

# ISO 22900-3:2009-06 (E)

## Road vehicles - Modular vehicle communication interface (MVCi) - Part 3: Diagnostic server application programming interface (D-Server API)

---

<b>Contents</b>		<b>Page</b>
Foreword .....		viii
Introduction .....		ix
1	Scope .....	1
2	Normative references .....	1
3	Terms, definitions, symbols and abbreviated terms .....	2
3.1	Terms and definitions .....	2
3.2	Abbreviated terms .....	3
3.3	Typographical conventions and mnemonics used in this part of ISO 22900 .....	4
3.4	Legends for used graphics .....	5
3.4.1	Hierarchical diagrams .....	5
3.4.2	Sequence diagrams .....	5
3.5	Stereotypes .....	6
4	General considerations .....	6
5	Specification release version information .....	6
6	Structure of MCD systems .....	7
7	Function block common MCD .....	10
7.1	MCD system object .....	10
7.2	Version information retrieval .....	11
7.3	Description of Terms .....	11
7.3.1	General .....	11
7.3.2	Client-controlled object .....	11
7.3.3	Location .....	11
7.3.4	Logical Link (LOGICAL-LINK) .....	12
7.3.5	Project .....	12
7.3.6	Server-controlled mutable object .....	13
7.3.7	Server-controlled object (shared object) .....	13
7.4	States of the MCD system object .....	13
7.5	State changes .....	15
7.6	Project configuration .....	16
7.7	Interface structure of MCD-server API .....	18
7.7.1	Separation in database and runtime side .....	18
7.7.2	Hierarchical model .....	19
7.8	Structure of the database .....	20
7.8.1	Overview .....	20
7.8.2	Associations of DbLocation for MCD .....	20
7.8.3	Database within the field Measurement and Calibration .....	22
7.8.4	Database within the field Diagnostics .....	22
7.9	Collections .....	22
7.9.1	Types and methods .....	22
7.9.2	RunTime collections .....	24
7.9.3	Database collections .....	26
7.9.4	Handling of collection of ASCIISTRING .....	28
7.10	EventHandler .....	28

7.10.1	Registering/deregistering of the EventHandlers .....	28
7.10.2	Methods of the EventHandlers .....	30
7.10.3	Eventfilter .....	35
7.11	Multi-Client capability .....	43
7.11.1	Requirements .....	43
7.11.2	Design .....	46
7.11.3	Proxy in Multi Client Architecture .....	47
7.11.4	Cooperation Level .....	50
7.11.5	Symbolic Names of Clients .....	52
7.11.6	Selection and de-selection of Project and VehicleInfo in a multi-client setting .....	52
7.11.7	Notification .....	53
7.11.8	Remove shared objects .....	54
7.11.9	Locking .....	55
7.12	Client Controlled Objects .....	57
7.13	Resource Release .....	58
7.13.1	Use cases .....	58
7.13.2	Requirements .....	59
7.13.3	Solution .....	59
7.14	Critical Section, Critical Groups of Methods .....	59
7.15	Result access .....	61
7.16	MCD value .....	62
7.16.1	Value types .....	62
7.16.2	Method getValue .....	63
7.16.3	Method setValue .....	63
7.16.4	Method createValue .....	64
7.17	Use cases .....	68
7.17.1	View .....	68
7.17.2	Instantiation of projects .....	68
7.17.3	Database access .....	71
7.17.4	Destruction .....	73
8	Function block Common MD .....	74
8.1	Collector .....	74
8.1.1	ERD .....	74
8.1.2	Concept .....	75
8.1.3	Result access .....	79
8.1.4	Collector usage in diagnostics .....	84
8.2	Use cases .....	85
8.2.1	Measurement with a collector - activation .....	85
8.2.2	Measurement with a collector - result access .....	87
8.2.3	Measurement with a collector - polling for results .....	90
9	Function block Diagnostics .....	91
9.1	Description of Terms .....	91
9.1.1	General .....	91
9.1.2	Access Key .....	91
9.1.3	Functional Class (FUNCTIONAL-CLASS) .....	91
9.1.4	Job (SINGLE-ECU-JOB, MULTIPLE-ECU-JOB) .....	91
9.1.5	Physical Interface Link .....	91
9.1.6	Physical Link .....	92
9.1.7	Physical Vehicle Link (PHYSICAL-VEHICLE-LINK) .....	92
9.2	Structuring of the function block Diagnostics .....	92
9.2.1	Separation in database and runtime side .....	92
9.2.2	Relation between Vehicle Connector Information Table and Logical Link Table .....	94
9.2.3	Hierarchical model .....	95
9.2.4	Entity Relationship Diagrams .....	97
9.3	System Properties .....	109
9.4	Diagnostic DiagComPrimitives and Services .....	109
9.4.1	Diagnostic DiagComPrimitives and States .....	109
9.4.2	Service overview .....	113
9.4.3	Non cyclic single shot diag service .....	120
9.4.4	Cyclic diag service .....	122

9.4.5	Repeated diag service .....	123
9.4.6	Repeated send only diag service .....	124
9.4.7	Repeated receive only diag service .....	125
9.4.8	Updating repetition parameters .....	125
9.4.9	Summary .....	126
9.4.10	Protocol parameters .....	127
9.4.11	Suppress Positive Response .....	140
9.5	Diagnostic variables .....	143
9.6	eEND_OF_PDU as RequestParameter .....	145
9.6.1	Database side .....	145
9.6.2	Runtime side .....	147
9.6.3	COMPCODE .....	148
9.7	Variable length parameters .....	149
9.8	Layer inheritance of services .....	151
9.8.1	Goal .....	151
9.8.2	Layer inheritance of services .....	151
9.8.3	Service handling on functional and physical locations .....	156
9.9	Variant Identification and Selection (VI / VIS) .....	158
9.9.1	Goal .....	158
9.9.2	Variant Identification Algorithm .....	158
9.9.3	General VI/VIS handling considerations .....	166
9.9.4	Deselecting of selected variants .....	168
9.9.5	Request and Response parameters of VI and VIS .....	168
9.9.6	Example Scenarios for VI and VIS .....	173
9.10	Base Variant Identification and Selection .....	183
9.11	Use Cases .....	190
9.11.1	Creation of LogicalLink and usage of DiagComPrimitives .....	190
9.11.2	Removal of communication objects .....	192
9.11.3	Service Handling .....	194
9.11.4	Result access .....	196
9.12	Filtering of results .....	225
9.12.1	Principle .....	225
9.12.2	Handling rules .....	229
9.13	Read DTC .....	230
9.14	Logical Link .....	234
9.14.1	General .....	234
9.14.2	Connection overview .....	234
9.14.3	State diagram of Logical Link .....	235
9.14.4	Logical Link examples .....	243
9.14.5	Gateway handling .....	248
9.14.6	Examples and Relations between Logical Links, Locations and relevant AccessKeys ....	250
9.15	Functional Addressing .....	256
9.16	Tables .....	258
9.16.1	General .....	258
9.16.2	Usage of tables within DiagComPrimitives .....	260
9.17	Dynamically Defined Local Id / Table Parameters (DDLID) .....	263
9.17.1	General .....	263
9.17.2	DDLID principle and requirements .....	263
9.17.3	Lifecycle .....	264
9.18	Internationalisation .....	272
9.18.1	Multi language support .....	272
9.18.2	Units .....	272
9.19	Special Data Groups .....	272
9.20	ECU Flash programming .....	273
9.20.1	Goal .....	273
9.20.2	Description of Terms for ECU-Reprogramming .....	274
9.20.3	Structure of the function block flash programming .....	275
9.20.4	Management of ECU-MEMs .....	288
9.20.5	Physical Memories .....	289
9.20.6	Executing flash sessions .....	291
9.21	Library .....	299
9.22	Java Jobs .....	300

9.22.1	General .....	300
9.22.2	General information Java Jobs .....	300
9.22.3	Types of Java Jobs .....	301
9.22.4	Handling of Java Jobs .....	304
9.22.5	Job execution .....	314
9.22.6	Job example .....	322
9.23	ECU configuration .....	332
9.23.1	General .....	332
9.23.2	ECU Configuration Database Part .....	332
9.23.3	ECU Configuration Runtime Part .....	336
9.23.4	Error Handling .....	339
9.23.5	Initialising an MCDConfigurationRecord .....	339
9.23.6	Offline versus Online Configuration .....	340
9.23.7	Uploading and Downloading Configuration Strings .....	341
9.24	Audiences and Additional Audiences .....	346
9.24.1	General .....	346
9.24.2	Audiences .....	348
9.24.3	Additional Audiences .....	348
9.25	Function Dictionary and Sub-Components .....	349
9.25.1	Terms and requirements .....	349
9.25.2	Functions and function group in ODX .....	349
9.25.3	Function dictionary data model description .....	351
9.25.4	Function dictionary usage scenario .....	353
9.25.5	Sub-Component data model description .....	355
9.25.6	Sub-Component usage scenario .....	356
9.26	ECU States .....	357
9.27	Monitoring vehicle bus traffic .....	360
9.28	Support of VCI module selection and other VCI module features in accordance with D-PDU API Standard .....	362
9.28.1	General .....	362
9.28.2	Definitions .....	362
9.28.3	General behaviour of D-PDU API related D-server methods .....	362
9.28.4	Overview of VCI module related classes .....	363
9.28.5	VCI module selection .....	364
9.28.6	MCDInterface .....	364
9.28.7	VCI module selection sequence .....	365
9.28.8	Interface status events .....	366
9.28.9	MCDInterfaceResource .....	367
9.28.10	Selection of an interface resource .....	367
9.28.11	Send Break Signal .....	368
9.28.12	MCDDbInterfaceCable .....	369
9.28.13	Accessing VCI module features .....	369
9.28.14	Adding Logical Links which are not found in the Vehicle Information .....	370
9.28.15	Behaviour of a MCD-server not supporting VCI Modules in accordance with D-PDU API Standard .....	371
9.29	Mapping of D-PDU API methods .....	372
9.29.1	General .....	372
9.29.2	Initialization and Selection of VCI Modules .....	372
9.29.3	Communication on a Logical Link .....	372
9.29.4	Handling of Communication Parameters .....	374
9.29.5	MCDStartCommunication and MCDStopCommunication .....	376
10	Error Codes .....	377
10.1	Principle .....	377
10.2	Description of the errors .....	379
10.2.1	Error free behaviour .....	379
10.2.2	Parameterisation errors .....	379
10.2.3	Runtime / ProgramViolation errors .....	379
10.2.4	Database errors .....	379
10.2.5	System errors .....	379
10.2.6	Communication errors .....	379
10.2.7	Job error .....	379

<b>Annex A (informative) Code examples .....</b>	<b>380</b>
<b>Annex B (normative) COMPUCODE template .....</b>	<b>387</b>
<b>Annex C (normative) Job Templates .....</b>	<b>388</b>
<b>Annex D (informative) Gateway Handling .....</b>	<b>392</b>
<b>Annex E (normative) Value reading and setting by string .....</b>	<b>393</b>
<b>Annex F (normative) Bus types .....</b>	<b>396</b>
<b>Annex G (normative) System parameter .....</b>	<b>398</b>
<b>Annex H (normative) Cooperation level .....</b>	<b>401</b>
<b>Annex I (normative) System properties .....</b>	<b>428</b>
<b>Annex J (informative) VCI Module selection sequence .....</b>	<b>431</b>
<b>Annex K (informative) Service example illustrations .....</b>	<b>432</b>
<b>Annex L (informative) Exemplary monitoring message format .....</b>	<b>437</b>
<b>Annex M (normative) Overview of the methods for isModifiedByOtherClient flag .....</b>	<b>439</b>
<b>Bibliography .....</b>	<b>447</b>