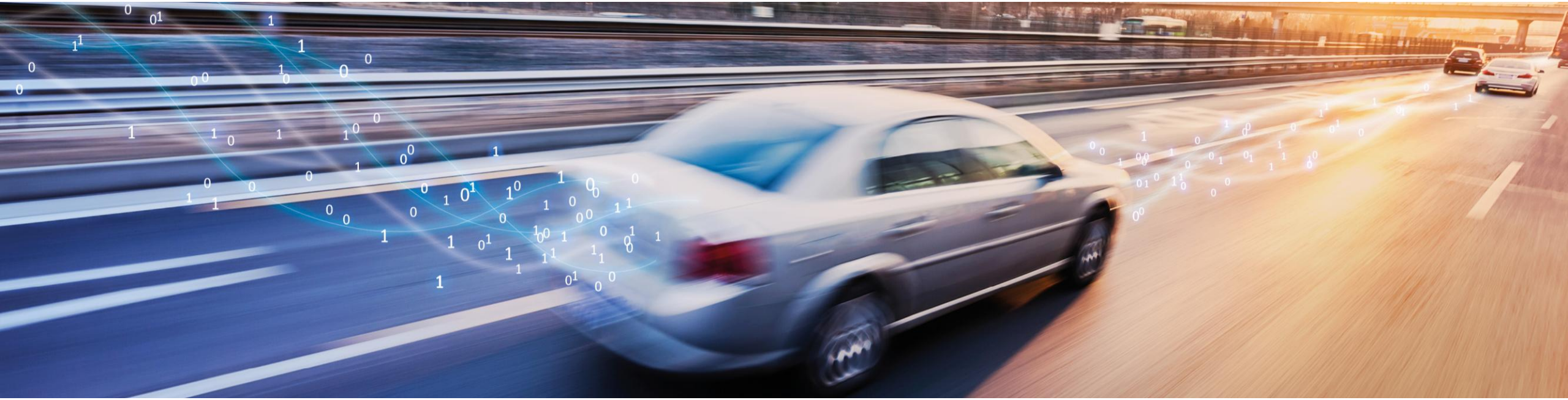
ASAM MDF v4.2.0

Release Presentation

03.07.2019



ASAM Association for Standardization of Automation and Measuring Systems

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Outlook

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Outlook

Introduction

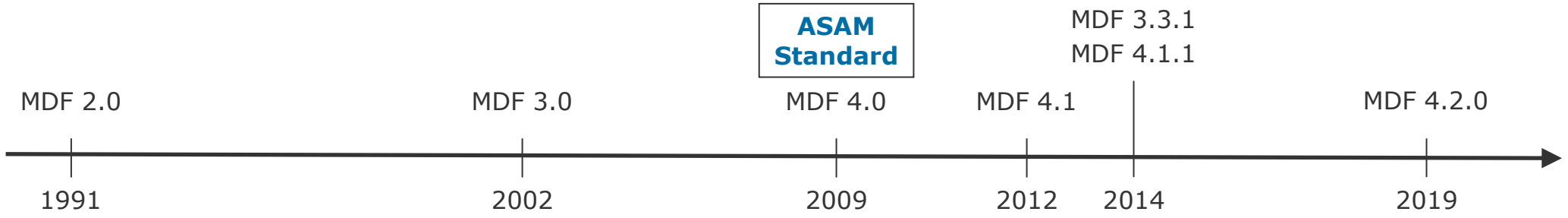
MDF – Measurement Data Format

- Binary file format to store measured or calculated data for post-measurement processing and long-term conservation.
- Common sources of the data to be stored are sensors, ECUs or bus monitoring systems.
- With MDF a high performance can be achieved for both writing and reading signal data.
- In addition to the plain measurement data, MDF also contains descriptive and customizable meta data within the same file.

Introduction

History

- 1990: MDF designed for use in the automotive industry
- 1991 until today: MDF versions 2.x and 3.x have successfully been used over many years and evolved to a de facto standard
- 2009: release of ASAM Common MDF 4.0.0 as result of a major update of the format and standardization by ASAM e.V.
- 2012: release of ASAM Common MDF 4.1.0 including three new associated standards
  - most important new features: compression of data, bus logging
- 2019: release of ASAM Common MDF 4.2.0
  - including new way to store data for enhanced read performance



Introduction

Key Concepts of MDF

- Compact binary format organized in loosely coupled blocks
- Measurement data stored in records according to sampling rate
- Record layout and general signal description given by channels
- Supports multiple and non-periodic sample rates
- Synchronization via master channel concept
- Special data types and meta information used in automotive area
- Data received (e.g. from ECU) can be stored "as is"
- Conversion rules for calculation of physical values from stored raw values
- Extension of meta information by XML or "attachments"
  (embedding or linking of other files)

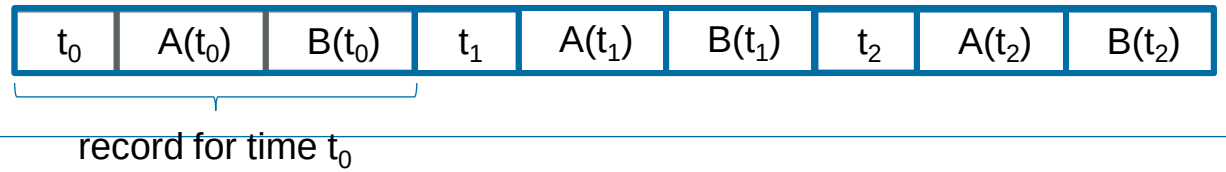
Agenda

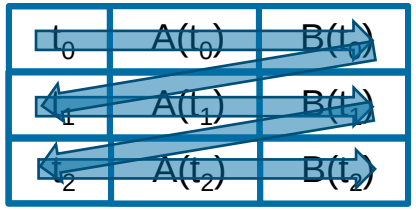
Motivation

Why MDF 4.2

- MDF as storage format for measurement data management systems (e.g. ASAM ODS)
  - data often already delivered as MDF => avoid conversion
  - compact storage including meta data

- Feedback from developers
  - reading signal data is not optimal due to storage in "records"
    - each record typically contains a timestamp and values of the signals acquired simultaneously (same sampling rate / bus message)
    - record layout defined by channel group and contained channels
      - so-called "row-oriented" storage
      - ideal for writing
      - good for reading all signals values at a specific time t_n
      - but: not ideal for reading all values of a specific signal

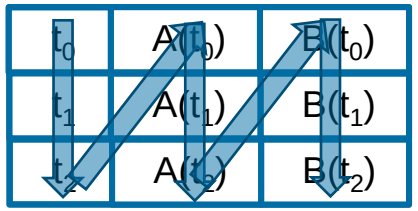




Motivation

Why MDF 4.2

- Other formats use a "column-oriented" storage
  - all values of a signal are stored "en bloc"
  - ideal for fast reading (in most programming languages)



- Idea
  - introduce a new "flavor" of MDF which stores signal values in column-oriented storage
  - must contain identical information
  - simple (offline) transformation similar to "sorting" of MDF file for faster seek & read

Write Once – Read Many

Agenda

New Features in MDF 4.2

Column-Oriented Storage in MDF 4.2

- MDF 4.2 introduces new layout options (in addition to unsorted / sorted)
  - Optimized for reading of single signals / small groups of signals
  - Can be used to store “column oriented”

- With column oriented storage, no channel data from signals that where not requested by the reader is fetched from disc. With the previous sorted configuration, all channel data from the channel group had to be fetched.
- Column oriented storage is not suitable for direct recording (unless in very simple use-cases) due to additional buffering requirements.

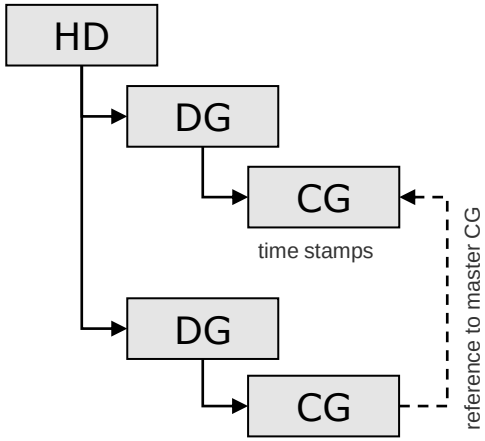
New Features in MDF 4.2

Additional Benefit of Column-Oriented Storage

- Use Case
  - Append a calculated signal to an existing file (with same raster as its input signal)

- Previous Solutions
  - Either duplicate time channel and write the new signal to a new channel group,
  - Or re-assemble the records for an existing group (re-write everything, difficult)

- New Possibility with MDF 4.2
  - Simply write the signal to a new channel group (using "column-oriented storage")
  - Refer to the "master channel group" => no duplication of time values

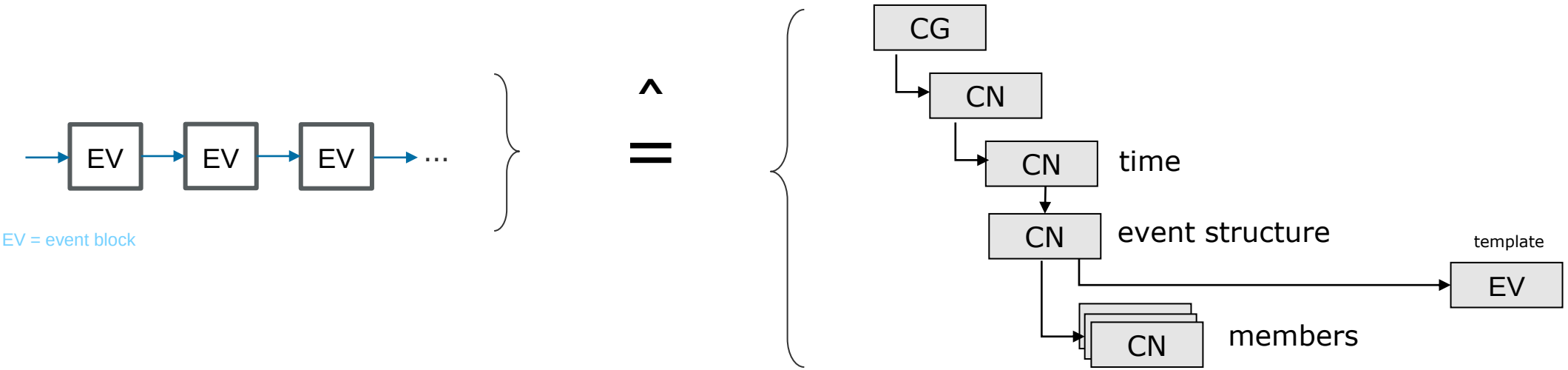


New Features in MDF 4.2

Further New Features to Improve Reading Performance

- Previously: events stored as linked list
  - OK for small number of events, but slow in case of large number (> 1000)
- MDF 4.2 offers an alternative way of storing events in channels ("event signals")
  - store events of same type in a structure and use a "template" event
  - channels are a proven mechanism to handle millions of samples
    - now open for events as well

=> loose a little bit of flexibility for the benefit of more efficient reading

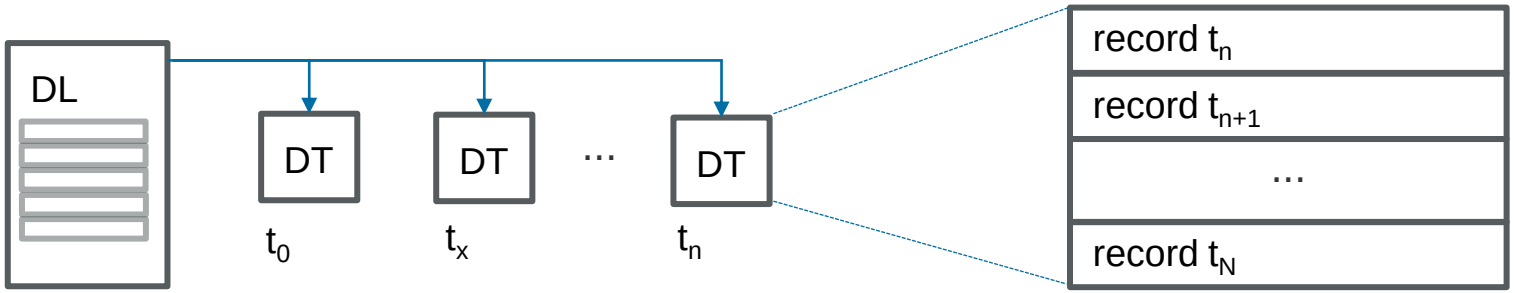


New Features in MDF 4.2

Further New Features to Improve Reading Performance

- Store time stamps for single parts of a distributed data block
  - faster seek of a time stamp by determining relevant partial data block

  => avoids unnecessary reads of partial blocks (esp. for compressed data)



Agenda

New Features in MDF 4.2

Additional New Features

- New Data Types
  - Complex Number
    - enables standard way to store complex number (real and imaginary part)
  - IEEE754 half precision float (2 bytes)
    - market requirement for compact storage of float values
    - retains MDFs capability to write data without transformation (write performance)
- New Conversion Rule
  - Bitfield Text Table
    - mapping of special conversion rule used in FIBEX / AUTOSAR

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	**Backward-Compatibility**
6	Relation to Other Standards
7	Outlook

Backward Compatibility

- ASAM MDF 4.1.0 is an extension of ASAM MDF 4.0.0
  - Backward compatibility: every valid MDF 4.0/4.1/4.11 file is also a valid MDF 4.2 files
  - Forward compatibility: old tools that only support MDF 4.0/4.1/4.11 should be able to read MDF 4.2 files while ignoring the new features.
  - Signal meta data for signals stored in column oriented storage is accessible in this case, but data isn’t.

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Outlook

Relation to other standards

ASAM General Expression Syntax (GES)

- Used for conversion rules and trigger conditions

ASAM Harmonized Objects (HO)

- Used for units

ASAM MCD-2 MC (ASAP2)

- Storage of properties defined in ASAP2

ASAM MCD-2 NET (FIBEX)

- Bitfield text table now supported

ASAM XIL

- Uses MDF for capturing measurement data

Relation to other standards

ASAM ODS

- The new storage layout features of MDF 4.2 allow the creation files that are easier to import into ODS
  - Invalidation bit area is separate and can be sized & filled according to ODS standard
  - COS allows fast reading of single signals – a major use-case of ODS

MDF4 files can be converted to fit well to ODS

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Outlook

Outlook

Summary of MDF

Compact storage of measurement data in binary format

- Capable to store huge amount of measurement data (file size practically not limited)
- MDF 4.1.0 introduces optional compression of the measurement data

High performance for writing and reading

- Data received from ECU can be stored without processing
- Easy writing by simply appending the records produces "unsorted" MDF files
- Loss-less re-organization of file ("sorting") to allow fast index-based access to samples
- Distributed data blocks introduced in MDF 4.0.0 allow direct writing of sorted MDF files
- MDF4.2 column-oriented storage allows optimization of the file for very efficient reading

Description of measurement data within the same file

- Important information in compact binary blocks, optional information in XML

Storage of customized meta information

- Generic XML tags allow easy display of custom information even by other tools

Outlook

Summary of MDF

Specialized for use cases and requirements in Automotive area

- Logging of ECU data and bus traffic
- Specialized data types and structures
- Compliance to other ASAM standards

MDF 4.x continues success of well-established MDF 3.x

- Support by tools continuously increases
- Major OEMs plan to establish MDF 4.x as company standard

MDF4.2 builds on this excellent foundation and extends it by selectively

- Improving performance
- Adding small features where experience shows need

Future versions will continue to follow that route once MDF4.2 has gained some acceptance

Thank you for your attention