

**NAME**

**fifolog\_create**, **fifolog\_writer**, **fifolog\_reader** - initialize, write, seek and extract data from a fifolog

**SYNOPSIS**

**fifolog\_create** [-l *record-size*] [-r *record-count*] [-s *size*] *file*

**fifolog\_reader** [-t] [-b *tstart*] [-B *Tstart*] [-e *tend*] [-E *Tend*] [-o *ofile*] [-R *regexp*] [-T *timefmt*] *file*

**fifolog\_writer** [-w *write-rate*] [-s *sync-rate*] [-z *compression*] *file*

**DESCRIPTION**

Fifologs provide a compact round-robin circular storage for recording text and binary information to permanent storage in a bounded and predictable fashion, time and space wise.

A fifolog can be stored either directly on a disk partition or in a regular file.

The input data stream is encoded, compressed and marked up with timestamps before it is written to storage, such that it is possible to seek out a particular time interval in the stored data, without having to decompress the entire logfile.

The **fifolog\_create** utility is used to initialize the first sector of a disk device or file system file to make it a fifolog and should be called only once.

Running **fifolog\_create** on an existing fifolog will reset it so that **fifolog\_reader** and **fifolog\_writer** will not see the previous contents. (The previous contents are not physically erased, and with a bit of hand-work all but the first record can be easily recovered.)

If the *file* does not already exist, **fifolog\_create** will attempt to create and truncate(2) it to the specified size, defaulting to 86400 records of 512 bytes if the **-r**, **-l** or **-s** options do not specify otherwise.

The **fifolog\_writer** utility will read standard input and write it to the end of the fifolog according to the parameters given.

Writes happen whenever the output buffer is filled with compressed data or when either of two timers expire, forcing a partially filled buffer to be written.

The first and faster timer, **-w write-rate**, forces available data to be written but does not flush and reset the compression dictionary. This timer is intended to minimize the amount of logdata lost in RAM in case of a crash and by default it fires 10 seconds after the previous write.

The second and slower timer, **-s sync-rate**, forces a full flush and reset of the compression engine and causes the next record written to be a synchronization point with an uncompressed timestamp, making it

possible to start reading the logfile from that record. By default this timer fires a minute after the previous sync.

The **-z** *compression* option controls the `zlib(3)` compression level; legal values are zero to nine which is the default.

The **fifo****log\_reader** utility will retrieve records from the `fifo`log according to the specified parameters and write them either to standard output or the file specified with **-o**.

It is possible to specify a start and end time to limit the amount of data **fifo****log\_reader** will report. The lower-case variants **-b** and **-e** take a *time\_t* value, whereas the upper-case variants **-B** and **-E** take human-readable specifications such as "1 hour ago".

The **-t** option forces timestamps to be formatted as "YYYYMMDDhhmmss" instead of as *time\_t*, and **-T** allows the specification of an `strftime(3)` formatting string.

Finally, records can be filtered such that only records matching the (REG\_BASIC) regular expression specified with **-R** are output.

## IMPLEMENTATION NOTES

The data stored in the `fifo`log consists of three layers, an outer layer that allows searches to synchronization points based on timestamps without having to decompress and decode the actual contents, a compression layer implemented with `zlib(3)`, and an inner serialization and timestamping layer.

The exact encoding is described in the *fifo*log.h file.

`Fifo`log is particularly well suited for use on Flash based media, where it results in much lower write-wear, than a file system with regular log files rotated with `newsyslog(8)` etc.

## EXAMPLES

Create a `fifo`log with 1024\*1024 records of 512 bytes:

```
fifo
```

log\_create -r 10m /tmp/fifo

Write a single record to this file:

```
date | fifo
```

log\_writer /tmp/fifo

Read it back with human readable timestamps:

```
fifolog_reader -t /tmp/fifolog
```

One particular useful use of **fifolog\_writer** is with `syslogd(8)` using a line such as this in `syslog.conf(5)`:

```
*.* |fifolog_writer /var/log/syslog_fifolog
```

## HISTORY

The `fifolog` tools have been liberated from an open source SCADA applications called "measured", which monitors and controls remote radio navigation transmitters for the Danish Air Traffic Control system.

## AUTHORS

The `fifolog` tools were written by Poul-Henning Kamp.