



Association for Standardisation of
Automation and Measuring Systems

ASAM MCD-3 MC

Application Programming Interface for
Measurement and Calibration Server

Part 1 of 2

Version 3.0.0

Date: 2011-09-23

Base Standard

© by ASAM e.V., 2011

Disclaimer

This document is the copyrighted property of ASAM e.V.
Any use is limited to the scope described in the license terms. The license terms can be viewed at www.asam.net/license

Table of contents

1	Introduction	9
2	Scope	10
2.1	Objective of the object model	10
2.1.1	Typographical conventions and mnemonics	10
2.2	Abbreviations	11
2.3	Legends for used graphics	11
2.3.1	Hierarchical diagrams	11
2.3.2	Sequence diagrams	11
2.4	Stereotypes	12
3	Terms and Definition	12
4	Structure of a MC-server	14
5	MC Server	16
5.1	MCD system object	16
5.2	Version information retrieval	17
5.3	Description of Terms	17
5.3.1	General	17
5.3.2	Client-controlled object	17
5.3.3	Location	17
5.3.4	Logical Link (LOGICAL-LINK)	17
5.3.5	Project	18
5.3.6	Server-controlled mutable object:	18
5.3.7	Server-controlled object (shared object)	18
5.4	States of the MCD system object	18
5.5	State changes	20
5.6	Project configuration	20
5.7	Interface structure of MC-server API	21
5.7.1	Hierarchical model overview	21
5.7.2	Separation in database and runtime side	25
5.8	Collections	27
5.8.1	Types and methods	27
5.8.2	Runtime collections	29
5.8.3	Data base collections	30
5.8.4	Handling of collection of ASCIISTRING	31
5.9	EventHandler	31
5.9.1	Registering/deregistering of the EventHandlers	31
5.9.2	Methods of the EventHandlers	33
5.9.3	Eventfilter	36
5.10	Multi-Client capability	41
5.10.1	Requirements	41
5.10.2	Design	43

5.10.3	Proxy in Multi Client Architecture	43
5.10.4	Cooperation Level	46
5.10.5	Symbolic Names of Clients	47
5.10.6	Selection and de-selection of Project in a multi-client setting	48
5.10.7	Notification	49
5.10.8	Remove shared objects	50
5.10.9	Locking	51
5.11	User interactions in the Application System User Interface in parallel to the MC-server	53
5.12	Client Controlled Objects	53
5.13	Resource Release	55
5.13.1	Use cases	55
5.13.2	Requirements	55
5.13.3	Solution	55
5.13.4	Multi-threading considerations	56
5.14	MCD value	56
5.14.1	Value types	56
5.14.2	Behaviour of <MC> MCDResponseParameter::getValue()	56
5.14.3	Method createValue	56
5.15	Entity Relationship Diagrams	57
5.15.1	Relation between DbProject and Logical Link Table	58
5.15.2	Associations of DbLocation	58
5.16	Parent functionality	63
5.17	Logical Link	63
5.17.1	DbBinary	63
5.17.2	Connection overview	63
5.17.3	State diagram of Logical Link	64
5.17.4	Logical Link example	66
5.18	Use cases	67
5.18.1	Instantiation of projects	68
5.18.2	Data base access	69
5.18.3	Destruction	71
5.18.4	Instantiation of Logical Links	71
5.18.5	Removal of Logical Links	73
6	Function block Measurement	74
6.1	Basic Rules	74
6.2	Collector	74
6.2.1	ERD	74
6.2.2	Concept	75
6.2.3	Result access	81
6.3	Use cases	87
6.3.1	Measurement with a collector - activation	87
6.3.2	Measurement with a collector - result access	89
6.3.3	Measurement with a collector - polling for results	92
6.4	Recorder	93
6.4.1	Concept	93
6.4.2	Recorder State Diagram	95
6.4.3	Recorder Result Access	98
6.4.4	Use Case Recorder	100

6.5	Watcher	102
6.5.1	Concept	102
6.5.2	Trigger	104
6.5.3	Use Case Work with Watcher	110
<u>7</u>	<u>Function block Calibration</u>	<u>112</u>
7.1	Kinds of Characteristics	112
7.2	Adjusting of Characteristics	113
7.3	Use case - adjusting / calibration	118
<u>8</u>	<u>Error Codes</u>	<u>121</u>
8.1	Principle	121
8.2	Description of the errors	122
8.2.1	Error free behaviour	122
8.2.2	Parameterisation errors	123
8.2.3	Runtime / ProgramViolation errors	128
8.2.4	Data base errors	149
8.2.5	System errors	150
8.2.6	Communication errors	151
<u>9</u>	<u>Appendix</u>	<u>152</u>
<u>A</u>	<u>Code example: collector result access</u>	<u>152</u>
<u>B</u>	<u>Value reading and setting by string</u>	<u>154</u>
<u>C</u>	<u>Cooperation Level</u>	<u>157</u>
<u>D</u>	<u>Overview of the methods for isModifiedByOtherClient Flag</u>	<u>166</u>
<u>E</u>	<u>Deprecated Methods</u>	<u>170</u>

1 Introduction

This Standard has been established in order to define a universal application programmer interface of a vehicle communication server application. Today's situation in the automotive market requires different vehicle communication interfaces for different vehicle OEMs supporting multiple communication protocols. However, until today, many vehicle communication interfaces are incompatible with regard to interoperability with multiple communication applications and vehicle communication protocols.

Implementation of the MC server concept supports overall cost reduction to the end user.