Document Title	Requirements on Diagnostic Extract Template
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	681

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R24-11

Document Change History

Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	• Cleaned up upstream traces
2023-11-23	R23-11	AUTOSAR Release Management	• No content changes
2022-11-24	R22-11	AUTOSAR Release Management	• Support for nested data structures • Remove superfluous requirement
2021-11-25	R21-11	AUTOSAR Release Management	• No content changes
2020-11-30	R20-11	AUTOSAR Release Management	• Add requirements for security event reporting
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Editorial changes
2017-12-08	4.3.1	AUTOSAR Release Management	• Editorial changes

△

2016-11-30	4.3.0	AUTOSAR Release Management	• Add requirements for OBD • Add requirements for the support of Fim • Add requirements for the support of J1939 • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2015-07-31	4.2.2	AUTOSAR Release Management	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	• Initial Release.

Disclaimer

Contents

References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] ISO 14229-1 – Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2006-12)
https://www.iso.org

[3] Specification of Diagnostic Communication Manager
AUTOSAR_CP_SWS_DiagnosticCommunicationManager

[4] Specification of Diagnostic Event Manager
AUTOSAR_CP_SWS_DiagnosticEventManager

[5] Motor Vehicle Pollution Control Devices
https://www.iso.org

[6] Specification of Function Inhibition Manager
AUTOSAR_CP_SWS_FunctionInhibitionManager

[7] ISO 15031-5 – Road vehicles – Communication between vehicle and external equipment for emission-related diagnostic – Part 5: Emission-related diagnostic services.
https://www.iso.org

1 Introduction

1.1 Scope of this document

This document collects the requirements on the Diagnostic Extract.

The main goal of the Diagnostic Extract is to exchange diagnostic data between the different parties involved in the diagnostic development process to support the automatic code generation process for diagnostic modules DCM and DEM.

Further, the Diagnostic Extract is used to support the distributed development process for diagnostic functionality.

Below, the key aspects for the usage of the Diagnostic Extract are mentioned again:

- Exchange of diagnostic data for DCM and DEM
- Support of distributed development for diagnostic functionality

1.2 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([1]).

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([1]).

1.3 Guidelines

Existing specifications shall be referenced (in form of a single requirement). Differences to these specifications are specified as additional requirements. All Requirements shall have the following properties:

- Redundancy
  Requirements shall not be repeated within one requirement or in other requirements.
- Clearness
  All requirements shall allow one possibility of interpretation only. Used technical terms that are not in the glossary must be defined.
- Atomicity
  Each Requirement shall only contain one requirement. A Requirement is atomic if it cannot be split up in further requirements.
- Testability
  Requirements shall be testable by analysis, review or test.
- Traceability
  The source and status of a requirement shall be visible at all times.

1.4 Requirements Tracing

Currently no requirements tracing is provided for this document. Requirement tracing will be included in later revision.

2 Requirements

2.1 Relation to AUTOSAR Features

The section describes a list of features that should be addressed by the requirements:

- [RS_Main_00300] AUTOSAR shall provide data exchange formats to support work-share in large inter and intra-company development groups

This is a short selection of features and main requirements which need to be fulfilled by the requirements on the Diagnostic Extract.

2.2 General Requirements

This chapter contains a collection of general requirements that apply for all aspects of the Diagnostic Extract.

[RS_DEXT_00001] Diagnostic data exchange ⌈

Description:	The diagnostic extract shall support the exchange of diagnostics-related information.
Rationale:	The configuration of the AUTOSAR diagnostics stack, in the vast majority of cases, is a joint effort with contributions from the OEM and the respective ECU supplier. For this purpose, the OEM and the supplier need to be able to exchange configuration information with as little friction as possible. In general, arbitrary combinations of particular OEMs and specific suppliers are possible. To make this work it is necessary to standardize the data exchanged between the affected parties to the necessary extent.
Use Case:	1. OEM delivers a partial configuration to a supplier that in turn is expected to fill in the missing pieces and/or review and (where applicable) overwrite configuration done by the OEM. 2. Supplier feeds back reviewed diagnostics configuration to the OEM. 3. Supplier delivers the final configuration of the diagnostics stack back to the OEM such that the latter is able to derive a tester configuration from this data. 4. Suppliers or OEMs use the Diagnostic Extract to exchange the information within the company between related responsible organization parts. By this means a distributed software development is supported within the company.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00044] Derivation of related ECU-C parameter ⌈

Description:	The Diagnostic Extract shall support the derivation of related ECU-C parameters for Dcm, Fim and Dem.
Rationale:	A concrete configuration of an AUTOSAR diagnostic stack on the level of ECUC is, by definition, not portable. It is very much focused on specific details that reflect the specific implementation of the particular diagnostic stack. This approach is reasonable because it allows for a deeper optimization of the software stack than a general and more abstract approach would support. However, the more abstract approach has other benefits in terms of being exchangeable across different organizations and, to some extent, even projects. Therefore, it is reasonable to define a general and more abstract way of configuring the diagnostic stack for which the ultimate goal is to derive the concrete and non-portable specific configuration to a large extent.
Use Case:	A user wants the specify configuration of the diagnostic stack in a general way that can be exchanged with project partners in the same way that e.g. a system description can be exchanged. The user wants to specify the diagnostic information on the same conceptual level as the system extract and the user wants to relate from the diagnostic configuration to elements of a system extract where applicable. Finally, someone wants to take this information that is described on a higher conceptual level and use if for the derivation of configuration information on the much more specific (and, consequently, less portable) level that is ECUC.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00048] Diagnostic Properties that are specific for one ECU ⌈

Description:	The Diagnostic Extract shall support the definition of diagnostic properties that are specific for a given ECU.
Rationale:	Some properties differ from ECU to ECU. In case the diagnostic extract covers multiple ECUs at the same time it is necessary to express their diagnostic properties individually.
Use Case:	The user wants to specify a diagnostic extract consisting of several ECUs and the user wants to define certain properties individually for each of the included ECUs.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00058] Indicate that an ECU supports ODB ⌈

Description:	The diagnostic extract shall allow for the definition of whether (and how) a given ECU supports OBD
Rationale:	There are certain switches to be set in the downstream configuration that depend on the information about the OBD capabilities of a given ECU.
Use Case:	The user wants to specify the applicability of OBD for a given ECU
Dependencies:	–
Supporting Material:	

⌋

[RS_DEXT_00059] Support for different protocols ⌈

Description:	The diagnostic extract shall support the definition of different diagnostic protocols (e.g. UDS, OBD, etc.) and their priority in relation to each other.
Rationale:	Different protocols shall be handled with different priorities. This requires a formal definition of a diagnostic protocol and its relation to further model elements that are already part of the diagnostic extract.
Use Case:	The user wants to define a diagnostic extract that supports different diagnostic protocols.
Dependencies:	–
Supporting Material:	

⌋

2.3 Requirements against the Support for UDS Diagnostic Services

This chapter contains a collection of requirements against the context of UDS diagnostic services according to [2].

[RS_DEXT_00003] SessionControl ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x10 (SessionControl).
Rationale:	The usage of different diagnostic sessions is very common and therefore needs to be supported by the Diagnostic Extract Template.
Use Case:	Support the switching from one diagnostic session to another.
Dependencies:	–

▽

△

Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00004] ECUReset ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x11 (ECUReset).
Rationale:	The ability to reset the server is crucial for conducing a diagnostic session.
Use Case:	The user wants to reset the connected server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00005] ClearDiagnosticInformation ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x14 (ClearDiagnosticInformation).
Rationale:	The service allows for clearing the diagnostic memory of a server. This is a frequently used functionality.
Use Case:	The user wants to clear diagnostic memory on the connected server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00006] ReadDTCInformation ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x19 (ReadDTCInformation).
Rationale:	The service allows for accessing the status of a diagnostic trouble code on the server. This is a frequently used functionality.
Use Case:	The user wants to access DTC information via a tester on the server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00007] ReadDataByIdentifier ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x22 (ReadDataByIdentifier).
Rationale:	The service allows for reading values on the server according to the definition of a given data identifier. This is a frequently used functionality.
Use Case:	The user wants to read the values associated with a given data identifier from the server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00008] ReadMemoryByAddress ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x23 (ReadMemoryByAddress).
Rationale:	The service allows for accessing the content of a piece of memory on the server. This is a frequently used functionality.
Use Case:	The user wants to read memory content from the diagnostic server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00009] SecurityAccess ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x27 (SecurityAccess).
Rationale:	This service allows for data and diagnostic services for which specific security restrictions apply.
Use Case:	The application of security restrictions limits the access to data and diagnostic services to authorized personnel. The restriction may be applied for safety and/or security reasons.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00010] CommunicationControl ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x28 (CommunicationControl).
Rationale:	This service allows for switching on and off the communication of certain messages (e.g. application-related communication).
Use Case:	The user wants to switch off normal communication messages.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00011] ReadDataByPeriodicIdentifier ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x2A (ReadDataByPeriodicIdentifier).
Rationale:	The service allows for requesting the periodic transmission of diagnostic data by the server according to the definition of a periodic data identifier.
Use Case:	The user wants to get access to diagnostic data that is transmitted periodically without the necessity to request each transmission individually.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00012] DynamicallyDefineDataIdentifier ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x2C (DynamicallyDefineDataIdentifier).
Rationale:	The service allows for the ad-hoc definition of a data identifier that can then be accessed by the respective diagnostic services.
Use Case:	In contrast to the case where data identifiers are defined in advance of a diagnostic session, this service allows for defining data identifiers while a diagnostic session is ongoing.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00013] WriteDataByIdentifier ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x2E (WriteDataByIdentifier).
Rationale:	The service allows for transmitting diagnostic data identified by the association with a diagnostic data identifier to a diagnostic server. This is a frequently used functionality.
Use Case:	The user wants to transmit data associated with a given data identifier to the diagnostic server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00014] IOControl ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x2F (IOControl).
Rationale:	The service allows for substituting values of the I/O layer with values provided by the diagnostic tester.
Use Case:	The user wants to bypass a sensor and feeds substitution values instead. The user wants to substitute values provided to a given actuator.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00015] RoutineControl ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x31 (RoutineControl).
Rationale:	The service can be used to execute specific code on the server according.
Use Case:	The user wants to execute code on the remote server in order to achieve a given functionality that goes beyond the capabilities provided by any of the “simple” data exchange services.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00016] RequestDownload ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x34 (RequestDownload).
Rationale:	This service has the ability to request the server to accept the transfer of a piece of data from the client (e.g. tester) to the server. Support for this service is the prerequisite for a support of the service described in [RS_DEXT_00018].
Use Case:	The user wants to transmit a piece of (mostly complex) data from the client to the server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00017] RequestUpload ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x35 (RequestUpload).
Rationale:	This service has the ability to request the server to accept the transfer of a piece of data from the server to a client (e.g. tester). Support for this service is the prerequisite for a support of the service described in [RS_DEXT_00018].
Use Case:	The user wants to transmit a piece of (mostly complex) data from the server to the client.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00018] TransferData ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x36 (TransferData).
Rationale:	This service is used to actually execute a data transfer between client and server.
Use Case:	The user wants to transmit a piece of (mostly complex) data between client and server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00019] RequestTransferExit ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x37 (RequestTransferExit).
Rationale:	The service can be taken to request the termination of a data transmission between server and client (independent of the direction of the data transmission).
Use Case:	The user wants to actively end a data transmission between client and server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00020] WriteMemoryByAddress ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x3D (WriteMemoryByAddress).
Rationale:	The service can be used to write a piece of data to the server's memory.
Use Case:	The user wants to overwrite the values in a given piece of server memory.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00021] ControlDTCSetting ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x85 (ControlDTCSetting).
Rationale:	The service can be used to control the updating of status bits of diagnostic trouble codes in the server.
Use Case:	The user wants to either stop or resume the updating of status bits of diagnostic trouble codes in the server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00022] ResponseOnEvent ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x86 (ResponseOnEvent).
Rationale:	The service can be used to control the behavior of the server with respect to the transmission of data in response to a given event.
Use Case:	The user wants to control the behavior of the server in terms of how the server sends response messages according to the existence of a given event.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00057] RequestFileTransfer ⌈

Description:	The Diagnostic Extract shall support the configuration of UDS service 0x38 (RequestFileTransfer).
Rationale:	The service RequestFileTransfer is part of the subset of UDS services supported by the Diagnostic Extract.
Use Case:	The user wants to specify a file transfer to/from the server.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00047] Custom Diagnostic Service ⌈

Description:	The Diagnostic Extract shall support the definition of custom diagnostic services.
Rationale:	In some cases diagnostic services beyond the set of services standardized in ISO14229 are needed.
Use Case:	The user wants to execute a diagnostic functionality that is not part of the specification of ISO 14229 [2]
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00049] Properties of individual diagnostic services ⌈

Description:	The Diagnostic Extract shall support the definition of properties that are specific for a given diagnostic service.
Rationale:	Some properties of diagnostic services need to be individually fine-tuned for every instance of a given class of diagnostic service, e.g. ReadDataByIdentifier.
Use Case:	The user wants to define specific properties differently for all instances of a given diagnostic service.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00050] Properties of all diagnostic services of a given kind ⌈

Description:	The Diagnostic Extract shall support the definition of properties that are common for all instances of a kind of diagnostic service.
Rationale:	Some properties of diagnostic services are common for all instances of the specific diagnostic service. If these were specifiable on an individual basis there would be potential for inconsistencies in the specification of different instances of the diagnostic service.
Use Case:	The user wants to define specific properties that are shared among all instances of a given diagnostic service.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00051] Subfunctions of Diagnostic Services ⌈

Description:	The Diagnostic Extract shall support the definition of subfunctions of diagnostic services.
Rationale:	The definition of subfunctions is an important part of the definition of diagnostic services. Also, the existence of subfunctions for certain diagnostic services is regulated by the applicable ISO 14229-1 [2].
Use Case:	The user wants to specify subfunctions for given diagnostic services.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00043] Description of data elements ⌈

Description:	The Diagnostic Extract shall support the description of data elements for diagnostic services.
Rationale:	Data elements represent a formal specification of the content of diagnostic messages exchanged between a client and a server. The formal definition of the elements of the messages allows for a clearer idea of what is actually exchanged. Plus, it is possible to check the consistency of data elements defined in the diagnostic extract to data elements in the application software to which the diagnostic service is finally connected to.
Use Case:	The user wants to create a fine-grained model of the content of diagnostic messages exchanged between client and server.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00038] Description of nested elements of DIDs and RIDs ⌈

Description:	The Diagnostic Extract shall support the usage of nested data elements for the DID sender/receiver and RID routine interaction between the Dcm and application software.
Rationale:	Nested data types are a commonly used case in AUTOSAR application software. Consequently, the AUTOSAR diagnostic stack and, by extension, the Diagnostic Extract needs to support this case for the interaction between the Dcm and application software.
Use Case:	The user wants to access a nested data type in a PortPrototype of the application to be used in the description of diagnostic content (e.g. the definition of a Diagnostic Data Identifier)
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Communication Manager [3].

⌋

[RS_DEXT_00039] Diagnostic Service Table ⌈

Description:	The Diagnostic Extract shall support the capability to specify a Diagnostic Service Table.
Rationale:	The Diagnostic Service Table represents the amount of services necessary for the applicable Diagnostic Protocol. The service table is a central aspect of the configuration of the diagnostic stack and shall therefore be supported by the Diagnostic Extract.
Use Case:	The user wants to define a specific Diagnostic protocol in the context of a project. As far as the Diagnostic Extract is concerned, this requires the definition of a Diagnostic Service Table.

▽

△

Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Communication Manager [3].

⌋

2.4 Requirements against Event Handling

This chapter contains a collection of requirements against the general context of diagnostic events and diagnostic trouble codes.

[RS_DEXT_00023] Configuration of events ⌈

Description:	The Diagnostic Extract shall support the configuration of diagnostic events.
Rationale:	The definition of diagnostic events is key for the configuration of the AUTOSAR Dem. Diagnostic events are rich with additional information that needs to be provided as part of the definition of the diagnostic event. Furthermore, diagnostic events have relations to other entities that also need to be expressed as part of the diagnostic extract.
Use Case:	The user wants to formally define the result of a diagnostic monitor. This is a typical task of (but not restricted to) the supplier's workflow.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00024] Configuration of DTCs ⌈

Description:	The Diagnostic Extract shall support the configuration of DTCs.
Rationale:	The diagnostic stack foresees the existence of entities called diagnostic trouble codes (DTC) that, from a simplified point of view, represent reportable incidents towards a diagnostic tester. In reality, a DTC has more complex features that need to be supported by the diagnostic extract as well.
Use Case:	The user wants too define a unique identifier that represents a given fault situation, which in turn can be reported to a diagnostic tester.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00026] Enable Conditions ⌈

Description:	The Diagnostic Extract shall support the configuration of Enable Conditions.
Rationale:	AUTOSAR foresees the existence of conditions that govern the processing of a diagnostic event. It shall be possible to define conditions that control how specific diagnostic events are enabled (i.e. will be processed) or disabled (i.e. processing will be blocked).
Use Case:	The user wants to define a diagnostic enable condition that impacts the processing of diagnostic events.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00027] Storage Conditions ⌈

Description:	The Diagnostic Extract shall support the configuration of Storage Conditions.
Rationale:	AUTOSAR foresees the existence of conditions that control whether or not a diagnostic event is stored in event memory. These conditions are part of the configuration of the diagnostic stack. It is therefore necessary to also be able to define storage conditions in the diagnostic extract.
Use Case:	The user wants to define storage conditions for a given event. These storage conditions shall later be taken to derive the configuration of the AUTOSAR diagnostic stack.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00028] Enable Condition Groups ⌈

Description:	The Diagnostic Extract shall support the collection of Enable Conditions by means of Enable Condition Groups.
Rationale:	The definition of an Enable Condition Group facilitates the handling of Enable Conditions in general. This applies in particular for Diagnostic Events that depend on a shared collection of enable conditions. It is therefore required that the Diagnostic Extract supports the definition of Enable Condition Groups that can later be taken to contribute to the configuration of the AUTOSAR diagnostic stack.
Use Case:	The user wants to group several Diagnostic Enable Conditions to a single Enable Condition Group that can then be used to express the relation of given Diagnostic Events to the definition of Enable Conditions in a single step.
Dependencies:	–

▽

△

Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00029] Storage Condition Groups ⌈

Description:	he Diagnostic Extract shall support the collection of Storage Conditions by means of Storage Condition Groups.
Rationale:	The definition of an Storage Condition Group facilitates the handling of Storage Conditions in general. This applies in particular for Diagnostic Events that depend on a shared collection of Storage conditions. It is therefore required that the Diagnostic Extract supports the definition of Storage Condition Groups that can later be taken to contribute to the configuration of the AUTOSAR diagnostic stack.
Use Case:	The user wants to group several Diagnostic Storage Conditions to a single Storage Condition Group that can then be used to express the relation of given Diagnostic Events to the definition of Storage Conditions in a single step.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00030] Assignment of Enable Condition Groups ⌈

Description:	The Diagnostic Extract shall support the assignment of Enable Condition Groups to an event.
Rationale:	A consequence of the existence of Diagnostic Enable Condition and Enable Condition Groups is that it should be possible to establish a relation of one with the other.
Use Case:	The user wants to collect several Diagnostic Enable Conditions in the context of one Enable Condition Group. This way, the configuration of the relation of Diagnostic Events to Enable Conditions can be significantly facilitated.
Dependencies:	[RS_DEXT_00028]
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00031] Assignment of Storage Condition Group ⌈

Description:	The Diagnostic Extract shall support the assignment of Storage Condition Groups to an event.
Rationale:	A consequence of the existence of Diagnostic Storage Condition and Storage Condition Groups is that it should be possible to establish a relation of one with the other.
Use Case:	The user wants to collect several Diagnostic Storage Conditions in the context of one Storage Condition Group. This way, the configuration of the relation of Diagnostic Events to Storage Conditions can be significantly facilitated.
Dependencies:	[RS_DEXT_00029]
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00032] Configuration of Extended Data Records ⌈

Description:	The Diagnostic Extract shall support the configuration of Extended Data Records.
Rationale:	Extended Data Records, like Freeze Frames, are used to store additional information related to a Diagnostic Event and shall therefore also be supported by the Diagnostic Extract.
Use Case:	The user wants to define additional information to be stored along with a given Diagnostic Event in event memory.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00033] Configuration of Snapshot Records ⌈

Description:	The Diagnostic Extract shall support the configuration of Snapshot Records (Freeze Frames).
Rationale:	Freeze Frames, like Extended Data Records, are used to store additional information related to a Diagnostic Event and shall therefore also be supported by the Diagnostic Extract.
Use Case:	The user wants to define the content of a Freeze frame to be stored along with a given Diagnostic Event in event memory.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00034] Description of Data Identifiers ⌈

Description:	The Diagnostic Extract shall support the description of Data Identifiers for Diagnostic Services.
Rationale:	The definition of Data Identifiers is a central part of the configuration of the AUTOSAR diagnostic stack.
Use Case:	The user wants to define a Diagnostic Data Identifier with a given content.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Communication Manager [3].

⌋

[RS_DEXT_00035] Description of Dynamic Data Identifiers ⌈

Description:	The Diagnostic Extract shall support the description of Dynamic Data Identifiers for Diagnostic Services.
Rationale:	The definition of Dynamic Data Identifiers is a central part of the configuration of the AUTOSAR diagnostic stack.
Use Case:	The user wants to specify that a Diagnostic Data Identifier can be used as a Diagnostic Dynamic Data Identifier.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Communication Manager [3].

⌋

[RS_DEXT_00036] Description of Routine Identifiers ⌈

Description:	The Diagnostic Extract shall support the description of Routine Identifiers for Diagnostic Services.
Rationale:	The definition of Routine Identifiers is a central part of the configuration of the AUTOSAR diagnostic stack.
Use Case:	The user wants to define a Routine Identifier associated with a given Diagnostic Routine. This typically extend to the modeling of the Diagnostic Routine itself.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Communication Manager [3].

⌋

[RS_DEXT_00037] Description of I/O Identifiers ⌈

Description:	The Diagnostic Extract shall support the description of I/O Identifiers for Diagnostic Services.
Rationale:	The definition of I/O Identifiers is a central part of the configuration of the AUTOSAR diagnostic stack.
Use Case:	The user wants to define an I/O Identifier. This typically extends to the modeling of the Diagnostic Data Identifier.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Communication Manager [3].

⌋

[RS_DEXT_00053] Debouncing of diagnostic events ⌈

Description:	The Diagnostic Extract shall support the specification of how Diagnostic Events shall be debounced.
Rationale:	Typically, a debouncing algorithm (as defined by the SWS Dem) is applied to the occurrence of Diagnostic Events in order to avoid the existence of false positives. This aspect shall be supported by the Diagnostic Extract as well.
Use Case:	The user wants to define how debouncing shall be applied to a given Diagnostic Event.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00054] Operation cycles ⌈

Description:	The Diagnostic Extract shall support the specification of operation cycles.
Rationale:	The AUTOSAR diagnostic stack supports the configuration of operation cycles. This aspect shall therefore be supported by the Diagnostic Extract.
Use Case:	The user wants to assign a specific operation cycle to a diagnostic event depending on the project needs.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00055] Aging ⌈

Description:	The Diagnostic Extract shall support the specification of aging.
Rationale:	It is common practice that events are removed from fault memory after a certain succession of “passed” reports for this event. This aspect shall be supported by the Diagnostic Extract.
Use Case:	The user wants to specify the way in which specific Diagnostic Events shall be subject to aging.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00056] Indicator ⌈

Description:	The Diagnostic Extract shall support the specification of indicators.
Rationale:	The diagnostic stack is supposed to support the signaling of certain Diagnostic Events to the driver of a vehicle. The signaling shall be modeled as a diagnostic indicator.
Use Case:	The user wants to specify how Diagnostic Events are indicated to the driver.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00045] Textual descriptions ⌈

Description:	The Diagnostic Extract shall support the capability to specify textual descriptions for Diagnostic Events and DTC attributes.
Rationale:	The ability to attach textual descriptions to the definition of diagnostic model elements allows for a better transport of knowledge about the purpose and the semantics of the respective model elements.
Use Case:	The user wants to provide a proper textual documentation of specific model elements related to the description of a Diagnostic Extract.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00078] Support for In Use Monitor Performance Ratio ⌈

Description:	The diagnostic extract shall support the definition of In Use Monitor Performance Ratio (IUMPR).
Rationale:	The definition of IUMPR is an integral part of the application of OBD.
Use Case:	The user wants to specify the IUMPR with respect to given diagnostic events.
Dependencies:	–
Supporting Material:	More information about this aspect can be found in the respective law publication [5].

⌋

[RS_DEXT_00080] Support for persisting Security Events

Status: DRAFT

⌈

Description:	In some cases, security events need to be persisted such that they can later be queried by a diagnostic tester.
Rationale:	The occurence of security events can be a very important part of a forensical analysis of an intrusion.
Use Case:	The user wants to specify which security events shall be persisted in the Security Event Memory that is part of the Dem
Dependencies:	–
Supporting Material:	

⌋

2.5 Requirements against Sessions and Security

This chapter contains a collection of requirements against the context of diagnostic sessions and security.

[RS_DEXT_00040] Diagnostic Sessions ⌈

Description:	The Diagnostic Extract shall support the capability to specify Diagnostic Sessions.
Rationale:	The configuration of Diagnostic Sessions represents an important angle of the configuration of the AUTOSAR diagnostic stack. Therefore, the Diagnostic Extract needs to support the modeling of this aspect.
Use Case:	The user wants to define Diagnostic Sessions as part of a project.
Dependencies:	–

▽

△

Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00041] Access Permissions ⌈

Description:	The Diagnostic Extract shall support the capability to specify Access Permissions.
Rationale:	The configuration of Access Permissions represents an important angle of the configuration of the AUTOSAR diagnostic stack. Therefore, the Diagnostic Extract needs to support the modeling of this aspect.
Use Case:	The user wants to define specific diagnostic functionality, e.g. DIDs or RIDs that require Access Permissions as part of a project.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00042] Security Levels ⌈

Description:	The Diagnostic Extract shall support the capability to specify Security Levels.
Rationale:	The configuration of Security Levels represents an important angle of the configuration of the AUTOSAR diagnostic stack. Therefore, the Diagnostic Extract needs to support the modeling of this aspect.
Use Case:	The user wants to define Security Levels as part of a project.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [2].

⌋

[RS_DEXT_00079] Support for environment conditions ⌈

Description:	The Diagnostic Extract shall support the formulation of environment conditions that enable the processing of diagnostic functionality.
Rationale:	Some diagnostic functionalities are only safe if the vehicle is in a certain state. Therefore, it should be possible to specify the conditions for executing diagnostic functionalities on the level of the diagnostic extract.
Use Case:	The user wants to execute a diagnostic functionality that requires the vehicle to be stopped (i.e. the condition is that vehicle speed == 0).
Dependencies:	–

▽

△

Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Communication Manager [3].

⌋

2.6 Requirements against the Support for Function Inhibition

This chapter contains a collection of requirements against the support for function inhibition in the diagnostic extract.

[RS_DEXT_00060] Function ⌈

Description:	The Diagnostic Extract shall support the definition of a function for the purpose of function inhibition. The function shall have an identification by which it can be identified further downstream the configuration.
Rationale:	In order to be able to define function inhibition, it is essential to have model elements that represent a function.
Use Case:	The user wants to specify a function that can then be used in the scope of function inhibition.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Function Inhibition Manager [6].

⌋

[RS_DEXT_00061] Relationship between functions and diagnostic events ⌈

Description:	The Diagnostic Extract shall support the relationship between a function and the diagnostic events that are no longer reported if the respective function is inhibited. It shall be possible to express the relationship between a function and the corresponding events on different levels of granularity. This means that it shall be possible to refer to single diagnostic events as well as to an entire collection of diagnostic events.
Rationale:	The disabling of reporting of diagnostic events is a core functionality of the Fim. It shall be possible to describe this aspect in the diagnostic extract
Use Case:	The user wants to specify how a specific function inhibits the reporting of a given collection of diagnostic events. Instead of having to refer to every single intended diagnostic event the user wants to have the ability to shortcut this effort by referring to a group of events formalized by means of the diagnostic extract.
Dependencies:	–

▽

△

Supporting Material:	More information can be found in the specification of the AUTOSAR Function Inhibition Manager [6].

⌋

[RS_DEXT_00062] Pre-configuration of the Fim when the Dem configuration is not yet available ⌈

Description:	At the point in time when the configuration of the Fim in the diagnostic extract is created it may be possible that the configuration of the Dem in the diagnostic extract does not exist yet. Therefore, there are no formalized representation of diagnostic events available that could be taken for the creation of the Fim's pre-configuration. Therefore, the diagnostic extract shall provide the ability to define placeholders for diagnostic events that can be used in the pre-configuration of the Fim and later be replaced by the real diagnostic events when the Dem configuration in the diagnostic extract becomes available.
Rationale:	The concept of a decentralized configuration supports the idea the certain parts of the diagnostic extract shall be defined independently from each other and then later merged to form the configuration of an entire diagnostic stack.
Use Case:	The user wants to model the relation between functions and diagnostic events before the definition of the actual diagnostic events is available.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Function Inhibition Manager [6].

⌋

2.7 Requirements against the Support for diagnostics on J1939

This chapter contains a collection of requirements against the support for diagnostics on J1939 in the diagnostic extract.

[RS_DEXT_00064] Definition of an SPN ⌈

Description:	The diagnostic extract shall support the definition of an SPN as well as the representation of a communication item in the system description along with its physical properties.
Rationale:	The so-called *Suspect Parameter Number* (SPN) is a prominent concept inside J1939 diagnostics. The SPN can also refer to physical properties.

▽

△

Use Case:	The user wants to specify a J1939 SPN by means of the diagnostic extract. The user wants to specify how the SPN is represented in the system description. The user wants to make sure that the intended physical properties are available in the system description.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00065] Definition of freeze frames on J1939 ⌈

Description:	The diagnostic extract shall support the definition of freeze frames (both regular and expanded) on J1939
Rationale:	The definition of freeze frames on J1939 is an important part of the configuration of a J1939 diagnostics stack.
Use Case:	The user wants to define the content of J1939 freeze frames.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00067] Definition of J1939 DTC ⌈

Description:	The diagnostic extract shall support the definition of a DTC in the domain of J1939 with all its specific properties.
Rationale:	The definition of a DTC is an important part of a diagnostic stack. The DTCs in the J1939 domain have specific properties that are clearly separate from the properties of e.g. UDS DTCs.
Use Case:	The user wants to specify a DTC within the configuration of the J1939 diagnostic stack in the diagnostic extract.
Dependencies:	–
Supporting Material:	–

⌋

2.8 Requirements against the Support for OBD Diagnostic Services

This chapter contains a collection of requirements against the context of OBD diagnostic services according to [7].

[RS_DEXT_00068] Definition of a Diagnostic Parameter Identifier ⌈

Description:	The diagnostic extract shall support the formal definition of diagnostic parameter identifiers (PID). The formal representation of the PID shall have a numerical identifier that shall be used for downstream configuration.
Rationale:	A PID plays a central role in the modeling of OBD services. The definition of PIDs is key for the support of OBD services.
Use Case:	The user wants to define a PID in order to model OBD services.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00069] Support for OBD Mode 0x01 (RequestCurrentPowertrainDiagnosticData) ⌈

Description:	The diagnostic extract shall support the modeling of the OBD mode 0x01, a.k.a. RequestCurrentPowertrainDiagnosticData.
Rationale:	The OBD mode 0x01 is a mandatory part of the OBD functionality and therefore a representation of this mode is required in the diagnostic extract.
Use Case:	The user wants to model the OBD service 0x01 in the diagnostic extract.
Dependencies:	[RS_DEXT_00068]
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00070] Support for OBD Mode 0x02 (RequestPowertrainFreezeFrameData) ⌈

Description:	The diagnostic extract shall support the modeling of the OBD mode 0x02, a.k.a. RequestPowertrainFreezeFrameData.
Rationale:	The OBD mode 0x02 is a mandatory part of the OBD functionality and therefore a representation of this mode is required in the diagnostic extract.
Use Case:	The user wants to model the OBD service 0x02 in the diagnostic extract.
Dependencies:	[RS_DEXT_00068]
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00071] Support for OBD ModeModes 0x03 / 0x07 / 0x0A (RequestEmissionRelatedDiagnosticTroubleCodes) ⌈

Description:	The diagnostic extract shall support the modeling of the OBD mode 0x03 / 0x07 / 0x0A, a.k.a. RequestEmissionRelatedDiagnosticTroubleCodes.
Rationale:	The OBD modes 0x03 / 0x07 / 0x0A are mandatory parts of the OBD functionality and therefore a representation of this mode is required in the diagnostic extract.
Use Case:	The user wants to model the OBD services 0x03 / 0x07 / 0x0A in the diagnostic extract.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00072] Support for OBD Mode 0x04 (ClearResetEmissionRelatedDiagnosticInformation) ⌈

Description:	The diagnostic extract shall support the modeling of the OBD mode 0x04, a.k.a. ClearResetEmissionRelatedDiagnosticInformation.
Rationale:	The OBD mode 0x04 is a mandatory part of the OBD functionality and therefore a representation of this mode is required in the diagnostic extract.
Use Case:	The user wants to model the OBD service 0x04 in the diagnostic extract.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00073] Support for OBD Mode 0x06 (RequestOnBoardMonitoringTestResults) ⌈

Description:	The diagnostic extract shall support the modeling of the OBD mode 0x06, a.k.a. RequestOnBoardMonitoringTestResults.
Rationale:	The OBD mode 0x06 is a mandatory part of the OBD functionality and therefore a representation of this mode is required in the diagnostic extract.
Use Case:	The user wants to model the OBD service 0x06 in the diagnostic extract.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00074] Support for OBD Mode 0x08 (RequestControlOfOnBoardDevice) ⌈

Description:	The diagnostic extract shall support the modeling of the OBD mode 0x08, a.k.a. RequestControlOfOnBoardDevice.
Rationale:	The OBD mode 0x08 is a mandatory part of the OBD functionality and therefore a representation of this mode is required in the diagnostic extract.
Use Case:	The user wants to model the OBD service 0x08 in the diagnostic extract.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00075] Support for OBD Mode 0x09 (RequestVehicleInformation) ⌈

Description:	The diagnostic extract shall support the modeling of the OBD mode 0x09, a.k.a. RequestVehicleInformation.
Rationale:	The OBD mode 0x09 is a mandatory part of the OBD functionality and therefore a representation of this mode is required in the diagnostic extract.
Use Case:	The user wants to model the OBD service 0x09 in the diagnostic extract.
Dependencies:	[RS_DEXT_00076]
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00076] Definition of Diagnostic Test Identifier ⌈

Description:	The diagnostic extract shall support the definition of a Diagnostic Test Identifier (TID)
Rationale:	The definition of a TID is required for the definition of the OBD model 0x09.
Use Case:	The user wants to define a TID that later shall be used in the definition of the OBD service 0x09.
Dependencies:	–
Supporting Material:	More information about this diagnostic service can be found in the respective ISO specification [7].

⌋

[RS_DEXT_00077] Description of the utilization of UDS for supporting WWH-OBD ⌈

Description:	The diagnostic extract shall specify how the UDS diagnostic services shall be utilized in order to implement WWH-OBD.
Rationale:	The implementation of WWH-OBD basically means to use UDS services to emulate the behavior of OBD services. In order to harmonize the usage of UDS services for this purpose the diagnostic extract shall provide a specification that eliminates ambiguous configuration variants as good as possible.
Use Case:	The user wants to utilize WWH-OBD on an ECU that in general supports the UDS diagnostic services. The user wants to take advantage of specifications that clarify how UDS services shall be configured in the diagnostic extract for this purpose.
Dependencies:	–
Supporting Material:	–

⌋

2.9 Requirements against the Mapping of Diagnostic Resources

This chapter contains a collection of requirements against the mapping of diagnostic resources onto other model elements.

[RS_DEXT_00052] Mapping of diagnostic services to the PortPrototypes of ApplicationSwComponentTypes ⌈

Description:	The Diagnostic Extract shall support the specification of how diagnostic services are mapped to the PortPrototypes of ApplicationSwComponentTypes
Rationale:	The mapping of diagnostic services to the the PortPrototypes of ApplicationSwComponentTypes connects the definition of diagnostic services with the actual application software that these services are about. Therefore, this mapping is an important step in the workflow and also provides essential information for the integration of the software on a given ECU.
Use Case:	The user wants to map diagnostic services to application software in order to express the conceptual connection between these two aspects of the ECU configuration.
Dependencies:	–
Supporting Material:	–

⌋

[RS_DEXT_00025] Combined Events ⌈

Description:	The Diagnostic Extract shall support the configuration of events mapping to a single DTC.
Rationale:	There are cases where the results of several diagnostic monitors contribute to a single diagnostic trouble code. For this purpose the diagnostic extract needs to be able to allow for an n:1 relationship between the model elements for diagnostic events and diagnostic trouble codes.
Use Case:	The user wants to wire the results of several diagnostic monitors to a single diagnostic trouble code.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Diagnostic Event Manager [4].

⌋

[RS_DEXT_00063] Relation between functions on Fim level and software-components ⌈

Description:	The diagnostic extract shall provide means to define the mapping of a function to a software-component.
Rationale:	The concept of a function with in the Fim is quite abstract and needs further clarification as to which part of the application software on an AUTOSAR ECU actually corresponds to it.
Use Case:	The user wants to specify which function in terms of the Fim is represented by which part of the application software.
Dependencies:	–
Supporting Material:	More information can be found in the specification of the AUTOSAR Function Inhibition Manager [6].

⌋

[RS_DEXT_00066] Mapping between a J1939 controller application and a software-component ⌈

Description:	The diagnostic extract shall support the specification of the mapping between the concept of a J1939 *controller application* and a part of the AUTOSAR application software represented by a single software-component.
Rationale:	The mapping between *controller application*s and software-component is an important part of “bridging the gap” between the technical domains of AUTOSAR and J1939. The different approaches to the definition of a functional unit shall be mapped onto each other.
Use Case:	The user wants to specify the mapping between a *controller application* and a part of the AUTOSAR application software.
Dependencies:	–

▽

△

Supporting Material:	–

⌋

2.10 Miscellaneous Requirements

This chapter contains a collection of miscellaneous requirements.

[RS_DEXT_00081] Support for updating the Reporting Mode of Security Events

Status: DRAFT

⌈

Description:	In some cases, it is necessary to update the reporting mode of certain security events via diagnostic services.
Rationale:	The reporting mode of security events can be a very important part of a forensical analysis of an intrusion.
Use Case:	The user wants to update the reporting mode of a given security event.
Dependencies:	–
Supporting Material:	

⌋

A Change history of AUTOSAR traceable items

A.1 Traceable item history of this document according to AUTOSAR Release R4.2.1

A.1.1 Added Requirements in R4.2.1

Id	Heading
[RS_DEXT_00001]	Diagnostic Data Exchange
[RS_DEXT_00002]	Distributed software development process
[RS_DEXT_00003]	SessionControl
[RS_DEXT_00004]	ECUReset
[RS_DEXT_00005]	ClearDiagnosticInformation
[RS_DEXT_00006]	ReadDTCInformation
[RS_DEXT_00007]	ReadDataByIdentifier
[RS_DEXT_00008]	ReadMemoryByAddress
[RS_DEXT_00009]	SecurityAccess
[RS_DEXT_00010]	CommunicationControl
[RS_DEXT_00011]	ReadDataByPeriodicIdentifier
[RS_DEXT_00012]	DynamicallyDefineDataIdentifier
[RS_DEXT_00013]	WriteDataByIdentifier
[RS_DEXT_00014]	IOControl
[RS_DEXT_00015]	RoutineControl
[RS_DEXT_00016]	RequestDownload
[RS_DEXT_00017]	RequestUpload
[RS_DEXT_00018]	TransferData
[RS_DEXT_00019]	RequestTransferExit
[RS_DEXT_00020]	WriteMemoryByAddress
[RS_DEXT_00021]	ControlDTCSetting
[RS_DEXT_00022]	ResponseOnEvent
[RS_DEXT_00023]	Configuration of events
[RS_DEXT_00024]	Configuration of DTCs
[RS_DEXT_00025]	Combined Events
[RS_DEXT_00026]	Enable Conditions
[RS_DEXT_00027]	Storage Conditions
[RS_DEXT_00028]	Enable Condition Groups
[RS_DEXT_00029]	Storage Condition Groups
[RS_DEXT_00030]	Assignment of Enable Condition Groups
[RS_DEXT_00031]	Assignment of Storage Condition Group
[RS_DEXT_00032]	Configuration of Extended Data Records
[RS_DEXT_00033]	Configuration of Snapshot Records
[RS_DEXT_00034]	Description of Data Identifiers
[RS_DEXT_00035]	Description of Dynamic Data Identifiers
[RS_DEXT_00036]	Description of Routine Identifiers
[RS_DEXT_00037]	Description of I/O Identifiers
[RS_DEXT_00038]	Description of array data types
[RS_DEXT_00039]	Diagnostic Service Table
[RS_DEXT_00040]	Diagnostic Sessions
[RS_DEXT_00041]	Access Permissions
[RS_DEXT_00042]	Security Levels
[RS_DEXT_00043]	Description of data elements

[RS_DEXT_00044]	Derivation of related ECU-C parameter
[RS_DEXT_00045]	Textual descriptions
[RS_DEXT_00046]	Variants
[RS_DEXT_00047]	Custom Diagnostic Service
[RS_DEXT_00048]	Diagnostic Properties that are specific for one ECU
[RS_DEXT_00049]	Properties of individual diagnostic services
[RS_DEXT_00050]	Properties of all diagnostic services of a given kind
[RS_DEXT_00051]	Subfunctions of Diagnostic Services
[RS_DEXT_00052]	Mapping of diagnostic services to the PortPrototypes of ApplicationSwComponentTypes
[RS_DEXT_00053]	Debouncing of diagnostic events
[RS_DEXT_00054]	Operation cycles
[RS_DEXT_00055]	Aging
[RS_DEXT_00056]	Indicator
[RS_DEXT_00057]	RequestFileTransfer

Table A.1: Added Traceables in R4.2.1

A.1.2 Changed Requirements in 4.2.1

none

A.1.3 Deleted Requirements in 4.2.1

none

A.2 Traceable item history of this document according to AUTOSAR Release R4.2.2

A.2.1 Added Requirements in 4.2.2

none

A.2.2 Changed Requirements in 4.2.2

Id	Heading
[RS_DEXT_00003]	SessionControl
[RS_DEXT_00004]	ECUReset
[RS_DEXT_00005]	ClearDiagnosticInformation
[RS_DEXT_00006]	ReadDTCInformation
[RS_DEXT_00007]	ReadDataByIdentifier
[RS_DEXT_00008]	ReadMemoryByAddress
[RS_DEXT_00009]	SecurityAccess
[RS_DEXT_00010]	CommunicationControl
[RS_DEXT_00011]	ReadDataByPeriodicIdentifier

[RS_DEXT_00012]	DynamicallyDefineDataIdentifier
[RS_DEXT_00013]	WriteDataByIdentifier
[RS_DEXT_00014]	IOControl
[RS_DEXT_00015]	RoutineControl
[RS_DEXT_00016]	RequestDownload
[RS_DEXT_00017]	RequestUpload
[RS_DEXT_00018]	TransferData
[RS_DEXT_00019]	RequestTransferExit
[RS_DEXT_00020]	WriteMemoryByAddress
[RS_DEXT_00021]	ControlDTCSetting
[RS_DEXT_00022]	ResponseOnEvent

Table A.2: Changed Traceables in 4.2.2

A.2.3 Deleted Requirements in 4.2.2

none

A.2.4 Added Constraints in 4.2.2

none

A.2.5 Changed Constraints in 4.2.2

none

A.2.6 Deleted Constraints in 4.2.2

none

A.3 Traceable item history of this document according to AUTOSAR Release R4.3.0

A.3.1 Added Requirements in 4.3.0

Id	Heading
[RS_DEXT_00058]	Indicate that an ECU supports ODB
[RS_DEXT_00059]	Support for different protocols
[RS_DEXT_00060]	Function
[RS_DEXT_00061]	Relationship between functions and diagnostic events
[RS_DEXT_00062]	Pre-configuration of the Fim when the Dem configuration is not yet available
[RS_DEXT_00063]	Relation between functions on Fim level and software-components
[RS_DEXT_00064]	Definition of an SPN

[RS_DEXT_00065]	Definition of freeze frames on J1939
[RS_DEXT_00066]	Mapping between a J1939 controller application and a software-component
[RS_DEXT_00067]	Definition of J1939 DTC
[RS_DEXT_00068]	Definition of a Diagnostic Parameter Identifier
[RS_DEXT_00069]	Support for OBD Mode 0x01 (RequestCurrentPowertrainDiagnosticData)
[RS_DEXT_00070]	Support for OBD Mode 0x02 (RequestPowertrainFreezeFrameData)
[RS_DEXT_00071]	Support for OBD ModeModes 0x03 / 0x07 / 0x0A (RequestEmissionRelated-DiagnosticTroubleCodes)
[RS_DEXT_00072]	Support for OBD Mode 0x04 (ClearResetEmissionRelatedDiagnosticInformation)
[RS_DEXT_00073]	Support for OBD Mode 0x06 (RequestOnBoardMonitoringTestResults)
[RS_DEXT_00074]	Support for OBD Mode 0x08 (RequestControlOfOnBoardDevice)
[RS_DEXT_00075]	Support for OBD Mode 0x09 (RequestVehicleInformation)
[RS_DEXT_00076]	Definition of Diagnostic Test Identifier
[RS_DEXT_00077]	Description of the utilization of UDS for supporting WWH-OBD
[RS_DEXT_00078]	Support for In Use Monitor Performance Ratio
[RS_DEXT_00079]	Support for environment conditions

Table A.3: Added Traceables in 4.3.0

A.3.2 Changed Requirements in 4.3.0

Id	Heading
[RS_DEXT_00001]	Diagnostic data exchange
[RS_DEXT_00023]	Configuration of events
[RS_DEXT_00024]	Configuration of DTCs
[RS_DEXT_00025]	Combined Events
[RS_DEXT_00026]	Enable Conditions
[RS_DEXT_00027]	Storage Conditions
[RS_DEXT_00028]	Enable Condition Groups
[RS_DEXT_00029]	Storage Condition Groups
[RS_DEXT_00030]	Assignment of Enable Condition Groups
[RS_DEXT_00031]	Assignment of Storage Condition Group
[RS_DEXT_00032]	Configuration of Extended Data Records
[RS_DEXT_00033]	Configuration of Snapshot Records
[RS_DEXT_00034]	Description of Data Identifiers
[RS_DEXT_00035]	Description of Dynamic Data Identifiers
[RS_DEXT_00036]	Description of Routine Identifiers
[RS_DEXT_00037]	Description of I/O Identifiers
[RS_DEXT_00038]	Description of array data types
[RS_DEXT_00039]	Diagnostic Service Table
[RS_DEXT_00040]	Diagnostic Sessions
[RS_DEXT_00041]	Access Permissions
[RS_DEXT_00042]	Security Levels
[RS_DEXT_00043]	Description of data elements
[RS_DEXT_00044]	Derivation of related ECU-C parameter
[RS_DEXT_00045]	Textual descriptions
[RS_DEXT_00046]	Variants
[RS_DEXT_00047]	Custom Diagnostic Service
[RS_DEXT_00048]	Diagnostic Properties that are specific for one ECU
[RS_DEXT_00049]	Properties of individual diagnostic services
[RS_DEXT_00050]	Properties of all diagnostic services of a given kind

[RS_DEXT_00051]	Subfunctions of Diagnostic Services
[RS_DEXT_00052]	Mapping of diagnostic services to the PortPrototypes of ApplicationSwComponentTypes
[RS_DEXT_00053]	Debouncing of diagnostic events
[RS_DEXT_00054]	Operation cycles
[RS_DEXT_00055]	Aging
[RS_DEXT_00056]	Indicator
[RS_DEXT_00057]	RequestFileTransfer

Table A.4: Changed Traceables in 4.3.0

A.3.3 Deleted Requirements in 4.3.0

Id	Heading
[RS_DEXT_00002]	Distributed software development process

Table A.5: Deleted Traceables in 4.3.0

A.3.4 Added Constraints in 4.3.0

none

A.3.5 Changed Constraints in 4.3.0

none

A.3.6 Deleted Constraints in 4.3.0

none

A.4 Traceable item history of this document according to AUTOSAR Release R4.3.1

A.4.1 Added Requirements in R4.3.1

none

A.4.2 Changed Requirements in R4.3.1

none

A.4.3 Deleted Requirements in R4.3.1

none

A.5 Traceable item history of this document according to AUTOSAR Release R4.4.0

A.5.1 Added Requirements in R4.4.0

none

A.5.2 Changed Requirements in R4.4.0

none

A.5.3 Deleted Requirements in R4.4.0

none

A.6 Traceable item history of this document according to AUTOSAR Release R19-11

A.6.1 Added Requirements in R19-11

none

A.6.2 Changed Requirements in R19-11

none

A.6.3 Deleted Requirements in R19-11

none

A.7 Traceable item history of this document according to AUTOSAR Release R20-11

A.7.1 Added Requirements in R20-11

Number	Heading
[RS_DEXT_00080]	Support for persisting Security Events
[RS_DEXT_00081]	Support for updating the Reporting Mode of Security Events

Table A.6: Added Requirements in R20-11

A.7.2 Changed Requirements in R20-11

none

A.7.3 Deleted Requirements in R20-11

none

A.8 Traceable item history of this document according to AUTOSAR Release R21-11

A.8.1 Added Requirements in R21-11

none

A.8.2 Changed Requirements in R21-11

none

A.8.3 Deleted Requirements in R21-11

none

A.9 Traceable item history of this document according to AUTOSAR Release R22-11

A.9.1 Added Requirements in R22-11

none

A.9.2 Changed Requirements in R22-11

Number	Heading
[RS_DEXT_00038]	Description of nested elements of DIDs and RIDs

Table A.7: Changed Requirements in R22-11

A.9.3 Deleted Requirements in R22-11

Number	Heading
[RS_DEXT_00046]	Variants

Table A.8: Deleted Requirements in R22-11

A.10 Traceable item history of this document according to AUTOSAR Release R23-11

A.10.1 Added Requirements in R23-11

none

A.10.2 Changed Requirements in R23-11

none

A.10.3 Deleted Requirements in R23-11

none

A.11 Traceable item history of this document according to AUTOSAR Release R24-11

A.11.1 Added Requirements in R24-11

none

A.11.2 Changed Requirements in R24-11

none

A.11.3 Deleted Requirements in R24-11

none