undertake any scheduled activity including making backups, and leave all apps other than Finder closed. When you're ready, open Stibium and set it up so:

- set the popup menu in the top row to 0x41
- next to it, tick ✓ **No Cache**
- set the number in the **repeats** box to 10
- unless you want to see each individual result, leave the **Verbose** checkbox unticked.

When you're happy those are correct, click on the **Write Series...** button in the middle row. This opens a window in which you should select your test folder and then click **Write**. Then leave your mouse/trackpad alone so that you don't disturb your Mac again until text appears below in the results.

The **write** box in the lower row of controls then displays the write speed for that test, and below that is a detailed report. Use the **Save Report...** command in the **File** menu to save that in a text file, but *not* inside your folder full of test files.

→ [concluded](#)

## *Gold Standard' tests concluded*

Now change the number in the **repeats** box from 10 to 1 ready for the read test, and quit Stibium. Restart your Mac, give it a minute or two to settle again, open Stibium, check that it's set ready for the read test with the **repeats** set to 1. Then click on the **Read Folder...** button in the middle row of controls.

In the window which then opens, select the folder containing the 160 test files which Stibium wrote previously and click **Read** to run the read test. Leave your mouse/trackpad alone so that you don't disturb your Mac again until text appears in the lower scrolling text view in Stibium, which then contains the results.

Use the **Save Report...** command again to save your read results. You're now free to open up your apps and resume normal work again. Unless you want to perform any more read tests on the original 160 test files, you can now drag that folder to the Trash.

In each of the result reports, look for the following:

- Check whether it's the read or write test first.
- Check that the number of files (n) is 160 in each.
- Look down through median transfer speeds given by file size (starting with 2000000 and ending with 2000000000). These may start rather low and rise by half way down the list. No median should be greatly different from its neighbours.
- There are four different estimates of overall transfer speed: the most reliable is the Theil-Sen regressed rate, calculated from the raw time results to yield an overall speed; the 20% trimmed mean should also be fairly reliable, although usually less than that from regression. The linear regressed rate can be very reliable too, unless there are outliers. Least reliable or representative in this test is the Average.
- Look for evidence of outliers. Sometimes individual tests run very slowly or worryingly fast. Next to the 20% trimmed mean is the 90% range across all 160 measurements. Check that those extremes don't look crazy. If they do, it may well be worth repeating the test(s) to see if they don't reduce and provide more reliable results.

If your results are looking good, they should be representative of the real-world performance of that disk.

# Series Write Tests

**Control:** click on the **Write Series...** button, select a destination folder on the disk to be tested and click **Write**.

**Purpose:** to measure the write transfer time on a series of files with sizes ranging from 2 MB to 2 GB.

## Key Settings:

popup menu in the middle of the top row sets the byte content to be written to the file, from 0x00 to 0x41 (hexadecimal)

**repeats** box sets how many sets of test files are written

**Random Sizes** checkbox must be unchecked (empty)

**Verbose** will report individual file results when ticked ✓

## Details:

Stibium writes a total of 16 files of sizes 2, 4, 6, 8, 10, 20, 40, 60, 80, 100, 200, 400, 600, 800 MB, 1, 2 GB, each containing the set byte contents. Each time it writes a set, the order of sizes is separately randomised to make this a random write test using fixed sizes. Files written are numbered in the order that they are written, not by size. The maximum setting for repeats is 1,000, which would write a total of 16,000 test files.

Each set of 16 test files requires about 54 GB of free disk space. Although each test file is given the extension .tiff, these aren't genuine TIFF files, but consist of the same single byte repeated throughout the entire file.

In some circumstances, file writes could be subject to interference by caching. Normally, this isn't the case, but unusually high transfer rates suggest that may have happened. Look for this in the range shown for the 20% trimmed mean, and in verbose listing of individual files.

→ [Series Read Tests](#)

# Series Read Tests – Read Folder

**Control:** click on the **Read Folder...** button, select the folder you wish Stibium to read and click **Read**.

**Purpose:** to measure the read transfer time on all files of 10 KB and greater size in a single folder, and any folders containing within it.

## Key Settings:

**repeats** box sets how many times the files in the folder are read

**No Cache** checkbox sets whether the 'no caching' read method is used, recommended to be ticked ✓

**Verbose** will report individual file results when ticked ✓

## Details:

Stibium reads all the files in the selected folder in an order which is determined by the file system. It repeats the reads according to the setting in the **repeats** box. The order of file reading appears quite random, and isn't the same across each repeat. This test is therefore a random read test using fixed sizes. The maximum setting for repeats is 1,000.

Because macOS has a habit of writing hidden files to folders, the smallest file to be read and its results included is 10 KB in size. All files smaller than that are ignored.

This test can be performed on *any* folder to which you have read access, including those in the System volume. It iterates through the contents of any folders inside the selected folder, and reads all files within those as well.

In some circumstances, repeated file reads could be subject to interference by caching. Normally, this isn't the case, but unusually high transfer rates suggest that may have happened. Look for this in the range shown for the 20% trimmed mean, and in verbose listing of individual files.

→ [Series Write Tests](#)

# Series Write & Read Tests

**Control:** click on the **Write & Read...** button, select the folder to be used to contain the files which are first written then read, and click **Write**.

**Purpose:** to measure the write and read transfer times on a series of files with sizes ranging from 2 MB to 2 GB.

## Key Settings:

popup menu in the middle of the top row sets the byte content to be written to the file, from 0x00 to 0x41 (hexadecimal)  
**repeats** box sets how many times the test files are *read*; it doesn't change the number of files *written*, though, which is fixed at 16, for a single cycle of write tests

**Random Sizes** checkbox must be unchecked (empty)

**Verbose** will report individual file results when ticked ✓

## Details:

Stibium writes a total of 16 files of sizes 2, 4, 6, 8, 10, 20, 40, 60, 80, 100, 200, 400, 600, 800 MB, 1, 2 GB, each containing the set byte contents. The order of sizes is randomised to make this part a random write test using fixed sizes. Files written are numbered in the order that they are written, not by size.

The 16 test files require about 54 GB of free disk space. Although each test file is given the extension .tiff, these aren't genuine TIFF files, but consist of the same single byte repeated throughout the entire file.

Once those files have been written, Stibium reads all the files in the selected folder in an order which is determined by the file system. It repeats the reads according to the setting in the **repeats** box. The order of file reading appears quite random, and isn't the same across each repeat. This part of the test is therefore a random read test using fixed sizes. The maximum setting for repeats is 1,000.

→ concluded



## *Series Write & Read Tests concluded*

Because macOS has a habit of writing hidden files to folders, the smallest file to be read and its results included is 10 KB in size. All files smaller than that are ignored.

In some circumstances, file writes could be subject to interference by caching. Normally, this isn't the case, but unusually high transfer rates suggest that may have happened. Look for this in the range shown for the 20% trimmed mean, and in verbose listing of individual files. More likely in this test is that files are cached after writing, so when they are read back some are read from cache and not disk. If that's the case, run separate write and read tests using multiple repeats, separated from the writes by restarting, if necessary.

→ [Series Write Tests](#)

→ [Series Read Tests](#)

# Random Write Tests

**Control:** tick the **Random Sizes** checkbox, specify the number of files to be written in the **length** box, then click on the **Write Series...** button, select a destination folder on the disk to be tested and click **Write**.

**Purpose:** to measure the write transfer time on a series of files with random sizes between 2 MB and 2 GB.

## Key Settings:

popup menu in the middle of the top row sets the byte content to be written to the file, from 0x00 to 0x41 (hexadecimal)

**repeats** box sets how many sets of test files are written

**length** sets the number of files to be written in each cycle; the total number written is **length** times **repeats**

**Random Sizes** checkbox must be ticked ✓

**Verbose** will report individual file results when ticked ✓

## Details:

Stibium writes the number of files set in the **length** box in random sizes between 2 MB and 2 GB, each containing the set byte contents, and repeats that using different random sizes for the number of sets in the repeats box. This a random write test using fixed sizes. Files written are numbered in the order that they are written, not by size. Settings for length can range between 5 and 100, and the maximum setting for repeats is 1,000, which can therefore write a total of 5 to 100,000 test files.

Although each test file is given the extension .tiff, these aren't genuine TIFF files, but consist of the same single byte repeated throughout the entire file.

In some circumstances, file writes could be subject to interference by caching. Normally, this isn't the case, but unusually high transfer rates suggest that may have happened. Look for this in the range shown for the 20% trimmed mean, and in verbose listing of individual files.

Because each file's size is chosen at random, it's extremely unlikely that any two will be of the same size. This limits the analysis of results compared with fixed size tests.

→ [Random Read Tests](#)

# Random Read Tests

**Control:** having generated a folder of test files in the Random Write test, click on the **Read Folder...** button, select that folder and click **Read**.

**Purpose:** to measure the read transfer time on all files of 10 KB and greater size in a single folder, and any folders containing within it.

## Key Settings:

**repeats** box sets how many times the files in the folder are read

**No Cache** checkbox sets whether the 'no caching' read method is used, recommended to be ticked ✓

**Verbose** will report individual file results when ticked ✓

## Details:

Stibium reads all the files in the selected folder in an order which is determined by the file system. It repeats the reads according to the setting in the **repeats** box. The order of file reading appears quite random, and isn't the same across each repeat. This test is therefore a random read test using random sizes. The maximum setting for repeats is 1,000.

Because macOS has a habit of writing hidden files to folders, the smallest file to be read and its results included is 10 KB in size. All files smaller than that are ignored.

This test can be performed on *any* folder to which you have read access, including those in the System volume. It iterates through the contents of any folders inside the selected folder, and reads all files within those as well.

In some circumstances, repeated file reads could be subject to interference by caching. Normally, this isn't the case, but unusually high transfer rates suggest that may have happened. Look for this in the range shown for the 20% trimmed mean, and in verbose listing of individual files.

→ [Random Write Tests](#)

# Single Size Write Tests

**Control:** set the size of the file(s) to be written using the box next to the Write Single... button and the popup menu of size units, click on the **Write Single...** button, give the file a name if prompted, select a destination folder on the disk to be tested and click **Save**.

**Purpose:** to measure the write transfer time on one of more files with a set size between 1 KB and 1,024 GB.

## Key Settings:

size box next to the Write Single... button and the popup menu of size units set the size of the file(s) to be written. Note size is given in 'decimal' units, where 1 KB is 1,000 bytes. Minimum size is 1 KB, maximum is 1,024 GB  
popup menu in the middle of the top row sets the byte content to be written to the file(s), from 0x00 to 0x41 (hexadecimal)  
**repeats** box sets how many test files of that size are written

## Details:

Stibium writes from 1 to 1,000 test files of the size determined by the settings in the box and units popup. If repeats is set to 1, just a single file is written, and you'll be prompted to save that by name. If repeats is set to 2 to 1,000, then you're prompted to select the name of a folder in which the test files will be written.

Although test files are given the extension .tiff, these aren't genuine TIFF files, but consist of the same single byte repeated throughout the entire file.

In some circumstances, file writes could be subject to interference by caching. Normally, this isn't the case, but unusually high transfer rates suggest that may have happened. Look for this in the range shown for the 20% trimmed mean, and in verbose listing of individual files.

→ [Single Size Read Tests](#)

# Single Size Read Tests

**Control:** click on the **Read Single...** button, select the file you wish Stibium to read and click **Read**.

**Purpose:** to measure the read transfer time on all files of 10 KB and greater size in a single folder, and any folders containing within it.

## Key Settings:

**No Cache** checkbox sets whether the 'no caching' read method is used, recommended to be ticked ✓

## Details:

Stibium reads the file selected and reports the result. To read all the files within a folder, use the Read Folder... button instead.

This test can be performed on *any* folder to which you have read access, including those in the System volume.

In some circumstances, repeated file reads could be subject to interference by caching. Normally, this isn't the case, but an unusually high transfer rates suggest that may have happened.

→ [Single Size Write Tests](#)

# Benchmarking

There's no such thing as a perfect benchmark. Performance tests should be designed to measure for a purpose. If your purpose is the selection of storage to be used in RAID arrays for heavily-loaded servers, then your benchmarks should look at measures which are relevant to that. This means that they'll probably need to precondition the SSDs for at least a couple of hours, during which their performance is likely to fall progressively. But those benchmarks won't tell you much about how an ordinary user of a Mac could select the most cost-effective storage for what they do.

Stibium sets out to look at SSD performance for regular users of Macs, both Intel and Apple Silicon models, writing and reading through macOS from normal everyday apps. Because its tests are almost infinitely flexible, you can tailor them to look at specific situations, such as an app which needs to work with file between 1 and 5 GB.

Its standard 'fixed size' and random size tests use files in the range 2 MB to 2 GB. Smaller file sizes are more difficult to test because they are prone to what appear to be very rapid transfer rates, which may reflect persistent caching or specific design features to greatly accelerate performance with smaller files. Above 2 GB, testing becomes slow even on fast SSDs, and apps likely to write or read larger files will normally use streams rather than single-call read or save.

Transfer speeds are limited by the slowest part in the chain between memory and the SSD's storage. This could be the SATA interface in an external SSD, or the connection between that and the Mac.

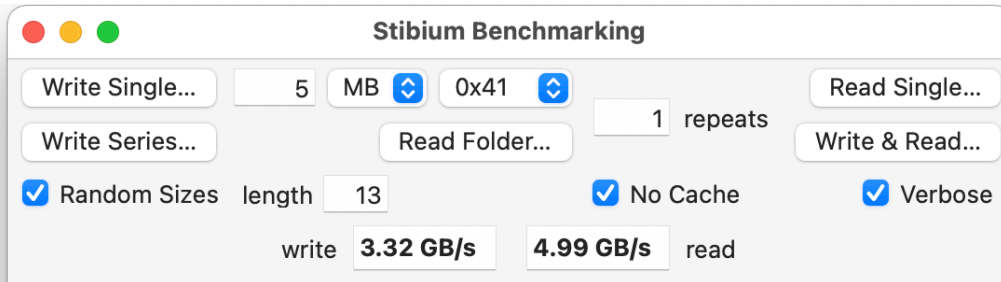
Standard maximum transfer rates include:

• SATA 6.0	600 MB/s
• USB 3.1 Gen 2	1.2 GB/s
• USB 3.2 Gen 2x2	2.4 GB/s
• PCI Express 2.0 (x4 link), including external	2 GB/s
• Thunderbolt 3, 2 links	5 GB/s
• NVMe M.2 and U.2 with PCI Express 4.0 (x4 link)	7.9 GB/s
• PCI Express 2.0 (x16 link)	8 GB/s
• PCI Express 4.0 (x16 link)	31.5 GB/s.

Further data [are here](#).

If you benchmark an SSD which is apparently exceeding the maximum transfer rate of its slowest link to the computer, then you should be extremely suspicious that something else is going on, such as caching. [→ Controls & Settings](#)

# Controls & Settings



The following settings are controlled, according to the test being run:

- file size, using the box next to the Write Single... button, given in 'decimal' bytes where 1 KB is 1,000 bytes, with the units popup next to it. These provide for sizes between 1 KB and 1,024 GB.
- byte content of test files written, using the popup menu in the top row. Options include 0x00, 0x88, 0x1A and 0x41, in hexadecimal.
- number of repeats of the test set or cycle, from 1 to 1,000.
- whether test files written are to be of Random Sizes, or the fixed sizes. In both cases, they range from 2 MB to 2 GB. Fixed sizes are 16: 2, 4, 6, 8, 10, 20, 40, 60, 80, 100, 200, 400, 600, 800 MB, 1, 2 GB.
- the length of random files to be written, from 5 to 100.
- whether the 'no caching' method is to be used when reading files. It's recommended that this is ticked ✓ so enabled.
- whether Verbose results are to be given, which list each result in CSV format when performing series and random tests.

The following buttons are available:

- Write Single... writes one or more files of exactly the same size
- Read Single... reads a single file
- Write Series... writes a standard set of 16 test files of pre-determined sizes, or of random sizes
- Read Folder... reads all the files of 10 KB size and greater in the selected folder, and folders inside it
- Write & Read... performs a Write Series test on a single group of 16 files of set sizes, followed by one or more reads of all files in that folder.

There are two results displayed:

- write gives the most recent Theil-Sen regressed write rate.
- read gives the most recent Theil-Sen regressed read rate.

[→ Test Methods](#)

# Test Methods

The major risk when trying to measure the performance of storage in Macs is that of caching by macOS. More recent versions of macOS are notorious for keeping data written to or read from files in memory for surprisingly long periods. This can even occur after a user has logged out and back in again, and sometimes it seems to occur even after a restart.

There several ways in which caching can be flushed and rendered ineffective. These include writing or reading large numbers of files of substantial size, logging out and back in, and restarting. The method suggested in the 'Gold Standard' test uses the first and the last, and appears quite reliable.

Caching and other accelerator effects increase the range of results, typically by adding outliers which appear to be of very high performance. For example, a SATA SSD connected by USB-C might generally deliver around 480 MB/s but in some individual tests could return speeds of more than the 600 MB/s which is a physical limit of its interface. Those outliers are the best indicator which should raise your suspicion that caching has contaminated its results.

The statistical techniques used in Stibium are designed to mitigate the effects of these outliers as much as possible. This is also necessary to cope with individual measurements which are significantly slower, perhaps as a result of another process accessing the same storage during that part of the test.

If you're concerned that outliers from caching or anything else are affecting your results, increase the number of tests, for instance by using more repeats, and separate write from read tests with a restart.

→ [Data Analysis](#)



# Data Analysis

Once each test run is complete, summary results are displayed in the two boxes marked write and read, according to the last test run. These values persist between tests even when you quit Stibium. That allows you to see the previous results after restarting your Mac. Full details are then given in the scrolling text view below.

Those currently given are:

- CSV for the test (w = write, r = regular read, r\* = no cache read), the size of the file in bytes, the time taken to read/write in seconds (taken from Mach time), and the transfer rate in bytes/s (verbose mode only, and for single tests).
- CSV for median values of transfer rate for each file size. These are most useful when repeating the standard range of file sizes, rather than using random sizes.
- The type of test performed and the number of files written or read, followed by a simple arithmetic average over all results of the transfer speed.
- A 20% trimmed mean transfer rate, with the minimum and maximum transfer rates found, in parentheses. This is a more robust estimate of the overall rate than the average or median. When small numbers of repeat measures are available, the median is given instead.
- A transfer rate calculated from a robust linear regression using the Theil-Sen method. This regression is performed on the original transfer times (as Y values) against file sizes (X), and is less susceptible to the effects of outliers. The overall transfer rate is then calculated as the inverse of that gradient, with the Y intercept being ignored as it's very small. Given in parentheses after this are the 90% confidence interval for that estimate, which exclude the highest 5% and lowest 5% of all gradients found using the Theil-Sen method.
- A transfer rate calculated using least-squares linear regression of time taken against file size. As with the robust regression, this isn't given when all the files written or read are of the same size, as linear regression can't then be performed.
- details of the test path, Mac model and hardware, macOS, version of Stibium, and the start and end times of that test.
- Those results aren't reloaded after opening Stibium; if you want a record of all the results currently in the scrolling text view, you must copy and paste them into another app, or export the text to a text file.

→ [Technical Information](#)

# Technical Information

## Source code used

To read a file:

```
let theFHandle = try FileHandle.init(forReadingFrom: url)
if #available(OSX 10.15.4, *) {
    theStart = mach_absolute_time()
    let theData = try theFHandle.readToEnd()
    try theFHandle.synchronize()
    theEnd = mach_absolute_time()
    let theSize = theData?.count ?? 0
    if theSize < 1 {
        theEnd = theStart
    }
    try theFHandle.close()
} else {
    theStart = mach_absolute_time()
    let theData = theFHandle.readDataToEndOfFile()
    theFHandle.synchronizeFile()
    theEnd = mach_absolute_time()
    let theSize = theData.count
    if theSize < 1 {
        theEnd = theStart
    }
    theFHandle.closeFile()
}
theTime = self.machToSecs(ticks: (theEnd - theStart))
```

To read a file with the no caching option:

```
theStart = mach_absolute_time()
let theData = try NSData.init(contentsOf: url, options: NSData.ReadingOptions.uncached)
theEnd = mach_absolute_time()
let theSize = theData.count
```

→ [concluded](#)

## Technical Information concluded

To write a file up to 2 GB in size:

```
let theFHandle = try FileHandle.init(forWritingTo: url)
if #available(OSX 10.15.4, *) {
    theStart = mach_absolute_time()
    try theFHandle.write(contentsOf: data)
    try theFHandle.synchronize()
    theEnd = mach_absolute_time()
    try theFHandle.close()
} else {
    theStart = mach_absolute_time()
    theFHandle.write(data)
    theFHandle.synchronizeFile()
    theEnd = mach_absolute_time()
    theFHandle.closeFile()
}
theTime = self.machToSecs(ticks: (theEnd - theStart))
```

To write a file over 2 GB in size:

```
theStart = mach_absolute_time()
try theData.write(to: theURL)
theEnd = mach_absolute_time()
theDuration = self.machToSecs(ticks: (theEnd - theStart))
```

## Acknowledgements

I'm very grateful to Ric Ford for his devoted testing and discussion, and to all those who have commented on the Eclectic Light Company blog.

→ [Change list](#)

# Change list

## 1.0

- added help reference
- added Theil-Sen regresssion
- added 20% trimmed means instead of overall medians
- improved regression to avoid generating NaN results
- improved control layout and changed labels on buttons
- made overall results persistent
- added multiple write tests of fixed size files.

## 1.0b6

- added support for random write tests
- added random ordering of standard write tests
- added Mac and other report details (thanks to Ric Ford)
- added write and read result boxes
- added median by file size analysis
- added linear regression analysis
- excluded small files fully
- added auto-update support
- added save report as text.

## 1.0b5

- changed file sizes used to avoid 'burst' effects
- checks for and handles zero times and errors correctly.

## 1.0b4

- added 'no caching' read option (thanks to Ric Ford)
- changed verbose output to CSV
- added analysis
- added multiple tests
- added repeats
- added preference saving.

→ [Details](#)

#### *1.0b3*

- changed timing to use Mach ticks
- added Help.

#### *1.0b2*

- using Date() for timing
- first public release.

#### *1.0b1*

- not released.

10 January 2021.