

# OTF Tools

Andreas Knüpfer

## Introduction

The Open Trace Format Library (OTF) comes with support tools that perform frequent tasks.

**otfmerge:** change the number of streams for an existing trace.

**otfmerge-mpi:** MPI version of otfmerge.

**otfauux:** add snapshot and statistics information to an otf trace.

**vtf2otf:** translate VTF3 traces to OTF.

**otf2vtf:** translate OTF to VTF3 backwards (limited functionality).

**otfdump:** print information about a tracefile in human readable format.

**otfcompress:** compress/decompress OTF traces.

**otfconfig:** show configure parameters for the existing OTF installation.

**otfprofile:** generate concise profile in Latex format for an OTF trace.

**otfshrink:** create a new otf file that only includes specified processes.

**otfinfo:** get basic information about a tracefile.

For all OTF tools the `-V` option will print the OTF version. See below for detailed description of each tool.

## otfmerge

The **otfmerge** tool allows to merge an existing OTF trace to a different number of streams. The `-n` option specifies the number of output streams. At maximum there will be as many output streams as there are trace processes. Setting `-n 0` will create the maximum number of streams automatically.

The output file name is set via the `-o` option. With `-f` it is possible to restrict the number of file handles used concurrently by `otfmerge`. This is necessary if the number of files exceeds the limit of file handles as set by the environment.

Via `-rb` and `-wb` the internal input resp. output buffer sizes per stream can be changed. However, the default buffer sizes should be suitable most of the time. The `-stats` and `-snaps` options allow to include statistics and snapshot records when merging. By default they are ignored.

Global definition records are copied to the output trace. Local definitions are also copied even though this invalidates the trace! Local definitions are not expected and should have been translated to global definitions beforehand by the resp. creator.

The following short help message is given when `otfmerge` is called with the `-h` option:

```
otfmerge - converter program of OTF library.
```

```
otfmerge [Options] <input file name>
```

### Options:

<code>-h, --help</code>	show this help message
<code>-V</code>	show OTF version
<code>-n &lt;n&gt;</code>	set number of streams for output set this to 0 for using one stream per process - standard is 1
<code>-f &lt;n&gt;</code>	set max number of filehandles available
<code>-o &lt;name&gt;</code>	namestub of the output file (default 'out')
<code>-rb &lt;size&gt;</code>	set buffersize of the reader
<code>-wb &lt;size&gt;</code>	set buffersize of the writer
<code>-stats</code>	cover statistics too
<code>-snaps</code>	cover snapshots too

-z <zlevel>	write compressed output zlevel reaches from 0 to 9 where 0 is no compression and 9 is the highest level
-l	write long OTF format
-p	show progress

## otfmerge-mpi

The otfmerge-mpi tool is the MPI version of the otfmerge tool.

otfmerge-mpi - an MPI implementation of otfmerge

otfmerge-mpi [options] <input file name>

### options:

-h, --help	show this help message
-V	show OTF version
-n <n>	set number of streams for output set this to 0 for using one stream per process - default is 1
-f <n>	set max number of filehandles available per rank
-o <name>	namestub of the output file (default 'out')
-rb <size>	set buffersize of the reader (for each rank)
-wb <size>	set buffersize of the writer (for each rank)
-stats	cover statistics too
-snaps	cover snapshots too
-z <zlevel>	write compressed output zlevel reaches from 0 to 9 where 0 is no compression and 9 is the highest level
-l	write long OTF format
-p	show progress

## otfaux

The `otfaux` tool appends auxiliary information to an existing OTF trace. The event records are read but not modified.

There are two kinds of auxiliary data. First, there are snapshot information that provide the complete status of a trace process at a given time stamp. This contains call stack information, pending messages, current performance counter values, etc. Second, there are statistics information accumulated from the beginning of the trace until the current time stamp. Statistics involve the number of calls, exclusive and inclusive time for per function resp. function group or accumulated message count and message volume for communication, etc. Statistics are always monotone increasing not unlike program profiles. Let  $S_a$  and  $S_b$  two statistics at time stamps  $a < b$  then  $S := S_b - S_a$  is the profile information for the time interval  $[a, b]$ .

Both, snapshots and statistics are generated at certain break point, which can be specified in several ways: First, `-n x` allows to have  $x$  break points distributed regularly over the trace's time interval. Second, `-p y` will generate a break point every  $y$  ticks starting from the beginning of the trace. If both options are given the one producing more break points wins. In addition break points can be specified with `-t z` which will add a single explicit break point regardless of `-n` and `-p` options.

If the `-g` switch is set then function statistics are replaced by function group statistics. This produces more terse output. The option `-v` switches on verbose mode which prints break point time stamps while processing.

In case there are auxiliary information already present the `-o` option forces `otfaux` to overwrite it. Otherwise `otfaux` exits with an error message. Via `-b` internal buffer size per stream can be adjusted although the default setting is suitable most of the time.

The `-h` switch provides the following short help message:

```
otfaux - append snapshots and statistics to existing
         otf traces at given 'break' time stamps
```

```
otfaux [Options] <file name>
```

Options:

```
-h, --help      show this help message
```

-V	show OTF version
-b <size>	buffer size for read and write operations
-n <n>	number of breaks (distributed regularly) if -p and -t are not set, the default for -n is 200 breaks
-p <p>	create break every 'p' ticks (if both, -n and -p are specified the one producing more breaks wins)
-t <t>	define (additional) break at given time stamp
-F	force overwrite old snapshots and statistics
-R	delete existing snapshots and statistics only
-f <n>	max number of filehandles output
--funcgroups	create functiongroup summaries instead of function summaries
--filegroups	create file group summaries instead of file summaries
-v	verbose mode, print break time stamps
-a	show advancing progress during operation
--snapshots	write ONLY snapshots but NO statistics
--statistics	write ONLY statistics but NO snapshots
-s a[,b]*	regard given streams only when computing statistics. expects a single token or comma separated list. this implies the '--statistics' option!
-l	list existing stream tokens

## **vtf2otf**

The `vtf2otf` tool translates a VTF3 trace to OTF. With `-o` the output file name is specified. If it has no `'.otf'` suffix already then it is appended automatically. This tool supports only those record types supported by OTF. Some deprecated or experimental VTF3 records are ignored.

The number of output streams to be generated is given with `-n n`. The `-f` option allows to restrict the number of file handles to be opened concurrently in case there are too many streams. Again, `-b` adjusts the output buffer size per stream if the default is not suitable. If the `-h` switch is set the following help message is provided:

```
vtf2otf - Convert VTF3 trace files to OTF format.
```

```
vtf2otf [Options] <input file name>
```

### Options:

<code>-h, --help</code>	show this help message
<code>-V</code>	show OTF version
<code>-o &lt;file&gt;</code>	output file
<code>-f &lt;n&gt;</code>	max count of filehandles
<code>-n &lt;n&gt;</code>	output stream count
<code>-b &lt;n&gt;</code>	size of the writer buffer
<code>-z &lt;n&gt;</code>	use zlib compression
<code>-io</code>	compute io events. This is neccessary for getting correct durations in IO-operations. Result of this step is a file with extra information. This file is used for creating correct duration- information in a normal run. If you do not have these extra -information-file, the duration of every IO-operation will be zero.

## otf2vtf

The `otf2vtf` tool performs the backward transformation from OTF to VTF3. Again, `-o` gives the VTF3 output file name including file suffix. Via `-b` OTF's input buffer size per stream can be adjusted if necessary.

With `-A` resp. `-B` the VTF3 sub-format can be set to ASCII (default) resp. binary. The `-h` switch produces a short help message like follows:

```
otf2vtf - Convert OTF trace files to VTF format.
```

```
otf2vtf [Options] <input file name>
```

Options:

<code>-h, --help</code>	show this help message
<code>-V</code>	show OTF version
<code>-o &lt;file&gt;</code>	output file
<code>-b &lt;n&gt;</code>	size of the reader buffer
<code>-A</code>	write VTF3 ASCII sub-format
(default)	
<code>-B</code>	write VTF3 binary sub-format

## otfdump

The `otfdump` tool prints information about a tracefile in human readable format.

```
otfdump - convert otf traces or parts of it into a  
human readable, long version
```

Options:

<code>-h, --help</code>	show this help message
<code>-V</code>	show OTF version
<code>-o &lt;file&gt;</code>	output file
	if the ouput file is unspecified the stdout will be used

```

--num <a> <b>  output only records no. [a,b]
--time <a> <b> output only records with time-
                stamp in [a,b]

--nodef      omit definition records
--noevent    omit event records
--nostat     omit statistic records
--nosnap     omit snapshot records
--nomarker   omit marker records

--nokeyvalue  omit key-value pairs

--procs <a>   show only processes <a>
              <a> is a space-separated list of
              process-tokens

--records <a> show only records <a>
              <a> is a space-separated list of
              record-type-numbers

              record-type-numbers can be found in
              OTF_Definitions.h (OTF_*_RECORD)

-s, --silent  do not display anything except
              the time otfdump needed to read
              the tracefile

```

## otfcompress

The `otfcompress` tool performs compression and decompression on traces.

```
otf(de)compress - compression program for single
                  OTF files.
```

Usage: `otf(de)compress [OPTIONS] <FILES>`

```
-h, --help  show this help message
```



-V	show OTF version
-c	compress (default action when called as 'otfcompress')
-d	decompress (default action if called as 'otfdecompress')
-k	keep original file (compressed resp. uncompressed)
-o <dir>	output directory (implicitly sets -k)
-[0-9]	use given compression level (default 4)
	0 - plain
	1 - minimum compression, fastest
	9 - maximum compression, slowest

## otfconfig

The `otfconfig` tool shows various installation parameters of OTF, which are important for developers.

`otf-config` - shows parameters of the otf configuration.

`otf-config` [Options]

Options:

-h, --help	show this help message
--version	show the otf version
--have-zlib	is zlib enabled
--includes	path to the otf headers
--libs	libline needed for linking otf
--sizes	print size of integer types

## otfprofile

The `otfprofile` tool creates a concise profile of an OTF trace in Latex format.

otfprofile - generates a profile of a trace in Latex or CSV format.

options:

-h, --help	show this help message
-b <x>	readbuffer size
-f <x>	max. number of filehandles to use
-i <file>	specify an input trace name
-csv <file>	specify an input csv-file trace name (as produced by otfprofiler before), don't use -i and -csv together
-o <path>	specify the path for the output files
-tex <x>	writes Latex output in different flavours: all,func,p2p,collop,none
-notex	disable Latex output
-nops	disable Postscript output
-var	also show statistic variance
-top <x>	max. number of functions shown (default 50)
-progress	show progress information
-sum	reads only summarized information, no events
-omp <x>	specify the number of threads which are used while reading the otf-file parallel Note: This option overrides the environ- ment variable OMP_NUM_THREADS, only useful if compiled with OpenMP support
-lite	ignore P2P and collective communication (saves memory for highly parallel cases)

## otfshrink

The otfshrink tool creates a new otf file that is reduced to specified processes.

otfshrink - creates a new otf file that only includes

specified processes

options:

-h, --help	show this help message
-i <file>	specify the input trace file
-o <file>	specify the output file
-l <list>	a space-separated list of processes to show, e.g. -l 1 2 3-4 8-5
-v <list>	a space-separated list of processes NOT to show, see -l for exact syntax
-s <mode>	display all selected processes, no files are created (simulation mode), modes: (l)ist, (r)ange or (t)able default: range

## otfinfo

The `otfinfo` tool is useful to get basic information about a tracefile.

`otfinfo` - program to get basic information of a trace.

`otfinfo` [Options] <input file name>

options:

-h, --help	show this help message
-V	show OTF version
-f <n>	set max number of filehandles available
-l <ilevel>	set the information level for the output (0 - 4) the default level is 1
-a	set the information level to 4
-p	show progress bar for reading event files