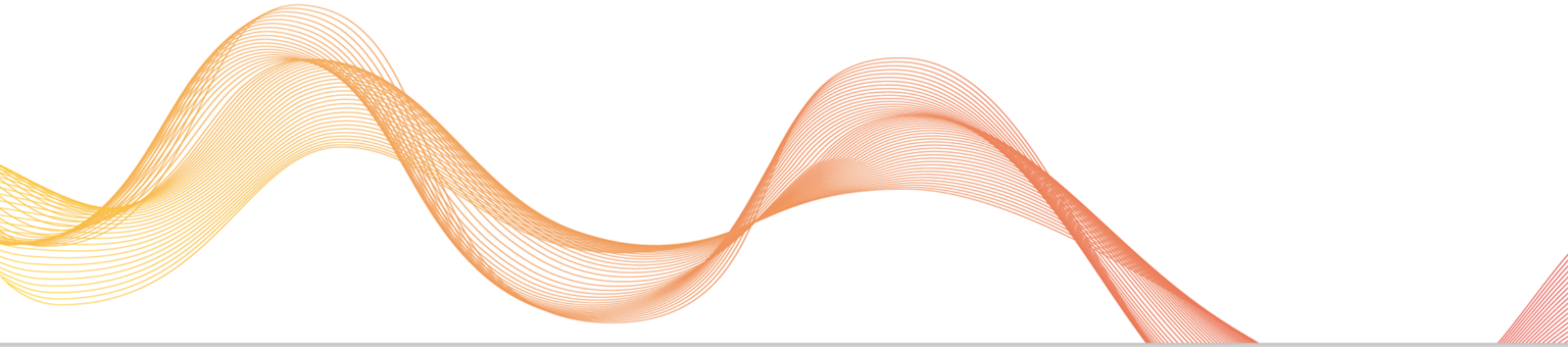
ASAM | Association for standardisation of automation and measuring systems



Release Presentation

P2013-04_AE_MCD-2_MC_BS_V1-7-0_FVD

ASAM MCD2-MC V1.7.0

Release Date: 2015 / 02 / 26

Introduction

- The focus of the standard is still the description of the ECU functionality from a measurement & calibration point of view
- The future development was based on compatibility to earlier versions
- Main Features
  - A **typedef** concept allows the reuse of common description objects. It provides the base for structured objects and reduces the size by removing redundancy.
  - The support of **structures** allows the MC tool to show the relationship of parameters / measurements
  - A **BLOB** as a new object type to handle data blocks as native information
  - A **transformer concept** allows complex calculation of parameter values by an external DLL
  - A **container** allows to combine A2L files and further ECU relevant descriptions in consistent way

Content

- Deliverables
- Marketing
- Main Features
- Keywords
- General
- Compatibility

Content

- **Deliverables**
- Marketing
- Main Features
- Keywords
- General
- Compatibility

Deliverables

- Specification document
  - New functionality included
  - Bug fixes included
  - Updated to newest format
- Revision History
- A2L example file
  - Explains the new functionality
- ASAP2 Checker
  - Update checker rules
  - Checker checks V1.5.1, V1.6.1, V1.7.0

Content

- Deliverables
- **Marketing**
- Main Features
- Keywords
- General
- Compatibility

Marketing

- The new version can be used without modifying existing tool chains
- The new transformer concept supports for ECU optimized code
- The new structure support allows to handle complex data as single object
- A container allows consistent transfer of all files belonging to one ECU
- The ASAP2 Checker strategy is continued

Content

- Deliverables
- Marketing
- **Main Features**
- Keywords
- General
- Compatibility

What’s New?

Type Definitions

- Type definitions for all measure and adjustable objects are now available
- They can be used to describe common properties of several objects without duplicating redundant information
- Furthermore, type definitions are prerequisite to define complex structures – they are used to describe the structure components
- New keywords for type definitions are TYPEDEF_MEASUREMENT, TYPEDEF_CHARACTERISTIC, TYPEDEF_AXIS and TYPEDEF_BLOB
- The usage of type definitions increases reusability and minimizes the size of the data model

What's New?

Type Definitions - INSTANCE

- New keyword INSTANCE is used to define measure and adjustable objects which are derived from a type definition.
- The referenced type definition determines the object type and all basic attributes of the measure or adjustable object
- If a lot of objects are of the same type but differ in a small set of attributes it is possible to use the same type definition and overrule some default attributes with keyword OVERWRITE for each special instance. This avoids a lot of similar type definitions.

What’s New?

Structures

- With the introduction of type definitions it is now possible to describe complex structures of measure or adjustable objects
- This allows the MC tool to display structured data and to represent the semantic togetherness of several measure or adjustable objects
- Structure types are defined with the new keywords TYPEDEF_STRUCTURE and STRUCTURE_COMPONENT
- Structure components are described by referencing other type definitions
- All types of type definitions are allowed to be referenced at a structure component – also other structure types. This allows to build nested structures without restrictions. However, it is not allowed to mix up measure components and adjustable components inside one structure

What’s New?

Structures

- Structure instances are defined using the new keyword INSTANCE
- To refer to other structure components of the same structure or to overwrite default attributes of the referenced type definition, the new keyword THIS was introduced

What’s New?

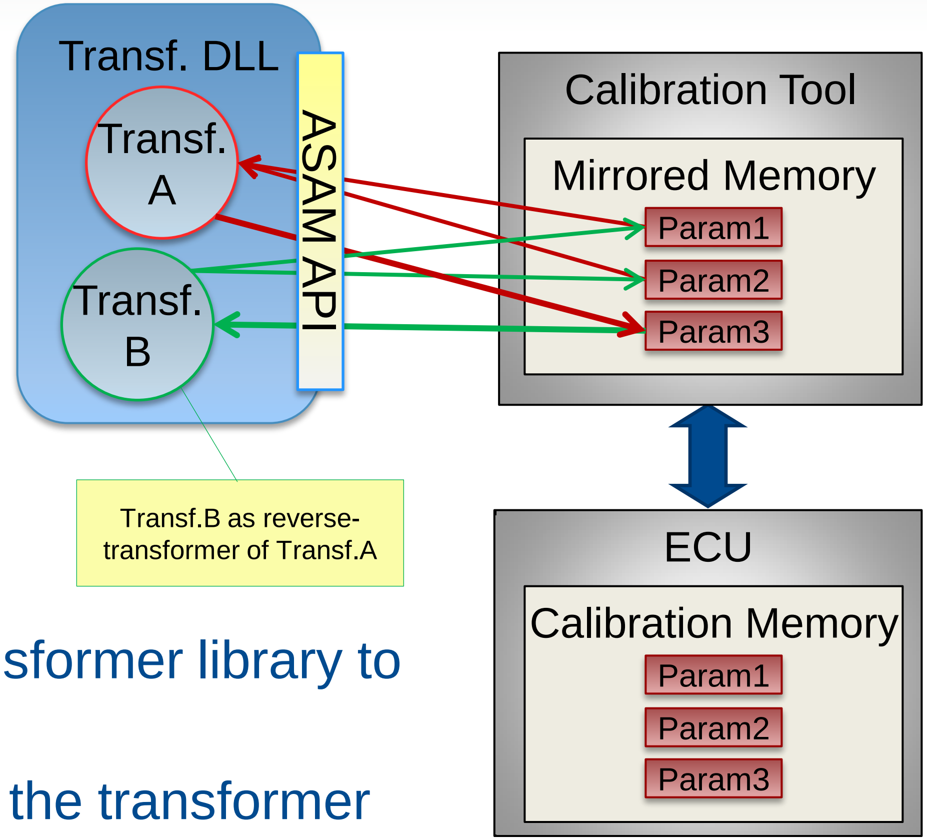
BLOB

- Special data object that can be used to handle domain specific data, which are processed inside the ECU in a dedicated way
- No special physical representation in tools
    - No computation method, no limits, etc.
- Technically, BLOBs can represent packed, encrypted or any other binary data
- Useable especially in conjunction with transformers
    - Calibration objects can be transferred into a BLOB
    - Calibration objects can be extracted from a BLOB
- BLOBs can be uploaded from and downloaded to ECU’s memory like any other adjustable calibration object

What's New?

Transformer Concept

- ECU parameter values can be calculated by an external transformer
- Arbitrary numbers of input and output parameters are possible
- The calibration tool loads the transformer library to perform the transformation
- A DLL API interface is defined for the transformer
- A2L Container can be used for exchange of A2L file including transformer DLL and other DLLs or data (hex files, Seed&Key DLL, etc)

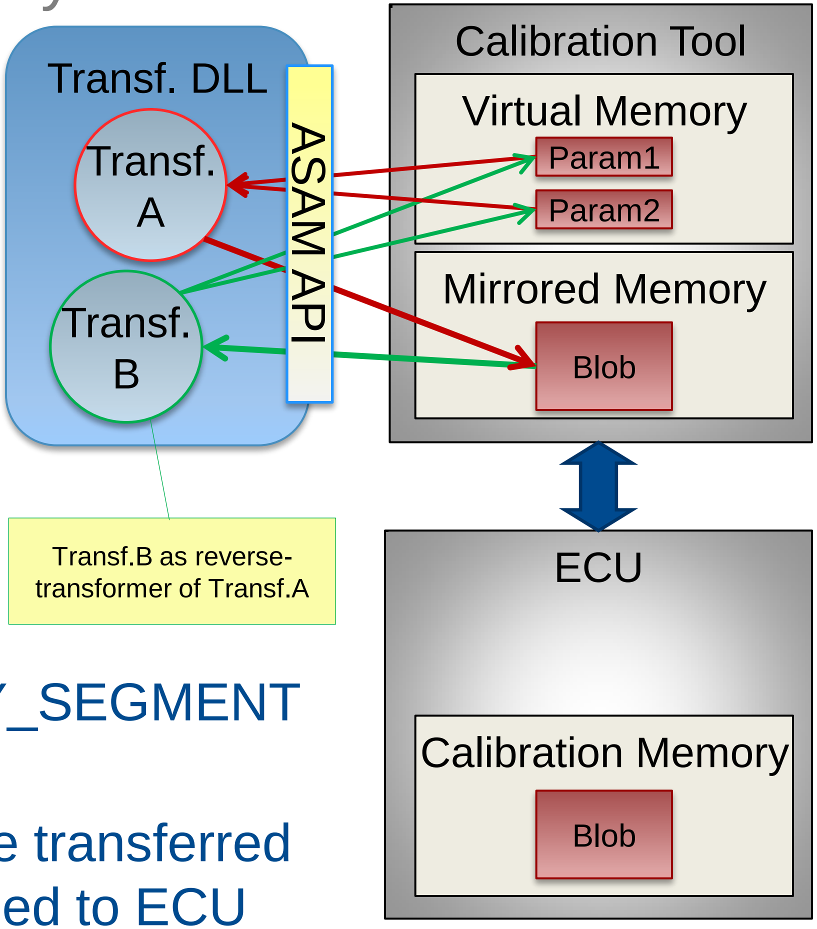


Transf.B as reverse-transformer of Transf.A

What's New?

Transformer Concept - Virtual Memory

- UseCase: Data for calibration is stored in a BLOB in ECU memory
- A transformer is used to transform BLOB data into human readable parameter values; these parameters are not available in ECU memory
- A virtual MEMORY_SEGMENT defines an address space not available in ECU memory
- Parameter values within virtual MEMORY_SEGMENT can be calibrated by the user
- After calibration, the parameter values are transferred back to the BLOB; the BLOB is downloaded to ECU memory



Transf.B as reverse-transformer of Transf.A

What’s New?

Container

- The container offers the possibility to pack all information belonging to a ECU in one single container
  - ZIP based container for files related to ECU
- Meta information how the single files belong to each other
  - Meta information based on ASAM CC standard
- Covers all ECU related files, not only A2L
  - A2L, HEX, Documentation, DLL, …
- Container content accessible with standard ZIP tool

Content

- Deliverables
- Marketing
- Main Features
- **Keywords**
- General
- Compatibility

What’s New?

ADDRESS_TYPE

- Until now, objects of e.g. type MEASUREMENT could only be addressed directly
- They can now be addressed indirectly
- The new and optional keyword ADDRESS_TYPE allows to specify the type of addressing for e.g. MEASUREMENT objects in the same way as for adjustable objects in A2L
  - PBYTE
  - PWORD
  - PLONG
  - PLONGLONG
  - DIRECT

What’s New?

BYTE_ORDER

- Two new byte orders are supported
  - MSB_FIRST_MSW_LAST
  - MSB_LAST_MSW_FIRST

- The new byte order allow to support special CPUs the use for data type long a mix between MSB_LAST and MSB_FIRST
(e.g. signal processors, combinations of CPUs for processing and data exchange, …)

What's New?

ENCODING

- Until now, CHARACTERISTIC of string type (type parameter = “ASCII” at CHARACTERISTIC) supported only single byte character strings
- To support international texts, the optional keyword ENCODING offers the possibility to handle multi byte character strings
- Possible values for ENCODING are
  - UTF8
  - UTF16
  - UTF32

  - If no ENCODING is specified, strings are interpreted as single byte character strings as before

What’s New?

FORMULA

- In formula text, new operators are supported:
  - hex(x)
  - phy(x)
- These operators can be used for the calculation of virtual or dependent parameters or virtual measure objects.
- hex(x) defines that the implementation value of the input object is used for the calculation, phy(x) defines that the physical value of the input object is used

What’s New?

MATRIX_DIM

- The arguments y and z are optional
- The arguments for the dimensions are no longer restricted to 3 dimensions

MATRIX_DIM (uint Dim)+

The number of Dim entries depends on the number of dimensions. For each dimension one Dim entry is necessary

Note: The dimensions value "0" shall not be used.

Note: The dimensions value "1" has a special meaning. This dimension has only one value (set of values). But for data exchange Objects (ASAP3, MCD3, CDF, …) this dimension is respected.

What’s New?

MODEL_LINK

- MODEL_LINK allows to add a further name string to reference to the name in the model.

- The MC tool may transfer the MODEL_LINK additionally to the short name to the data exchange file. This allows to transfer the values from the data exchange file back to the model

What's New?

STATIC_ADDRESS_OFFSETS

- STATIC_ADDRESS_OFFSETS describes a special behavior for record layouts
- This keyword is used to indicate that the start addresses of axes and function values of an adjustable object do not change when removing or inserting axis points. All record layout elements are located at the address offsets as for the max. number of axis points specified at the calibration object - independent of the current number of axis points
- In difference to STATIC_RECORD_LAYOUT the FNC_VALUES are handled dynamically. I.e., the addresses of the single data cells change if the dimension of the map changes

Content

- Deliverables
- Marketing
- Main Features
- Keywords
- **General**
- Compatibility

What’s New?

Name Spaces

- Name spaces are now clearly defined and described in a separate section of the specification
- Inside one name space all identifiers must be unique, duplicates are not allowed
- The following name spaces are defined:
    - FUNCTION
    - GROUP
    - VARIANT
    - MEASUREMENT, CHARACTERISTIC, AXIS_PTS, BLOB, INSTANCE
    - TRANSFORMER,
    - COMPU_METHOD
    - COMPU_VTAB, COMPU_VTAB_RANGE, COMPU_TAB
    - MEMORY_SEGMENT
    - RECORD_LAYOUT
    - TYPEDEF

What’s New?

Reference to Array Elements and Structure Components

- Up to version 1.6.1, only whole objects could be referenced by their object name – e.g. a measure object as input quantity of an axis
- Now it is possible to reference also single array elements or structure components
- A fixed syntax with ‘[‘ and ‘]’ for array indices and ‘.’ for separation of structure components must be used to describe the full path of the referenced elements
- Array elements and structure components can be referenced as input quantity for axes and as input quantity for the calculation of virtual and dependent objects

What’s New?

AML

- The syntax of the ASAP2 meta language was extended due to commonly used applications
- TAGGED_STRUCT
  - The tagged_struct keyword now additionally supports a list of names grouped with /begin - /end
- ENUM
  - The numerical values of enumerators in an enum definition are now defined as 32 bit signed integer values.
- INT64
  - Support of new base types int64 and uint64 (signed and unsigned 64 bit integer values)

Content

- Deliverables
- Marketing
- Main Features
- Keywords
- General
- **Compatibility**

Compatibility

- Downward compatible to earlier ASAP2 versions
  - All former keywords are supported
  - Only extension with new keywords
  - Structures are optional, existing description by single objects still valid
- Checker
  - Checker for new ASAM MCD-2 MC V1.7.0 will be developped