DTCExtendedData contains DTC parameter values, which have been identified at the time of the request. A typical use of DTCExtendedData is to store dynamic data associated with the DTC, for example:

- DTC B1 Malfunction Indicator counter which conveys the amount of time (number of engine operating hours) during which the OBD system has operated while a malfunction is active;
- DTC occurrence counter, counts number of driving cycles in which "testFailed" has been reported;
- DTC aging counter, counts number of driving cycles since the fault was latest failed excluding the driving cycles in which the test has not reported "testPassed" or "testFailed";
- specific counters for OBD (e.g. number of remaining driving cycles until the "check engine" lamp is switched off if driving cycle can be performed in a fault free mode);
- time of last occurrence (etc.);
- test failed counter, counts number of reported "testFailed" and possible other counters if the validation is performed in several steps;
- uncompleted test counters, counts numbers of driving cycles since the test was latest completed (i.e. since the test reported "testPassed" or "testFailed");
- retrieve the number of DTCs matching a client defined severity mask;
- retrieve the list of DTCs matching a client defined severity mask record;
- retrieve severity information for a client defined DTC;
- retrieve the status of all DTCs supported by the server;
- retrieve the first DTC failed by the server;
- retrieve the most recently failed DTC within the server;
- retrieve the first DTC confirmed by the server;
- retrieve the most recently confirmed DTC within the server;
- retrieve all current "prefailed" DTCs which have or have not yet been detected as "pending" or "confirmed";
- retrieve DTCExtendedData associated with a client defined DTCExtendedData record status out of the DTC memory;
- retrieve the list of DTCs out of a user defined DTC memory matching a client defined DTC status mask;
- retrieve user defined DTC memory DTCExtendedData record data for a client defined DTC mask;
- retrieve user defined DTC memory DTCSnapshotRecord data for a client defined DTC mask out of the user defined DTC memory;
- retrieve DTC inform

This service uses a SubFunction to determine which type of diagnostic information the client is requesting. Further details regarding each SubFunction parameter are provided in the following subclauses.

This service makes use of the following terms:

- **Enable criteria:** Server/vehicle manufacturer/system supplier specific criteria used to control when the server actually performs a particular internal diagnostic.
- Test pass criteria: Server/vehicle manufacturer/system supplier specific conditions, that define, whether a system being diagnosed is functioning properly within normal, acceptable operating ranges (e.g. no failures exist and the diagnosed system is classified as "OK").
- **Test failure criteria:** Server/vehicle manufacturer/system supplier specific failure conditions that define, whether a system being diagnosed has failed the test.
- Confirmed failure criteria: Server/vehicle manufacturer/system supplier specific failure conditions that define whether the system being diagnosed is definitively problematic (confirmed), warranting storage of the DTC record in long term memory.
- **Occurrence counter:** A counter maintained by certain servers that records the number of instances in which a given DTC test reported a unique occurrence of a test failure.
- Aging: A process whereby certain servers evaluate past results of each internal diagnostic to determine if a confirmed DTC can be cleared from long-term memory, for example in the event of a calibrated number of failure free cycles.

A given DTC value (e.g. 080511_{16}) shall never be reported more than once in a positive response to readDTCInformation with the exception of reading DTCSnapshotRecords, where the response may contain multiple DTCSnapshotRecords for the same DTC.

When using paged-buffer-handling to read DTCs (especially for SubFunction = reportDTCByStatusMask), it is possible that the number of DTCs can decrease while creating the response. In such a case the response shall be filled up with DTC 000000_{16} and DTC status 00_{16} . The client shall treat these DTCs as not present in the response message.

IMPORTANT — The server and the client shall meet the request and response message behaviour as specified in 8.7.

12.3.1.2 Retrieving the number of DTCs that match a client defined status mask (SubFunction = 01₁₆ reportNumberOfDTCByStatusMask)

A client can retrieve a count of the number of DTCs matching a client defined status mask by sending a request for this service with the SubFunction set to reportNumberOfDTCByStatusMask. The response to this request contains the DTCStatusAvailabilityMask, which provides an indication of DTC status bits that are supported by the server for masking purposes. Following the DTCStatusAvailabilityMask the response contains the DTCFormatIdentifier which reports information about the DTC formatting and encoding. The DTCFormatIdentifier is followed by the DTCCount parameter which is a 2-byte unsigned numeric number containing the number of DTCs available in the server's memory based on the status mask provided by the client.

12.3.1.3 Retrieving the list of DTCs that match a client defined status mask (SubFunction = 02₁₆ reportDTCByStatusMask)

The client can retrieve a list of DTCs, which satisfy a client defined status mask by sending a request with the SubFunction byte set to reportDTCBvStatusMask. This SubFunction allows the client to request

the server to rep

The evaluation shall be done as follows: The server shall perform a bit-wise logical AND-ing operation between the mask specified in the client's request and the actual status associated with each DTC supported by the server. In addition to the DTCStatusAvailabilityMask, the server shall return all DTCs for which the result of the AND-ing operation is non-zero (i.e. (statusOfDTC & DTCStatusMask) != 0). 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. If no DTCs within the server match the masking criteria specified in the client's request, no DTC or status information shall be provided following the DTCStatusAvailabilityMask byte in the positive response message.

DTC status information shall be cleared upon a successful ClearDiagnosticInformation request from the client (see DTC status bit definitions in D.2 for further descriptions on the DTC status bit handling in case of a ClearDiagnosticInformation service request reception in the server).

12.3.1.4 Retrieving DTCSnapshot record identification (SubFunction = 03₁₆ reportDTCSnapshotIdentification)

A client can retrieve DTCSnapshot record identification information for all captured DTCSnapshot with request for this service the SubFunction records bv sending а set to reportDTCSnapshotIdentification. The server shall return the list of DTCSnapshot record identification information for all stored DTCSnapshot records. Each item the server places in the response message for a single DTCSnapshot record shall contain a DTCRecord [containing the DTC number (high, middle, and low byte)] and the DTCSnapshot record number. In case multiple DTCSnapshot records are stored for a single DTC then the server shall place one item in the response for each occurrence, using a different DTCSnapshot record number for each occurrence (used for the later retrieval of the record data).

A server can support the storage of multiple DTCSnapshot records for a single DTC to track conditions present at each occurrence of the DTC. Support of this functionality, definition of "occurrence" criteria, and the number of DTCSnapshot records to be supported shall be defined by the system supplier/vehicle manufacturer.

DTCSnapshot record identification information shall be cleared upon a successful ClearDiagnosticInformation request from the client. It is in the responsibility of the vehicle manufacturer to specify the rules for the deletion of stored DTCs and DTCSnapshot data in case of a memory overflow (memory space for stored DTCs and DTCSnapshot data completely occupied in the server).

12.3.1.5 Retrieving DTCSnapshot record data for a client defined DTC mask (SubFunction = 04₁₆ reportDTCSnapshotRecordByDTCNumber)

A client can retrieve captured DTCSnapshot record data for a client defined DTCMaskRecord in conjunction with a DTCSnapshot record number by sending a request for this service with the SubFunction set to reportDTCSnapshotRecordByDTCNumber. The server shall search through its supported DTCs for an exact match with the DTCMaskRecord specified by the client (containing the DTC number (high, middle, and low byte)). The UserDefDTCSnapshotRecordNumber parameter provided in the client's request shall specify a particular occurrence of the specified DTC for which DTCSnapshot record data is being requested.

NOTE 1 The UserDefDTCSnapshotRecordNumber does not share the same address space as the DTCStoredDataRecordNumber.

If the server supports the ability to store multiple DTCSnapshot records for a single DTC (support of this functionality to be defined by system supplier/vehicle manufacturer), then it is recommended that the server also implements the reportDTCSnapshotIdentification SubFunction parameter. It is recommended that the client first requests the identification of DTCSnapshot records stored using the SubFunction parameter reportDTCSnapshotIdentification before requesting a specific DTCSnapshotRecordNut

It is also recommended to support the SubFunction parameter reportDTCSnapshotRecordIdentification in order to give the client the opportunity to identify the stored DTCSnapshot records directly instead of parsing through all stored DTCs of the server to determine if a DTCSnapshot record is stored.

It shall be the responsibility of the system supplier/vehicle manufacturer to define whether DTCSnapshot records captured within such servers store data associated with occurrence information of a failure as part of the ECU documentation.

Along with the DTC number and statusOfDTC, the server shall return a single predefined DTCSnapshotRecord in response to the client's request, if a failure has been identified for the client defined DTCMaskRecord and DTCSnapshotRecordNumber parameters (DTCSnapshotRecordNumber unequal FF_{16}).

NOTE 2 The exact failure criteria is defined by the system supplier/vehicle manufacturer.

The DTCSnapshot record may contain multiple data-parameters that can be used to reconstruct the vehicle conditions (e.g. B+, RPM, time-stamp) at the time of the failure occurrence.

The vehicle manufacturer shall define format and content of the DTCSnapshotRecord. The data reported in the DTCSnapshotRecord first of all contains a dataIdentifier to identify the data that follows. This dataIdentifier/data combination can be repeated within the DTCSnapshotRecord.The usage of one or multiple dataIdentifiers in the DTCSnapshotRecord allows for the storage of different types of DTCSnapshotRecords for a single DTC for different occurrences of the failure. A parameter which indicates the number of record DataIdentifiers contained within each DTCSnapshotRecord shall be provided with each DTCSnapshotRecord to assist data retrieval.

The server shall report one DTCSnapshot record in a single response message, except the client has set the UserDefDTCSnapshotRecordNumber to FF_{16} , because this shall cause the server to respond with all DTCSnapshot records stored for the client defined DTCMaskRecord in a single response message. The DTCAndStatusRecord is only included one time in the response message. If the client has used FF_{16} for the parameter DTCSnapshotRecordNumber in its request, the server shall report all DTCSnapshot records captured for the particular DTC in numeric ascending order.

The server shall negatively respond if the DTCMaskRecord or DTCSnapshotRecordNumber parameters specified by the client are invalid or not supported by the server. This is to be differentiated from the case in which the DTCMaskRecord and/or DTCSnapshotRecordNumber parameters specified by the client are indeed valid and supported by the server, but have no DTCSnapshot data associated with it (e.g. because a failure event never occurred for the specified DTC or record number). The server shall send the positive response containing only the DTCAndStatusRecord [echo of the requested DTC number (high, middle, and low byte) plus the statusOfDTC].

DTCSnapshot information shall be cleared upon a successful ClearDiagnosticInformation request from the client. It is in the responsibility of the vehicle manufacturer to specify the rules for the deletion of stored DTCs and DTCSnapshot data in case of a memory overflow (memory space for stored DTCs and DTCsnapshot data completely occupied in the server).

12.3.1.6 Retrieving DTCStoredData record data for a client defined record number (SubFunction = 05₁₆ reportDTCStoredDataByRecordNumber)

A client can retrieve captured DTCStoredData record data for a DTCStoredDataRecordNumber by sending a request for this service with the SubFunction set to reportDTCStoredDataByRecordNumber. The server shall search through its stored DTCStoredData records for the match of the client provided record number.

The DTCStoredDataRecordNumber does not share the same address space as the DTCSnapshotRecordNumber.

ISO 14229-1:2020(E)

It shall be the responsibility of the system supplier/vehicle manufacturer to define whether DTCStoredData records captured within such servers store data associated with the first or most recent occurrence of a failure.

NOTE The exact failure criteria is defined by the system supplier/vehicle manufacturer.

The DTCStoredData record may contain multiple data-parameters that can be used to reconstruct the vehicle conditions (e.g. B+, RPM, time-stamp) at the time of the failure occurrence.

The vehicle manufacturer shall define format and content of the DTCStoredDataRecordNumber. The data reported in the DTCStoredDataRecord first of all contains a dataIdentifier to identify the data that follows. This dataIdentifier/data combination can be repeated within the DTCStoredDataRecord. The usage of one or multiple dataIdentifiers in the DTCStoredDataRecord allows for the storage of different types of DTCStoredDataRecords for a single DTC for different occurrences of the failure. A parameter which indicates the number of record DataIdentifiers contained within each DTCStoredDataRecord shall be provided with each DTCStoredDataRecord to assist data retrieval.

The server shall report one DTCStoredDataRecord in a single response message, except the client has set the DTCStoredDataRecordNumber to FF_{16} , because this shall cause the server to respond with all DTCStoredDataRecords stored in a single response message.

In case the client requested to report all DTCStoredDataRecords by record number, then the DTCAndStatusRecord shall be repeated in the response message for each stored DTCStoredDataRecord.

The server shall negatively respond if the DTCStoredDataRecordNumber parameters specified by the client are invalid or not supported by the server. This shall be differentiated from the case in which the DTCStoredDataRecordNumber parameters specified by the client are indeed valid and supported by the server, but have no DTCStoredDataRecord data associated with it (e.g. because a failure event never occurred for the specified record number). In this case, the server shall send the positive response containing only the DTCStoredDataRecordNumber (echo of the requested record number).

DTCStoredDataRecord information shall be cleared upon a successful ClearDiagnosticInformation request from the client. It is in the responsibility of the vehicle manufacturer to specify the rules for the deletion of stored DTCs and DTCStoredDataRecord data in case of a memory overflow (memory space for stored DTCs and DTCStoredDataRecord data completely occupied in the server).

12.3.1.7 Retrieving DTCExtendedData record data for a client defined DTC mask and a client defined DTCExtendedData record number (SubFunction = 06₁₆ reportDTCExtDataRecordByDTCNumber)

A client can retrieve DTCExtendedData for a client defined DTCMaskRecord in conjunction with a DTCExtendedData record number by sending a request for this service with the SubFunction set to reportDTCExtDataRecordByDTCNumber. The server shall search through its supported DTCs for an exact match with the DTCMaskRecord specified by the client [containing the DTC number (high, middle, and low byte)]. In this case, the DTCExtDataRecordNumber parameter provided in the client's request shall specify a particular DTCExtendedData record of the specified DTC for which DTCExtendedData is being requested.

Along with the DTC number and statusOfDTC, the server shall return a single predefined DTCExtendedData record in response to the client's request (DTCExtDataRecordNumber unequal to FE_{16} or FF_{16}).

The vehicle manufacturer shall define format and content of the DTCExtDataRecord. The structure of the data reported in the DTCExtDataRecord is defined by the DTCExtDataRecordNumber in a similar way to the definition of data within a record DataIdentifier. Multiple DTCExtDataRecordNumbers and associated DTCExtDataRecords may be included in the response. The usage of one or multiple DTCExtDataRecordNumbers allows for the storage of different types of DTCExtDataRecords for a single DTC.

The server shall report one DTCExtendedData record in a single response message, except the client has set the DTCExtDataRecordNumber to FE_{16} or FF_{16} , because this shall cause the server to respond with all DTCExtendedData records stored for the client defined DTCMaskRecord in a single response message.

The server shall negatively respond if the DTCMaskRecord or DTCExtDataRecordNumber parameters specified by the client are invalid or not supported by the server. This includes the case where a DTCExtDataRecordNumber of FE₁₆ is sent by the client, but no OBD extended data records (90₁₆ to EF₁₆) are supported by the server. This shall be differentiated from the case in which the DTCMaskRecord and/or DTCExtDataRecordNumber parameters specified by the client are indeed valid and supported by the server, but have no DTC extended data associated with it (e.g. because of memory overflow of the extended data). In case of reportDTCExtDataRecordByDTCNumber the server shall send the positive response containing only the DTCAndStatusRecord [echo of the requested DTC number (high, middle, and low byte) plus the statusOfDTC].

Clearance of DTCExtendedData information upon the reception of a ClearDiagnosticInformation service is specified in 11.2.1. It is in the responsibility of the vehicle manufacturer to specify the rules for the deletion of stored DTCs and DTC extended data in case of a memory overflow (memory space for stored DTCs and DTC extended data completely occupied in the server).

12.3.1.8 Retrieving the number of DTCs that match a client defined severity mask record (SubFunction = 07₁₆ reportNumberOfDTCBySeverityMaskRecord)

A client can retrieve a count of the number of DTCs matching a client defined severity status mask record bv sending reauest for this service with the SubFunction а set to reportNumberOfDTCBySeverityMaskRecord. The server shall scan through all supported DTCs, performing a bit-wise logical AND-ing operation between the mask record specified by the client with the actual information of each stored DTC.

(((statusOfDTC & DTCStatusMask) != 0) && ((severity & DTCSeverityMask) != 0)) == TRUE

For each AND-ing operation yielding a TRUE result, the server shall increment a counter by 1. If the client specifies a status mask within the mask record 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. Once all supported DTCs have been checked once, the server shall return the DTCStatusAvailabilityMask and resulting 2-byte count to the client.

If no DTCs within the server match the masking criteria specified in the client's request, the count returned by the server to the client shall be 0. The reported number of DTCs matching the DTC status mask is valid for the point in time when the request was made. There is no relationship between the reported number of DTCs and the actual list of DTCs read via the SubFunction reportDTCByStatusMask, because the request to read the DTCs is done at a different point in time.

12.3.1.9 Retrieving severity and functional unit information that match a client defined severity mask record (SubFunction = 08₁₆ reportDTCBySeverityMaskRecord)

The client can retrieve a list of DTC severity and functional unit information, which satisfy a client defined severity mask record by sending a request with the SubFunction byte set to reportDTCBySeverityMaskRecord. This SubFunction allows the client to request the server to report all DTCs with a certain severity and status that are "testFailed" OR "confirmed" OR "etc.". The evaluation shall be done as follows.

The server shall perform a bit-wise logical AND-ing operation between the DTCSeverityMask and the DTCStatusMask specified in the client's request and the actual DTCSeverity and statusOfDTC associated with each DTC supported by the server.

In addition to the DTCStatusAvailabilityMask, server shall return all DTCs for which the result of the AND-ing operation in TDUE

(((statusOfDTC & DTCStatusMask) !=0) && ((severity & DTCSeverityMask) != 0)) == TRUE

If the client specifies a status mask within the mask record 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. If no DTCs within the server match the masking criteria specified in the client's request, no DTC or status information shall be provided following the DTCStatusAvailabilityMask byte in the positive response message.

12.3.1.10Retrieving severity and functional unit information for a client defined DTC (SubFunction = 09₁₆ reportSeverityInformationOfDTC)

A client can retrieve severity and functional unit information for a client defined DTCMaskRecord by sending a request for this service with the SubFunction set to reportSeverityInformationOfDTC. The server shall search through its supported DTCs for an exact match with the DTCMaskRecord specified by the client [containing the DTC number (high, middle, and low byte)].

12.3.1.11Retrieving the status of all DTCs supported by the server (SubFunction = 0A₁₆ reportSupportedDTC)

A client can retrieve the status of all DTCs supported by the server by sending a request for this service with the SubFunction set to reportSupportedDTCs. The response to this request contains the DTCStatusAvailabilityMask, which provides an indication of DTC status bits that are supported by the server for masking purposes. Following the DTCStatusAvailabilityMask, the response also contains the listOfDTCAndStatusRecord, which contains the DTC number and associated status for every diagnostic trouble code supported by the server.

12.3.1.12Retrieving the first/most recent failed DTC (SubFunction = 0B₁₆ reportFirstTestFailedDTC, SubFunction = 0D₁₆ reportMostRecentTestFailedDTC)

The client can retrieve the first/most recent failed DTC from the server by sending a request with the SubFunction byte set to "reportFirstTestFailedDTC" or "reportMostRecentTestFailedDTC", respectively. Along with the DTCStatusAvailabilityMask, the server shall return the first or most recent failed DTC number and associated status to the client.

No DTC/status information shall be provided following the DTCStatusAvailabilityMask byte in the positive response message if there were no failed DTCs logged since the last time the client requested the server to clear diagnostic information. Also, if the status of only one DTC is failed since the last time the client requested the server to clear diagnostic information, the one failed DTC shall be returned to both reportFirstTestFailedDTC and reportMostRecentTestFailedDTC requests from the client.

Record of the first/most recent failed DTC shall be independent of the ageing process of confirmed DTCs.

As mentioned above, first/most recent failed DTC information shall be cleared upon a successful ClearDiagnosticInformation request from the client (see DTC status bit definitions in D.2 for further descriptions on the DTC status bit handling in case of a ClearDiagnosticInformation service request reception in the server).

12.3.1.13Retrieving the first/most recently detected confirmed DTC (SubFunction = 0C₁₆ reportFirstConfirmedDTC,SubFunction = 0E₁₆ reportMostRecentConfirmedDTC)

The client can retrieve the first/most recent confirmed DTC from the server by sending a request with the SubFunction byte set to "reportFirstConfirmedDTC" or "reportMostRecentConfirmedDTC", respectively. Along with the DTCStatusAvailabilityMask, the server shall return the first or most recent confirmed DTC number and associated status to the client.

No DTC/status information shall be provided following the DTCS logged since the last time the client requested the server to This is a preview. Click here to purchase the full publication. Ince the last time the

the client requested the server to clear diagnostic information the one confirmed DTC shall be returned to both reportFirstConfirmedDTC and reportMostRecentConfirmedDTC requests from the client.

The record of the first confirmed DTC shall be preserved in the event that the DTC failed at one point in the past, but then satisfied aging criteria prior to the time of the request from the client (regardless of any other DTCs that become confirmed after the aforementioned DTC is confirmed). Similarly, record of the most recently confirmed DTC shall be preserved in the event that the DTC was confirmed at one point in the past, but then satisfied aging criteria prior to the time of the request from the client (assuming no other DTCs is confirmed after the aforementioned DTC failed).

As mentioned above, first/most recent confirmed DTC information shall be cleared upon a successful ClearDiagnosticInformation request from the client.

12.3.1.14Retrieving a list of "prefailed" DTC status (SubFunction = 14₁₆ reportDTCFaultDetection-Counter)

The client can retrieve a list of all current "prefailed" DTCs which have or have not yet been detected as "pending" or "confirmed" at the time of the client's request. The intention of the DTCFaultDetectionCounter is a simple method to identify a growing or intermittent problem which cannot be identified/read by the statusOfDTC byte of a particular DTC. The internal implementation of the DTCFaultDetectionCounter shall be vehicle manufacturer specific. The use case of "prefailed" DTCs is to speed up failure detection during testing in the manufacturing plants for DTCs that require a maturation time unacceptable to manufacturing testing. Service has a similar use case after repairing or installing new components.

12.3.1.15Retrieving a list of DTCs with "permanent DTC" status (SubFunction = 15₁₆ reportDTCWithPermanentStatus)

The client can retrieve a list of DTCs with "permanent DTC" status as described in 3.12.

The SubFunction 15_{16} will be replaced by SubFunction 55_{16} in the future. In case there is a need for PermanentDTC implementation it is recommended to use the 55_{16} SubFunction.

12.3.1.16Retrieving DTCExtendedData record data for a client defined DTCExtendedData record number (SubFunction = 16₁₆ reportDTCExtDataRecordByRecordNumber)

A client can retrieve DTCExtendedData for a client defined DTCExtendedData record number by sending а request for this service with the SubFunction set to reportDTCExtDataRecordByRecordNumber. The server shall search through all supported DTCs for exact matches with the DTCExtDataRecordNumber specified by the client. In this case, the DTCExtDataRecordNumber parameter provided in the client's request shall specify a particular DTCExtendedData record for all supported DTCs for which DTCExtendedData is being requested.

The server shall return a DTCExtendedData record along with the DTC number and statusOfDTC for each supported DTC that contains data for the requested DTCExtDataRecordNumber.

The vehicle manufacturer shall define the format and content of the DTCExtDataRecord. The structure of the data reported in the DTCExtDataRecord is defined by the DTCExtDataRecordNumber in a similar way to the definition of data within a record DataIdentifier.

The server shall negatively respond if the DTCExtDataRecordNumber parameter specified by the client is invalid or not supported by the server.

Clearance of DTCExtendedData information upon the reception of a ClearDiagnosticInformation service is specified in 11.2.1. It is in the responsibility of the vehicle manufacturer to specify the rules for the deletion of stored DTCs and DTC extended data in case of a memory overflow (memory space for stored DTCs and DTC extended data completely occupied in the server).

12.3.1.17 Retrieving the list of DTCs out of the server's user defined DTC memory that match a client defined DTC status mask (SubFunction = 17_{16} reportUserDefMemoryDTCByStatusMask)

The client can retrieve a list of DTCs from a user defined memory, which satisfy a client defined status mask by sending a request with the SubFunction byte set to reportUserDefMemoryDTCByStatusMask. This SubFunction allows the client to request the server to report all DTCs that are "testFailed" or "confirmed" or "etc." from the user defined memory.

The evaluation shall be done as follows: the server shall perform a bit-wise logical AND-ing operation between the mask specified in the client's request and the actual status associated with each DTC supported by the server in that user defined memory. In addition to the DTCStatusAvailabilityMask, the server shall return all DTCs for which the result of the AND-ing operation is non-zero (i.e. (statusOfDTC & DTCStatusMask) != 0) in that specific memory. 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. If no DTCs within the server match the masking criteria specified in the client's request in that specific memory, no DTC or status information shall be provided following the DTCStatusAvailabilityMask byte in the positive response message.

DTC status information shall be cleared either by a service 14_{16} ClearDTC service with the memorySelection parameter set to the applicable memory or by manufacturer specific conditions (e.g. a routine control request from the client).

12.3.1.18Retrieving user defined memory DTCSnapshot record data for a client defined DTC mask and a client defined DTCSnapshotNumber out of the DTC user defined memory (SubFunction = 18₁₆ reportUserDefMemoryDTCSnapshotRecordByDTCNumber)

A client can retrieve captured DTCSnapshot record data for a client defined DTCMaskRecord in conjunction with a DTCSnapshot record number and a user defined memory identifier by sending a SubFunction request for this service with the set to reportUserDefMemoryDTCSnapshotRecordByDTCNumber. The server shall search through its supported DTCs for an exact match with the DTCMaskRecord specified by the client [containing the DTC number (high, middle, and low byte)]. The DTCSnapshotRecordNumber parameter provided in the client's request shall specify a particular occurrence of the specified DTC and the defined memory for which DTCSnapshot record data is being requested.

NOTE 1 The DTCSnapshotRecordNumber does not share the same address space the as DTCStoredDataRecordNumber.

It shall be the responsibility of the system supplier/vehicle manufacturer to define whether DTCSnapshot records captured within such servers store data associated with the first or most recent occurrence of a failure.

Along with the DTC number and statusOfDTC, the server shall return a single predefined UserDefDTCSnapshotRecordNumber from the specific user memory in response to the client's request, if а failure has been identified for the client defined DTCMaskRecord and UserDefDTCSnapshotRecordNumber parameters (UserDefDTCSnapshotRecordNumber unequal FF_{16}) and that specific memory.

NOTE 2 The exact failure criteria is defined by the system supplier/vehicle manufacturer.

The DTCSnapshot record may contain multiple data-parameters that can be used to reconstruct the vehicle conditions (e.g. B+, RPM, time-stamp) at the time of the failure occurrence.

The vehicle manufacturer shall define format and content of the DTCSnapshotRecord in the user defined memory (i.e. the content of the DTCSnapshotRecords can differ between different memories) records. The data reported in the DTCSnapshotRecord first of all contains a dataIdentifier to identify the d

data that follows. DTCSnapshotRecord. The usage of one or multiple dataIdentifiers in the DTCSnapshotRecord in the user defined memory allows for the storage of different types of DTCSnapshotRecords for a single DTC for different occurrences of the failure. A parameter which indicates the number of record DataIdentifiers contained within each DTCSnapshotRecord shall be provided with each DTCSnapshotRecord to assist data retrieval.

The server shall report one DTCSnapshot record in a single response message, except if the client has set the UserDefDTCSnapshotRecordNumber to FF_{16} , because this shall cause the server to respond with all DTCSnapshot records stored for the client defined DTCMaskRecord and the user defined memory in a single response message. The DTCAndStatusRecord is only included one time in the response message.

The server shall negatively respond if the DTCMaskRecord, UserDefDTCSnapshotRecordNumber, UserDefMemory parameters specified by the client are invalid or not supported by the server. This is to differentiated from the which the DTCMaskRecord be case in and/or UserDefDTCSnapshotRecordNumber parameters specified by the client are indeed valid and supported by the server for that specific memory, but have no DTCSnapshot data associated with it (e.g. because a failure event never occurred for the specified DTC or record number). The server shall send the positive response containing only the DTCAndStatusRecord [echo of the requested DTC number (high, middle, and low byte) plus the statusOfDTC].

DTCSnapshot information shall be cleared upon a manufacturer specific conditions (e.g a routine control) request from the client. It is in the responsibility of the vehicle manufacturer to specify the rules for the deletion of stored DTCs and DTCSnapshot data in case of a memory overflow (memory space for stored DTCs and DTCsnapshot data completely occupied in the server for that specific memory).

12.3.1.19Retrieving user defined memory DTCExtendedData record data for a client defined DTC mask and a client defined DTCExtendedData record number out of the DTC memory (SubFunction = 19₁₆ reportUserDefMemoryDTCExtDataRecordByDTCNumber)

A client can retrieve DTCExtendedData for a client defined DTCMaskRecord in conjunction with a DTCExtendedData record number and a UserDefMemoryIdenitfier by sending a request for this service with the SubFunction set to reportUserDefMemoryDTCExtDataRecordByDTCNumber. The server shall search through its supported DTCs for an exact match with the DTCMaskRecord specified by the client [containing the DTC number (high, middle, and low byte)] and the UserDefMemoryIdentifier. In this case the DTCExtDataRecordNumber parameter provided in the client's request shall specify a particular DTCExtendedData record of the specified DTC for which DTCExtendedData is being requested.

Along with the DTC number and statusOfDTC, the server shall return a single predefined DTCExtendedData record in response to the client's request (DTCExtDataRecordNumber unequal to FE_{16} or FF_{16}).

The vehicle manufacturer shall define format and content of the UserDefDTCExtDataRecord. The structure of the data reported in the DTCExtDataRecord is defined by the DTCExtDataRecordNumber for that specific user defined memory in a similar way to the definition of data within a record DataIdentifier. Multiple DTCExtDataRecordNumbers and associated DTCExtDataRecords may be included in the response. The usage of one or multiple DTCExtDataRecordNumbers allows for the storage of different types of DTCExtDataRecords for a single DTC.

The server shall report one DTCExtendedData record in a single response message, except the client has set the DTCExtDataRecordNumber to FE_{16} or FF_{16} , because this shall cause the server to response with all DTCExtendedData records stored for the client defined DTCMaskRecord out of the user defined memory in a single response message.

The server shall negatively respond if the DTCMaskRecord or DTCExtDataRecordNumber parameters specified by the client are invalid or not supported by the server or not in that specific memory. This shall be differen **ExtDataRecordNumber**

parameters specified by the client are indeed valid and supported by the server, but have no DTC extended data associated with it (e.g. because of memory overflow of the extended data). In case of reportDTCExtDataRecordByDTCNumber the server shall send the positive response containing only the DTCAndStatusRecord [echo of the requested DTC number (high, middle, and low byte) plus the statusOfDTC].

DTCExtendedDataRecord information shall be cleared upon a manufacturer specific conditions by either a service 14_{16} ClearDiagnosticInformation with the memorySelection parameter set to the applicable memory or by a routine control request from the client.

12.3.1.20Retrieving the list of all DTCs that supports an specific DTCExtendedDataRecord (SubFunction = 1A₁₆ reportSupportedDTCExtDataRecord)

A client can retrieve the list of all DTCs that supports a specific DTCExtendedDataRecord by sending a request for this service with the SubFunction set to reportDTCExtendedDatatIdentification. The server shall search through its supported DTCs. If а DTC support the requested DTCExtendedDataRecordNumber it should be included in the response. Along with the DTC number(s) and the statusOfDTC, the server shall return the DTCExtendedDataRecordNumber for that DTC(s). The server shall negatively respond if the DTCExtendedDataRecord parameter specified by the client is invalid or not supported by the server.

12.3.1.21Retrieving the list of VOBD DTCs from a functional group that match a client defined status mask (SubFunction = 42₁₆ reportWWHOBDDTCByMaskRecord)

The implementation and usage of DTCS everityMask (with severity and class) is defined in ISO 27145- $3^{[22]}$.

12.3.1.22Retrieving a list of VOBD DTCs with "permanent DTC" status (SubFunction = 55₁₆ reportWWHOBDDTCWithPermanentStatus)

The client can retrieve a list of VOBD DTCs with the "permanent DTC" status as described in 3.12.

12.3.1.23Retrieve DTC information for a client defined DTCReadinessGroupIdentifier (SubFunction = 56₁₆ reportDTCInformationByDTCReadinessGroupIdentifier)

A client can retrieve DTC information for a client defined DTC Readiness Group Identifier by sending a request of this service with the SubFunction set to reportDTCInformationByDTCReadinessGroupIdentifer. The server shall search through its supported DTCs for an exact match with the DTCReadinessGroupIdentifier specified by the client. Along with the DTC number(s) and the statusOfDTC, the server shall return the DTCReadinessGroupIdentifier for those DTCs.

The server shall negatively respond if the DTCReadinessGroupIdentifier parameter specified by the client is invalid or not supported by the server.

12.3.2 Request message

12.3.2.1 Request message definition

Table 302 specifies the structure of the ReadDTCInformation request message based on the used SubFunction parameter.