

Copyright (C) 2004 Gavin Stewart, XDT

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

## README for diskWriggler

- Summary.
- Availability.
- Basic usage.
- Parameters explained.
- Compiling from source.
- Feedback or bug notification.
- Copying and License information.

### Summary.

diskWriggler is a benchmark tool for testing filesystem storage throughput of film or video frames as sequential files or as frames contained in one large file.

The expected audience is systems engineers working in the film and post-production industries.

### Availability.

diskWriggler is available as a source tarball, or as precompiled binaries for various platforms at:  
<http://www.xdt.com.au/Products/diskWriggler/>

### Basic Usage.

Executing diskWriggler without parameters at a command prompt will present a usage screen. Version 1.0.0 appears as:

```
$ ./diskWriggler
Parameters required!
```

Version: 1.0.0

usage: diskWriggler -f|PAL|NTSC|HD|2K|4K [<options>] -n <n> -o <path>

Frame type (Required, one only):

-f <filename> : Source frame file.	-HD : 1920x1080 10bit 4:4:4
-res <nxn> : eg. 720x576.	-2K : 2048x1556 10bit 4:4:4
-PAL : 720x576 8bit 4:2:2	-4K : 4096x3112 10bit 4:4:4
-NTSC : 720x486 8bit 4:2:2	

Frame type/size overrides (if requirements different to defaults above).

-bit <n> : bits / pix / channel	-C : frames in Container, packed
-samp <n:n:n> : 4:4:4, 4:2:2 or 4:2:0	deflt is file/frame, padded
-pack : packed bits	-BS <n> : Block Size in bytes
-pad : padded bits	deflt is frame size

Reading/Writing options (some options will work together):

-A : Async.	-S : Sync.
-D : DirectIO (Unbuffered).	-DR : DirectIO only for Read.

## Options :

-t : Time (profile).

## Required Parameters:

-n <num\_frames> | -o <out\_path>

The minimum usage requires a "Frame type", -n <num\_frames> and -o <out\_path>. It is suggested that the -t option be used to show a statistics line approximately every 5 seconds.

## Minimal example:

```
$ ./diskWriggler -2K -t -n 300 -o /raid/tmp
```

## Parameters explained.

Note that all parameters are case sensitive, and whitespace must exist between all parameters.

## Frame type:

-f <filename>

The provided <filename> is used as the source frame. This allows any existing file (.cin, .sgi, .dpx, etc) to be used and becomes useful if the frame contains other information (overheads) that are not factored in when the frame is generated from the following options.

-res <nxn>

Generate a source frame of the provided resolution.

-PAL, NTSC, HD, 2K, 4K

These are existing profiles that are a quick way to get started.

## Frame type/size overrides:

(These are useful to override any defaults of the "Frame type"s above).

-bit <n>

Specify the number of bits per pixel per channel. Commonly this is 8, 10, 12 or 16.

-samp <n:n:n>

Specify the colour sampling used. If the colour is RGB, then use 4:4:4

-pack

Pack frame bits to save space. This appears to be common in container files like Quicktime.

-pad

Pad frame bits out to the nearest byte. This is an expensive waste of bits, and appears common in file-per-frame formats.

-C

Put all frames into one file. This is to simulate "container" files like Quicktime's "mov"s and Microsoft's "avi"s. The default is to generate one file per frame.

-BS <n>

Specify a specific block size (in bytes) to use when reading and writing frames. The default is the frame size.

## Reading/Writing options:

-A

Asynchronous. Does nothing on win32.

-S

Asynchronous. Does nothing on win32.

-D

DirectIO, Unbuffered, NoCache. The exact meaning is platform dependent, but in general means skipping the operating systems buffers and performing data transfers directly from RAM to Disk (and vice versa). Typically this provides the highest throughput for large streaming files.

-DR

Same as above, for the file reads only (not writes). This was for

an interesting issue with Windows XP and a Windows Server 2003 network share. The unbuffered writes were very poor.

**Options:**

- t  
This causes a statistics line to be printed approximately every 5 seconds. This is a good idea to see what is really happening rather than a blank screen and disk drive lights on hard!

**Required Parameters:**

- n <n>  
The number of frames to write. It is suggested that the number of frames be large enough so that the amount of data written/read is larger than what will fit in the operating systems cache.
- o <out\_path>  
The output path of frames to write. Under win32 this may include the drive letter as well: i.e. r:\tmp

**Compiling from source.**  
-----

diskWiggler is known to compile under the following environments:

**Linux:**

- RedHat 8.0, 9.0

**Windows XP:**

- Using MinGW-3.1.0 and MSYS-1.0.10 (<http://www.mingw.org/>)

**Mac OSX:**

- 10.3.6 Using Xcode (<http://developer.apple.com/tools/macosxtools.html>)

Download the source, unpack the tar or zip file, type "make" in the src directory.

**Feedback or bug notification.**  
-----

Please direct all feedback to the XDT R&D department: [rnd@xdt.com.au](mailto:rnd@xdt.com.au)

**Copying and License information.**  
-----

diskWiggler is released under the GNU General Public License version 2. The full text of which can be found in the file "COPYING" in this package.