

Trace Viewer User Manual



by
Fondazione Bruno Kessler
Embedded Systems Unit

This document is part of the distribution package of the viewers toolset.

Copyright © 2023 by FBK

Chapter 1

Trace Viewer

The trace viewer (Figure 1.1) is a sophisticated tool designed to facilitate the analysis and comprehension of system behavior through the visualization of trace generated by NUXMV or xSAP. It takes a trace file (XML format) and shows it. Its tabular format presents a comprehensive view of variable states and their temporal evolution. It allows also to specify filters which can be used to limit the symbols spaces and/or the steps which are show; for further details about trace format, please refer to the NUXMV manual (Chapter 4.7).¹

	Name	step 1	step 2	step 3	step 4	step 5	step
1	deadlock	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2	fork[0]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
3	fork[1]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
4	fork[2]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
5	fork[3]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
6	fork[4]	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
7	phil_0.grabs_left_fork	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
8	phil_0.grabs_right_fork	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
9	phil_0.releases_left_fork	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
10	phil_0.releases_right_fork	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
11	phil_0.state	idle	idle	idle	idle	idle	need_left
12	phil_1.grabs_left_fork	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
13	phil_1.grabs_right_fork	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
14	phil_1.releases_left_fork	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
15	phil_1.releases_right_fork	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

Total steps: 13

Figure 1.1: The trace viewer

- File Import: Users can import trace files in XML format for visualization and analysis.

¹Visit the website: [NUXMV Website](http://www.nuxmv.org)

- **Symbol Space and Step Filters:** The viewer allows users to specify filters to limit the symbol spaces or steps displayed in the trace.
- **Color-Coded Representation:** The viewer utilizes a color scheme to enhance trace analysis.

To enhance readability, the viewer employs a consistent color scheme. By employing these intuitive visual cues, the trace viewer provides an organized and aesthetically pleasing interface for in-depth analysis, aiding in tasks such as debugging, performance optimization, and insightful observations.

- state variables are represented in black
- input variables in blue
- combinatorial variables in green
- loop steps are visually distinguished by a prominent red header
- timed variables are easily identifiable with a distinctive pink background that encompasses the entire cell
- additionally, fault variables, crucial for error detection, are highlighted in vivid red, encompassing their respective cells

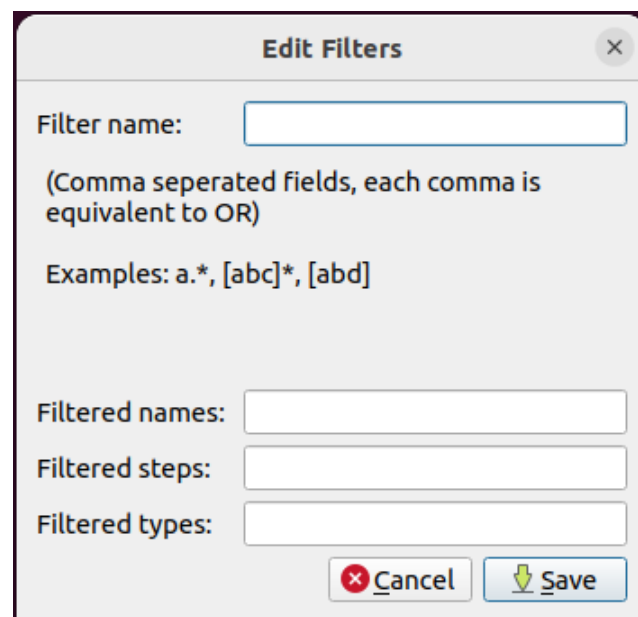


Figure 1.2: The trace viewer filter

To effectively use the filter (Figure 1.2), follow these guidelines:

Names: Enter a regular expression pattern to filter names based on specific patterns or criteria. You can utilize special characters, metacharacters, and regular expression syntax for advanced matching. The regular expression patterns `^f.*`, `^d.*` match names that start with "f" or "d".

Steps: Specify a range of steps by entering the starting and ending steps separated by a dash (-). If the ending step is omitted or left empty, the filter will be applied from the starting step until the last step. For instance, if you want to filter names from step 1 to step 5 (inclusive), enter **1-5** in the steps field. If you want to filter names starting from step 3 until the last step, you can enter **3-** in the steps field.

Type: Select the desired variable types for filtering: "combinatorial," "input," or "state." You can filter names based on one or more variable types. If filtering by multiple types, separate them with commas (,), treating each comma as an OR operator. For example, to filter names for both "combinatorial" and "state" variable types, enter **combinatorial,state**.

By applying these guidelines, you can effectively use the filter to narrow down and identify names that match your specific criteria, based on regular expression patterns, step ranges, and variable types.

1.1 Prerequisites

pip latest available version

Install corresponding package.

PySide2 Version 5.15.2.1 or later

Install corresponding package.

1.2 Trace viewer

To run the trace viewer, invoke script:

```
$> python view_trace.py -h

usage: view_trace.py [-h] [-t TRACE-FILE] [-v] [-ft FAILURES-FILE] [-tv TIMED-VAR]

Trace Viewer

optional arguments:
  -h, --help            show this help message and exit
  -t TRACE-FILE, --trace-file TRACE-FILE
                        The trace file name
  -v, --version          show programs version number and exit
  -ft FAILURES-FILE, --failure-variables FAILURES-FILE
                        The failure variables file name
  -tv TIMED-VAR, --timed-variable TIMED-VAR
                        The timed variable name
```