



## PRODUCT DATA SHEET



### TRACE-CHECK accelerates in-depth analyses

TRACE-CHECK is a tool that automatically and thoroughly analyzes measurement data. It is specifically useful, wherever you need test results in the form of reports, which are easy to read and to interpret – no matter where the data comes from or into which tool chain you plan to integrate the tool.

#### Key features at a glance

- Easy analysis specification via
  - Triggered analyses
  - Timing diagrams
  - Python interface
- Support for all common recording formats
- High reusability of analyses
- Intuitive graphical user interface
- Clear presentation of results
  - Transition to the interactive SignalViewer
  - Plots enriched with result data

#### Supported formats

Bus description:

- ARXML (Classic Platform) 4.1.1 to 4.3.1
- ARXML (Adaptive Platform) to 10/18
- DBC
- FIBEX to 4.1.1/FIBEX for Ethernet 4.1.2
- LIN Description File (LDF)

ECU description:

- ASAP2 Database (A2L)
- Executable and Linkable Format (ELF)
- Intel HEX
- Motorola S19

Raw data trace formats:

- ASC, BLF, TTL: CAN and FlexRay recordings
- PCAP: Ethernet recordings

Signal-based trace formats:

- CSV
- ERG: CarMaker
- ECAL 5.0, 5.1
- MAT: MATLAB/Simulink, ControlDesk
- MDF 3.0, 3.1, 3.2, 3.3, 4.0, 4.1
- RDB: VTD
- STI, STZ 2.0.1 ASAM XiL-API
- TDMS: National Instruments

#### Interfaces

**Automation interfaces** (COM) enable all TRACE-CHECK work steps to be controlled, say for a seamless and fully automatic operation in an existing tool chain. On request we can also provide an adapted version of the **ECU-TEST Jenkins plug-in**.

#### Means of specification

Various means of specification are available for the user to formalize requirements in the form of analysis components:

Any logical expressions from signal names, package variables and internal functions can be used in **trigger blocks** and **calculation steps**.

With **timing diagrams** both simple and complex signal relationships can be clearly and at the same time formally described.

The **Python interface** allows the users to implement the analysis specifications themselves. In doing so, they can focus on the actual requirements, as many tasks – from processing traces via the interpolation of different time axes to generating report entries – are carried out automatically by TRACE-CHECK.

For the efficient processing of the signal data the user is provided with all the functionalities of the program libraries NumPy and SciPy.

In addition, TRACE-CHECK provides various methods for synchronizing multiple recordings:

- EqualnessMatching
- ExpectationMatching
- Offset