



Elektrobit

# EB tresos<sup>®</sup> AutoCore Generic 8 A2L documentation

product release 8.8.0



Elektrobit Automotive GmbH  
Am Wolfsmantel 46  
91058 Erlangen, Germany  
Phone: +49 9131 7701 0  
Fax: +49 9131 7701 6333  
Email: [info.automotive@elektrobit.com](mailto:info.automotive@elektrobit.com)

## Technical support

<https://www.elektrobit.com/support>

## Legal disclaimer

Confidential information.

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks, and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2020, Elektrobit Automotive GmbH.

# Table of Contents

1. Overview of EB tresos AutoCore Generic 8 A2L documentation .....	4
2. ACG8 A2L release notes .....	5
2.1. Overview .....	5
2.2. Scope of the release .....	5
2.2.1. Configuration tool .....	5
2.2.2. AUTOSAR modules .....	5
2.2.3. EB (Elektrobit) modules .....	5
2.2.4. MCAL modules and EB tresos AutoCore OS .....	6
2.3. Module release notes .....	6
2.3.1. A2L module release notes .....	6
2.3.1.1. Change log .....	6
2.3.1.2. New features .....	9
2.3.1.3. EB-specific enhancements .....	9
2.3.1.4. Deviations .....	10
2.3.1.5. Limitations .....	10
2.3.1.6. Open-source software .....	10
3. ACG8 A2L user's guide .....	11
3.1. Overview .....	11
3.2. Generating the A2L files .....	11
3.3. Generating swc.A2L and bsw.A2L files .....	12
3.4. ACG8 A2L elements generation .....	12
3.4.1. Measurements .....	13
3.4.2. Characteristics .....	14
3.4.2.1. Subcomponent Characteristic Axis Description .....	16
3.4.2.2. Subcomponent Axis RecordLayout .....	17
3.4.2.3. Subcomponent Characteristic Axis Types .....	17
3.4.2.4. Subcomponent Characteristic Axis Point .....	18
3.4.3. Unit .....	18
3.4.3.1. Unit types .....	19
3.4.3.1.1. Unit conversion .....	19
3.4.3.1.2. SI_EXPONENTS .....	19
3.4.4. Control unit management data .....	20
3.4.5. Record layout .....	20
3.4.6. CompuMethods .....	21
3.4.7. Data type calculation .....	22
3.5. Generation of A2L files for XCP configuration .....	23



# 1. Overview of EB tresos AutoCore Generic 8 A2L documentation

Welcome to the EB tresos AutoCore Generic 8 A2L (ACG8 A2L) product documentation.

This document provides:

- ▶ [Chapter 2, “ACG8 A2L release notes”](#): release notes for the ACG8 A2L module
- ▶ [Chapter 3, “ACG8 A2L user's guide”](#): containing background information and instructions

## 2. ACG8 A2L release notes

### 2.1. Overview

This chapter provides the ACG8 A2L product specific release notes. General release notes that are applicable to all products are provided in the EB tresos AutoCore Generic documentation. Refer to the general release notes in addition to the product release notes documented here.

### 2.2. Scope of the release

#### 2.2.1. Configuration tool

Your release of EB tresos AutoCore is compatible with the release of the EB tresos Studio configuration tool:

- ▶ EB tresos Studio: 27.1.0 b200625-0900

#### 2.2.2. AUTOSAR modules

The following table lists the AUTOSAR modules that are part of this ACG8 A2L release.

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
No AUTOSAR modules available				

Table 2.1. Hardware-Independent Modules specified by the AUTOSAR standard

#### 2.2.3. EB (Elektrobit) modules

The following table lists all modules which are part of this release but are not specified by the AUTOSAR standard. These modules include tooling developed by EB or they may hold files shared by all other modules.

Module name	Module version	Supplier
<a href="#">A2L</a>	2.6.3	Elektrobit Automotive GmbH

Table 2.2. Modules not specified by the AUTOSAR standard

## 2.2.4. MCAL modules and EB tresos AutoCore OS

For information about MCAL modules and OS, refer to the respective documentation, which is available as PDF at `$TRESOS_BASE/doc/3.0_EB_tresos_AutoCore_OS` and `$TRESOS_BASE/doc/5.0_MCAL_modules`<sup>1</sup>. It is also available in the online help in EB tresos Studio. Browse to the folders `EB tresos AutoCore OS` and `MCAL modules`.

## 2.3. Module release notes

### 2.3.1. A2L module release notes

- ▶ Module version: 2.6.3.B337087
- ▶ Supplier: Elektrobit Automotive GmbH

#### 2.3.1.1. Change log

This chapter lists the changes between different versions.

##### Module version 2.6.3

2020-06-19

- ▶ Improved the A2L Generator error handling when parsing limits of `PHYS-CONSTRS` or `INTERNAL-CONSTRS` elements by providing a helpful error message when an exception occurs

##### Module version 2.6.2

2020-02-21

- ▶ Updated the A2L Generator to support 64-bit data types with native declaration `signed long long` and `unsigned long long`
- ▶ ASCA2L-199 Fixed known issue: A2L Generator generates an incorrect address granularity for XCP
- ▶ ASCA2L-200 Fixed known issue: A2L Generator sets an incorrect XCP protocol version

---

<sup>1</sup>`$TRESOS_BASE` is the location at which you installed EB tresos Studio.



### Module version 2.6.1

2019-10-11

- ▶ Extended the A2L Generator to support the generation of unique names for compu methods
- ▶ ASCA2L-187 Fixed known issue: A2L Generator generates a string literal instead of an integer value for `XcpEventChannelTimeUnit`

### Module version 2.6.0

2019-06-14

- ▶ ASCA2L-180 Fixed known issue: A2L Generator generates incorrect `COEFFS_LINEAR` for `LINEAR` compu method

### Module version 2.5.0

2019-02-15

- ▶ Updated the A2L Generator to support the generation of A2L files for XCP configuration without a configured transport layer
- ▶ Extended the A2L Generator to support the generation of `COMPU_METHOD` type `LINEAR` for given `compuInternalToPhys` and physical data constraint

### Module version 2.4.0

2018-10-26

- ▶ Extended the A2L Generator to support the generation of `CHARACTERISTICS` for `MCDataInstance` of type categories `CURVE`, `MAP` and `CUBOID` for axis type `COM-AXIS`.

### Module version 2.3.0

2018-06-22

- ▶ Extended the A2L Generator to support the generation of `COMPU_METHOD` instances of category `FORM`
- ▶ Extended the A2L Generator to support the generation of the parameters `LowerLimit` or `UpperLimit` when the physical data constraint is provided. It supports `COMPU_METHODS` `IDENTICAL`, `LINEAR`, and `RAT_FUNC` implementation to handle the physical data representation.

### Module version 2.2.0

2018-02-16

- ▶ Extended the A2L Generator to support the large numerical value specifications
- ▶ Extended the A2L Generator to support the generation of XCP configuration over CAN when `CanIfRx-PduCanIdRange` is configured

### Module version 2.1.0

2017-09-22

- ▶ Extended the A2L Generator to generate only one `MEASUREMENT` or `CHARACTERISTIC` with `MATRIX_DIM` parameter for arrays that contain primitive type elements
- ▶ Updated the A2L Generator to generate the value of `DISPLAY_IDENTIFIER` same as name of the MC-element
- ▶ Updated the A2L Generator to generate the `SYSTEM_CONSTANT` parameter for ECU management data (`MOD_PAR`)
- ▶ Extended the A2L Generator to support the generation of A2L content from XCP configurations
- ▶ A2L Generator supports the generation of `MEASUREMENT` or `CHARACTERISTIC` for `MCDataInstance` of type category `BOOLEAN`
- ▶ Extended the A2L Generator to support the generation of `UNIT` keywords for `MEASUREMENT`, `CHARACTERISTIC`, and `COMPU_METHOD` instances
- ▶ Extended the A2L Generator to support the generation of A2L files for XCP configuration without `MCDataInstances`

### Module version 2.0.0

2017-03-31

- ▶ Extended the A2L Generator to support the generation of `COMPU_METHOD` instances of category `LINEAR`
- ▶ Extended the A2L Generator to support the generation of optional parameter `READ_WRITE` for `MEASUREMENT` and `READ_ONLY` for `CHARACTERISTIC` if applicable
- ▶ Extended the A2L Generator to support the generation of `COMPU_METHOD` instances of category `RAT_FUNC`
- ▶ Extended the A2L Generator unattended wizard to provide configurable options to generate only MC-elements specified in the user-defined flat map. It also provides the option to generate only MC-elements which can be traced to hierarchical model using their name. It also provides the option to generate all the MC-elements with their name set to the short name of the `MCDataInstance`
- ▶ Extended the A2L Generator unattended wizard to provide configurable options to enable and disable the generation of MC-elements related to basic software modules
- ▶ Extended the A2L Generator to generate MC-elements related to application software components into an `swc.a2l` file and generate MC-elements related to basic software modules into an `bsw.a2l` file



- ▶ Updated the A2L Generator to generate the value for the parameter `MAX_DIFF` correctly

### Module version 1.2.0

2016-11-04

- ▶ Generated additional characteristics and measurements if flatmap references are available for the `MCDataInstance`
- ▶ Extended the A2L Generator to support the generation of `COMPU_METHODS` of category `TAB_VERB` and `IDENTICAL`
- ▶ ASCA2L-41 Fixed known issue: The A2L Generator generates empty lines for the parameters `UpperLimit` and `LowerLimit`
- ▶ Extended the A2L Generator to log a `Warning` message instead of an `Error` message, when the project does not contain a valid system model or when the system model does not contain any `MCDataInstances`
- ▶ ASCA2L-40 Fixed known issue: The A2L Generator logs a null pointer exception for invalid or unsupported native declarations
- ▶ ASCA2L-55 Fixed known issue: A2L Generator logs a `java.lang.IllegalArgumentException` when handling a generic `ARRAY` containing composite elements

### Module version 1.0.1

2016-02-11

- ▶ Internal module improvement. This module version update does not affect module functionality

### Module version 1.0.0

2015-06-19

- ▶ Internal module improvement. This module version update does not affect module functionality
- ▶ Initial release of A2L Generator

## 2.3.1.2. New features

- ▶ The A2L Generator is updated to support 64-bit data types.

## 2.3.1.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

#### 2.3.1.4. Deviations

This module is not part of the AUTOSAR specification.

#### 2.3.1.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- ▶ The A2L Generator supports only the generation of `COMPU_METHOD` instances of the categories `TAB_VERB`, `LINEAR`, `IDENTICAL`, `RAT_FUNC`, and `FORM`.
- ▶ The parameter `LongIdentifier` for `MEASUREMENT` and `CHARACTERISTIC` is not handled.
- ▶ The parameter `Resolution` for `MEASUREMENT` is not handled. A default value 0 is generated.
- ▶ The parameter `Accuracy` for `MEASUREMENT` is not handled. A default value 0 is generated.
- ▶ The A2L Generator supports only the generation of configuration related to XCP Interface over CAN, CANFD, FlexRay, and Ethernet.
- ▶ The `FORM COMPU_METHOD` supports only the generation of Formula and Inverse Formula in the form of Text Content.
- ▶ The A2L Generator supports `COMPU_METHODS` `IDENTICAL`, `LINEAR`, and `RAT_FUNC` to handle a physical representation value.
- ▶ The A2L Generator supports the generation of `CHARACTERISTICS` for `MCDataInstance` of type categories `CURVE`, `MAP`, and `CUBOID` for axis type `COM-AXIS` only if axis description `SW-CALPRM-AXIS` is available.
- ▶ The A2L Generator skips the generation of the corresponding `MEASUREMENT/CHARACTERISTIC` for `COMPU_METHOD` type `FORM` when `compuInternalToPhys` and a physical representation value is given.

#### 2.3.1.6. Open-source software

Open-source software information is not available for this module.

## 3. ACG8 A2L user's guide

### 3.1. Overview

This module generates A2L files for all MC-SUPPORT data inside the system model.

### 3.2. Generating the A2L files

To generate the A2L files, use an unattended wizard. Use this unattended wizard to transform all MC-SUPPORT data in the system model.

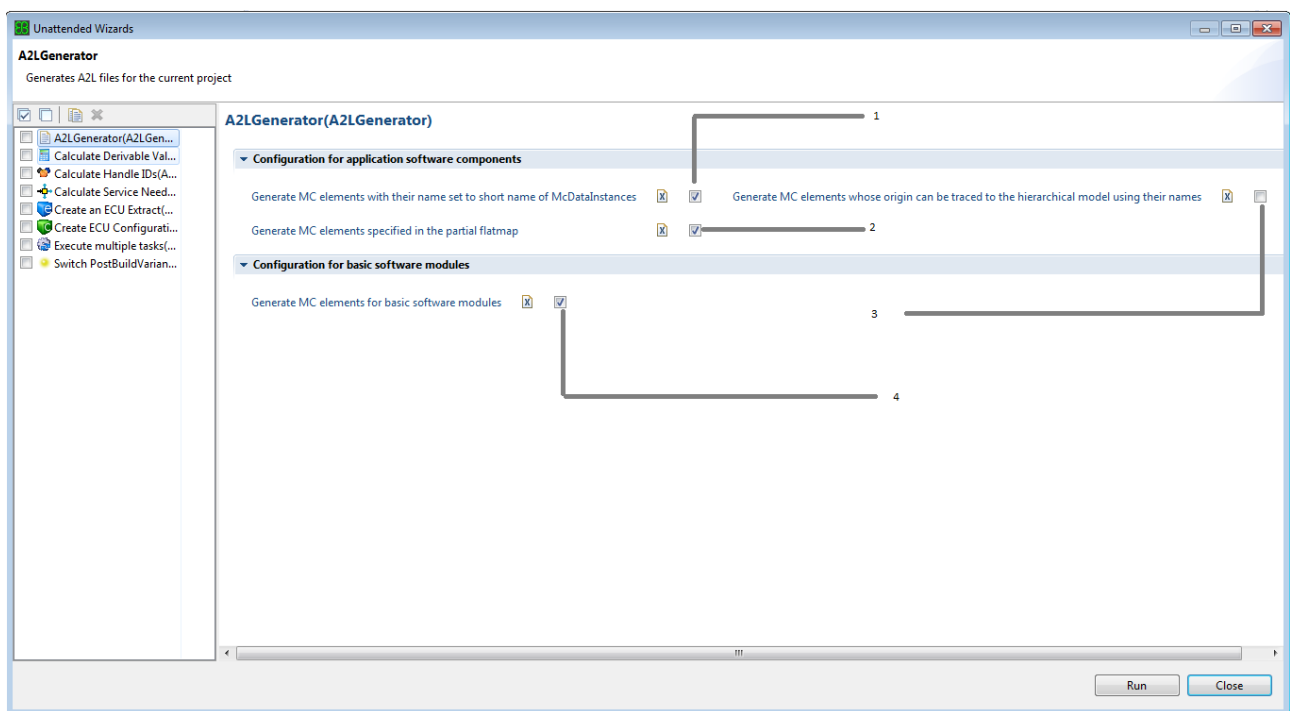


Figure 3.1. Unattended wizard for generating the A2L files

The unattended wizard provides configurable options to influence the generation of measurements and characteristics that belongs to application software components.

Options	Descriptions
Option 1	Generates measurements and characteristics for every MC-DATA-INSTANCE.

Options	Descriptions
Option 2	Generates measurements and characteristics only for <code>MC-DATA-INSTANCE</code> elements that have an equivalent <i>user defined</i> flat map entry.
Option 3	Generates only measurements and characteristics for which a traceable name can be generated using the <code>FLAT-INSTANCE-DESCRIPTOR</code> referenced by <code>MC-DATA-INSTANCE</code> . The name is computed based on context elements of <code>UPSTREAM-REFERENCE-IREF</code> which are aggregated by the <code>FLAT-INSTANCE-DESCRIPTOR</code> .

Table 3.1. Configurable options for generating measurements and characteristics for software components

#### NOTE



#### Using more than one options

If you select more than one option specified in the above table, the A2L Generator could generate two identical measurements or characteristics element for a given `MC-DATA-INSTANCE`. These measurements or characteristics differ only by their name and other parameters remain the same.

You can also enable and disable the generation of measurements and characteristics related to basic software modules using the configurable option 4.

#### NOTE



#### Invoke the A2L Generator

Make sure to invoke the generators for all modules you need A2L data for first, before you invoke the A2L Generator.

When you invoke the generator for a project that does not contain any valid system model or if there are no `MC-DATA-INSTANCE` instances in the system model, warning messages are logged and the A2L files are not generated.

## 3.3. Generating swc.A2L and bsw.A2L files

A2L Generator generates the measurements and characteristics related to application software components into an `swc.a2l` file.

A2L Generator generates the measurements and characteristics related to basic software modules into an `swc.a2l` file.

## 3.4. ACG8 A2L elements generation

The following chapter explain how system model parameters are converted to A2L elements and how their parameters get their values.

### 3.4.1. Measurements

For every MC-DATA-INSTANCE aggregated by MC-VARIABLE-INSTANCES one or more MEASUREMENT elements are generated. If the value of the MC-DATA-INSTANCE/CATEGORY is VALUE or BOOLEAN, then a MEASUREMENT is generated. If the value of the MC-DATA-INSTANC EMC/CATEGORY is STRUCTURE or ARRAY, subelements of MC-DATA-INSTANCE are iterated recursively to get the list of MC-DATA-INSTANCE instances whose CATEGORY evaluates to VALUE. For every such subelement a MEASUREMENT is generated.

The A2L MEASUREMENT has many parameters and the following mapping table describes how these parameters get their values.

A2L parameters	Mapping descriptions
Name	The value of this parameter is set to MC-DATA_INSTANCE/SHORT-NAME.
LongIdentifier	The value of this parameter is set to "Comment/description".
DataType	The value of this parameter is set based on the table specified in <a href="#">Section 3.4.7, "Data type calculation"</a> .
Conversion	The value of this parameter is set based on the conversion table specified in <a href="#">Section 3.4.6, "CompuMethods"</a> .
Resolution	The value of this parameter is set to 0.
Accuracy	The value of this parameter is set to 0.
LowerLimit	<p>The value of this parameter is set to the value of DATA-CONSTR-RULES/DATA-CONSTR-RULE/INTERNAL-CONSTRS/LOWER-LIMIT aggregated by MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/. Note that this value is not checked for validity.</p> <p>The value of this parameter is set to the corresponding internal constraint value of DATA-CONSTR-RULES/DATA-CONSTR-RULE/PHYSICAL-CONSTRS/LOWER-LIMIT aggregated by MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/.</p>
UpperLimit	<p>The value of this parameter is set to the value of DATA-CONSTR-RULES/DATA-CONSTR-RULE/INTERNAL-CONSTRS/UPPER-LIMIT aggregated by MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/. Note that this value is not checked for validity.</p> <p>The value of this parameter is set to the corresponding internal constraint value of DATA-CONSTR-RULES/DATA-CONSTR-RULE/PHYSICAL-CONSTRS/UPPER-LIMIT aggregated by MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/.</p>

A2L parameters	Mapping descriptions
	ERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/.
BIT_MASK	The value of this parameter is set based on the MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/SW-BIT-REPRESENTATION.
ECU_ADDRESS	The value of this parameter is set to 0x.
ECU_ADDRESS_EXTENSION	The value of this parameter is set to 0x.
DISPLAY_IDENTIFIER	The value of this parameter is set to MC-DATA_INSTANCE/SHORT-NAME.
READ_WRITE	This optional parameter is set if the value of MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/SW-CALIBRATION-ACCESS is READ-WRITE.
SYMBOL_LINK	The value of this parameter is set to MC-DATA_INSTANCE/SYMBOL.

Table 3.2. Mapping of MEASUREMENT parameters to the system model parameters

No MEASUREMENT is generated and a warning is logged for an MC-DATA-INSTANCE that refers to a DATA-CONSTR where either LOWER-LIMIT or UPPER-LIMIT is not specified.

If the value of MC-DATA\_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/SW-CALIBRATION-ACCESS is NOT-ACCESSIBLE, then the equivalent measurement is not generated.

## 3.4.2. Characteristics

For every MC-DATA-INSTANCE aggregated by MC-PARAMETER-INSTANCES, one or more CHARACTERISTIC elements are generated. If the value of the MC-DATA-INSTANCE/CATEGORY is VALUE,BOOLEAN, CURVE (1-dimensional array with axes),MAP (2-dimensional array with axes) or CUBOID (3-dimensional array with axes) , then a CHARACTERISTIC is generated. If value of the MC-DATA-INSTANCE/CATEGORY is STRUCTURE or ARRAY, the subelements of MC-DATA-INSTANCE are iterated recursively to get the list of MC-DATA-INSTANCE elements whose CATEGORY evaluates to VALUE and for every such subelement a CHARACTERISTIC is generated.

The A2L CHARACTERISTIC has many parameters and the following mapping table describes how these parameters get their values.

A2L parameters	Mapping descriptions
Name	The value of this parameter is set to MC-DATA_INSTANCE/SHORT-NAME.

A2L parameters	Mapping descriptions
LongIdentifier	The value of this parameter is set to “Comment/description”.
DataType	The value of this parameter is set based on the table specified in <a href="#">Section 3.4.7, “Data type calculation”</a> .
Address	The value of this parameter is set to 0x.
Deposit	The value of this parameter is set based on the table specified in <a href="#">Section 3.4.5, “Record layout”</a> .
MaxDiff	The value of this parameter is set to the difference between UpperLimit and LowerLimit.
Conversion	The value of this parameter is set based on the conversion table specified in <a href="#">Section 3.4.6, “CompuMethods”</a> .
LowerLimit	<p>The value of this parameter is set to the value of DATA-CONSTR-RULES/DA-TA-CONSTR-RULE/INTERNAL-CONSTRS/LOWER-LIMIT aggregated by MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/. Note that this value is not checked for validity.</p> <p>The value of this parameter is set to the corresponding internal constraint value of DATA-CONSTR-RULES/DATA-CONSTR-RULE/PHYSICAL-CONSTRS/LOWER-LIMIT aggregated by MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/.</p>
UpperLimit	<p>The value of this parameter is set to the value of DATA-CONSTR-RULES/DA-TA-CONSTR-RULE/INTERNAL-CONSTRS/UPPER-LIMIT aggregated by MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/. Note that this value is not checked for validity.</p> <p>The value of this parameter is set to the corresponding internal constraint value of DATA-CONSTR-RULES/DATA-CONSTR-RULE/PHYSICAL-CONSTRS/UPPER-LIMIT aggregated by MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/.</p>
BIT_MASK	The value of this parameter is set based on the MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF/SW-BIT-REPRESENTATION.
DISPLAY_IDENTIFIER	The value of this parameter is set to MC-DATA_INSTANCE/SHORT-NAME.

A2L parameters	Mapping descriptions
READ_ONLY	This optional parameter is set if the value of MC-DATA_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/SW-CALIBRATION-ACCESS is READ-ONLY.
SYMBOL_LINK	The value of this parameter is set to MC-DATA_INSTANCE/SYMBOL.
AXIS_DESCR	The value of this parameter is set based on the table specified in <a href="#">Section 3.4.2.1, "Subcomponent Characteristic Axis Description"</a> .

Table 3.3. Mapping of CHARACTERISTIC parameters to the system model parameters

No CHARACTERISTIC is generated and a warning is logged for an MC-DATA-INSTANCE that refers to a DATA-CONSTR where either LOWER-LIMIT or UPPER-LIMIT is not specified.

If the value of MC-DATA\_INSTANCE/RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/SW-CALIBRATION-ACCESS is NOT-ACCESSIBLE, then the equivalent characteristic is not generated.

### 3.4.2.1. Subcomponent Characteristic Axis Description

The Axis Description elements are created based on the axis types such as CURVE/MAP/CUBOID present in the Characteristic . The following table describes AXIS\_DESCR parameters and their description.

Parameters	Descriptions
CalprmAxisCategoryEnum	The value of this parameter is set based on table specified in <a href="#">Section 3.4.2.3, "Subcomponent Characteristic Axis Types"</a>
InputQuantity	Reference to the data record for description of the input quantity . If there is no input quantity assigned, parameter 'InputQuantity' should be set to "NO_INPUT_QUANTITY".
Conversion	Reference to the relevant record of the description of the conversion method (see COMPU_METHOD). If there is no conversion method, as in the case of CURVE_AXIS, the parameter '_Conversion' should be set to —NO_COMPU_METHOD".
MaxAxisPoints	Maximum number of axis points




	<p><b>NOTE</b></p>  <p>The measurement and calibration system can change the dimensions of a characteristic. That means, increase or decrease the number of axis points. The number of axis points may not be increased by random, as the address range is reserved for each characteristic in the ECU program by the measurement and calibration system. This cannot be changed.</p>
LowerLimit	Plausible range of axis point values, lower limit
UpperLimit	Plausible range of axis point values, upper limit
AXIS_PTS_REF	Reference to the AXIS_PTS record for description of the axis points distribution.
DEPOSIT	The axis points of a characteristic can be deposited in two different ways: a) The individual axis point values are deposited as absolute values. b) The individual axis point are stored as differences. Each axis point value is determined from the adjacent axis point (predecessor). Where the standard value does not apply this parameter can be used to specify the axis point deposit.

Table 3.4. Axis Description and Parameter description

### 3.4.2.2. Subcomponent Axis RecordLayout

The `AxisRecordLayout` elements are created based on the data type of the `Characteristic`. The following table describes the `AxisRecordLayout` parameter and their values used.

NAME	Value of this parameter set to ShortName of the Recordlayout
AXIS_PTS_X / _Y / _Z / _4 / _5	Describes where the X, Y, Z, Z4 or Z5 axis points are deposited in memory.
NO_AXIS_PTS_X / _Y / _Z / _4 / _5	Description of the number of axis points in an adjustable object
FNC_VALUES	This keyword describes how the table values (function values) of the adjustable object are deposited in memory

Table 3.5. Mapping of data types to record layouts

### 3.4.2.3. Subcomponent Characteristic Axis Types

This Section specifies the possible values of the category property within `SwCalprmAxis`.

AXIS TYPE	Description
-----------	-------------

COM_AXIS	In COM_AXIS ,input and output axis definition are stored within this CURVE, MAP, CUBOID. COM_AXIS is an shared axis, that means this axis can be used multiple times by different CURVEs, MAPs, CUBOIDs.
----------	--

Table 3.6. Axis Types and Parameter description

#### NOTE



#### Axis Types Support

Supported Axis types: COM\_AXIS

Unsupported Axis types: STD\_AXIS, RES\_AXIS,CURVE\_AXIS, FIX\_AXIS

### 3.4.2.4. Subcomponent Characteristic Axis Point

This section provides describes axis point and supported parameters.

Parameter	Description
Name	Unique identifier in the ECU program.Value of this parameter set to the short-name of the mc-datainstance
LongIdentifier	The value of this parameter hardcoded to "comments/description"
Address	The address of the adjustable object in the emulation memory.
InputQuantity	Reference to the data record for description of the input quantity, see MEASUREMENT). If there is no input quantity assigned, parameter 'InputQuantity' should be set to "NO_INPUT_QUANTITY".
MaxDiff	Maximum float with respect to the adjustment of a table value
Conversion	Reference to the relevant record of the description of the conversion method. If there is no conversion method, the parameter __Conversion' should be set to —NO_COMPU_METHOD".
MaxAxisPoints	The maximum number of axis points.
LowerLimit	Plausible range of axis point values, lower limit.
UpperLimit	Plausible range of axis point values, upper limit.
Deposit	Reference to the relevant data record for description of the record layout.

Table 3.7. Axis Point Parameter description

### 3.4.3. Unit

The UNIT keyword for the description of units.

The A2L UNIT has many parameters and the following mapping table describes how these parameters get their values.

UNIT parameters	Mapping descriptions
Name	The value of this parameter is set to the <code>ShortName</code> attribute of the <code>UNIT</code> attribute of <code>McDataInstance</code> .
LongName	The value of this parameter is set to the <code>LongName</code> attribute of <code>McDataInstance</code> based on the user option.
DisplayName	The value of this parameter is set to the <code>SHORT-NAME</code> attribute of <code>McDataInstance</code> .
Category	For more information, see the section <a href="#">Section 3.4.3.1, “Unit types”</a> .
REF_UNIT	REF_UNIT is used to reference to <code>EXTENDED_SI</code> unit from unit of type <code>DERIVED</code> .
UNIT_CONVERSION	For more information, see the section <a href="#">Section 3.4.3.1.1, “Unit conversion”</a> .
SI_EXPONENTS	For more information, see the table <a href="#">Section 3.4.3.1.2, “SI_EXPONENTS”</a> .

### 3.4.3.1. Unit types

The following table describes `UNIT` categories:

Unit type	Description
DERIVED	UNIT derived from another unit referenced by the optional parameter <code>REF_UNIT</code> .
EXTENDED_SI	EXTENDED_SIUNIT which describes real units is to refer to SI units.

#### 3.4.3.1.1. Unit conversion

The following table describes `UNIT_CONVERSION`:

Parameter	Description
FactorSiToUnit	The value of this parameter is set to the <code>FactorSiToUnit</code> specification of the linear relationship between two units given by describing the conversion from the referenced <code>UNIT</code> to the derived unit.
OffsetSiToUnit	The value of this parameter is set to the <code>OffsetSiToUnit</code> specification of the linear relationship between two units given by describing the conversion from the referenced <code>UNIT</code> to the derived unit.

#### 3.4.3.1.2. SI\_EXPONENTS

The following table describes `SI_EXPONENTS`:

Parameter	Description
Length	Exponent of the base dimension <code>Length</code> with unit <i>meter</i> required to define an EXTENDED_SI UNIT.
Mass	Exponent of the base dimension <code>Mass</code> with unit <i>kilogram</i> required to define an EXTENDED_SI UNIT.
Time	Exponent of the base dimension <code>Time</code> with unit <i>second</i> required to define an EXTENDED_SI UNIT.
ElectricCurrent	Exponent of the base dimension <code>ElectricCurrent</code> with unit <i>ampere</i> required to define an EXTENDED_SI UNIT.
Temperature	Exponent of the base dimension <code>thermodynamic Temperature</code> with unit <i>kelvin</i> required to define an EXTENDED_SI UNIT.
AmountOfSubstance	Exponent of the base dimension <code>AmountOfSubstance</code> with unit <i>mole</i> required to define an EXTENDED_SI UNIT.
LuminousIntensity	Exponent of the base dimension <code>LuminousIntensity</code> with unit <i>candela</i> required to define an EXTENDED_SI UNIT.

### 3.4.4. Control unit management data

The `MOD_PAR` keyword describes the management data to be specified for a device. `MOD_PAR` has many parameters and the following mapping table shows the supported parameters.

MOD_PAR parameter	Mapping description
SYSTEM_CONSTANT	<p>The value of this parameter is set to the <code>measurement system constant value attribute</code> of <code>McSupportData</code>. The attributes of <code>SYSTEM_CONSTANT</code> are as follows:</p> <p>Name: The value of this parameter is set to the <code>ShortName</code> of <code>measurement system constant value attribute</code> of <code>McSupportData</code>.</p> <p>Value: The value of this parameter is set to the <code>VALUE</code> of <code>measurement system constant value attribute</code> of <code>McSupportData</code>.</p>

### 3.4.5. Record layout

The `RECORD_LAYOUT` elements are generated based on the data type of the `CHARACTERISTIC`. The following table describes the mapping between the data type and the `RECORD_LAYOUT`

Data Type	Record layout
UBYTE	RL.FNC.UBYTE
SBYTE	RL.FNC.SBYTE
UWORD	RL.FNC.UWORD
SWORD	RL.FNC.SWORD
ULONG	RL.FNC.ULONG
SLONG	RL.FNC.SLONG
A_UINT64	RL.FNC.A_UINT64
A_INT64	RL.FNC.A_INT64
FLOAT32_IEEE	RL.FNC.FLOAT32_IEEE
FLOAT64_IEEE	RL.FNC.FLOAT64_IEEE

Table 3.8. Mapping of data types to record layouts

### 3.4.6. CompuMethods

The A2L Generator supports the generation of following compu methods:

Compu method	Description
TAB_VERB	If the value of COMPU-METHOD/CATEGORY referred by the MC-DATA_INSTANCE is TEXTTABLE, then a CompuMethod of type TAB_VERB is generated.
IDENTICAL	If the value of COMPU-METHOD/CATEGORY referred by the MC-DATA_INSTANCE is IDENTICAL, then a CompuMethod of type IDENTICAL is generated.
LINEAR	If the value of COMPU-METHOD/CATEGORY referred by the MC-DATA_INSTANCE is LINEAR, then a CompuMethod of type LINEAR is generated. If the value of COMPU-METHOD/CATEGORY referred by the MC-DATA_INSTANCE is SCALE_LINEAR_AND_TEXTTABLE, then a CompuMethod of type LINEAR is generated. The generated CompuMethod refers a compu tab via STATUS_STRING_REF.
RAT_FUNC	If the value of COMPU-METHOD/CATEGORY referred by the MC-DATA_INSTANCE is RAT_FUNC, then a CompuMethod of type RAT_FUNC is generated.
FORM	If the value of COMPU-METHOD/CATEGORY referred by the MC-DATA_INSTANCE is FORM, then a CompuMethod of type FORM is generated.
NO_COMPU_METHOD	For other categories, the default value NO_COMPU_METHOD is set.

Table 3.9. Type of COMPU\_METHODS supported

### 3.4.7. Data type calculation

The data types are calculated based on the `RESULTING-PROPERTIES/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/BASE-TYPE-REF` aggregated by the `MC-DATA-INSTANCE`. The following table describes how the parameter `DataType` is mapped to the `BASE-TYPE` element.

BaseType	DataType
signed char	SBYTE
void	ULONG
signed short	SWORD
signed short int	SWORD
short	SWORD
signed long	SLONG
signed long int	SLONG
long	SLONG
signed int	SLONG
int	SLONG
unsigned char	UBYTE
unsigned short	UWORD
unsigned short int	UWORD
unsigned long	ULONG
unsigned int	ULONG
unsigned long int	ULONG
float	FLOAT32_IEEE
double	FLOAT64_IEEE
float64	FLOAT64_IEEE
float32	FLOAT32_IEEE
boolean	SBYTE
unsigned long long int	A_UINT64
unsigned long long	A_UINT64
signed long long int	A_INT64
signed long long	A_INT64
long long int	A_INT64

BaseType	DataType
long long	A_INT64

Table 3.10. Mapping of data types to record layouts

## 3.5. Generation of A2L files for XCP configuration

The A2L Generator supports the generation of A2L files for XCP configuration. It supports generation of `IF_` - `DATA` for XCP over CAN, XCP over CANFD, XCP over Ethernet, and XCP over FlexRay.