# Trace Analyser Console Application

Enxhi Ferko ([enxhi.ferko@mdh.se](mailto:enxhi.ferko@mdh.se))

Last edited : 2021/01/17

This document contains a short description for the Trace Analyser console application and few test cases performed on it.

## Console App Functionality

The console app takes as input two files, BasicWW.eaxml and LogFile.csv. The first file, BasicWW.eaxml is the EAST-ADL model for a vehicle windshield wiper system and contains all relevant information for the system elements, signal names, port names and constraints.

In the second file, LogFile.csv, are listed all signal values in different timestamps.

The goal of the application is to check the log file and inform if the timing constraints defined in the model are met or not.

Detailed description of the algorithm on how the log file is checked for Age Constraint and Delay Constraint, with a generic example can be found in LogFileTest9Dec.xlsx

## Code Structure Explanation.

* ***ConstraintProvider.cs*** – takes as argument path to file BasicWW2.eaxml and parse the file. It stores all the info in the following helper classes.
  + ***Constraint.cs*** – is an abstract class to store all relevant info related with a specific constraint.
    - ***DelayConstraint.cs, AgeConstraint.cs*** – Implemented separately just in case different methods would be needed in the future.
      * ***StimulusResponse.cs*** – Used to store a stimulus response couple for a constraint.
* ***LogStream.cs*** – this class provides a stream to read logs from. Logs are enqueued one by one, processed and then dequeued if certain criteria are met. Queue<Log> QueuedLogs is used to store the logs.
  + ***Log.cs*** – This class extend Dictionary<string, string> and is used to store one row of the csv

## Console App Test Cases

For each test case LogFile1.csv is modified to test several aspects of the app. All its versions with corresponding modification of the file will be motivated and explained shortly, visualizing with corresponding excel files.

Mapping the signals names in the file with stimulus and response of Age and Delay Constraint.

***Delay Constraint*** has only one stimulus signal and one response signal

Stimulus – Wiping\_HMI\_rqst

Response – WipingMotorAngle

***Age Constraint*** can have chains of signals between stimulus and response. Therefore, two couples of stimulus and response are taken into consideration for testing.

Stimulus1 – wiperParkStatusOut

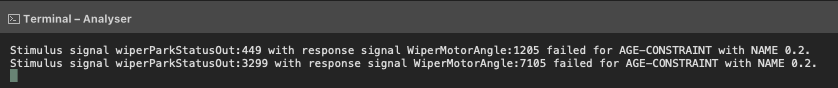
Response1 – WiperMotorAngle

Stimulus2 – wiperMotorLowSpeed

Response2 – bWiperMotorLowSpeed (I added this in the log file to test the chain in age constraints)

The causality order in this case would be: Stimulus1 -> Stimulus2 -> Response2 -> Response1

Test1 – Test AgeConstraint Chain for causality order (code for delay constraint is commented)



Test passed In Test1.xlsx document (Test1.csv in project folder) cells where the value is changed are highlighted with green. The casuality order is met, but the constraint is not met because difference of the timestamps is greater than constraint value

Test2 – Test DelayConstraint (code for age constraint is commented)

Graphical user interface, text

Description automatically generated

Test passed In Test2.xlsx document (Test2.csv in project folder) cells where the value is changed are highlighted with green. There are 15 couples of stimulus response as shown in the terminal.

Test3 – Test AgeConstraint and Delay constraint together

Text

Description automatically generated

Test passed In Test3.xlsx document (Test3.csv in project folder) cells where the value is changed for Delay Constraint are highlighted in green, while cells that change value and pertain to Age Constraint chain are highlighted in red. We have tested here when one signal is response for both constraints (at timestamp 1205 WiperMotorAngle is response for both constraints)