

# ISO 22900-2:2009-02 (E)

## Road vehicles - Modular vehicle communication interface (MVCI) - Part 2: Diagnostic protocol data unit application programming interface (D-PDU API)

---

<b>Contents</b>		<b>Page</b>
Foreword .....		vi
Introduction .....		vii
1	Scope .....	1
2	Normative references .....	1
3	Terms and definitions .....	2
4	Symbols and abbreviated terms .....	2
5	Specification release version information .....	4
5.1	Specification release version location .....	4
5.2	Specification release version .....	4
6	Modular VCI use cases .....	4
6.1	OEM merger .....	4
6.2	OEM cross vehicle platform ECU(s) .....	4
6.3	Central source diagnostic data and exchange during ECU development .....	5
6.4	OEM franchised dealer and aftermarket service outlet diagnostic tool support .....	5
7	Modular VCI software architecture .....	5
7.1	Overview .....	5
7.2	Modular VCI D-Server software .....	6
7.3	Runtime format based on ODX .....	7
7.4	MVCI protocol module software .....	7
7.5	MVCI protocol module configurations .....	7
8	D-PDU API use cases .....	8
8.1	Overview .....	8
8.2	Use case 1: Single MVCI protocol module .....	8
8.3	Use case 2: Multiple MVCI protocol modules supported by same D-PDU API implementation .....	9
8.4	Use case 3: Multiple MVCI protocol modules supported by different D-PDU API implementations .....	10
9	Diagnostic protocol data unit (D-PDU) API .....	11
9.1	Software requirements .....	11
9.1.1	General requirements .....	11
9.1.2	Vehicle protocol requirements .....	12
9.1.3	Timing requirements for protocol handler messages .....	12
9.1.4	Serialization requirements for protocol handler messages .....	14
9.1.5	Compatibility requirements .....	15
9.1.6	Timestamp requirements .....	16
9.2	API function overview and communication principles .....	17
9.2.1	Terms used within the D-PDU API .....	17
9.2.2	Function overview .....	17
9.2.3	General usage .....	19
9.2.4	Asynchronous and synchronous communication .....	21
9.2.5	Usage of resource locking and resource unlocking .....	22

9.2.6	Usage of ComPrimitives .....	22
9.3	Tool integration .....	38
9.3.1	Requirement for generic configuration .....	38
9.3.2	Tool integrator - use case .....	38
9.4	API functions - interface description .....	40
9.4.1	Overview .....	40
9.4.2	PDUConstruct .....	40
9.4.3	PDUDeconstruct .....	41
9.4.4	PDUIoCtl .....	42
9.4.5	PDUGetVersion .....	43
9.4.6	PDUGetStatus .....	44
9.4.7	PDUGetLastError .....	45
9.4.8	PDUGetResourceStatus .....	47
9.4.9	PDUCreateComLogicalLink .....	48
9.4.10	PUDestroyComLogicalLink .....	50
9.4.11	PDUConnect .....	51
9.4.12	PDUDisconnect .....	53
9.4.13	PDULockResource .....	54
9.4.14	PDUUnlockResource .....	55
9.4.15	PDUGetComParam .....	56
9.4.16	PDUSetComParam .....	63
9.4.17	PDUStartComPrimitive .....	65
9.4.18	PDUCancelComPrimitive .....	69
9.4.19	PDUGetEventItem .....	70
9.4.20	PUDestroyItem .....	71
9.4.21	PDURegisterEventCallback .....	72
9.4.22	EventCallback prototype .....	74
9.4.23	PDUGetObjectid .....	75
9.4.24	PDUGetModuleids .....	76
9.4.25	PDUGetResourceids .....	78
9.4.26	PDUGetConflictingResources .....	79
9.4.27	PDUGetUniqueRespIdTable .....	80
9.4.28	PDUSetUniqueRespIdTable .....	82
9.4.29	PDUModuleConnect .....	87
9.4.30	PDUModuleDisconnect .....	88
9.4.31	PDUGetTimestamp .....	89
9.5	I/O control section .....	90
9.5.1	IOCTL API command overview .....	90
9.5.2	PDU_IOCTL_RESET .....	92
9.5.3	PDU_IOCTL_CLEAR_TX_QUEUE .....	93
9.5.4	PDU_IOCTL_SUSPEND_TX_QUEUE .....	93
9.5.5	PDU_IOCTL_RESUME_TX_QUEUE .....	94
9.5.6	PDU_IOCTL_CLEAR_RX_QUEUE .....	94
9.5.7	PDU_IOCTL_READ_VBATT .....	95
9.5.8	PDU_IOCTL_SET_PROG_VOLTAGE .....	95
9.5.9	PDU_IOCTL_READ_PROG_VOLTAGE .....	96
9.5.10	PDU_IOCTL_GENERIC .....	97
9.5.11	PDU_IOCTL_SET_BUFFER_SIZE .....	97
9.5.12	PDU_IOCTL_GET_CABLE_ID .....	98
9.5.13	PDU_IOCTL_START_MSG_FILTER .....	98
9.5.14	PDU_IOCTL_STOP_MSG_FILTER .....	100
9.5.15	PDU_IOCTL_CLEAR_MSG_FILTER .....	101
9.5.16	PDU_IOCTL_SET_EVENT_QUEUE_PROPERTIES .....	101
9.5.17	PDU_IOCTL_SEND_BREAK .....	102
9.5.18	PDU_IOCTL_READ_IGNITION_SENSE_STATE .....	103
9.6	API functions -- error handling .....	104
9.6.1	Synchronous error handling .....	104
9.6.2	Asynchronous error handling .....	104
9.7	Installation .....	104
9.7.1	Generic description .....	104
9.7.2	Windows installation process .....	105
9.7.3	Linux installation process .....	106

9.7.4	Selecting MVCI protocol modules .....	106
9.8	Application notes .....	106
9.8.1	Interaction with the MDF .....	106
9.8.2	Accessing additional hardware features for MVCI protocol modules .....	106
9.8.3	Documentation and information provided by MVCI protocol module vendors .....	107
9.8.4	Performance Testing .....	107
10	Using the D-PDU API with existing applications .....	108
10.1	SAE J2534-1 and RP1210a existing standards .....	108
11	Data structures .....	108
11.1	API functions -- data structure definitions .....	108
11.1.1	Abstract basic data types .....	108
11.1.2	Definitions .....	109
11.1.3	Bit encoding for UNUM32 .....	109
11.1.4	API data structures .....	110
Annex A (normative) D-PDU API compatibility mappings .....		123
Annex B (normative) D-PDU API standard ComParams and protocols .....		141
Annex C (informative) D-PDU API manufacturer specific ComParams and protocols .....		209
Annex D (normative) D-PDU API constants .....		211
Annex E (normative) Application defined tags .....		225
Annex F (normative) Description files .....		226
Annex G (informative) Resource handling scenarios .....		269
Annex H (informative) D-PDU API partitioning .....		274
Annex I (informative) Use case scenarios .....		278
Annex J (normative) OBD protocol initialization .....		310
Bibliography .....		325