Table 265 — Request message definition - sub-function = reportUserDefMemoryDTCSnapshotRecordByDTCNumber

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Request SID	M	0x19	RDTCI
#2	sub-function = [reportType = reportUserDefMemoryDTCSnapshotRecordByDTCNumber]	M	0x18	LEV_ RUDMDTCSSBDTC
	DTCMaskRecord[] = [			DTCMREC_
#3	DTCHighByte	M	0x00 – 0xFF	DTCHB
#4	DTCMiddleByte	M	0x00 – 0xFF	DTCMB
#5	DTCLowByte]	M	0x00 – 0xFF	DTCLB
#6	DTCSnapshotRecordNumber	M	0x00 – 0xFF	DTCSSRN
#7	MemorySelection	M	0x00 – 0xFF	MEMYS

Table 266 defines the structure of the ReadDTCInformation request message based on the used sub-function parameter.

Table 266 — Request message definition - sub-function = reportUserDefMemoryDTCExtDataRecordByDTCNumber

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Request SID	M	0x19	RDTCI
#2	sub-function = [reportType = reportUserDefMemoryDTCExtDataRecordByDTCNumber]	M	0x19	LEV_ RUDMDTCEDRBDN
	DTCMaskRecord[] = [			DTCMREC_
#3	DTCHighByte	M	0x00 – 0xFF	DTCHB
#4	DTCMiddleByte	M	0x00 – 0xFF	DTCMB
#5	DTCLowByte]	M	0x00 – 0xFF	DTCLB
#6	DTCExtDataRecordNumber	M	0x00 – 0xFF	DTCEDRN
#7	MemorySelection	M	0x00 – 0xFF	MEMYS

Table 267 defines the structure of the ReadDTCInformation request message based on the used sub-function parameter.

Table 267 — Request message definition - sub-function = reportWWHOBDDTCByMaskRecord

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Request SID	M	0x19	RDTCI
#2	sub-function = [reportType = reportWWHOBDDTCByMaskRecord]	M	0x42	LEV_ ROBDDTCBMR
#3	FunctionalGroupIdentifier	M	0x00 – 0xFF	FGID
	DTCSeverityMaskRecord[] = [			DTCSVMREC_
#4	DTCStatusMask	M	0x00 – 0xFF	DTCSM
#5	DTCSeverityMask]	M	0x00 – 0xFF	DTCSVM

Table 268 defines the structure of the ReadDTCInformation request message based on the used sub-function parameter.

Table 268 — Request message definition - sub-function = reportWWHOBDDTCWithPermanentStatus

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Request SID	M	0x19	RDTCI
#2	sub-function = [reportType = reportWWHOBDDTCWithPermanentStatus]	M	0x55	LEV_ RWWHOBDDTCWPS
#3	FunctionalGroupIdentifier	M	0x00 – 0xFF	FGID

11.3.2.2 Request message sub-function parameter $Level (LEV_) definition

The sub-function parameters are used by this service to select one of the DTC report types specified in Table 269. Explanations and usage of the possible levels are detailed below (suppressPosRspMsgIndicationBit (bit 7) not shown).

Table 269 — Request message sub-function definition

Bits 6 – 0	Description	Cvt	Mnemonic
0x00	**ISOSAEReserved** This value is reserved by this document for future definition.	M	ISOSAERESRVD
0x01	**reportNumberOfDTCByStatusMask** This parameter specifies that the server shall transmit to the client the number of DTCs matching a client defined status mask.	U	RNODTCBSM
0x02	**reportDTCByStatusMask** This parameter specifies that the server shall transmit to the client a list of DTCs and corresponding statuses matching a client defined status mask.	U	RDTCBSM
0x03	**reportDTCSnapshotIdentification** This parameter specifies that the server shall transmit to the client all DTCSnapshot data record identifications (DTC number(s) and DTCSnapshot record number(s)).	U	RDTCSSI
0x04	**reportDTCSnapshotRecordByDTCNumber** This parameter specifies that the server shall transmit to the client the DTCSnapshot record(s) associated with a client defined DTC number and DTCSnapshot record number (0xFF for all records).	U	RDTCSSBDTC
0x05	**reportDTCStoredDataByRecordNumber** This parameter specifies that the server shall transmit to the client the DTCStoredDatarecord(s) associated with a client defined DTCStoredData record number (0xFF for all records).	U	RDTCSDBRN
0x06	**reportDTCExtDataRecordByDTCNumber** This parameter specifies that the server shall transmit to the client the DTCExtendedData record(s) associated with a client defined DTC number and DTCExtendedData record number (0xFF for all records, 0xFE for all OBD records).	U	RDTCEDRBDN
0x07	**reportNumberOfDTCBySeverityMaskRecord** This parameter specifies that the server shall transmit to the client the number of DTCs matching a client defined severity mask record.	U	RNODTCBSMR

Table 269 — *(continued)*

Bits 6 – 0	Description	Cvt	Mnemonic
0x08	**reportDTCBySeverityMaskRecord** This parameter specifies that the server shall transmit to the client a list of DTCs and corresponding statuses matching a client defined severity mask record.	U	RDTCBSMR
0x09	**reportSeverityInformationOfDTC** This parameter specifies that the server shall transmit to the client the severity information of a specific DTC specified in the client request message.	U	RSIODTC
0x0A	**reportSupportedDTC** This parameter specifies that the server shall transmit to the client a list of all DTCs and corresponding statuses supported within the server.	U	RSUPDTC
0x0B	**reportFirstTestFailedDTC** This parameter specifies that the server shall transmit to the client the first failed DTC to be detected by the server since the last clear of diagnostic information. Note that the information reported via this sub-function parameter shall be independent of whether or not the DTC was confirmed or aged.	U	RFTFDTC
0x0C	**reportFirstConfirmedDTC** This parameter specifies that the server shall transmit to the client the first confirmed DTC to be detected by the server since the last clear of diagnostic information. The information reported via this sub-function parameter shall be independent of the aging process of confirmed DTCs (e.g. if a DTC ages such that its status is allowed to be reset, the first confirmed DTC record shall continue to be preserved by the server, regardless of any other DTCs that become confirmed afterwards).	U	RFCDTC
0x0D	**reportMostRecentTestFailedDTC** This parameter specifies that the server shall transmit to the client the most recent failed DTC to be detected by the server since the last clear of diagnostic information. Note that the information reported via this sub-function parameter shall be independent of whether or not the DTC was confirmed or aged.	U	RMRTFDTC
0x0E	**reportMostRecentConfirmedDTC** This parameter specifies that the server shall transmit to the client the most recent confirmed DTC to be detected by the server since the last clear of diagnostic information. Note that the information reported via this sub-function parameter shall be independent of the aging process of confirmed DTCs (e.g. if a DTC ages such that its status is allowed to be reset, the first confirmed DTC record shall continue to be preserved by the server assuming no other DTCs become confirmed afterwards).	U	RMRCDTC
0x0F	**reportMirrorMemoryDTCByStatusMask** This parameter specifies that the server shall transmit to the client a list of DTCs out of the DTC mirror memory and corresponding statuses matching a client defined status mask.	U	RMMDTCBSM

Table 269 — *(continued)*

Bits 6 – 0	Description	Cvt	Mnemonic
0x10	**reportMirrorMemoryDTCExtDataRecordByDTCNumber** This parameter specifies that the server shall transmit to the client the DTCExtendedData record(s) - out of the DTC mirror memory - associated with a client defined DTC number and DTCExtendedData record number (0xFF for all records, 0xFE for all OBD records) DTCs.	U	RMMDEDRBDN
0x11	**reportNumberOfMirrorMemoryDTCByStatusMask** This parameter specifies that the server shall transmit to the client the number of DTCs out of mirror memory matching a client defined status mask.	U	RNOMMDTCBSM
0x12	**reportNumberOfEmissionsOBDDTCByStatusMask** This parameter specifies that the server shall transmit to the client the number of emissions-related OBD DTCs matching a client defined status mask. The number of OBD DTCs reported shall only be those which are required to be compatible with emissions-related legal requirements.	U	RNOOEOBDDTCBSM
0x13	**reportEmissionsOBDDTCByStatusMask** This parameter specifies that the server shall transmit to the client a list of emissions-related OBD DTCs and corresponding statuses matching a client defined status mask. The list of OBD DTCs reported shall only be those which are required to be compatible with emissions-related legal requirements.	U	ROBDDTCBSM
0x14	**reportDTCFaultDetectionCounter** This parameter specifies that the server shall transmit to the client a list of current "prefailed" DTCs which have or have not yet been detected as "pending" or "confirmed". The intention of the DTCFaultDetectionCounter is a simple method to identify a growing or intermittent problem which can not be identified / read by the statusOfDTC byte of a particular DTC. The internal implementation of the DTCFaultDetectionCounter shall be vehicle manufacturer specific (e.g., number of bytes, signed versus unsigned, etc.) but the reported value shall be a scaled 1 byte signed value so that +127 (0x7F) represents a test result of "failed" and any other non-zero positive value represents a test result of "prefailed". However DTCs with DTCFaultDetectionCounter with the value +127 shall not be reported according to below stated rule. The DTCFaultDetectionCounter shall be incremented by a vehicle manufacturer specific amount each time the test logic runs and indicates a fail for that test run. A reported DTCFaultDetectionCounter value greater than zero and less than +127 (i.e., 0x01 – 0x7E) indicates that the DTC enable criteria was met and that a non completed test result prefailed at least in one condition or threshold. Only DTCs with DTCFaultDetectionCounters with a non-zero positive value less than +127 (0x7F) shall be reported. The DTCFaultDetectionCounter shall be decremented by a vehicle manufacturer specific amount each time the test logic runs and indicates a pass for that test run. If the DTCFaultDetectionCounter is decremented to zero or below the DTC shall no longer be reported in the positive response message. The value of the DTCFaultDetectionCounter shall not be maintained between operation cycles. If a ClearDiagnosticInformation service request is received the DTCFaultDetectionCounter value shall be reset to zero for all DTCs. Additional reset conditions shall be defined by the vehicle manufacturer. Refer to D.5 for example implementation details.	U	RDTCFDC

Table 269 — *(continued)*

Bits 6 – 0	Description	Cvt	Mnemonic
0x15	**reportDTCWithPermanentStatus** This parameter specifies that the server shall transmit to the client a list of DTCs with "permanent DTC" status as described in 3.1.	U	RDTCWPS
0x16	**reportDTCExtDataRecordByRecordNumber** This parameter specifies that the server shall transmit to the client the DTCExtendedData records associated with a client defined DTCExtendedData record number less than 0xF0.	U	RDTCEDBR
0x17	**reportUserDefMemoryDTCByStatusMask** This parameter specifies that the server shall transmit to the client a list of DTCs out of the user defined DTC memory and corresponding statuses matching a client defined status mask.	U	RUDMDTCBSM
0x18	**reportUserDefMemoryDTCSnapshotRecordByDTCNumber** This parameter specifies that the server shall transmit to the client the DTCSnapshot record(s) – out of the user defined DTC memory - associated with a client defined DTC number and DTCSnapshot record number (0xFF for all records).	U	RUDMDTCSSBDTC
0x19	**reportUserDefMemoryDTCExtDataRecordByDTCNumber** This parameter specifies that the server shall transmit to the client the DTCExtendedData record(s) – out of the user defined DTC memory - associated with a client defined DTC number and DTCExtendedData record number (0xFF for all records).	U	RUDMDTCEDRBDN
0x1A – 0x41	**ISOSAEReserved** This value is reserved by this document for future definition.	M	ISOSAERESRVD
0x42	**reportWWHOBDDTCByMaskRecord** This parameter specifies that the server shall transmit to the client a list of WWH OBD DTCs and corresponding status and severity information matching a client defined status mask and severity mask record.	U	RWWHOBDDTCBMR
0x43 – 0x54	**ISOSAEReserved** This value is reserved by this document for future definition.	M	ISOSAERESRVD
0x55	**reportWWHOBDDTCWithPermanentStatus** This parameter specifies that the server shall transmit to the client a list of WWH OBD DTCs with "permanent DTC" status as described in 3.1.	U	RWWHOBDDTCWPS
0x56 – 0x7F	**ISOSAEReserved** This value is reserved by this document for future definition.	M	ISOSAERESRVD

11.3.2.3 Request message data-parameter definition

Table 270 specifies the data-parameters of the request message.

Table 270 — Request data-parameter definition

Definition
DTCStatusMask The DTCStatusMask contains eight (8) DTC status bits. The definitions for each of the eight bits can be found in D.2. This byte is used in the request message to allow a client to request DTC information for the DTCs whose status matches the DTCStatusMask. A DTCs status matches the DTCStatusMask if any one of the DTCs actual status bits is set to '1' and the corresponding status bit in the DTCStatusMask is also set to '1' (i.e., if the DTCStatusMask is bit-wise logically ANDed with the DTCs actual status and the result is non-zero, then a match has occurred). If the client specifies a status mask that contains bits that the server does not support, then the server shall process the DTC information using only the bits that it does support.
DTCMaskRecord [DTCHighByte, DTCMiddleByte, DTCLowByte] DTCMaskRecord is a 3-byte value containing DTCHighByte, DTCMiddleByte and DTCLowByte, which together represent a unique identification number for a specific diagnostic trouble code supported by a server. The definition of the 3-byte DTC number allows for several ways of coding DTC information. It can either be done — by using the decoding of the DTCHighByte, DTCMiddleByte and DTCLowByte according to the ISO 15031-6 [12] specification. This format is identified by the DTCFormatIdentifier = SAE_J2012-DA_DTCFormat_00, or — by using the decoding of the DTCHighByte, DTCMiddleByte and DTCLowByte according to this part of ISO 14229 which does not specify any decoding method and therefore allows a vehicle manufacturer defined decoding method. This format is identified by the DTCFormatIdentifier = ISO_14229-1_DTCFormat, or — by using the decoding of the DTCHighByte, DTCMiddleByte and DTCLowByte according to the SAE J1939-73 [19] specification. This format is identified by the DTCFormatIdentifier = SAE_J1939-73_DTCFormat, or — by using the decoding of the DTCHighByte, DTCMiddleByte and DTCLowByte according to the ISO 11992-4 [5] specification. This format is identified by the DTCFormatIdentifier = ISO_11992-4_DTCFormat. — by using the decoding of the DTCHighByte, DTCMiddleByte and DTCLowByte according to the ISO 27145-2 [16] specification. This format is identified by the DTCFormatIdentifier = SAE_J2012-DA_WWH-OBD_DTCFormat.
DTCSnapshotRecordNumber DTCSnapshotRecordNumber is a 1-byte value indicating the number of the specific DTCSnapshot data record requested for a client defined DTCMaskRecord via the reportDTCSnapshotByDTCNumber sub-function. DTCSnapshot data record number 0x00 shall be reserved for legislated purposes (e.g., WWH-OBD). DTCSnapshot records in range of 0x01 through 0xFE shall be available for vehicle manufacturer specific usage. A value of 0xFF requests the server to report all stored DTCSnapshot data records at once.
DTCStoredDataRecordNumber DTCStoredDataRecordNumber is a 1-byte value indicating the number of the specific DTCStoredDataRecord requested via the reportDTCStoredDataByRecordNumber sub-function. DTCStoredDataRecordNumber 0x00 shall be reserved for legislated purposes. DTCStoredData records in range of 0x01 through 0xFE shall be available for vehicle manufacturer specific usage. A value of 0xFF requests the server to report all stored DTCStoredData data records at once.

Table 270 — *(continued)*

Definition
DTCExtDataRecordNumber DTCExtDataRecordNumber is a 1-byte value indicating the number of the specific DTCExtendedData record requested for a client defined DTCMaskRecord via the reportDTCExtDataRecordByDTCNumber and reportDTCExtDataRecordByRecordNumber sub-function. For emissions-related servers (OBD compliant ECUs) the DTCExtDataRecordNumber 0x00 shall be reserved for future OBD use. The following DTCExtDataRecordNumber ranges are reserved: — A value of 0x00 is reserved by ISO/SAE. — A value of 0x01 – 0x8F requests the server to report the vehicle manufacturer specific stored DTCExtendedData records. — A value of 0x90 – 0xEF requests the server to report legislated OBD stored DTCExtendedData records. — A value of 0xF0 – 0xFD is reserved by ISO/SAE for future reporting of groups in a single response message. — A value of 0xFE requests the server to report all legislated OBD stored DTCExtendedData records in a single response message. — A value of 0xFF requests the server to report all stored DTCExtendedData records in a single response message.
DTCSeverityMaskRecord [DTCSeverityMask, DTCStatusMask] DTCSeverityMaskRecord is a 2-byte value containing the DTCSeverityMask and the DTCStatusMask (see D.3 and D.2).
DTCSeverityMask The DTCSeverityMask contains three DTC severity bits. The definitions for each of the three bits can be found in D.3. This byte is used in the request message to allow a client to request DTC information for the DTCs whose severity definition matches the DTCSeverityMask. A DTCs severity definition matches the DTCSeverityMask if any one of the DTCs actual severity bits is set to '1' and the corresponding severity bit in the DTCSeverityMask is also set to '1' (i.e., if the DTCSeverityMask is bit-wise logically ANDed with the DTCs actual severity and the result is non-zero, then a match has occurred).
FunctionalGroupIdentifier The FunctionalGroupIdentifier has been introduced to distinguish commands sent by the test equipment between different functional system groups within an electrical architecture which consists of many different ECUs. If an ECU has implemented software of the emissions system as well as other systems which may be inspected during an I/M test it is important that only the DTC information of the requested functional system group is reported. An I/M test should not be failed because another functional system group has DTC information stored. The FunctionalGroupIdentifiers are specified in D.5.
MemorySelection This parameter shall be used to address the respective user defined DTC memory when retrieving DTCs.

11.3.3 Positive response message

11.3.3.1 Positive response message definition

Positive response(s) to the service ReadDTCInformation requests depend on the sub-function in the service request.

Table 271 defines the positive response message format of the sub-function parameter.

Table 271 — Response message definition - sub-function = reportNumberOfDTCByStatusMask, reportNumberOfDTCBySeverityMaskRecord, reportNumberOfMirrorMemoryDTCByStatusMask, reportNumberOfEmissionsOBDDTCByStatusMask

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Response SID	M	0x59	RDTCIPR
#2	reportType = [reportNumberOfDTCByStatusMask reportNumberOfDTCBySeverityMaskRecord reportNumberOfMirrorMemoryDTCByStatusMask reportNumberOfEmissionsOBDDTCByStatusMask]	M	 0x01 0x07 0x11 0x12	LEV_ RNODTCBSM RNODTCBSMR RNOMMDTCBSM RNOOEOBDDTCBSM
#3	DTCStatusAvailabilityMask	M	0x00 – 0xFF	DTCSAM
#4	DTCFormatIdentifier = [SAE_J2012-DA_DTCFormat_00 ISO_14229-1_DTCFormat SAE_J1939-73_DTCFormat ISO_11992-4_DTCFormat SAE_J2012-DA_DTCFormat_04]	M	 0x00 0x01 0x02 0x03 0x04	DTCFID_ J2012-DADTCF00 14229-1DTCF J1939-73DTCF 11992-4DTCF J2012-DADTCF04
 #5 #6	DTCCount[] = [DTCCountHighByte DTCCountLowByte]	 M M	 0x00 – 0xFF 0x00 – 0xFF	DTCC_ DTCCHB DTCCLB

Table 272 defines the positive response message format of the sub-function parameter.

Table 272 — Response message definition - sub-function = reportDTCByStatusMask, reportSupportedDTCs, reportFirstTestFailedDTC, reportFirstConfirmedDTC, reportMostRecentTestFailedDTC, reportMostRecentConfirmedDTC, reportMirrorMemoryDTCByStatusMask, reportEmissionsOBDDTCByStatusMask, reportDTCWithPermanentStatus

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Response SID	M	0x59	RDTCIPR
#2	reportType = [reportDTCByStatusMask reportSupportedDTCs reportFirstTestFailedDTC reportFirstConfirmedDTC reportMostRecentTestFailedDTC reportMostRecentConfirmedDTC reportMirrorMemoryDTCByStatusMask reportEmissionsOBDDTCByStatusMask reportDTCWithPermanentStatus]	M	 0x02 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x13 0x15	LEV_ RDTCBSM RSUPDTC RFTFDTC RFCDTC RMRTFDTC RMRCDTC RMMDTCBSM ROBDDTCBSM RDTCWPS
#3	DTCStatusAvailabilityMask	M	0x00 – 0xFF	DTCSAM
 #4 #5 #6 #7 #8 #9 #10 #11 : #n-3 #n-2 #n-1 #n	DTCAndStatusRecord[] = [DTCHighByte#1 DTCMiddleByte#1 DTCLowByte#1 statusOfDTC#1 DTCHighByte#2 DTCMiddleByte#2 DTCLowByte#2 statusOfDTC#2 : DTCHighByte#m DTCMiddleByte#m DTCLowByte#m statusOfDTC#m]	 C_1 C_1 C_1 C_1 C_2 C_2 C_2 C_2 : C_2 C_2 C_2 C_2	 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF : 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF	DTCASR_ DTCHB DTCMB DTCLB SODTC DTCHB DTCMB DTCLB SODTC : DTCHB DTCMB DTCLB SODTC

Table 272 — *(continued)*

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
C_1: This parameter is only present if DTC information is available to be reported. C_2: This parameter is only present if reportType = reportSupportedDTCs, reportDTCByStatusMask, reportMirrorMemoryDTCByStatusMask, reportEmissionsOBDDTCByStatusMask, reportDTCWithPermanentStatus and more than one DTC information is available to be reported.				

Table 273 defines the positive response message format of the sub-function parameter.

Table 273 — Response message definition - sub-function = reportSnapshotIdentification

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Response SID	M	0x59	RDTCIPR
#2	reportType = [reportDTCSnapshotIdentification]	M	0x03	LEV_ RDTCSSI
#3 #4 #5	DTCRecord[]#1 = [DTCHighByte#1 DTCMiddleByte#1 DTCLowByte#1]	C_1 C_1 C_1	0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF	DTCASR_ DTCHB DTCMB DTCLB
#6	DTCSnapshotRecordNumber#1	C_1	0x00 – 0xFF	DTCSSRN
:	:	:	:	:
#n-3 #n-2 #n-1	DTCRecord[]#m = [DTCHighByte#m DTCMiddleByte#m DTCLowByte#m]	C_2 C_2 C_2	0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF	DTCASR_ DTCHB DTCMB DTCLB
#n	DTCSnapshotRecordNumber#m	C_2	0x00 – 0xFF	DTCSSRN
C_1: The DTCRecord and DTCSnapshotRecordNumber parameter is only present if at least one DTCSnapshot record is available to be reported. C_2: The DTCRecord and DTCSnapshotRecordNumber parameter is only present if more than one DTCSnapshot record is available to be reported.				

Table 274 defines the positive response message format of the sub-function parameter.

Table 274 — Response message definition - sub-function = reportDTCSnapshotRecordByDTCNumber

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Response SID	M	0x59	RDTCIPR
#2	reportType = [reportDTCSnapshotRecordByDTCNumber]	M	0x04	LEV_ RDTCSSBDTC
#3 #4 #5 #6	DTCAndStatusRecord[] = [DTCHighByte DTCMiddleByte DTCLowByte statusOfDTC]	M M M M	0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF	DTCASR_ DTCHB DTCMB DTCLB SODTC
#7	DTCSnapshotRecordNumber#1	C_1	0x00 – 0xFF	DTCSSRN
#8	DTCSnapshotRecordNumberOfIdentifiers#1	C_1	0x00 – 0xFF	DTCSSRNI

Table 274 — *(continued)*

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
	DTCSnapshotRecord[]#1 = [			DTCSSR_
#9	dataIdentifier#1 byte#1 (MSB)	C_1	0x00 – 0xFF	DIDB11
#10	dataIdentifier#1 byte#2 (LSB)	C_1	0x00 – 0xFF	DIDB12
#11	snapshotData#1 byte#1	C_1	0x00 – 0xFF	SSD11
:	:	C_1	:	:
# 11+(p-1)	snapshotData#1 byte#p	C_1	0x00 – 0xFF	SSD1p
:	:	:	:	:
#r-(m-1)-2	dataIdentifier#w byte#1 (MSB)	C_2	0x00 – 0xFF	DIDB21
#r-(m-1)-1	dataIdentifier#w byte#2 (LSB)	C_2	0x00 – 0xFF	DIDB22
#r-(m-1)	snapshotData#w byte#1	C_2	0x00 – 0xFF	SSD21
:	:	C_2	:	:
#r	snapshotData#w byte#m]	C_2	0x00 – 0xFF	SSD2m
:	:	:	:	:
#t	DTCSnapshotRecordNumber#x	C_3	0x00 – 0xFF	DTCSSRN
#t+1	DTCSnapshotRecordNumberOfIdentifiers#x	C_3	0x00 – 0xFF	DTCSSRNI
	DTCSnapshotRecord[]#x = [			DTCSSR_
#t+2	dataIdentifier#1 byte#1 (MSB)	C_3	0x00 – 0xFF	DIDB11
#t+3	dataIdentifier#1 byte#2 (LSB)	C_3	0x00 – 0xFF	DIDB12
#t+5	snapshotData#1 byte#1	C_3	0x00 – 0xFF	SSD11
:	:	C_3	:	:
#t+5+(p-1)	snapshotData#1 byte#p	C_3	0x00 – 0xFF	SSD1p
:	:	:	:	:
#n-(u-1)-2	dataIdentifier#w byte#1 (MSB)	C_4	0x00 – 0xFF	DIDB21
#n-(u-1)-1	dataIdentifier#w byte#2 (LSB)	C_4	0x00 – 0xFF	DIDB22
#n-(u-1)	snapshotData#w byte#1	C_4	0x00 – 0xFF	SSD21
:	:	C_4	:	:
#n	snapshotData#w byte#u]	C_4	0x00 – 0xFF	SSD2u

C_1: The DTCSnapshotRecordNumber and the first dataIdentifier/snapshotData combination in the DTCSnapshotRecord parameter is only present if at least one DTCSnapshot record is available to be reported.

C_2/C_4 There are multiple dataIdentifier/snapshotData combinations allowed to be present in a single DTCSnapshotRecord. This can e.g. be the case for the situation where a single dataIdentifier only references an integral part of data. When the dataIdentifier references a block of data then a single dataIdentifier/snapshotData combination can be used.

C_3: The DTCSnapshotRecordNumber and the first dataIdentifier/snapshotData combination in the DTCSnapshotRecord parameter is only present if all records are requested to be reported (DTCSnapshotRecordNumber set to 0xFF in the request) and more than one record is available to be reported.

Table 275 defines the positive response message format of the sub-function parameter.

Table 275 — Response message definition - sub-function = reportDTCStoredDataByRecordNumber

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	ReadDTCInformation Response SID	M	0x59	RDTCIPR
#2	reportType = [reportDTCStoredDataByRecordNumber]	M	0x05	LEV_ RDTCSDBRN
#3	DTCStoredDataRecordNumber#1	M	0x00 – 0xFF	DTCSDRN
	DTCAndStatusRecord[]#1 = [			DTCASR_
#4	DTCHighByte	C_1	0x00 – 0xFF	DTCHB
#5	DTCMiddleByte	C_1	0x00 – 0xFF	DTCMB
#6	DTCLowByte	C_1	0x00 – 0xFF	DTCLB
#7	statusOfDTC]	C_1	0x00 – 0xFF	SODTC

Table 275 — *(continued)*

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#8	DTCStoredDataRecordNumberOfIdentifiers#1	C_1	0x00 – 0xFF	DTCSDRNI
	DTCStoredDataRecord[]#1 = [			DTCSDR_
#9	dataIdentifier#1 byte#1 (MSB)	C_1	0x00 – 0xFF	DIDB11
#10	dataIdentifier#1 byte#2 (LSB)	C_1	0x00 – 0xFF	DIDB12
#11	DTCstoredData#1 byte#1	C_1	0x00 – 0xFF	DSD11
:	:	:	:	:
#11+(p-1)	DTCstoredData#1 byte#p	C_1	0x00 – 0xFF	DSD1p
:	:	:	:	:
#r-(m-1)-2	dataIdentifier#w byte#1 (MSB)	C_2	0x00 – 0xFF	DIDB21
#r-(m-1)-1	dataIdentifier#w byte#2 (LSB)	C_2	0x00 – 0xFF	DIDB22
#r-(m-1)	DTCstoredData#w byte#1	C_2	0x00 – 0xFF	DSD21
:	:	:	:	:
#r	DTCstoredData#w byte#m]	C_2	0x00 – 0xFF	DSD2m
:	:	:	:	:
#t	DTCStoredDataRecordNumber#x	C_3	0x00 – 0xFF	DTCSDRN
	DTCAndStatusRecord[]#x = [			DTCASR_
#t+1	DTCHighByte	C_3	0x00 – 0xFF	DTCHB
#t+2	DTCMiddleByte	C_3	0x00 – 0xFF	DTCMB
#t+3	DTCLowByte	C_3	0x00 – 0xFF	DTCLB
#t+4	statusOfDTC]	C_3	0x00 – 0xFF	SODTC
#t+5	DTCStoredDataRecordNumberOfIdentifiers#x	C_3	0x00 – 0xFF	DTCSDRNI
	DTCStoredDataRecord[]#x = [			DTCSDR_
#t+6	dataIdentifier#1 byte#1 (MSB)	C_3	0x00 – 0xFF	DIDB11
#t+7	dataIdentifier#1 byte#2 (LSB)	C_3	0x00 – 0xFF	DIDB12
#t+8	DTCstoredData#1 byte#1	C_3	0x00 – 0xFF	DSD11
:	:	:	:	:
#t+8+(p-1)	DTCstoredDataa#1 byte#p	C_3	0x00 – 0xFF	DSD1p
:	:	:	:	:
#n-(u-1)-2	dataIdentifier#w byte#1 (MSB)	C_4	0x00 – 0xFF	DIDB21
#n-(u-1)-1	dataIdentifier#w byte#2 (LSB)	C_4	0x00 – 0xFF	DIDB22
#n-(u-1)	DTCstoredData#w byte#1	C_4	0x00 – 0xFF	DSD21
:	:	:	:	:
#n	DTCstoredData#w byte#u]	C_4	0x00 – 0xFF	DSD2u

C_1: The DTCAndStatusRecord and the first dataIdentifier/DTCStoredData combination in the DTCStoredDataRecord parameter is only present if at least one DTCStoredData record is available to be reported.

C_2/C_4 There are multiple dataIdentifier/DTCStoredData combinations allowed to be present in a single DTCStoredDataRecord. This can e.g. be the case for the situation where a single dataIdentifier only references an integral part of data. When the dataIdentifier references a block of data then a single dataIdentifier/DTCStoredData combination can be used.

C_3: The DTCStoredDataRecordNumber, DTCAndStatusRecord, and the first dataIdentifier/DTCStoredData combination in the DTCStoredDataRecord parameter is only present if all records are requested to be reported (DTCStoredDataRecordNumber set to 0xFF in the request) and more than one record is available to be reported.

Table 276 defines the positive response message format of the sub-function parameter.