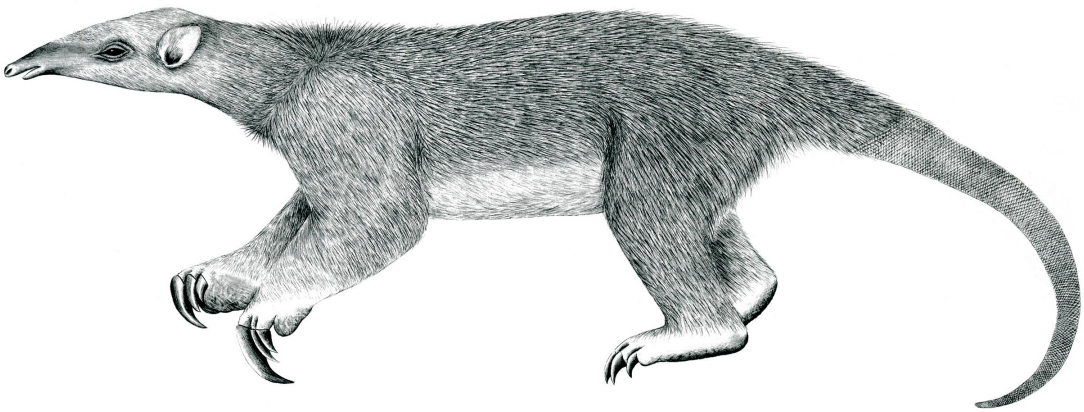
Tranalyzer2

binSink

Binary Output



Tranalyzer Development Team

Contents

1 binSink

1.1 Description

The binSink plugin is one of the basic output plugin for Tranalyzer2. It uses the output prefix (`-w` option) to generate a binary flow file with suffix `_flows.bin`. All standard output from every plugin is stored in binary format in this file.

1.2 Dependencies

1.2.1 External Libraries

If gzip compression is activated (`BFS_GZ_COMPRESS=1`), then **zlib** must be installed.

		BFS_GZ_COMPRESS=1
Ubuntu:	`sudo apt-get install`	`zlib1g-dev`
Arch:	`sudo pacman -S`	`zlib`
Gentoo:	`sudo emerge`	`zlib`
openSUSE:	`sudo zypper install`	`zlib-devel`
Red Hat/Fedora[1]**:**	`sudo dnf install`	`zlib-devel`
macOS[2]**:**	`brew install`	`zlib`

1.2.2 Core Configuration

This plugin requires the following core configuration:

- *$T2HOME/tranalyzer2/src/tranalyzer.h*:
  - `BLOCK_BUF=0`

1.3 Configuration Flags

The following flags can be used to control the output of the plugin:

Name	Default	Description
`BFS_GZ_COMPRESS`	0	Compress (gzip) the output
`BFS_SFS_SPLIT`	1	Split the output file (Tranalyzer `-W` option)
`BFS_FLOWS_SUFFIX`	`"_flows.bin"`	Suffix to use for the output file

1.3.1 Environment Variable Configuration Flags

The following configuration flags can also be configured with environment variables (`ENVCNTRL>0`):

- `BFS_FLOWS_SUFFIX`

[1] If the `dnf` command could not be found, try with `yum` instead
[2] Brew is a packet manager for macOS that can be found here: https://brew.sh

1.4 Post-Processing

1.5 t2b2t

The program `t2b2t` can be used to transform binary Tranalyzer files generated by the binSink or `socketSink` plugin into text or JSON files. The converted files use the same format as the ones generated by the `txtSink` or `jsonSink` plugin.

The program can be found in `$T2HOME/utils/t2b2t` and can be compiled by typing `make`.

The use of the program is straightforward:

- bin→txt: `t2b2t -r FILE_flows.bin -w FILE_flows.txt`
- bin→JSON: `t2b2t -r FILE_flows.bin -j -w FILE_flows.json`
- bin→compressed txt: `t2b2t -r FILE_flows.bin -c -w FILE_flows.txt.gz`
- bin→compressed JSON: `t2b2t -r FILE_flows.bin -c -j -w FILE_flows.json.gz`

If the `-w` option is omitted, the destination is inferred from the input file, e.g., the examples above would produce the same output files with or without the `-w` option. Note that `-w -` can be used to output to stdout.
Additionally, the `-n` option can be used **not** to print the name of the columns as the first row.
Try `t2b2t -h` for more information.

1.6 Custom File Output

- `PREFIX_flows.bin`: Binary representation of Tranalyzer output