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

Document Status	Final
Part of AUTOSAR Release	4.2.2

Document Change History

Release	Changed by	Description
4.2.2	AUTOSAR Release Management	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
4.2.1	AUTOSAR Release Management	• Initial Release.

Disclaimer

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

Bibliography

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Unified diagnostic services (UDS) - Part 1: Specification and requirements (Release 2006-12)
http://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
- [RS_BRF_01112] AUTOSAR shall offer interfaces to boot loaders.
- [RS_BRF_01440] AUTOSAR services shall support system diagnostic functionality.

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 ⌈

Type:	valid
Description:	The Diagnostic Extract shall support Diagnostic Data Exchange for diagnostic demands.
Rationale:	tbd
Use Case:	Diagnostic data exchange
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00002] Distributed software development process ⌈

Type:	valid
Description:	The Diagnostic Extract shall support a distributed software development process for diagnostic demands.
Rationale:	tbd
Use Case:	Support of diagnostic data exchange based on standardized AUTOSAR data format
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00044] Derivation of related ECU-C parameter ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the derivation of related ECU-C parameter for DCM and DEM.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00046] Variants ⌈

Type:	valid
Description:	The Diagnostic Extract shall support variants.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

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

Type:	valid
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:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

2.3 Requirements against the Support for DiagnosticServices

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

[RS_DEXT_00003] SessionControl ⌈

Type:	valid

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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid

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 ⌈

Type:	valid
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 ⌈

Type:	valid
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 ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of UDS service 0x38 (RequestFileTransfer).
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00047] Custom Diagnostic Service ⌈

Type:	valid
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:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00049] Properties of individual diagnostic services ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the definition of properties that are specific for a given diagnostic service.
Rationale:	
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

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

Type:	valid
Description:	The Diagnostic Extract shall support the definition of properties that are common for all instances of a kind of diagnostic service.
Rationale:	
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00051] Subfunctions of Diagnostic Services ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the definition of subfunctions of diagnostic services.
Rationale:	
Use Case:	tbd
Dependencies:	–

Supporting Material:	–

⌋()

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

Type:	valid
Description:	The Diagnostic Extract shall support the specification of how diagnostic services are mapped to the PortPrototypes of ApplicationSwComponentTypes
Rationale:	
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00043] Description of data elements ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the description of data elements for diagnostic services.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

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 ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of events.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00024] Configuration of DTCs ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of DTCs.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00025] Combined Events ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of Combined Events.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00026] Enable Conditions ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of Enable Conditions.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00027] Storage Conditions ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of Storage Conditions.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00028] Enable Condition Groups ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of Enable Condition Groups.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00029] Storage Condition Groups ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of Storage Condition Groups.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00030] Assignment of Enable Condition Groups ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the assignment of Enable Condition Groups to an event.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00031] Assignment of Storage Condition Group ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the assignment of Storage Condition Group to an event.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00032] Configuration of Extended Data Records ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of Extended Data Records.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00033] Configuration of Snapshot Records ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the configuration of Snapshot Records.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00034] Description of Data Identifiers ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the description of Data Identifiers for diagnostic services.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00035] Description of Dynamic Data Identifiers ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the description of Dynamic Data Identifiers for diagnostic services.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00036] Description of Routine Identifiers ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the description of Routine Identifiers for diagnostic services.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00037] Description of I/O Identifiers ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the description of I/O Identifiers for diagnostic services.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00038] Description of array data types ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the description of array data types for diagnostic services.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00039] Diagnostic Service Table ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the capability to specify a Diagnostic Service Table.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00053] Debouncing of diagnostic events ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the specification of how diagnostic services shall be debounced.
Rationale:	
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00054] Operation cycles ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the specification of operation cycles for diagnostic purposes.
Rationale:	
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00055] Aging ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the specification of aging for diagnostic purposes.
Rationale:	
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00056] Indicator ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the specification of indicators for diagnostic purposes.
Rationale:	
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00045] Textual descriptions ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the capability to specify textual descriptions for event and DTC attributes.
Rationale:	tbd
Use Case:	tbd
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 ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the capability to specify Diagnostic Sessions.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00041] Access Permissions ⌈

Type:	valid
Description:	The Diagnostic Extract shall support the capability to specify Access Permissions.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

[RS_DEXT_00042] Security Levels ⌈

Type:	valid

Description:	The Diagnostic Extract shall support the capability to specify Security Levels.
Rationale:	tbd
Use Case:	tbd
Dependencies:	–
Supporting Material:	–

⌋()

A History of Constraints and Specification Items

A.1 Constraint History of this Document according to AUTOSAR R4.2.1

A.1.1 Added Traceables 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 Added Traceables in 4.2.1

none

A.1.3 Deleted Traceables in 4.2.1

none

A.2 Constraint History of this Document according to AUTOSAR R4.2.2

A.2.1 Added Traceables in 4.2.2

none

A.2.2 Changed Traceables 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 Traceables 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