Elektrobit

Q_LAH80126

Implementation Matrix

Revision 1.3, Status: released

Elektrobit Automotive GmbH
Am Wolfsmantel 46
D-91058 Erlangen
GERMANY

Phone: +49 9131 7701-0
Fax: +49 9131 7701-6333
http://www.elektrobit.com

Table of Contents

1. Implementation Matrix

Id:	80126-A1
Version:	NA
Description:	1 Scope

Id:	80126-A2
Version:	NA
Description:	This document is aimed at electronics developers, manufacturers of diagnostic and test equipment, system engineers, and control unit suppliers involved in programming and updating control unit programs with respect to design, development, and implementation. All control units with programmable memories that are programmed with the diagnostic equipment of the Volkswagen Group and diagnosed via UDS must comply with these specifications in full.

Id:	80126-A3
Version:	NA
Description:	The purpose of this document is to describe in full the programming of electronic control units in the automobile using standard specifications and to standardize the programming sequence between the client and the server in Development, Production, and After-Sales Service for the purchaser. A further goal is to achieve uniformity so that standard software can be used.

Id:	80126-A4
Version:	NA
Description:	In the interests of standardization, the terminology from the specification [Q-LAH 80124] is used.

Id:	80126-A1125
Version:	NA
Description:	All personal interpretations and extensions related to this document or any of the referenced documents must be agreed with representatives of the Volkswagen diagnostic special function department and the responsible representative of the manufacturing planning department and documented in the ECU specification (BT-LAH).

Id:	80126-A5
Version:	NA

Description:	2 Glossar

Id:	80126-A6
Version:	NA
Description:	All the terms, definitions and abbreviations from the documents listed in the Section "Normative references" apply, as well as the following:

Id:	80126-A7
Version:	NA
Description:	Table 1 - Glossary:

Id:	80126-A803
Version:	NA
Description:	For an explanation of other terms not listed here, refer to the diagnosis glossary [Diagglossar]

Id:	80126-A8
Version:	NA
Description:	3 Requirements

Id:	80126-A9
Version:	NA
Description:	3.1 General information

Id:	80126-A10
Version:	NA
Description:	A server must be programmable while integrated in the vehicle network or as a standalone server without further conditions (e.g., restbus simulation, etc.) and without further interventions by the diagnostic tester according to this specification. This requirement does not affect bus signals which are mandatory for the pyhsical operation (e.g. FlexRay bus requires a minimum number of 'coldstarter' ECUs).

Id:	80126-A11
Version:	NA
Description:	Example: If there is no terminal status, this is regarded as fulfilled and the control unit can be flashed as a stand-alone control unit.

Id:	80126-A1337

Version:	NA
Description:	If there are boundary conditions that prevent the programming of the server in the vehicle network, the boundary conditions for services listed in [Q-LAH 80124] under chapter 17 shall be observed. Deviations shall be agreed upon with Q, P and customer service representatives and shall be documented in the BT-LAH.

Id:	80126-A12
Version:	NA
Description:	Other programming interfaces or methods not based on this specification are not permitted.

Id:	80126-A13
Version:	NA
Description:	3.2 Server

Id:	80126-A14
Version:	NA
Description:	3.2.1 General

Id:	80126-A15
Version:	NA
Description:	In order to ensure programming in the control unit network, the following requirements must be observed for programmable as well as non-programmable servers:

Id:	80126-A19
Version:	NA
Description:	Each server must implement the DiagnosticSessionControl (extendedSession) (10 03 hex) service and the required session switchover.

Id:	80126-A20
Version:	NA
Description:	3.2.2 Non-programmable servers

Id:	80126-A21
Version:	NA
Description:	For every non-programmable server, the following requirements exist:

Id:	80126-A22
Version:	NA
Description:	A non-programmable server must be designated by its identification data.

Id:	80126-A23
Version:	NA
Description:	A non-programmable server must reject the DiagnosticSessionControl (ProgrammingSession) service (10hex), Subfunction 0x02-ECUProgrammingSession with NRC=0x12 SubFunctionNotSupported (12 hex).

Id:	80126-A24
Version:	NA
Description:	Formerly programmable servers that are then converted to a "mask" must be handled like "nonprogrammable servers".

Id:	80126-A927
Version:	NA
Description:	Note: In the case of changeover to a mask, the following must be noted: • The control unit rejects the reverting back to the ProgrammingSession with NRC 0x12 • The identifier 0xF1DF indicates that the control unit is not programmable.

Id:	80126-A25
Version:	NA
Description:	Formerly programmable servers that are then converted to a "mask" must be handled like "nonprogrammable servers".

Id:	80126-A26
Version:	NA
Description:	3.2.3 Programmable servers

Id:	80126-A27
Version:	NA
Description:	The following requirements apply to programmable servers:

Id:	80126-A28
Version:	NA
Description:	A consistent and programmable server must provide for at least two partitions (bootloader and application).

Id:	80126-A29
Version:	NA
Description:	A programmable server must ensure the compatibility of SW/SW, SW/HW, and SW/data structure (e.g., EEPROM data).

Id:	80126-A30
Version:	NA
Description:	The joint use of program code for the communication functions (com stack) by both the application and the boot loader is not permitted.

Id:	80126-A31
Version:	NA
Description:	A programmable server must always guarantee that programming can occur, unless unfulfilled programming pre-conditions (following/implementing [Q-LAH 80124]) or error statuses (e.g., flash or EEPROM defect) do not permit programming.

Id:	80126-A900
Version:	NA
Description:	Note: The technical implementation of the programming pre-conditions must be specified in the ECU component specification (BT-LAH) jointly by the supplier and the control unit developer. If the requirements of [Q-LAH 80124] are not fulfilled a consultation of P and KD is obligatory.

Id:	80126-A32
Version:	NA
Description:	A programmable server whose boot loader can be replaced must be able to update the boot loader with the programming sequence described in these Performance Specifications. (see also chapter "Optional Bootloader-Update").

Id:	80126-A33
Version:	NA
Description:	A programmable server must be able to update the application with the programming sequence described in these Performance Specifications.

Id:	80126-A34
Version:	NA
Description:	A programmable server must be able to update the application data with the programming sequence described in these Performance Specifications.

Id:	80126-A35
Version:	NA
Description:	A programmable server must guarantee the service of a new programming session when an active programming session was interrupted previously (e.g caused by a voltage drop).

Id:	80126-A36
Version:	NA
Description:	A programmable server must guarantee programmability even after a ground interruption while programming is running.

Id:	80126-A37
Version:	NA
Description:	A programmable server must guarantee programmability even after a communication interruption while programming is running.

Id:	80126-A38
Version:	NA
Description:	A programmable server must guarantee programmability even after a flashing operation is canceled due to overvoltage.

Id:	80126-A39
Version:	NA
Description:	A programmable server must guarantee programmability even after a flashing operation is canceled due to undervoltage.

Id:	80126-A1082
Version:	NA
Description:	A programmable server must guarantee programmability after all kinds of interruptions during flash programming.

Id:	80126-A40
Version:	NA
Description:	3.3 Software

Id:	80126-A703
Version:	NA

Description:	3.3.1 Compression

Id:	80126-A704
Version:	NA
Description:	A compression method is used to shorten the transmission times of the tester software to the control unit to be programmed. It is also used to safe resources and reduce the amount of data size for data exchange within production and service departents.

Id:	80126-A700
Version:	NA
Description:	All programmable software parts must be transferred in compressed form to the control unit and must be decompressed in the control unit (in the boot loader).

Id:	80126-A705
Version:	NA
Description:	The data to be programmed must already be stored as compressed data in the flash container. See [Q-LAH 80128]

Id:	80126-A701
Version:	NA
Description:	The LZSS algorithm with a dictionary size of 1023 bytes must be used as the compression/decompression algorithm. A newer compression-/decompression algorithm with higher compression ratio is allowed.

Id:	80126-A1004
Version:	NA
Description:	Usage of other compression-/decompression algorithms must be agreed with the VOLKSWAGEN AG diagnostic functions department

Id:	80126-A702
Version:	NA
Description:	To coordinate the availability of standardsoftware for your specific ECU, read the Stanardsoftare specification [Q-LAH 893.910.A] or please contact one of this departments: standardsoftware@volkswagen.de standardsoftware@audi.de standardsoftware@porsche.de

Id:	80126-A41
Version:	NA

Description:	3.3.2 Software distribution

Id:	80126-A42
Version:	NA
Description:	The SW distribution in the server must be performed by the supplier in coordination with the purchaser.

Id:	80126-A716
Version:	NA
Description:	Control units, which need to be (re-)programmed on the production line must coordinate the following topics with representatives of the responsible manufacturing planning department and document in the ECU component specification (BT-LAH): * the maximum programming time for individual logical blocks * the maximum programming time for the complete software * the state of the received ECU (erased, default-application with diagnostics capability)

Id:	80126-A717
Version:	NA
Description:	Through suitable specification of the block sizes for the logical blocks, the interruptibility of the flash operation and, thus, the ability to distribute the blocks to multiple production cycles must be achieved.

Id:	80126-A1190
Version:	NA
Description:	For ECUs that must be calibrated using online remote update, the maximum flashing time and the data size for individual logical blocks and for the entire software package must be agreed upon with representatives of the appropriate Online Remote Update and Diagnostics departments and the department responsible for the "flashing" function.

Id:	80126-A43
Version:	NA
Description:	The memory of a server is divided into at least two SW areas.

Id:	80126-A44
Version:	NA
Description:	One area, the SW application, contains the functions and data required for the normal operation of the server and thus guarantees the functions in the vehicle. This part of the SW is stored in the flashable, non-volatile memory.

Id:	80126-A45
Version:	NA
Description:	An additional SW area, the flash boot loader, contains the functions required to replace the software in the flashable, non-volatile memory.

Id:	80126-A46
Version:	NA
Description:	Both SW areas each contain their own software module for the diagnostic and communication services.

Id:	80126-A924
Version:	NA
Description:	Note: For control unit families that have the same software but use different part numbers, the part number can also be formed as a separate logical block. The basic software then contains a default part number that could be overwritten by the additional part number block.

Id:	80126-A925
Version:	NA
Description:	The distribution of the part numbers as a separate logical block must be agreed with the engineering department responsible for flashing.

Id:	80126-A47
Version:	NA
Description:	A logical block represents one or several coherent address areas of the memory.

Id:	80126-A48
Version:	NA
Description:	The following figure (80126-A49) shows the distribution of the server memory, independent of the physical hardware architecture.

Id:	80126-A49
Version:	NA
Description:	Figure 1 - Exemplary ECU memory distribution

Id:	80126-A1098
Version:	NA

Description:	Every logical block in the server must unique identifiable. (see also chapter "Logistic data").

Id:	80126-A50
Version:	NA
Description:	3.3.3 Block segmentation

Id:	80126-A51
Version:	NA
Description:	A logical block does not necessarily have to comprise a contiguous address space. Segmentation of the block is possible by taking the relevant architecture and the existing physical memory into consideration.

Id:	80126-A52
Version:	NA
Description:	Figure 80126-A53 shows an example of the distribution (code segments) of a logical block, depending on a sample physical hardware architecture (memory segments).

Id:	80126-A53
Version:	NA
Description:	Figure 2 - Segmentation of logical Blocks

Id:	80126-A931
Version:	NA
Description:	A logical block is addressed by a block index.

Id:	80126-A54
Version:	NA
Description:	The segments of a logical block must be sorted in ascending order of addresses for the transfer.

Id:	80126-A932
Version:	NA
Description:	Each segment of a logical block receives a segment index for addressing purposes.

Id:	80126-A933

Version:	NA
Description:	The addressing of a segment (e.g., ReqDownload) is carried out using the logical address of the segment, which is composed of a block index and segment index.

Id:	80126-A934
Version:	NA
Description:	Example: Logical Block 01 with three segments - Segment 01 – LogAddress 0101 - Segment 02 – LogAddress 0102 - Segment 03 – LogAddress 0103 Block index 0x01 is used for addressing purposes for erasing the logical block (see RoutineControl-EraseMemory). The logical address of the segment to be written is used in each case to write the segments (see RequestDownload).

Id:	80126-A871
Version:	NA
Description:	It must be possible to program all programmable logical blocks independently of one another. If the SW-architecture requires dependencies in between the logical blocks please coordinate this with the responsible ECU contact.

Id:	80126-A886
Version:	NA
Description:	The programmable logical blocks can be programmed in any order. The boot loader must accept any order.

Id:	80126-A55
Version:	NA
Description:	3.3.4 Application SW

Id:	80126-A56
Version:	NA
Description:	The following requirements are defined for application software:

Id:	80126-A57
Version:	NA
Description:	The application SW must be programmable independently of the application data and are therefore organized as one or more stand-alone logical blocks.

Id:	80126-A58
Version:	NA

Description:	The application SW must verify the consistency for the application data.

Id:	80126-A60
Version:	NA
Description:	It must be possible to identify the application SW according to [Q-LAH 80125].

Id:	80126-A61
Version:	NA
Description:	3.3.5 Application data

Id:	80126-A62
Version:	NA
Description:	The following requirements are defined for the application data:

Id:	80126-A63
Version:	NA
Description:	The application data must be programmable independent of the application SW and, thus, be organized as one or more stand-alone logical blocks.

Id:	80126-A64
Version:	NA
Description:	It must be possible to identify the application data according to [Q-LAH 80125].

Id:	80126-A65
Version:	NA
Description:	3.3.6 Bootloader-SW

Id:	80126-A66
Version:	NA
Description:	The following requirements are defined for the boot loader software:

Id:	80126-A67
Version:	NA
Description:	The SW of a non-replaceable boot loader must be in a protected area of the memory, such that the server can always be programmed reliably. A SW or HW protection mechanism must be used to protect the software from being erased or overwritten. If the utilized microcontroller supports a HW protection, this must be used.

Id:	80126-A69
Version:	NA
Description:	It must be possible to identify the boot loader SW according to [Q-LAH 80125].

Id:	80126-A856
Version:	NA
Description:	The relevant boot loader version ID must be returned for the boot loader software with the identifier "0xF1AB-VW Logical Software Block Version".

Id:	80126-A936
Version:	NA
Description:	Logical blocks, accessed by identifier "0xF1AB-VW Logical Software Block Version", which return the value 0x41 46 46 45 (7 Bit-ASCII character = 'AFFE') are not allowed to be reprogrammed with partial programming. The flashjob must skip these blocks and leave them unchanged.

Id:	80126-A857
Version:	NA
Description:	The appropriate boot loader version must be output for the boot loader SW by means of the

Id:	80126-A1133
Version:	NA
Description:	DataIdentifier "0xF1AB-VW Logical Software Block Version."

Id:	80126-A1136
Version:	NA
Description:	In certain cases ECUs stay always in the bootloader after the icar production (e.g. ehen car is built only partly for distribution, datasets or application are not fully programmed.)

Id:	80126-A1134
Version:	NA
Description:	To avoid stranded vehicles caused by an empty battery all KL30 ECUs require Sleep/Wakeup mode functionality in the Bootloader.

Id:	80126-A1192
Version:	NA

Description:	All flashable ECUs must implement passive network management.

Id:	80126-A1327
Version:	NA
Description:	Information regarding passive management can be found in [NM-CAN] and [NM-FlexRay].

Id:	80126-A1137
Version:	NA
Description:	The ECU is allowed to enter Sleep-Mode after all Lock-times and ECU-specific afterrun timers are expired.

Id:	80126-A1138
Version:	NA
Description:	The Lock-times and ECU-specific afterrun times have to be documented in the ECU component specification (BT-LAH).

Id:	80126-A1139
Version:	NA
Description:	ECUs which stay in Sleep-Mode should wakeup if they receive a diagnostic request.

Id:	80126-A1140
Version:	NA
Description:	For detailed requirements related to Sleep/Wakeup-Mode please refer to chapter 17 "Display of a server-diagnosis in the network" [Q-LAH 80124]

Id:	80126-A1135
Version:	NA
Description:	The detailled requirements for the Sleep/Wakeup-Modes are not part of this specification and have to be clarified with the function department for the network powermanagement and shall be documented in the ecu component specification (BT-LAH)

Id:	80126-A75
Version:	NA
Description:	3.4 Programming documentation

Id:	80126-A76

Version:	NA
Description:	The following requirements are defined for data used for documenting programming:

Id:	80126-A912
Version:	NA
Description:	The following identifiers are used to document the programming: 0x0407-VW Logical Software Block Counter Of Programming Attempts, 0x040F-VW Logical Software Block Lock Value, 0xF1AB-VW Logical Software Block Version, 0xF15B-Fingerprint And Programming Date Of Logical Software Blocks, 0xF187-VW Spare Part Number, 0xF189-VW Application Software Version Number, 0xF1DF-ECU Programming Information, and, especially for ECUs with central diagnostic access (Gateway): 0xF190-Vehicle Identification Number, Note: this dataIdentifier must be implemented in the bootloader of a Gateway ECU, so that the VIN can be read also after a interrupted flashprogramming session. (See the example in Appendix C)

Id:	80126-A77
Version:	NA
Description:	The data used to document the programming and the associated data identifiers and data definitions must be realized according to [Q-LAH 80125].

Id:	80126-A78
Version:	NA
Description:	The data used for documenting the programming must be stored in non-volatile memory.

Id:	80126-A79
Version:	NA
Description:	The following identification data must be reserved for the number of logical blocks: 0x0407-VW Logical Software Block Counter Of Programming Attempts, 0x040F-VW Logical Software Block Lock Value, 0xF1AB-VW Logical Software Block Version, 0xF15B-Fingerprint And Programming Date Of Logical Software Blocks.

Id:	80126-A1130
Version:	NA
Description:	The number of logical blocks must be identical for the referenced identifiers in 80126-A79.

Id:	80126-A80
Version:	NA
Description:	A server must supply the same information for the data used to document the programming, irrespective of the current operating mode (application or boot loader).

Id:	80126-A81
Version:	NA
Description:	In the course of programming, the data of the programming of application and boot loader must be adjusted to the status of the server, as appropriate (see Section "State administration for programming").

Id:	80126-A82
Version:	NA
Description:	3.4.1 Programming counter and blocking values

Id:	80126-A684
Version:	NA
Description:	The boot loader must always be identified as the first block, irrespective of whether or not the boot loader itself is programmable.

Id:	80126-A685
Version:	NA
Description:	A boot loader that has not yet be programmed or that is not be programmed must return the default values for test detection (car workshopcode(WSC)), programming date, and programming counter. (See 80126-A98, 80126-714 and 80126-A85).

Id:	80126-A83
Version:	NA
Description:	Each logical block must have a programming counter.

Id:	80126-A84
Version:	NA
Description:	The programming counter returns the number of already performed programming attempts.

Id:	80126-A85

Version:	NA
Description:	The programming counter must be set to "0000" in the as-received condition. For "Not programmable" memory blocks

Id:	80126-A86
Version:	NA
Description:	The programming counter must be incremented each time this logical block is erased.

Id:	80126-A1174
Version:	NA
Description:	If blocks have already been completely erased, the programming counter must not be incremented if there is a new erasing process.

Id:	80126-A87
Version:	NA
Description:	The programming counter must be stored in the non-volatile memory.

Id:	80126-A88
Version:	NA
Description:	The programming counter must be retained for the service life of the server.

Id:	80126-A90
Version:	NA
Description:	Each logical block must have a blocking value.

Id:	80126-A91
Version:	NA
Description:	The lock value describes the maximum possible number of programming attempts and is a hardware-dependent constant.

Id:	80126-A92
Version:	NA
Description:	A blocking value with the value "0" means that the logical block can be programmed without limitations.

Id:	80126-A910
Version:	NA

Description:	If the programming counter has reached the maximum counter value of 0xFFFF with a blocking value of "0", the programming counter must remain at this value. However, programming continues to be possible.

Id:	80126-A93
Version:	NA
Description:	If the programming counter reaches the blocking value (provided that the blocking value is > 0) before the erase routine is executed, the erasing and thus the server programming must be rejected.

Id:	80126-A94
Version:	NA
Description:	The blocking value must be stored in the non-volatile memory.

Id:	80126-A687
Version:	NA
Description:	If, in the case of a programmable block, the blocking value is inferior to 100, a detailed technical justification must be provided to the purchaser.

Id:	80126-A926
Version:	NA
Description:	A blocking value >= 1000 must be implemented for development control units to allow for automated flash tests during development. For series production the blocking values have to be agreed with the responsible "Flashprogramming" function department at Volkswagen and have to be documented in the ECU component specifications (BT-LAH).

Id:	80126-A686
Version:	NA
Description:	The order of the block identifiers of all logical blocks must always appear in the same order, starting with the boot loader, for the identifiers (0x0407, 0x040F, 0xF15B, 0xF1AB). Example: Server with three logical blocks: 1. Block: Bootloader 2. Block: Application 3. Block: Dataset

Id:	80126-A95
Version:	NA
Description:	3.4.2 Tester serial numbern

Id:	80126-A96

Version:	NA
Description:	Each logical block must have a 'tester serial number' (= fingerprint).

Id:	80126-A97
Version:	NA
Description:	The tester serial number indicates the identifier of the client which last programmed the corresponding logical block.

Id:	80126-A98
Version:	NA
Description:	The tester serial number must be set to 0x000000000000 in the as-received condition.

Id:	80126-A99
Version:	NA
Description:	The client must transfer its serial number to the server as part of the programming sequence.

Id:	80126-A100
Version:	NA
Description:	The tester serial number must be stored in non-volatile memory each time this logical block is erased.

Id:	80126-A101
Version:	NA
Description:	See also Sections "ReadDataByIdentifier (Fingerprint)" and "WriteDataByIdentifier (Fingerprint)".

Id:	80126-A102
Version:	NA
Description:	3.4.3 Programming date

Id:	80126-A103
Version:	NA
Description:	Each logical block must have a programming date.

Id:	80126-A104

Version:	NA
Description:	The programming date provides the point in time (year, month, day) that programming was performed by the client.

Id:	80126-A105
Version:	NA
Description:	The programming date must be set to the date of manufacture of the control unit in the as-received condition. Alternative: Fixed valid date 01.01.(20)00

Id:	80126-A106
Version:	NA
Description:	The client must transfer the programming date to the server as part of the programming sequence.

Id:	80126-A107
Version:	NA
Description:	The programming date must be stored in the non-volatile memory each time this logical block is erased.

Id:	80126-A108
Version:	NA
Description:	See also Sections "ReadDataByIdentifier (Fingerprint)" and "WriteDataByIdentifier (Fingerprint)".

Id:	80126-A109
Version:	NA
Description:	3.4.4 Logistical data

Id:	80126-A110
Version:	NA
Description:	The logistical data are broken down into two areas. One area is used for hardware identification (HW logistical data). The other area is used for identification of the logical blocks and the complete software (SW logistical data).

Id:	80126-A111
Version:	NA
Description:	The HW logistical data identifies the hardware of the server and must not be changed as a result of programming. The following listing makes no claim of

	completeness. * 0xF191-VW ECU Hardware Number * 0xF1A3-VW ECU Hardware Version Number

Id:	80126-A112
Version:	NA
Description:	The SW logistical data identifies the version of the SW of the server and must be updated by programming. * 0xF187-VW Spare Part Number * 0xF189-VW Application Software Version Number * 0xF19E-ASAM ODX File Identifier * 0xF1A2-ASAM ODX File Version * 0xF1DF-ECU Programming Information Further requirements must be documented in the ECU component specification (BT-LAH).

Id:	80126-A911
Version:	NA
Description:	In order for the identifiers named in 80126-A112 to be updated after programming, the corresponding information must be stored in the programming data.

Id:	80126-A986
Version:	NA
Description:	The following DataIdentifier are mandatory for the update-programming and have to be implemented: 0xF190-Vehicle Identification Number 0xF17C-VW FAZIT Identification String

Id:	80126-A113
Version:	NA
Description:	For the logistical data,[Q-LAH 80125] must be implemented.

Id:	80126-A114
Version:	NA
Description:	The following requirements are defined:

Id:	80126-A115
Version:	NA
Description:	It must be possible to explicitly identify each logical block of the server.

Id:	80126-A849
Version:	NA
Description:	For each logical block (regardless of whether the block is programmable), the following logistical data must be output (see also [Q-LAH 80125]): * 0x0407-VW

	Logical Software Block Counter Of Programming Attempts, * 0x040F-VW Logical Software Block Lock Value, * 0xF15B-Fingerprint And Programming Date Of Logical Software Blocks, * 0xF1AB-VW Logical Software Block Version..

Id:	80126-A116
Version:	NA
Description:	The logistical data must be stored in the non-volatile memory of the respective logical block.

Id:	80126-A117
Version:	NA
Description:	A programmable server must be able to supply the logistical data of the last "valid" programmed software from a separate backup (non-volatile memory) under any circumstances. These are: * 0xF187-VW Spare Part Number * 0xF189-VW Application Software Version Number * 0xF19E-ASAM ODX File Identifier * 0xF1A2-ASAM ODX File Version

Id:	80126-A794
Version:	NA
Description:	The logistical data 0xF187-VW Spare Part Number 0xF189-VW Application Software Version Number 0xF1DF-ECU Programming Information in the backup must be updated immediately after the successful compatibility/consistency check.

Id:	80126-A119
Version:	NA
Description:	The following logistical data in the backups of all logical blocks must be updated immediately after the successful integrity check (checkMemory) of the respective block. * 0xF15B-Fingerprint And Programming Date Of Logical Software Blocks * 0xF1AB-VW Logical Software Block Version

Id:	80126-A120
Version:	NA
Description:	A programmable server must provide the same information for the logistical data from physically identical memory addresses, irrespective of the current operating state of the server (application or boot loader).

Id:	80126-A848
Version:	NA

Description:	3.4.4.1 Logistical data in the as-received condition

Id:	80126-A850
Version:	NA
Description:	This section describes the logistical data of the server in its as-received condition from the Tier-1. In this scope the server is completely programmed and the software is operational

Id:	80126-A851
Version:	NA
Description:	The DataIdentifer "0x0407-VW Logical Software Block Counter Of Programming Attempts" must be coded according to 80126-A85.

Id:	80126-A852
Version:	NA
Description:	The DataIdentifer "0x040F-VW Logical Software Block Lock Value"must be coded according to 80126-A92.

Id:	80126-A853
Version:	NA
Description:	The DataIdentifer "0xF15B-Fingerprint And Programming Date Of Logical Software Blocks" of a logical Block must be coded according to 80126-A105, 80126-A714, 80126-A98 and 80126-A385.

Id:	80126-A854
Version:	NA
Description:	The ProgrammingState within data itentifier "0xF15B-Fingerprint And Programming Date Of Logical Software Blocks" (80126-A385) must be coded with 0x00 (valid) in the as-received condition in the case of valid software.

Id:	80126-A855
Version:	NA
Description:	Dataidentifier DataIdentifer "0xF1AB-VW Logical Software Block Version" must be coded with the respectively valid block version. In the case of logical blocks for boot loader, 80126-A856 and 80126-A857 apply.

Id:	80126-A877
Version:	NA

Description:	In the as-received condition, the r DataIdentifer "0xF1AB-VW Logical Software Block Version" for erased or nonwritten blocks must be coded as specified in 80126-A694.

Id:	80126-A858
Version:	NA
Description:	3.4.4.2 Logistical data following cancelation of a flash operation

Id:	80126-A859
Version:	NA
Description:	The following section describes the logistical data to be output by a control unit after a canceled flash operation in the boot loader.

Id:	80126-A861
Version:	NA
Description:	The DataIdentifier "0xF1DF-ECU Programming Information" indicates that the server was not programmed consistently. The server starts in the boot loader. The application is not started

Id:	80126-A860
Version:	NA
Description:	The DataIdentifer "0xF15B-Fingerprint And Programming Date Of Logical Software Blocks" must identify those blocks which were not coded caused by a canceled flash operation as invalid (ProgrammingState). Please refer to 80126-A385.

Id:	80126-A862
Version:	NA
Description:	For logical blocks that are still invalid after the canceled flash operation, the DataIdentifer "0xF1AB-VW Logical Software Block Version" must be coded according to 80126-A694.

Id:	80126-A863
Version:	NA
Description:	After the canceled flash operation, the identifier "0xF19E-ASAM ODX File Identifier" must continue to be coded with the last valid value

Id:	80126-A864
Version:	NA

Description:	After the canceled flash operation, the identifier "0xF1A2-ASAM ODX File Version" must continue to be coded with the last valid value.

Id:	80126-A865
Version:	NA
Description:	After the canceled flash operation, the foloowing identifiers remain unchanged and reflect the last valid programming state: 0xF187-VW Spare Part Number 0xF189-VW Application Software Version Number 0xF191-VW ECU Hardware Number 0xF1A3-VW ECU Hardware Version Number

Id:	80126-A121
Version:	NA
Description:	3.5 Safety

Id:	80126-A122
Version:	NA
Description:	3.5.1 Safety concept

Id:	80126-A123
Version:	NA
Description:	Safety requirements that extend beyond the concepts described here must be defined jointly with the purchaser in the individual case.

Id:	80126-A124
Version:	NA
Description:	3.5.2 Programming pre-conditions (checkProgrammingPreConditions)

Id:	80126-A125
Version:	NA
Description:	For a server, situations can exist in which programming is not permissible (e.g. engine running, undervoltage, activation of actuators, etc.).

Id:	80126-A1120
Version:	NA
Description:	OBD-ECU's include specific coding preconditions from the requirement specification "OBD Requirements for emission relevant ECUs".

Id:	80126-A126

Version:	NA
Description:	In order to ensure that a server is in a safe state prior to programming, the client checks the programming pre-conditions of the server immediately prior to programming.

Id:	80126-A128
Version:	NA
Description:	The type and number of programming pre-conditions are specific to the particular server and depend on the function and on the safety concept of the server application.

Id:	80126-A1116
Version:	NA
Description:	To ensure start up during car production (means: car is partly assembled) all preconditions (Type and Number) have to be agreed with the production department and must be documented in the ECU component specification (BT-LAH) mandatory.

Id:	80126-A1330
Version:	NA
Description:	The programming preconditions implemented in the server must be transmitted via the DID "0x0448- Programming_preconditions "(see [Q-LAH 80125])

Id:	80126-A1182
Version:	NA
Description:	Predefined programming preconditions from the manufacturers' software initiative (HIS) (1 to 127) and Volkswagen/Audi (128 to 191) ranges must be used for relevance.

Id:	80126-A1183
Version:	NA
Description:	It is not permissible to use a deviating value from the supplier-specific range (192 to 255) for a predefined programming precondition.

Id:	80126-A129
Version:	NA
Description:	Before programming a server, the client must request the programming pre-conditions from the server and check them.

Id:	80126-A130
Version:	NA
Description:	Programming by the client is only permitted when the server signals all conditions as fulfilled.

Id:	80126-A131
Version:	NA
Description:	If a server signals programming pre-conditions as not fulfilled, it must reject a switch to the ProgrammingSession. The negative response have to occur according to 80126-A804.

Id:	80126-A132
Version:	NA
Description:	A server must reject a request for a change to the ECUProgrammingSession if unfulfilled programming preconditions are present, regardless of the fact whether the client has not queried them at all or has not queried them recently.

Id:	80126-A134
Version:	NA
Description:	If information on a pre-condition is not available during the check (server uninstalled, server partly installed, server in the boot loader, etc.), the pre-condition is deemed to have been fulfilled.

Id:	80126-A135
Version:	NA
Description:	Note: To ensure that the programming pre-conditions do not block programming in the event of uninstalled servers or a partially installed harness in the vehicle (lack of signal, CAN message), the following procedure must be used:

Id:	80126-A805
Version:	NA
Description:	1. If the signal of a programming pre-condition is not present when the programming-conditions are queried (signal timeout, defined substitute value, or message obsolete), the requirements from[Q-LAH 80124], V2.0 and later versions, chapter "General Requirements for the check of auxiliary conditions".

Id:	80126-A806
Version:	NA

Description:	A server must also reject a request to switch to the ProgrammingSession if unfulfilled programming pre-conditions are present and these have not been queried or are not currently being queried by the client.

Id:	80126-A136
Version:	NA
Description:	3.5.3 Access authorization (seed and key)

Id:	80126-A137
Version:	NA
Description:	The requirements of [Q-LAH 80124 Appendix A] must be implemented

Id:	80126-A138
Version:	NA
Description:	3.5.4 Integrity/authenticity check (checkMemory)

Id:	80126-A139
Version:	NA
Description:	In order to ensure that the data to be programmed (logical block) is transferred without errors and stored correctly in the memory, a check of the programmed memory area is required.

Id:	80126-A140
Version:	NA
Description:	To this end, for example, test methods such as CRC32 or digital signatures, etc., serve to determine the data integrity, authenticity, and consequently the validity.

Id:	80126-A142
Version:	NA
Description:	Control units with more stringent requirements regarding protection of flash data against manipulation must perform the check based on electronic signatures. The purchaser summarizes the technical and organizational measures for this under the term "flash data security" (FDS). The need to implement these requirements must be clarified between the supplier and purchaser and must be documented in the Component Performance Specifications.

Id:	80126-A711
Version:	NA

Description:	The check sum computation must be carried out according to HIS Security Class DDD with Ethernet-CRC32 (width=32 poly=0x04c11db7 init=0xffffffff refin=true refout=true xorout=0xffffffff). The following generator polynomial must be used with the initial value shown below: Generator polynomial: G(X) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1 Initwert: 0xFFFFFFFF Siehe dazu auch [HIS SECMOD].

Id:	80126-A144
Version:	NA
Description:	The client itself must not compute the check value that it transmits for the check.

Id:	80126-A145
Version:	NA
Description:	The client itself must not compute the check value that it transmits for the check.

Id:	80126-A146
Version:	NA
Description:	Alternatively, the CRC32 value can also be contained in the data stream of the logical block. In this case the value for the check request must be coded with 0x00000000.

Id:	80126-A872
Version:	NA
Description:	If control units support flash data security [FDS], all encrypted/compressed data require a combination of CRC32 and signature for the checksums. See also [FDS].

Id:	80126-A954
Version:	NA
Description:	While flashing of compressed data, the checksum-calculation need to be done always with uncompressed data to exclude decompression errors.

Id:	80126-A147
Version:	NA
Description:	The check of the transferred data (logical block) is not initiated in the server until after the data have been completely transferred by the client with the check request. For transmission acknowledgment it is mandatory for single- and multi-core processors that all transferred data, in all processors, is fully written to the flash-memoy.

Id:	80126-A148
Version:	NA
Description:	All bytes of the transferred data must be included completely and exclusively in the check. Exception: If there is necessary logistical data in the data stream of the logical block, this is not included in the check. All relevant Dataareas have to be agreed with the responsible function department for "flash programming" and must be documentend in the ECU component specification (BT-LAH)

Id:	80126-A149
Version:	NA
Description:	The application must only be started if all logical blocks are "valid". Otherwise, the control unit must remain in the boot loader.

Id:	80126-A915
Version:	NA
Description:	Exception: Optional logical blocks do not have to be included in this check. See 80126-A914

Id:	80126-A150
Version:	NA
Description:	Control units to which the requirements according to "Flash Data Security" apply must be developed according to [FDS] and the documents referenced there.

Id:	80126-A1005
Version:	NA
Description:	When downloading drivers, make sure that after a negative check memory test of a driver any further download attempt in Flash or RAM memory areas must be rejected by the bootloader. Only after a hard reset (0x11 01), Cl. 15 exchange or other restart the boot manager must re EraseMemory- or Request download request will be answered positively.

Id:	80126-A1006
Version:	NA
Description:	Figure 3 - Example of a typical driver downloads

Id:	80126-A151
Version:	NA
Description:	See also section "checkMemory"

Id:	80126-A897
Version:	NA
Description:	To coordinate the availability of standardsoftware for your specific ECU, read the Stanardsoftare specification [Q-LAH 893.910.A] or please contact one of this departments: standardsoftware@volkswagen.de standardsoftware@audi.de standardsoftware@porsche.de

Id:	80126-A152
Version:	NA
Description:	3.5.5 Compatibility/consistency check (checkProgrammingDependencies)

Id:	80126-A153
Version:	NA
Description:	For a server, situations can arise due to programming which can put the operational reliability of the server at risk through incompatibility/inconsistency between the SW/SW or SW/HW

Id:	80126-A154
Version:	NA
Description:	In order to ensure that a server is not in an undefined state after programming, the client requires the server to check the compatibility/consistency between the SW/SW and SW/HW.

Id:	80126-A155
Version:	NA
Description:	As part of this, the server checks whether or not the individual logical blocks are complete and compatible with one another. In addition, a check is made to determine whether the software is compatible with the hardware version (e.g., variants of sensors/actuators) and other data structures (e.g., EEPROM data).

Id:	80126-A156
Version:	NA
Description:	The method used to check compatibility/consistency must be specified by the supplier in consultation with the purchaser.

Id:	80126-A157
Version:	NA
Description:	A method for checking the compatibility/consistency must be implemented without exception.

Id:	80126-A158
Version:	NA
Description:	Solely the server is responsible for carrying out the check.

Id:	80126-A159
Version:	NA
Description:	The check of compatibility/consistency is not initiated in the server until all of the logical blocks to be programmed have been transmitted completely by the client with the check request. With partial programming, the test requirement could be carried out if the integrity / authenticity check (check memory) the last is done block to be transferred.

Id:	80126-A160
Version:	NA
Description:	The check for consistency can only produce a positive result if all logical blocks are "valid".

Id:	80126-A914
Version:	NA
Description:	Exception: If logical blocks are present (e.g., optional blocks, blocks reserved for future applications) that do not belong to the application and do not affect its functionality, these can be removed from the compatibility check. Example: Additional fonts for a display control unit are flashed in an optional block. If this block is not present or is invalid, default fonts are used. The application remains functional nevertheless. This exception must be agreed with the responsible function department for flash programming and must be documented in the ECU component specification (BT-LAH).

Id:	80126-A161
Version:	NA
Description:	The entire server structure (HW and SW) must be taken into consideration for the check.

Id:	80126-A162
Version:	NA
Description:	If the check does not occur or it fails (consistency of the server is "invalid"), the application must not be started and the server must remain in the boot loader.

Id:	80126-A163

Version:	NA
Description:	See also Section "checkProgrammingDependencies".

Id:	80126-A706
Version:	NA
Description:	3.5.5.1 Example mechanism for a consistency check

Id:	80126-A707
Version:	NA
Description:	For checking the consistency or compatibility of the application and data sets, the server can use the following mechanism, for example:

Id:	80126-A708
Version:	NA
Description:	Each logical block contains an additional version number (2-byte) that is read out by the boot loader. This 2-byte version number is located at the end of every data file (binary file, e.g., of the application or data set). The two bytes are coded as follows:

Id:	80126-A1096
Version:	NA
Description:	Figure 4 - Example consistency check

Id:	80126-A1097
Version:	NA
Description:	Figure 80126-A710 shows an example consistency check of 3 logical blocks. * In the left-hand case, all major indices are identical, and thus it must be possible to successfully perform the consistency check. * In the right-hand case, the major indices differ and the consistency check fails. At the same time, the programming status in DataIdentifier "0xF1DF-ECU Programming Information" must be updated in both cases.

Id:	80126-A1210
Version:	NA
Description:	The primary objective of downgrade protection is to ensure that the updated software versions that fix known security vulnerabilities cannot be overwritten by old software versions, in order to exploit the known security vulnerabilities contained therein.

Id:	80126-A1211
Version:	NA
Description:	Appendix E "Downgrade protection" contains the requirements for the boot loader and the flash containers for implementing a downgrade protection version number - also referred to as the "RFS version number" or "RfsV" - for ECU software. The RfsV can be increased, for example, in the event of a security mechanism update in the software. If the new software version contains no security update, the RfsV of the previous software version can be retained. If the RfsV has not been increased, then any software with the same or higher RfsV can be flashed. The old RfsV is compared with the new RfsV for this purpose, and the flashing process is completed only if the comparison is successful.

Id:	80126-A1212
Version:	NA
Description:	Application example:

Id:	80126-A1213
Version:	NA
Description:	All servers that implement FDS must implement the downgrade protection. If a Security risk analysis shows that the RFS is not necessary, a removal is thus allowed. The removal shall be documented in the BT-LAH.

Id:	80126-A1214
Version:	NA
Description:	If downgrade protection is required for an ECU, then the requirements of Appendix E "Downgrade protection" apply.

Id:	80126-A1323
Version:	NA
Description:	Boot loader data sets are not protected by the downgrade protection mechanism.

Id:	80126-A167
Version:	NA
Description:	3.6 Diagnosis container (DIAG-LAYER)

Id:	80126-A168
Version:	NA

Description:	See [Q-LAH 80128]

Id:	80126-A169
Version:	NA
Description:	3.7 Flash data container (ECU-MEM)

Id:	80126-A170
Version:	NA
Description:	See [Q-LAH 80128]

Id:	80126-A171
Version:	NA
Description:	4 Session-Management

Id:	80126-A172
Version:	NA
Description:	The session management is described in the document [Q-LAH 80124].

Id:	80126-A173
Version:	NA
Description:	For programming, the following sessions are required both in the application as well as in the boot loader: - DefaultSession - ExtendedDiagnosticSession

Id:	80126-A174
Version:	NA
Description:	In addition, the following is required in the boot loader only: - ECUProgrammingSession

Id:	80126-A175
Version:	NA
Description:	4.1 DefaultSession

Id:	80126-A176
Version:	NA
Description:	After PowerOn, a server is by definition in the DefaultSession [Q-LAH 80124].

Id:	80126-A177

Version:	NA
Description:	4.2 ExtendedDiagnosticSession

Id:	80126-A178
Version:	NA
Description:	This session is used to access programming. The following actions are performed here: - Reading out of the programming pre-conditions - Freezing and blocking of the event log entries - Blocking of the inter-server communication

Id:	80126-A179
Version:	NA
Description:	4.3 ProgrammingSession

Id:	80126-A180
Version:	NA
Description:	This provides the entire functionality for server programming.

Id:	80126-A181
Version:	NA
Description:	4.4 Diagnostic services and sessions

Id:	80126-A182
Version:	NA
Description:	The basic structure and the requirements for executing the services are described in Section "Diagnostic services".

Id:	80126-A184
Version:	NA
Description:	The table shows all the services required for programming for boot loader and application of a server.

Id:	80126-A185
Version:	NA
Description:	Table 3 (Part 1) – UDS-Services of flash programming 1) $01 = DefaultSession $02 = ProgrammingSession $03 = ExtendedDiagnosticSession

Id:	80126-A186

Version:	NA
Description:	Table 3 (Part 2) – UDS-Services of flash programming 1) This refers only to the flash-relevant DataIdentifier (e.g. Fingerprint). Writing of other DataIdentifiers (e.g. Coding) in the application is out of scope. 2) Applies only to Non-OBD ECUs.

Id:	80126-A679
Version:	NA
Description:	Legend for Table 1: "P": = Physical addressing "F": = Functional addressing "C1": = Server-specific ("conditional"), to be implemented by all CAN ECUs mandatory "C²": = Server-specific ("conditional"), if requested by function departmend (purchaser). "M" = Must be implemented ("Mandatory") "-": = Not allowed, means the server replies with a negative response and corresponding response code in the case of physical addressing. The server ignores the UDS-service in the case of functional addressing only optional. "+": = This UDS-service is only allowed after a preceding Security Access and Write Fingerprint. The fingerprint itself is only permitted after Security Access. The selection of the error code corresponds to the availability within the bootloader or within the application. Example: The bootloader does not have to know whether the command "$27 03" is allowed in the application (in a particular session).

Id:	80126-A187
Version:	NA
Description:	The UDS diagnostic services in tables 80126-A185 and 80126-A186 and the specifications described there are mandatory for the implementation.

Id:	80126-A188
Version:	NA
Description:	The server must support the UDS-diagnostic services in tables 80126-A185 and 80126-A186 both physically and functionally (as long as they are not marked with "*").

Id:	80126-A952
Version:	NA
Description:	The UDS-diagnostic services marked with "*" in tables 80126-A185 and 80126-A186 can be implemented optionally.

Id:	80126-A689
Version:	NA

Description:	The UDS-diagnostic services marked with "*" in tables 80126-A185 and 80126-A186 are not allowed to be used by clients.

Id:	80126-A189
Version:	NA
Description:	"The implementation of other UDS services and routine identifier in the boot loader, which are not listed in the tables 80126-A185 and A186-80126 (e.g. WriteMemoryByAddress, ReadMemoryByAddress, etc.) is not permitted. Exceptions must be agreed by the supplier with the client and documented in the ECU component specification (BT-LAH).

Id:	80126-A190
Version:	NA
Description:	The UDS services Communication Control (28 hex) ControlDTCSetting (85 hex) and TesterPresent (3e Hex) shalle be sent by default as functional adressed messag by the client. (Exceptions are possible, eg. As turning off the Mute during parallels flashing of a single control unit in the network).

Id:	80126-A690
Version:	NA
Description:	The UDS services DiagnosticSessionControl (10hex) CommunicationControl (28hex) ControlDTCSetting (85hex) und TesterPresent (3Ehex) serve to support the vehicle-wide diagnostic management (bus idle, deactivate/activate event log).

Id:	80126-A191
Version:	NA
Description:	Note: The execution of the individual UDS-services is dependent to some extent on the previous execution of other UDS-services or on other boundary conditions. Information about this is provided in the description of the individual services in Section "Negative Response Codes" of these requirement specifications as well as in [Q-LAH 80124]. Unless explicitly described otherwise, there are no dependencies. Especially detailed information on the meaning of the diagnostic sessions is presented in [Q-LAH 80124], specifically in Section "Status transition diagram of the diagnostic session (server)".

Id:	80126-A192
Version:	NA
Description:	4.5 Session transition

Id:	80126-A193
Version:	NA
Description:	The state transitions of the sessions in application, boot manager and boot loader are included in [Q LAH_80124-6616].

Id:	80126-A200
Version:	NA
Description:	As depcited in [Q LAH_80124-6616] and described afterwards. the state transitions havet to be implemented

Id:	80126-A201
Version:	NA
Description:	The transition into the Bootloader ECUProgrammingSession shall be reachable from the application ExtendedDiagnosticSession or Bootloader ExtendedDiagnosticSession.

Id:	80126-A947
Version:	NA
Description:	All other transitions from NonDefaultSessions from the application to the bootloader ECUProgrammingSession can be implemented optionally. It should be noted that the switching mechanisms: Routine Control (31hex), Subfunction 0xFF01-CheckProgrammingDependencies ControlDTCSettings (85hex) Communication Control (28hex) be carried out only in the Extended diagnostic session.

Id:	80126-A202
Version:	NA
Description:	The direct transit from the Application Defaultsession to the bootloader ECUProgrammingSession is not allowed. The server must quit such a request with NRC = 0x7E (SubFunctionNotSupportedInActiveDiagnosticSession) reject.

Id:	80126-A204
Version:	NA
Description:	The UDS services ECU Reset "(11 hex), Subfunction 0x01-Hard Reset" and "Diagnostics essionControl (10hex), Subfunction 0x01-Default Session" are allowed in the bootloader anytime. The only exception is described in 80126-A271..

Id:	80126-A206

Version:	NA
Description:	5 Programming sequence

Id:	80126-A207
Version:	NA
Description:	The programming sequence is divided into two phases.

Id:	80126-A208
Version:	NA
Description:	Programming Phase 1: This phase is "mandatory" and is used for programming of: ● Boot loader SW (must be specified by the supplier in agreement with the purchaser) and/or ● Application SW and/or ● Application data

Id:	80126-A210
Version:	NA
Description:	Programming Phase 1 can again be divided into three sections. These are: ● Pre-programming ● Programming execution ● Post-programming

Id:	80126-A209
Version:	NA
Description:	Programming Phase 2: This phase is optional and can be initiated immediately after programming phase 1 by the client. The actions demanded in Phase 2 are processed by the application and are thus not the task of the boot loader; consequently, they are not part of these Performance Specifications. Example: When starting from the boot loader in the new application consistency between hardware and new software is tested.

Id:	80126-A211
Version:	NA
Description:	5.1 Pre-programming

Id:	80126-A212
Version:	NA
Description:	The sequence presented in Figure VW80126-A213 shows the programming preparation.

Id:	80126-A213
Version:	NA

Description:	Figure 5 - Pre-Programming

Id:	80126-A214
Version:	NA
Description:	In the following, whenever reference is made to servers (plural), the information also applies to a single server.

Id:	80126-A215
Version:	NA
Description:	After the startup the server is in the Default Session (application or bootloader). The client requests the server to change in the Extended diagnostic session on (Step (a) in image 80126-A213) by functional adressing.

Id:	80126-A216
Version:	NA
Description:	The client determines the identification data. This step is optional and is provided as an example at this point. On principle, the following requirement applies:

Id:	80126-A217
Version:	NA
Description:	The determination of the identification data must be possible at any stage of programming. As an exception, only the sequence RequestDownload (34Hex), TransferData (36hex), and RequestTransferExit (37hex), allowable (Step (b) in figure 80126-A213).

Id:	80126-A218
Version:	NA
Description:	The client checks the programming pre-conditions of the server (Step (c) in figure 80126-A213) either by physical or functional adressing.

Id:	80126-A219
Version:	NA
Description:	The client sends a request to the server to freeze and block their event logs by functional adressing. (Step (d) in figure 80126-A213)

Id:	80126-A220
Version:	NA
Description:	The client sends a functional request to the servers to block the inter-server communication. (Step (e) in figure 80126-A213). Note: If the server contains a

	component of the engine immobilizer function, a successful engine immobilizer authentication must have been performed prior to blocking inter-server communication.

Id:	80126-A221
Version:	NA
Description:	A server that communicates directly via FlexRay discontinues its application messages on receipt of the UDS-Service CommunicationControl service (disable non-diagnostic communication) and prevents the filling of the communication slots.

Id:	80126-A222
Version:	NA
Description:	Note: Empty slots are not routed from the vehicle gateway onto connected CAN busses.

Id:	80126-A223
Version:	NA
Description:	The sequence described applies to the application and the boot loader equally, i.e., a server that is started in the boot loader (because there is no executable application vailable) behaves exactly as it does in application mode. Therefore the client does not have to differentiate when handling the servers.

Id:	80126-A224
Version:	NA
Description:	The client must always execute the pre-programming sequence first, before it puts the server in the boot loader.

Id:	80126-A225
Version:	NA
Description:	The steps of the pre-programming sequence must be implemented by the client in the manner described.

Id:	80126-A226
Version:	NA
Description:	The steps of the pre-programming sequence "Routine Control checkProgrammingPreConditions", "Control DTC Setting", and "Communication Control" are optional for the server.

Id:	80126-A228
Version:	NA
Description:	The following Figures show the behavior of client and server in detail:

Id:	80126-A229
Version:	NA
Description:	5.1.1 Switch to the ExtendedSession

Id:	80126-A230
Version:	NA
Description:	Figure 6 - Pre-Programming: Transition to ExtendedDiagnosticSession

Id:	80126-A232
Version:	NA
Description:	The server is in the DefaultSession state and is waiting for requests from the client.

Id:	80126-A234
Version:	NA
Description:	The servers deliver the response and perform the switch. In its reply, each server must transfer the timing (P2 and P2*) for the ExtendedSession to the client. The timing must be dimensioned such that any server programming and the required reversion to the boot loader is taken into consideration. Network latency times are taken into consideration by the client. On the topic of UDS-Timing, see also Section "UDS Layer".

Id:	80126-A235
Version:	NA
Description:	5.1.2 Determination of the programming pre-conditions

Id:	80126-A236
Version:	NA
Description:	Figure 7 - Pre-Programming: Determination of the programming pre-conditions

Id:	80126-A238
Version:	NA
Description:	The servers are in the ExtendedDiagnosticSession status and are waiting for requests from the client. a) The client sends a request for communication of the

	programming pre-conditions. b) The servers deliver the status of the programming pre-conditions. For servers that do not deliver an "empty list", programming is only possible after eliminating the "programming obstacles".

Id:	80126-A239
Version:	NA
Description:	5.1.3 Preparation for programming

Id:	80126-A240
Version:	NA
Description:	Figure 8 - Pre-Programming: Preparation of programming

Id:	80126-A242
Version:	NA
Description:	a) The servers are in the ExtendedDiagnosticSession status and are waiting for requests from the client. b) The client sends a functionally addressed request without response (SuppressPositiveResponse Bit = 1) to freeze and block the event log entries. The servers perform the blocking and deliver a response message only if an error occurs. c) The client sends a functionally addressed request without response (SuppressPositiveResponse Bit = 1) to block inter-server communication. The servers perform the blocking and deliver a response message only if an error occurs.

Id:	80126-A243
Version:	NA
Description:	Note: Since the client has not requested any responses, it must observe the time P3Client_Func until the next request. In return, a server must guarantee the execution of the required action within this time. If it cannot do this, it must report this situation to the client with a negative response.

Id:	80126-A244
Version:	NA
Description:	5.2 Programming execution

Id:	80126-A245
Version:	NA
Description:	The following depicts the programming sequence.

Id:	80126-A246

Version:	NA
Description:	Figure 9 - Programming execution

Id:	80126-A1187
Version:	NA
Description:	The sequence RequestDownload, TransferData, RequestTransferExit, and RoutineControl/CheckMemory will be discussed below. Repetitions within this sequence (e.g., multiple TransferData requests) are part of the sequence.

Id:	80126-A1188
Version:	NA
Description:	A sequence begins with a positive response to a RequestDownload request.

Id:	80126-A247
Version:	NA
Description:	ad. a) Switch to the ECUProgrammingSession: To perform the programming, the client prompts the server to switch to the ECUProgrammingSession. Programming takes place in the boot loader of the server.

Id:	80126-A248
Version:	NA
Description:	Following an accepted request to switch to the ProgrammingSession, the application must make all preparations to guarantee trouble-free programming operation. In this process it must end all routines and functions that influence programming and ensure that the server is in a safe state. See Section "Programming pre-conditions (checkProgrammingPreConditions)".

Id:	80126-A249
Version:	NA
Description:	The switch to the boot loader must then be initiated with a restart (see Section "DiagnosticSessionControl (10 hex)").

Id:	80126-A786
Version:	NA
Description:	Additional requirements for the switch to the ProgrammingSession can be found in Section "Switch to the ProgrammingSession".

Id:	80126-A250
Version:	NA

Description:	Ad. b) Access authorization: The client must authenticate itself for the purposes of programming on the server. Without valid access authorization, neither the client nor the server may continue programming. See Section "Access authorization (Seed and Key)" and "ReadDataByIdentifier (Fingerprint)".

Id:	80126-A251
Version:	NA
Description:	The access authorization (SecurityAccess) must be implemented according to [Q-LAH 80124], and Appendix A.

Id:	80126-A252
Version:	NA
Description:	An activation must be canceled by each session switch. This also applies to a session-change in the currently active session.

Id:	80126-A253
Version:	NA
Description:	An activation must be canceled by each ECUReset.

Id:	80126-A254
Version:	NA
Description:	An activation must be canceled by each TesterPresent Timeout (S3Server).

Id:	80126-A255
Version:	NA
Description:	Ad. c) Fingerprint: This is used in order to document in the server who has programmed the relevant logical block and when. See Section "WriteDataByIdentifier (2E hex)".

Id:	80126-A256
Version:	NA
Description:	Writing of the fingerprint is mandatory. Without a valid fingerprint, neither the client nor the server is allowed to accept a continuation of programming.

Id:	80126-A257
Version:	NA
Description:	For additional requirements, see Sections "Tester serial number" and "Programming date".

Id:	80126-A1068
Version:	NA
Description:	Ad. d) Partial Servers that support the Partial Programming must read before starting the update programming the identifier "0xF1AB-VW Logical Software Block Version" once from the control unit (This is part of the Java Job "SinglJob_CheckOwnIdent" and not part of the Flash Job). Before writing a logical block the client checks whether the re-written logical block from the flash container is already stored on the server. For this, the 'VW Logical Software Block' Version is read from the Flashcontainer and compared with the value returnd from ECU via the Identifier 0xF1AB. If they are equal, the Block won't be updated and the next Block will be compared. For additional verifcation security there extists an optional RoutineControl Service - Verify_partial_software_checksum.

Id:	80126-A258
Version:	NA
Description:	Ad. e) Load erase routine: In the event that the routines required to erase the memory that is to be programmed are not a fixed part of the boot loader, these must be loaded subsequently. There is also the option here to load a complete flash driver containing the erase and write routines. See also "Transfer memory block".

Id:	80126-A259
Version:	NA
Description:	Ad. f) Check erase routine:

Id:	80126-A260
Version:	NA
Description:	If an erase routine or a flash driver are loaded subsequently, these must also be checked. See "Check memory block".

Id:	80126-A261
Version:	NA
Description:	Ad. g) Erase memory block: This step enables the memory area of the logical block subsequently intended for programming to be erased. See Section "erase-Memory".

Id:	80126-A262
Version:	NA

Description:	When the erase routine is executed, the server must set the addressed logical block to "invalid" just before the erase operation.

Id:	80126-A263
Version:	NA
Description:	When the erase routine is executed, the server must set the server consistency to "invalid" just before the erase operation (0xF1DF = 0x44).

Id:	80126-A264
Version:	NA
Description:	If the logical block is set to "invalid" and, thus, the server is inconsistent, it must ensured that the start of an application is not enabled. See Section "Compatibility/consistency check (checkProgrammingDependencies)".

Id:	80126-A694
Version:	NA
Description:	When the erase routine is executed, the server must set the SoftwareVersion of the addressed logical block "0xF1AB-VW Logical Software Block Version" to the initial value "----" (0x2d2d2d2d) just before the erase operation.

Id:	80126-A873
Version:	NA
Description:	When the erase routine is executed, the server must update the fingerprint (0xF15B-Fingerprint And Programming Date Of Logical Software Blocks) just before the erase operation.

Id:	80126-A1334
Version:	NA
Description:	Writing services (0x2E) are not allowed after the EraseMemory routine until the flash sequence (services 0x34, 0x36, 0x37) or until their end. Servers have to answer write attempts with NRC = 0x22 (ConditionsNotCorrect).

Id:	80126-A265
Version:	NA
Description:	Ad. h) Load write routine: In the event that the routines required for programming are not a fixed part of the boot loader, these must be loaded subsequently. This step is optional. See "Transfer memory block".

Id:	80126-A266

Version:	NA
Description:	Ad. i) Check write routine:

Id:	80126-A267
Version:	NA
Description:	If a write routine is loaded subsequently, it must be checked.

Id:	80126-A268
Version:	NA
Description:	This step is optional, if no write routine is loaded subsequently. See "Check memory block".

Id:	80126-A269
Version:	NA
Description:	i) Transfer memory block: The RequestDownload, TransferData, and TransferExit sequence (see Sections "RequestDownload (34 hex)", "TransferData (36 hex)", "RequestTransferExit (37 hex)") is used to transfer memory data (for Flash, EEPROM, or RAM memory) of a logical block (see Sections "Software distribution" and "Block segmentation").

Id:	80126-A270
Version:	NA
Description:	The following steps are mandatory when transferring data to the server: 1. The client sends the service 0x34 for the first segment. 2. The server responds. 3. The client sends the UDS-service 0x36 in order to download the segment. 4. The server responds. 5. The client repeats the UDS-service 0x36 until the complete segment that was described by UDS-service 0x34 has been downloaded. 6. The client sends the service 0x37 in order to end downloading of the segment. 7. The server responds. 8. The client immediately sends the UDS-service 0x34 in order to download the next segment, if one still exists for the logical block. Steps 1–7 are performed for all segments of a logical block.

Id:	80126-A271
Version:	NA
Description:	No other services other than TesterPresent are permitted during the error-free execution of the RequestDownload, TransferData, and TransferExit services. The server rejects a non-allowed request with NRC 0x24 (RequestSequenceError). Whe NRC = 0x24 is not defined for the service, the request must be rejected with NRC = 0x22 (conditionsNotCorrect).

Id:	80126-A1155
Version:	NA
Description:	Immediately after the positive response of the UDS-Service Request download from the server is a repetion of the Service Request Download client allowed if there was no TransferData or TransferExit request transmitted in the meanwhile (e.g. When the is a new tester-request because the positive response of the ECUs got lost).

Id:	80126-A272
Version:	NA
Description:	Ad. k) Check memory block: This check aims to ensure that the data to be programmed (logical block) have been transferred without errors and stored correctly in the memory. To this end, test methods serve to determine the data integrity and authenticity and, consequently, the validity. For additional requirements, see Section "Integrity/Authenticity Check (checkMemory)".

Id:	80126-A273
Version:	NA
Description:	After executing the check routine and receiving a positive result, the server must set the affected logical block to "valid".

Id:	80126-A691
Version:	NA
Description:	Starting from this time, the software version of the logical block "0xF1AB-VW Logical Software Block Version" must output the current programmed software version of the block.

Id:	80126-A692
Version:	NA
Description:	Starting from this time, the validity information of the logical block within the dataidentifier "0xF15B-Fingerprint And Programming Date Of Logical Software Blocks" must return the information block is "valid".

Id:	80126-A274
Version:	NA
Description:	Only the check routine may set the relevant logical block in 0xF15B to "valid".

Id:	80126-A275
Version:	NA

Description:	Ad. I) Check programming: The purpose of this check is to ensure the compatibility/consistency between SW/SW and SW/HW. See Section "Compatibility/consistency check (checkProgrammingDependencies)".

Id:	80126-A276
Version:	NA
Description:	After executing the check with a positive result, the server must set the consistency of the server to "valid" (0xF1DF = 0x40).

Id:	80126-A277
Version:	NA
Description:	Only the check routine may set the consistency of the server to "valid".

Id:	80126-A278
Version:	NA
Description:	The consistency of the server must be updated simultaneously with the internal state administration in the diagnostic coding [Q-LAH 80125], as well.

Id:	80126-A923
Version:	NA
Description:	After a successful consistency check, the control unit must update the values for the DataIdentifier "0xF189-VW Application Software Version Number" and "0xF187-VW Spare Part Number".

Id:	80126-A279
Version:	NA
Description:	To ensure that a newly programmed application can implement the required checks/reorganizations of the data structures (EEPROM data, taught data, adaptive data, coding, etc.) with respect to the content and/or address location, the following requirements must be fulfilled:

Id:	80126-A280
Version:	NA
Description:	After executing the check with a positive result, the server must remember that it has been newly programmed. Reprogramming must be detected only if at least 1 bit has been deleted or written in the flashing process [A: 80126-A315].

Id:	80126-A1186

Version:	NA
Description:	The execution of the CheckProgrammingDependencies routine may be omitted only if the "preliminary erasing of blocks" option is used (see section 6.7.2.4).

Id:	80126-A281
Version:	NA
Description:	See Sections "Compatibility/consistency check (checkProgrammingDependencies)" and "checkProgrammingDependencies".

Id:	80126-A286
Version:	NA
Description:	Ad. m) After the consistency check (no matter if successful or not), the client must initiate a server restart using the service ECUReset (hardReset). Once this is done, the server is in the “new” application, if programming was successful. If programming was unsuccessful, the server is still in the boot loader.

Id:	80126-A788
Version:	NA
Description:	Additional requirements for exiting the boot loader can be found in the Section "Exiting the ECUProgrammingSession".

Id:	80126-A938
Version:	NA
Description:	Requirement: 1. RequestDownloads to addresses which are outside of the block must acknowledge with NRC = 0x31 (RequestOutOfRange) to reject. 2. Erase commands for one of the following blocks need first to reject with NRC = 0x24 (sequenceError) before the checkMemory-Command can return with positve result. If the following Block is a driver, the RequestDownload- Request for this driver must also reject with NRC = 0x24 (sequenceError).

Id:	80126-A974
Version:	NA
Description:	Before the server accepts EraseMemory-Requests or RequestDownload-Requests for the peristant memory, he must perform the check for validity of a reloadable driver. Only after positive result of this test the erase or write operations are permitted on the persistent store.

Id:	80126-A975
Version:	NA

Description:	The memory (RAM) area of a reloadable driver must be erased immediately when the check memory routine of the driver has been answered with negative result.

Id:	80126-A287
Version:	NA
Description:	5.2.1 Switch to the ProgrammingSession

Id:	80126-A288
Version:	NA
Description:	Figure 10 - Programming execution: Switch to the ECUProgrammingSession

Id:	80126-A787
Version:	NA
Description:	The following description of the individual steps refers to the representation in Figure 80126-A288.

Id:	80126-A289
Version:	NA
Description:	Ad. a) The server is in the ApplicationExtendedDiagnosticsSession state and is waiting for requests from the client.

Id:	80126-A290
Version:	NA
Description:	Ad. b) The client requests the servers to be programmed to change into the ProgrammingSession.

Id:	80126-A291
Version:	NA
Description:	With the DiagnosticSessionControl (ProgrammingSession) request, a server must check immediately prior to changing to the ProgrammingSession to ensure that no programming-conditions exist that would not allow programming.

Id:	80126-A292
Version:	NA
Description:	Ad. c) Before the restart, the following requirement exists for the server, even if it is not likely to exceed the time P2CAN_Server:

Id:	80126-A293

Version:	NA
Description:	Independently of the normal ResponsePending behavior, a server must respond with a ResponsePending (NRC 0x78) immediately prior to the restart for switching to the ECUProgrammingSession.

Id:	80126-A294
Version:	NA
Description:	Ad. d) In order to minimize the period for which the server cannot communicate, the following requirement exists:

Id:	80126-A295
Version:	NA
Description:	The preparatory measures and sequences in the server must be organized such that the restart and, thus, the switch to the boot loader will be performed as quickly as possible.

Id:	80126-A296
Version:	NA
Description:	Ad. e) After the restart, the server is in the boot manager; it registers the programming request there and starts the boot loader.

Id:	80126-A297
Version:	NA
Description:	Ad. f) As its first action, the boot loader supplies the response for the completed switch into the ECUProgrammingSession and must thus fulfill the following requirement:

Id:	80126-A298
Version:	NA
Description:	The server must provide the timing (P2CAN_Server, P2*CAN_Server) of the ProgrammingSession, with the positive response to the DiagnosticSessionControl (programmingSession) request.

Id:	80126-A299
Version:	NA
Description:	Ad. g) After the positive response, the server is in the ProgrammingSession of the boot loader and is ready for programming.

Id:	80126-A300

Version:	NA
Description:	The server must start the S3 timer for the TesterPresent handling immediately on the switch to the ProgrammingSession of the boot loader.

Id:	80126-A301
Version:	NA
Description:	5.2.2 Exiting the ProgrammingSession

Id:	80126-A302
Version:	NA
Description:	Figure 11 - Programming execution: Exiting the ECUProgrammingSession

Id:	80126-A789
Version:	NA
Description:	The following descriptions refer to the representations in Figure 80126-A302.

Id:	80126-A303
Version:	NA
Description:	Ad. a) The server is in the ProgrammingSession state and is waiting for requests from the client.

Id:	80126-A304
Version:	NA
Description:	The ECUProgrammingSession can only be exited via S3 timeout, ECUReset, DiagnosticSessionControl (defaultSession), or PowerOn-Reset.

Id:	80126-A305
Version:	NA
Description:	After an S3 timeout, ECUReset, or DiagnosticSessionControl (defaultSession) request, the server must perform a restart.

Id:	80126-A306
Version:	NA
Description:	Ad. b) The client requests the server to quit the ECUProgrammingSession:

Id:	80126-A307
Version:	NA

Description:	A server must remember the reason (ECUReset (hardReset) or DiagnosticSessionControl (defaultSession)) why the ECUProgrammingSession is exited. This is necessary so that the correct positive response can be given after the restart.

Id:	80126-A308
Version:	NA
Description:	Ad. c) Before the restart, the following requirement exists for the server, even if it is not likely to exceed the time P2CAN_Server:

Id:	80126-A309
Version:	NA
Description:	Independently of the normal ResponsePending behavior, a server must respond with a negative response NRC=0x78 (RequestCorrectlyReceived-ResponsePending) immediately prior to the restart for exiting the ECUProgrammingSession.

Id:	80126-A310
Version:	NA
Description:	Ad. d) In order to minimize the period for which the server cannot communicate, the following requirement exists:

Id:	80126-A311
Version:	NA
Description:	The preparatory measures and sequences in the server must be organized such that the restart and thus the exiting of the ProgrammingSession will be performed as quick as possible.

Id:	80126-A312
Version:	NA
Description:	The preparatory measures and sequences in the server must be organized such that the restart and thus the exiting of the ProgrammingSession will be performed as quick as possible.

Id:	80126-A314
Version:	NA
Description:	Ad. f, g) Once a newly programmed application has implemented the required checks/reorganization measures for the data structures (EEPROM data, taught data, adaptive data, coding etc.) with respect to the content and/or address location, the following requirements must be fulfilled:

Id:	80126-A315
Version:	NA
Description:	Following execution of the self-test and the resulting consequences (event log entry, writing of default values, etc.), the flag indicating the new programming must be removed.

Id:	80126-A313
Version:	NA
Description:	Afterward, the server (application or boot loader) supplies the response for the exiting of the ECUProgrammingSession.

Id:	80126-A870
Version:	NA
Description:	If the server cannot send the positive response for the exiting of the ECUProgrammingSession within time P2*CAN_Server, the server must sustain the communication by sending further cyclic requestCorrectlyReceivedResponsePending (NRC 0x78) responses.

Id:	80126-A866
Version:	NA
Description:	After the positive response for the exiting of the ProgrammingSession is sent, the control unit must immediately be able to receive new requests.

Id:	80126-A316
Version:	NA
Description:	5.3 Post-programming

Id:	80126-A317
Version:	NA
Description:	Afterward, the server (application or boot loader) supplies the response for the exiting of the ECUProgrammingSession.

Id:	80126-A318
Version:	NA
Description:	Figure 12 - Post-Programming

Id:	80126-A320
Version:	NA

Description:	The following points describe the individual steps in Figure 80126-A318. a) The client requests the servers by functional adressing for transition into Extended diagnostic session. b) (Optional) The client requests the server by physical or functional adressing mode to lock the event memory (is z. B. the parallels flashing of several ECUs used). c) (Optional) The client requests the server by physical or functional adressing mode to stop the inter-server communication (e.g. is used in parallel flashing of multiple ECUs). d) The client sends a functional request to the server to start the inter-server communication. e) The client sends a functional request to the server to release the event logs.

Id:	80126-A321
Version:	NA
Description:	All programmable servers of a vehicle that support diagnosis must also support the postprogramming sequence.

Id:	80126-A322
Version:	NA
Description:	The client must execute the post-programming sequence, when no further server remains to be programmed.

Id:	80126-A323
Version:	NA
Description:	The steps of the post-programming "CommunicationControl" and "ControlDTCSetting" sequence are optional for the server. They may be omitted or performed in a different order but shall never influence the coding process

Id:	80126-A324
Version:	NA
Description:	The post-programming sequence steps must be implemented by the client in the manner described.

Id:	80126-A325
Version:	NA
Description:	The sequence described applies to the application and the boot loader equally, i.e., a server that is started in the boot loader because there is no executable application available must behave exactly like an application.

Id:	80126-A326
Version:	NA

Description:	5.4 State administration for programming

Id:	80126-A327
Version:	NA
Description:	This Section describes the coordination of programming between the boot loader and the application, as well as the steps required to update a check structure for programming.

Id:	80126-A329
Version:	NA
Description:	The following statuses must be considered for coordination of the programming between the boot loader and application: ● A functional application (C1) is (not) present "validity" status* ● A programming request (C2) exists "programming request" status** ● Server was successfully programmed "programmed" flag*** * There are two status values for this status: - Consistent: The server can be used without restrictions for operation in the vehicle. - Inconsistent: The server must not be used for operation in the vehicle. ** The server has executed the pre-programming sequence and received a request to switch to the ProgrammingSession. *** The server has successfully performed the consistency check (checkProgrammingDependencies) at the end of the programming and at least 1 bit has been deleted or written in the flashing process.

Id:	80126-A330
Version:	NA
Description:	In the following description it is assumed that the status of the server is "consistent" (C1) and without "programming request" (C2). The "programmed" flag for the new programming status is not set.

Id:	80126-A808
Version:	NA
Description:	Appendix B contains Figure 80126-A793, which describes the state administration during the flash sequence.

Id:	80126-A331
Version:	NA
Description:	For the referencing used below, see Section "Programming execution", Figure 80126-A246.

Id:	80126-A332

Version:	NA
Description:	Ad. a) Transfer of the programming request, application → boot loader ● The client sends the programming request (DiagnosticSessionControl(ProgrammingSession) ● The application then sets the "Programming request" status (C2) and initiates the required server restart. ● The boot manager (see Section "Session Transition") registers the "programming request" and starts the boot loader instead of the still functional application. ● After the boot loader starts, it cancels the "programming request".

Id:	80126-A333
Version:	NA
Description:	Ad. c) Storage of the serial number and programming date The boot loader receives the serial number (RepairShopCodeOrSerialNumber) and the programming date (ProgrammingDate) and stores these temporarily (in volatile memory).

Id:	80126-A334
Version:	NA
Description:	Ad. f) Update of the check structure prior to programming The erasing of a logical block or of the entire server software renders the server non-functional. Accordingly the state of the server and the data of the check structure must be updated when the erase process starts. The order described for this must be observed. ● For the boot manager, the lack of a functional software must be stored for a possible interruption or faulty programming. The "Validity" status (C1) must be set to "inconsistent". ● The programming counter of the relevant logical block must be incremented. (If a blocking value exists, it must be checked.) ● The temporarily stored serial number of the client must be stored in non-volatile memory for the logical block involved. ● The temporarily stored programming date must be stored in non-volatile memory for the logical block involved.

Id:	80126-A335
Version:	NA
Description:	Ad. k) After a successful consistency check, the boot loader must set the "programmed" flag if at least 1 bit has been deleted or written in the flashing process.. If the check was not successful, the "programmed" flag must be set to false. EB tresos Bootloader calls callbacks allowing the management of this data on Customer side.

Id:	80126-A790
Version:	NA

Description:	After a successful consistency check, the boot loader must set the "Validity" status (C1) to "consistent". If the check was not successful, the "Validity" status must be set to "inconsistent". EB tresos Bootloader calls callbacks allowing the management of this data on Customer side.

Id:	80126-A791
Version:	NA
Description:	In both cases, the server must wait for the restart request by the client.

Id:	80126-A336
Version:	NA
Description:	Ad. m) Update of the check structure after programming When the restart request is received and the ResponsePending response is issued, the boot loader performs the restart and the boot manager then checks the states.

Id:	80126-A809
Version:	NA
Description:	If the "Validity" status (C1) is "inconsistent", the application has not been successfully programmed and the boot manager starts the boot loader in the DefaultSession.

Id:	80126-A810
Version:	NA
Description:	If the application has been consistently programmed (C1 = consistent), the boot manager starts the application in the DefaultSession.

Id:	80126-A811
Version:	NA
Description:	When the application starts, the "programmed" flag is evaluated. If the flag indicates a new programming of the application, it first runs through an initialization phase.

Id:	80126-A812
Version:	NA
Description:	During the initialization phase, the application performs the required checks/reorganizations of the data structures (EEPROM data, learned data, adaptive data, coding, etc.) with respect to content and/or address position.

Id:	80126-A813

Version:	NA
Description:	The flag for the new programming is then canceled and the data (see [VW 80125]) of the software that was just programmed are transferred to the non-volatile server memory provided for this purpose.

Id:	80126-A814
Version:	NA
Description:	The flag for the new programming is then canceled and the data (see [VW 80125]) of the software that was just programmed are transferred to the non-volatile server memory provided for this purpose.

Id:	80126-A815
Version:	NA
Description:	If the duration of the initialization phase is longer than the time P2*CAN_Server returned in the SessionControl ProgrammingSession, the application must extend the response time by sending further ResponsePending messages.

Id:	80126-A337
Version:	NA
Description:	6 Diagnostic Services

Id:	80126-A338
Version:	NA
Description:	The services, the positive and negative responses, their basic static structure, and the dynamic behavior must be implemented according to Q-LAH 80124] and [VW-ISO-TP].

Id:	80126-A339
Version:	NA
Description:	The coordination of the programming between the boot loader and the application, as well as the necessary steps to update the programming documentation (forwarding of the programming request, acceptance of the programming date and tester serial number, incrementing of the programming counter, etc.), are described in Section "Documentation of programming" and are not mentioned in the following sections.

Id:	80126-A340
Version:	NA

Description:	Each service has a separate "Negative response" section in which, in addition to the requirements for the behavior in the event of error, the general error behavior must always be noted (see Section "Error general behavior").

Id:	80126-A341
Version:	NA
Description:	6.1 DiagnosticSessionControl (10 hex)

Id:	80126-A345
Version:	NA
Description:	The services, the positive and negative responses, their basic static structure, and the dynamic behavior must be implemented according to Q-LAH 80124] and [VW-ISO-TP].

Id:	80126-A342
Version:	NA
Description:	6.1.1 Request

Id:	80126-A343
Version:	NA
Description:	The following parameters must be implemented for the request.

Id:	80126-A344
Version:	NA
Description:	Table 4 - DiagnosticSessionControl, diagnosticSessionType

Id:	80126-A346
Version:	NA
Description:	6.1.2 Positive response

Id:	80126-A347
Version:	NA
Description:	The response is to be implemented with the following parameters.

Id:	80126-A348
Version:	NA
Description:	Table 5 - DiagnosticSessionControl, sessionParameterRecord

Id:	80126-A1163
Version:	NA
Description:	After a positive response for the transition into the bootloader or exit the bootloader all internal actions (eg. B. writes to non-volatile storage) must be completed before sending the positive response, so that the server is fully operational and able to communicate.

Id:	80126-A350
Version:	NA
Description:	6.1.3 Negative response

Id:	80126-A351
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A352
Version:	NA
Description:	ConditionsNotCorrect

Id:	80126-A804
Version:	NA
Description:	ConditionsNotCorrect

Id:	80126-A353
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A354
Version:	NA
Description:	6.2 ECUReset (11 hex)

Id:	80126-A358
Version:	NA
Description:	The UDS-Service ECUReset (11hex) must be implemented according to [VW-ISO-TP], [Q-LAH 80124] and with the required parameters.

Id:	80126-A355
Version:	NA

Description:	6.2.1 Request

Id:	80126-A356
Version:	NA
Description:	The request must be implemented with the following parameters.

Id:	80126-A357
Version:	NA
Description:	Table 7 - ECUReset, resetType

Id:	80126-A695
Version:	NA
Description:	The coding values, adaptation values, and parameterization values must be maintained after flashing.

Id:	80126-A696
Version:	NA
Description:	If 80126-A695 is not possible due to technical circumstances (incompatible coding layout, different amount of parameterization data, different value ranges, etc.), the control unit must behave like a non-coded/parameterized control unit and apply appropriate identification.

Id:	80126-A921
Version:	NA
Description:	TheDataIdentifier 0xF15A-Fingerprint is only used for the documentation of the updated (flashed) logical blocks.

Id:	80126-A359
Version:	NA
Description:	6.2.2 Positive response

Id:	80126-A715
Version:	NA
Description:	In addition to the Standards named in VW80126-A360, the additional requirements set forth in VW80126-A307 and VW80126-A309 apply to the response behavior of the ECUReset in the boot loader.

Id:	80126-A982

Version:	NA
Description:	After a positive ECUreset-response to exit the bootloader, an additional update is not allowed during startup or afterrun. All writes to non-volatile storage must be completed before sending the positive response, so that the server is immediately fully functional and able to communicate.

Id:	80126-A361
Version:	NA
Description:	6.2.3 Negative response

Id:	80126-A362
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A363
Version:	NA
Description:	ConditionsNotCorrect

Id:	80126-A364
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A365
Version:	NA
Description:	6.3 ReadDataByIdentifier (22 hex)

Id:	80126-A368
Version:	NA
Description:	The UDS-Service ReadDataByIdentifier (22hex) must be implemented according to [VW-ISO-TP] and [Q-LAH 80124].

Id:	80126-A366
Version:	NA
Description:	The "DataIdentifiers" according to [Q-LAH 80125] that are relevant for the programming must be implemented..

Id:	80126-A367
Version:	NA

Description:	6.3.1 Request

Id:	80126-A922
Version:	NA
Description:	The maximum number of supportive identifier for each request also applies in the bootloader and is defined in [Q LAH_80124-2252].

Id:	80126-A371
Version:	NA
Description:	6.3.3 Negative response

Id:	80126-A372
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A373
Version:	NA
Description:	ConditionsNotCorrect

Id:	80126-A816
Version:	NA
Description:	NRC31

Id:	80126-A817
Version:	NA
Description:	NRC31

Id:	80126-A818
Version:	NA
Description:	NRC33

Id:	80126-A374
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A375
Version:	NA

Description:	6.3.3 0xF15B-Fingerprint And Programming Date Of Logical Software Blocks

Id:	80126-A376
Version:	NA
Description:	6.3.3.1 Request

Id:	80126-A377
Version:	NA
Description:	The following DataIdentifier is defined for reading out the fingerprint(s):

Id:	80126-A379
Version:	NA
Description:	Table 10 - ReadDataByIdentifier, Fingerprint, DataIdentifier

Id:	80126-A380
Version:	NA
Description:	Data Identifier " fingerprint " must be implemented to read the or the fingerprint of a programmable server .

Id:	80126-A381
Version:	NA
Description:	6.3.3.2 Positive Response

Id:	80126-A382
Version:	NA
Description:	The positive response is defined as follows:

Id:	80126-A383
Version:	NA
Description:	Table 11 - ReadDataByIdentifier, Fingerprint, response definition

Id:	80126-A384
Version:	NA
Description:	The parameters of the dataRecord for the fingerprint are defined as follows:

Id:	80126-A385
Version:	NA
Description:	Table 12 - ReadDataByIdentifier, Fingerprint, dataRecord

Id:	80126-A714
Version:	NA
Description:	The ReadDataByIdentifier service must output the value 000101 (corresponds to 01.01.00) as the initial value for the programmingDate parameter.

Id:	80126-A1172
Version:	NA
Description:	The programmingState 0xFF must not be used as a debug value.

Id:	80126-A1173
Version:	NA
Description:	As soon as a block has been completely erased and is immediately rewritable, the programmingState must have the value 0xFF.

Id:	80126-A386
Version:	NA
Description:	For reading the fingerprint(s) of a server for programming, the response must be implemented in the manner described.

Id:	80126-A387
Version:	NA
Description:	Note: On writing the fingerprint (2E F1 5A), the combination of the date and tester serial number is transferred only once. On reading the fingerprint (22 F1 5B), a list of n combinations of date, tester serial number, and programming status is provided.

Id:	80126-A388
Version:	NA
Description:	6.3.4.3 Negative response

Id:	80126-A389
Version:	NA
Description:	The error behavior Section "Negative response" must be observed.

Id:	80126-A390
Version:	NA
Description:	6.4 SecurityAccess (27 hex)

Id:	80126-A395
Version:	NA
Description:	The UDS-Service SecurityAccess (27hex) must be implemented according to [VW-ISO-TP], [Q-LAH 80124], and Appendix A, and with the required parameters.

Id:	80126-A391
Version:	NA
Description:	6.4.1 Request

Id:	80126-A392
Version:	NA
Description:	The request must be implemented with the following parameters.

Id:	80126-A394
Version:	NA
Description:	Table 13 - SecurityAccess, SecurityLevel

Id:	80126-A396
Version:	NA
Description:	The use of "securityAccessDataRecord" is not permitted.

Id:	80126-A399
Version:	NA
Description:	6.4.3 Negative response

Id:	80126-A400
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A401
Version:	NA
Description:	ConditionsNotCorrect

Id:	80126-A819
Version:	NA
Description:	requestSequenceError

Id:	80126-A820
Version:	NA
Description:	invalidKey

Id:	80126-A1112
Version:	NA
Description:	ExceededNumberOfAttemps

Id:	80126-A821
Version:	NA
Description:	requiredTimeDelayNotExpired

Id:	80126-A402
Version:	NA
Description:	Note: By default, an attempt counter for 2 attempts and a blocking time (after the three attempts) of 10 minutes must be realized.

Id:	80126-A403
Version:	NA
Description:	The information that a blocking time has not yet expired must be retained on a switch from the boot loader to the application. If the boot loader is exited before the blocking time expires, the blocking time will be restarted in the application.

Id:	80126-A404
Version:	NA
Description:	Figure 13 - Blocking Time in Application and Bootloader

Id:	80126-A405
Version:	NA
Description:	The blocking times of boot loader and application must be identical.

Id:	80126-A406
Version:	NA
Description:	The information that a blocking time for programming is active should be deleted when the blocking time has expired, either in the bootloader or in the application. The expiry of the blocking time must be possible in parallel to the ECUProgrammingSession either in the Default Session and the ExtendedDiagnosticSession (in the boot loader as well as in the application).

Id:	80126-A407
Version:	NA
Description:	A programming pre-condition that prevents a switch from the application to the boot loader must be implemented in every server, if a blocking time is active for the security access of the programming.

Id:	80126-A408
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A409
Version:	NA
Description:	6.5 CommunicationControl (28 hex)

Id:	80126-A414
Version:	NA
Description:	The UDS-Service CommunicationControl (28hex) must be implemented according to [VW-ISO-TP], [Q-LAH 80124], and with the required parameters.

Id:	80126-A410
Version:	NA
Description:	6.5.1 Request

Id:	80126-A411
Version:	NA
Description:	The following parameters must be implemented for the request.

Id:	80126-A412
Version:	NA
Description:	Table 15 - CommunicationControl, controlType

Id:	80126-A413
Version:	NA
Description:	Table 16 - CommunicationControl, communicationType

Id:	80126-A417
Version:	NA

Description:	6.1.2 Negative response

Id:	80126-A418
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A419
Version:	NA
Description:	ConditionsNotCorrect

Id:	80126-A822
Version:	NA
Description:	ConditionsNotCorrect

Id:	80126-A823
Version:	NA
Description:	requestOutOfRange

Id:	80126-A420
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A421
Version:	NA
Description:	6.6 WriteDataByIdentifier (2Ehex)

Id:	80126-A422
Version:	NA
Description:	6.6.1 0xF15A-Fingerprint

Id:	80126-A423
Version:	NA
Description:	The service described here serves to transmit the fingerprint at the start of programming.

Id:	80126-A424
Version:	NA

Description:	It must be ensured that the WriteDataByIdentifier service (Fingerprint) is executed only after successful access authorization.

Id:	80126-A425
Version:	NA
Description:	A plausibility check of the test serial number or the date is not required.

Id:	80126-A427
Version:	NA
Description:	6.6.1.1 Request

Id:	80126-A428
Version:	NA
Description:	The following DataIdentifier is defined for writing the fingerprint:

Id:	80126-A429
Version:	NA
Description:	Table 18 - WriteDataByIdentifier, Fingerprint, DataIdentifier

Id:	80126-A430
Version:	NA
Description:	The structure of the request is defined as follows:

Id:	80126-A431
Version:	NA
Description:	Table 19 - WriteDataByIdentifier, Fingerprint, request definition

Id:	80126-A432
Version:	NA
Description:	The parameters of the dataRecord for the fingerprint are defined as follows:

Id:	80126-A433
Version:	NA
Description:	dataRecord format

Id:	80126-A434
Version:	NA

Description:	dataRecord format

Id:	80126-A435
Version:	NA
Description:	6.6.1.2 Positive Response

Id:	80126-A436
Version:	NA
Description:	The structure is defined as follows:

Id:	80126-A437
Version:	NA
Description:	Table 21 - WriteDataByIdentifier, Fingerprint, response definition

Id:	80126-A438
Version:	NA
Description:	After a transfer of the fingerprint to the server to be programmed, the server must supply a positive response in the manner described

Id:	80126-A439
Version:	NA
Description:	6.6.1.3 Negative Response

Id:	80126-A440
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A441
Version:	NA
Description:	ConditionsNotCorrect

Id:	80126-A845
Version:	NA
Description:	requestOutOfRange

Id:	80126-A846
Version:	NA

Description:	securityAccessDenied

Id:	80126-A847
Version:	NA
Description:	generalProgrammingFailure

Id:	80126-A442
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A443
Version:	NA
Description:	6.6.2 0x0410-Bootloader TP Blocksize

Id:	80126-A444
Version:	NA
Description:	The service described here is used to parameterize the CAN ISO transport protocol [VW-ISO-TP] in the bootloader. Control units that have no direct connection to a CAN bus (e.g., FlexRay control units) do not require this service.

Id:	80126-A959
Version:	NA
Description:	The DataIdentifier shall be written is in all Non Default Session without precondition (e.g. no fingerprint), because the value is not overwritten in the server and only temporarily valid during the active session. He must remain in the boot loader when you change the application so that the new value can be used in the ECUProgrammingSession. Only after a relapse in the default session, ECUReset or terminal change the default value must be assumed again.

Id:	80126-A445
Version:	NA
Description:	The function of the service is described in Section "Network layer".

Id:	80126-A446
Version:	NA
Description:	6.6.2.1 Request

Id:	80126-A447

Version:	NA
Description:	The following DataIdentifier is defined for writing the parameter:

Id:	80126-A448
Version:	NA
Description:	Table 23 - Parameter BS_Bootloader

Id:	80126-A449
Version:	NA
Description:	The structure of the request is defined as follows:

Id:	80126-A450
Version:	NA
Description:	request structure

Id:	80126-A451
Version:	NA
Description:	6.6.2.2 Positive Response

Id:	80126-A452
Version:	NA
Description:	The structure is defined as follows:

Id:	80126-A453
Version:	NA
Description:	Table 25 - Positive Response WriteDataByIdentifier BS_Blocksize

Id:	80126-A454
Version:	NA
Description:	6.6.2.3 Negative Response

Id:	80126-A455
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A456
Version:	NA

Description:	conditionsNotCorrect

Id:	80126-A844
Version:	NA
Description:	requestOutOfRange

Id:	80126-A457
Version:	NA
Description:	Section "Error general behavior" must also be observed

Id:	80126-A458
Version:	NA
Description:	6.7 RoutineControl (31hex)

Id:	80126-A464
Version:	NA
Description:	The UDS-Service RoutineControl (31hex) service must be implemented according to [VW-ISO-TP] and [Q-LAH 80124] and with the required parameters.

Id:	80126-A459
Version:	NA
Description:	This service allows the client to start processing routines of the server.

Id:	80126-A460
Version:	NA
Description:	The following value is defined for the RoutineControlType.

Id:	80126-A461
Version:	NA
Description:	Table 27 - RoutineControl, RoutineControlType

Id:	80126-A462
Version:	NA
Description:	The following values have been defined for the routineIdentifier.

Id:	80126-A463
Version:	NA

Description:	Table 28 - RoutineControl, routineIdentifier

Id:	80126-A465
Version:	NA
Description:	The following requirements must be taken into consideration when implementing the service:

Id:	80126-A466
Version:	NA
Description:	The processing of the function must start with the request (startRoutine) of the client and end with the response of the server.

Id:	80126-A467
Version:	NA
Description:	The use of "stopRoutine" and "requestRoutineResult" is not permitted for the above-named RoutineIdentifiers.

Id:	80126-A468
Version:	NA
Description:	The result of the function (positive same as negative) must be delivered with a positive response.

Id:	80126-A469
Version:	NA
Description:	If the execution of the function is not possible, a negative response must be delivered.

Id:	80126-A470
Version:	NA
Description:	The further structure of the request (routineControlOptionRecord) and the response (routineStatusRecord) is detailed in the following Sections.

Id:	80126-A471
Version:	NA
Description:	6.7.1 0x0203-Check Programming Preconditions (Prüfung der Programmiervorbedingungen)

Id:	80126-A481

Version:	NA
Description:	The UDS-Service RoutineControl (31hex), checkProgrammingPreConditions (0x0203) service must be implemented according to [VW-ISO-TP] and [Q-LAH 80124] and with the required parameters.

Id:	80126-A472
Version:	NA
Description:	This function enables the client to call up the state of the programming pre-conditions in the server.

Id:	80126-A473
Version:	NA
Description:	For further requirements, see Section "Programming pre-conditions #(checkProgrammingPreConditions)".

Id:	80126-A964
Version:	NA
Description:	A recent overview of existing programming pre-conditions can be accessed at the function department which is responsible for flash ECU programming. (refer to Diagnostic-portal of VOLKSWAGEN AG).

Id:	80126-A476
Version:	NA
Description:	The programming pre-conditions which shall be implemented for the server are determined by the responsible department for the ECU. (aspects: legislation, protection of the ECU and safety pf)

Id:	80126-A962
Version:	NA
Description:	Each server must have the programming condition 129 dez "blocking time for programming is active" implement.

Id:	80126-A477
Version:	NA
Description:	Required programming pre-conditions not listed in Table 27 must be specified in consultation with the purchaser's department responsible for flashing.

Id:	80126-A693

Version:	NA
Description:	All programming pre-conditions implemented in the control unit must be documented in the ODX data (ECU variant). See [Q-LAH 80128].

Id:	80126-A478
Version:	NA
Description:	6.7.1.1 Request

Id:	80126-A479
Version:	NA
Description:	The structure is defined as follows:

Id:	80126-A480
Version:	NA
Description:	Table 30 - RoutineControl, checkProgrammingPreConditions, request definition

Id:	80126-A482
Version:	NA
Description:	6.7.1.2 Positive Response

Id:	80126-A483
Version:	NA
Description:	The structure is defined as follows:

Id:	80126-A484
Version:	NA
Description:	Table 31 - RoutineControl, checkProgrammingPreConditions, positive response definition

Id:	80126-A485
Version:	NA
Description:	The routineStatusRecord of the response of a server consists of a list of unfulfilled programming pre-conditions. If all conditions are fulfilled, the list is empty. If conditions are not fulfilled, the list contains a programmingConditionIdentifier for every unfulfilled condition.

Id:	80126-A487
Version:	NA

Description:	6.7.1.3 Protokollbeispiele

Id:	80126-A488
Version:	NA
Description:	Request: client → server Table 32 - RoutineControl, checkProgrammingPreConditions, protocol example 1a

Id:	80126-A489
Version:	NA
Description:	Response: server → client Table 33 - RoutineControl, checkProgrammingPre-Conditions, Protokolexample 1b The programming pre-conditions 04, 01, 02, 0C reportet as "not fulfilled".

Id:	80126-A490
Version:	NA
Description:	Request: client → server Table 34 - RoutineControl, checkProgrammingPreConditions, protocol example 2a

Id:	80126-A491
Version:	NA
Description:	Response: server → client Table 35 - RoutineControl, checkProgrammingPre-Conditions, Protocol example 2b The list is empty. That reflects that all programming pre-conditions are met.

Id:	80126-A492
Version:	NA
Description:	6.7.2 0xFF00-Erase Memory (Löschen des Programmspeichers)

Id:	80126-A501
Version:	NA
Description:	The RoutineControl "0xFF00-Erase Memory" – eraseMemory service must be implemented according to [VW-ISO-TP], [Q-LAH 80124] and with the required parameters.

Id:	80126-A493
Version:	NA
Description:	This function allows the client to start an erase routine of the server.

Id:	80126-A494

Version:	NA
Description:	The following requirements must be taken into consideration when implementing the service:

Id:	80126-A495
Version:	NA
Description:	It must be ensured that the eraseMemory service is executed only after successful access authorization and the receipt of a tester serial number and programming date. If one of these conditions is not fulfilled, the server must reject further programming.

Id:	80126-A942
Version:	NA
Description:	During an active sequence Request Download, Data Transfer, transfer exit and check memory a EraseMemory with the NRC must be 0x24 rejected.

Id:	80126-A496
Version:	NA
Description:	6.7.1.3 Protokollbeispiele

Id:	80126-A497
Version:	NA
Description:	The request to erase is defined as follows:

Id:	80126-A498
Version:	NA
Description:	Table 36 - RoutineControl, eraseMemory, request definition

Id:	80126-A499
Version:	NA
Description:	Table 37 - RoutineControl, eraseMemory, addressAndLengthFormatIdentifier

Id:	80126-A500
Version:	NA
Description:	Table 38 - RoutineControl, eraseMemory, memoryAddress

Id:	80126-A502
Version:	NA

Description:	Note: Logical blocks can only be erased in their entirety. Individual segments cannot be erased.

Id:	80126-A503
Version:	NA
Description:	6.7.2.2 Positive Response

Id:	80126-A504
Version:	NA
Description:	The positive response for erasing is defined as follows:

Id:	80126-A505
Version:	NA
Description:	Table 39 - RoutineControl, eraseMemory, positive response definition

Id:	80126-A506
Version:	NA
Description:	Table 40 - RoutineControl, eraseMemory, routineResult * Here only the result of erasing is returned. Events or conditions that have not allowed erasing to take place must be answered with a negative response.

Id:	80126-A508
Version:	NA
Description:	If on receipt of the "eraseMemory" command the logical block has already been completely erased or is writable, the server must send the positive response immediately without physically erasing the memory. The positive response is made independent of the memory used (RAM, EEPROM, FLASH).

Id:	80126-A796
Version:	NA
Description:	In the case of 80126-A508, the programming date and the tester serial number of the logical block must also be backed up. See 80126-A100 and 80126-A107.

Id:	80126-A509
Version:	NA
Description:	6.7.2.2 Positive Response

Id:	80126-A919

Version:	NA
Description:	In the following sample report, a logical block (block index 0x01) consisting of 3 segments is erased. The logical addresses for these segments are specified with: '- 0101 (Block index 01, Segment index 01) - 0102 (Block index 01, Segment index 02) - 0103 (Block index 01, Segment index 03)

Id:	80126-A510
Version:	NA
Description:	Example Request: client → server Table 41 - RoutineControl, eraseMemory, Protokolexample, Erase request with block index 0x01

Id:	80126-A511
Version:	NA
Description:	Example: Response: server → client Table 42 - RoutineControl, eraseMemory, Protocol example . The result of the erase operation is "correctResult

Id:	80126-A1176
Version:	NA
Description:	To be able to eliminate the time for erasing the flash memory during production, there is the option of preliminary erasing of individual blocks of the server at an upstream processing location.

Id:	80126-A1177
Version:	NA
Description:	Calling the EraseMemory routine with the highest possible memoryAddress (0xFF, 0xFFFF, etc.) erases all logical blocks besides the boot loader (see 80126-A500).

Id:	80126-A1178
Version:	NA
Description:	Therefore, the highest possible memoryAddress (0xFF, 0xFFFF, etc.) may not be used as the address for a regular logical block.

Id:	80126-A1179
Version:	NA
Description:	During the preliminary erasing, all logistical data from 80126-A112, 80126-A849 and 80126-A861 must be handled as during an interruption in the flashing process.

Id:	80126-A1180
Version:	NA
Description:	For the preliminary erasing of individual blocks, the modified sequence diagram from 80126-A246 applies.

Id:	80126-A1322
Version:	NA
Description:	If downgrade protection is implemented in the server, a valid pseudo flash driver for all logical blocks must be transferred for the purpose of preliminary erasing.

Id:	80126-A512
Version:	NA
Description:	6.7.3 0x0202-Check Memory

Id:	80126-A524
Version:	NA
Description:	The RoutineControl Identifier "0x0202-Check Memory" – checkMemory service must be implemented according to [VW-ISO-TP], [Q-LAH 80124] and with the required parameters.

Id:	80126-A513
Version:	NA
Description:	This service allows the client to start a check routine of the server.

Id:	80126-A514
Version:	NA
Description:	It must be ensured that the checkMemory service is executed only after successful access authorization. If this condition is not fulfilled, the server must reject further programming.

Id:	80126-A515
Version:	NA
Description:	For additional requirements, see Section "Integrity/authenticity check (check-Memory)".

Id:	80126-A516
Version:	NA

Description:	6.7.3.1 Request

Id:	80126-A517
Version:	NA
Description:	The request to check is defined as follows:

Id:	80126-A518
Version:	NA
Description:	Table 43 - RoutineControl, checkMemory, request definition:

Id:	80126-A523
Version:	NA
Description:	Table 44 - RoutineControl, checkMemory, value

Id:	80126-A526
Version:	NA
Description:	6.7.3.2 Positive Response

Id:	80126-A527
Version:	NA
Description:	The positive response for checking is defined as follows:

Id:	80126-A528
Version:	NA
Description:	Table 45 - RoutineControl, checkMemory, positive response definition

Id:	80126-A530
Version:	NA
Description:	Table 46 - RoutineControl, checkMemory, routineResult * Only result of check is returend. Events or conditions that prohibited a check have to be acknowledged with a negative response code.

Id:	80126-A532
Version:	NA
Description:	6.7.3.3 Protokollbeispiele

Id:	80126-A533
Version:	NA

Description:	Beispiel: Request: client → server Table 47 - RoutineControl, checkMemory, protocol example, Request check with Signature 0x03FABC65...4245894C (128 Byte long signature)

Id:	80126-A1160
Version:	NA
Description:	Beispiel: Request: client → server Table 82 - RoutineControl, checkMemory, protocol example, Request Check with Signature 0x30AFCB56...245498C4 (384 Byte long Signature)

Id:	80126-A534
Version:	NA
Description:	Example: Response: server → client Table 48 -RoutineControl, checkMemory, Protocol example, Response. The result of the check is: "correctResult".

Id:	80126-A535
Version:	NA
Description:	6.7.4 0xFF01-Check Programming Dependencies

Id:	80126-A542
Version:	NA
Description:	The RoutineControl – "0xFF01-Check Programming Dependencies"service must be implemented according to [VW-ISO-TP], [Q-LAH 80124] and with the required parameters.

Id:	80126-A536
Version:	NA
Description:	Beispiel: Request: client → server Table 82 - RoutineControl, checkMemory, protocol example, Request Check with Signature 0x30AFCB56...245498C4 (384 Byte long Signature)

Id:	80126-A537
Version:	NA
Description:	It must be ensured that the checkProgrammingDependencies service is executed only after successful access authorization and the receipt of a tester serial number and programming date. If one of these conditions is not fulfilled, the server must reject further programming.

Id:	80126-A538

Version:	NA
Description:	For additional requirements, see Section "Compatibility/consistency check (checkProgrammingDependencies)".

Id:	80126-A539
Version:	NA
Description:	6.7.4.1 Request

Id:	80126-A540
Version:	NA
Description:	The request is defined as follows:

Id:	80126-A541
Version:	NA
Description:	Table 49 - RoutineControl, checkProgrammingDependencies, request definition

Id:	80126-A543
Version:	NA
Description:	6.7.4.2 Positive Response

Id:	80126-A544
Version:	NA
Description:	The positive response is defined as follows:

Id:	80126-A545
Version:	NA
Description:	Table 50 - RoutineControl, checkProgrammingDependencies, positive response definition

Id:	80126-A546
Version:	NA
Description:	Table 51 - RoutineControl, checkProgrammingDependencies, routineResult * The result of the check is returned. Events or Conditions, that prohibit a check or actions which could not be started after a check, have to be reported with a negative response

Id:	80126-A548
Version:	NA

Description:	6.7.4.3 Protocol examples

Id:	80126-A549
Version:	NA
Description:	Table 52 - RoutineControl, checkProgrammingDependencies, protocol example Request Check

Id:	80126-A550
Version:	NA
Description:	Table 53 - RoutineControl, checkProgrammingDependencies, Protocol Example Response. Das result of the check is: "correctResult".

Id:	80126-A1025
Version:	NA
Description:	6.7.5 0x0544-Verify_partial_software_checksum

Id:	80126-A1026
Version:	NA
Description:	The routine Identifer "0x0544-Verify_partial_software_checksum" allows the client to start a test routine of the server.

Id:	80126-A1027
Version:	NA
Description:	This routine is optional to implement for servers that support the partial programming. The decision about necessarity and the usage depends on the function responsible ECU department.

Id:	80126-A1028
Version:	NA
Description:	It shall be ensured that the RoutineIdentifier is executed only after a successful access authorization. When the routine is called in the application, the access authorization shall be made according to [SFD]. When the routine is called in the application, the access authorization shall be made by means of the SA2 procedure described in Annex A. If no access authorization is available, the server must refuse the execution of this routine.

Id:	80126-A1029
Version:	NA

Description:	The RoutineIdentifier shall be implemented in the application and in the bootloader in the ExtendedDiagnosticSession (see [Q-LAH 80125]).

Id:	80126-A1030
Version:	NA
Description:	The usage of the routine identifier is not permitted during an active RequestDownload, TransferData or RequestTransferExit.

Id:	80126-A1031
Version:	NA
Description:	A RoutineIdentifier shall not cause an abort of an active update programming session at any time.

Id:	80126-A1038
Version:	NA
Description:	Polling of a routine identifier is allowed in any order independently of the memory address.

Id:	80126-A1205
Version:	NA
Description:	In the case of a check request from the client, the server must calculate the checksum for the concerned block again. A comparison between the value received in the check request and a value stored from the last calculation of the CheckMemory routine is not allowed. In this case, exceptions are possible, but they must be agreed upon with the Diagnostic department and documented in the BT-LAH, e.g. when it comes to relatively big memory areas that should be calculated again.

Id:	80126-A1339
Version:	NA
Description:	Either the CRC32 procedure of the CheckMemory routine or the procedure with Calibration Verification Numbers (CVN) according to [Q-LAH 80124] shall be used for the calculation of the checksum value.

Id:	80126-A1033
Version:	NA
Description:	For further requirements see chapter "Parallel Programming"

Id:	80126-A1034

Version:	NA
Description:	6.7.5.1 Request

Id:	80126-A1035
Version:	NA
Description:	The Request ist defined as followes:

Id:	80126-A1036
Version:	NA
Description:	Table 54 - RoutineControl, Verify_partial_software_checksum, request definition

Id:	80126-A1041
Version:	NA
Description:	Table 55 - RoutineControl, Verify_partial_software_checksum, lengthIdentifier

Id:	80126-A1043
Version:	NA
Description:	Table 56 - RoutineControl, Verify_partial_software_checksum, memoryAdress

Id:	80126-A1044
Version:	NA
Description:	Table 56 - RoutineControl, Verify_partial_software_checksum, memoryAdress

Id:	80126-A1039
Version:	NA
Description:	6.7.5.2 Positive Response

Id:	80126-A1046
Version:	NA
Description:	The positive response is defined as follows:

Id:	80126-A1045
Version:	NA
Description:	Table 58 - RoutineControl, Verify_partial_software_checksum, positive response definition

Id:	80126-A1040
Version:	NA

Description:	Table 59 - RoutineControl, Verify_partial_software_checksum, routineVerification

Id:	80126-A1047
Version:	NA
Description:	6.7.5.3 Protocol examples

Id:	80126-A1048
Version:	NA
Description:	Example: Request: client -> server

Id:	80126-A1049
Version:	NA
Description:	Table 60 - RoutineControl, Verify_partial_software_checksum, Protokollbeispiel Request

Id:	80126-A1050
Version:	NA
Description:	Example Response: server -> client

Id:	80126-A1051
Version:	NA
Description:	Table 61 - RoutineControl, Verify_partial_software_checksum, Protokollbeispiel Response

Id:	80126-A551
Version:	NA
Description:	6.7.6 Common NRC for RoutineControl

Id:	80126-A552
Version:	NA
Description:	The following NRC have to be implemented:

Id:	80126-A553
Version:	NA
Description:	conditionsNotCorrect

Id:	80126-A825

Version:	NA
Description:	conditionsNotCorrect

Id:	80126-A827
Version:	NA
Description:	requestSequenceError

Id:	80126-A828
Version:	NA
Description:	requestOutOfRange

Id:	80126-A829
Version:	NA
Description:	requestOutOfRange

Id:	80126-A830
Version:	NA
Description:	securityAccesssDenied

Id:	80126-A831
Version:	NA
Description:	generalProgrammmingFailure

Id:	80126-A554
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A555
Version:	NA
Description:	6.8 RequestDownload (34hex)

Id:	80126-A559
Version:	NA
Description:	The RequestDownload service must be implemented according to [VW-ISO-TP], [Q-LAH 80124] and with the required parameters.

Id:	80126-A556

Version:	NA
Description:	6.8.1 Request

Id:	80126-A557
Version:	NA
Description:	As a supplement to the definition of the request in [ISO 14229-1], the following definition is specified for the "memoryAddress" parameter:

Id:	80126-A937
Version:	NA
Description:	A download for a logical block has to be completed (CheckMemory i.O.) before a new RequestDownload is allowed, otherwise a renewed request Download the NRC = 0x22 should be rejected.

Id:	80126-A558
Version:	NA
Description:	Table 63 - RequestDownload, memoryAddress

Id:	80126-A698
Version:	NA
Description:	Table 64 - RequestDownload, DataFormatIdentifier

Id:	80126-A699
Version:	NA
Description:	The RequestDownload - DataFormatIdentifier has to be parameterized according Table 80126-A698.

Id:	80126-A562
Version:	NA
Description:	6.8.2 Negative Response

Id:	80126-A563
Version:	NA
Description:	The following NRCs have to be implemented

Id:	80126-A564
Version:	NA

Description:	conditionsNotCorrect

Id:	80126-A832
Version:	NA
Description:	conditionsNotCorrect

Id:	80126-A833
Version:	NA
Description:	conditionsNotCorrect

Id:	80126-A834
Version:	NA
Description:	requestOutOfRange

Id:	80126-A835
Version:	NA
Description:	securityAccessDenied

Id:	80126-A1317
Version:	NA
Description:	uploadDownloadNotAccepted

Id:	80126-A1318
Version:	NA
Description:	uploadDownloadNotAccepted 2

Id:	80126-A836
Version:	NA
Description:	uploadDownloadNotAccepted

Id:	80126-A565
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A681
Version:	NA
Description:	6.8.3 Protocol Example

Id:	80126-A683
Version:	NA
Description:	Programming request with logical address 0x0201 (Block index 02, Segment index 01) and block size 0xFA000. The data are compressed (LZSS method) and sent unencrypted. Example: Request: Server -> Client Table 66 - RequestDownload (Server Request) Example: Response Server -> Client Table 67- RequestDownload (Client Response), The response delivers the maximum data quantity of 996 bytes (SID + 994 bytes of flash data) for the data transfer

Id:	80126-A566
Version:	NA
Description:	6.9 TransferData (36hex)

Id:	80126-A569
Version:	NA
Description:	The UDS-Service TransferData (36hex) must be implemented according to [VW-ISO-TP] and [Q-LAH 80124].

Id:	80126-A567
Version:	NA
Description:	It must be ensured that the TransferData service is executed only after successful access authorization. If this is not available, the server must refuse further programming.

Id:	80126-A572
Version:	NA
Description:	6.9.1 Negative Response

Id:	80126-A573
Version:	NA
Description:	The structure of the negative response must be implemented according to [Q-LAH 80124]

Id:	80126-A574
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A575

Version:	NA
Description:	requestSequenceError

Id:	80126-A837
Version:	NA
Description:	requestSequenceError

Id:	80126-A838
Version:	NA
Description:	requestOutOfRange

Id:	80126-A839
Version:	NA
Description:	transferDataSuspended

Id:	80126-A840
Version:	NA
Description:	generalProgrammingFailure

Id:	80126-A841
Version:	NA
Description:	wrongBlockSequenceCounter

Id:	80126-A842
Version:	NA
Description:	voltageTooHigh

Id:	80126-A843
Version:	NA
Description:	voltageTooLow

Id:	80126-A576
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A577
Version:	NA

Description:	6.10 RequestTransferExit (37hex)

Id:	80126-A580
Version:	NA
Description:	The UDS-Service RequestTransferExit (37hex) must be implemented according to [VW-ISO-TP] and [Q-LAH 80124].

Id:	80126-A578
Version:	NA
Description:	It must be ensured that the RequestTransferExit service is executed only after successful access authorization. If this is not available, the server must refuse further programming

Id:	80126-A583
Version:	NA
Description:	6.10.1 Negative Response

Id:	80126-A584
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A585
Version:	NA
Description:	requestSequenceError

Id:	80126-A802
Version:	NA
Description:	requestSequenceError

Id:	80126-A586
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A587
Version:	NA
Description:	6.11 TesterPresent (3Ehex)

Id:	80126-A588

Version:	NA
Description:	The UDS-Service TesterPresent (3Ehex) must be implemented according to [VW-ISO-TP] and [Q-LAH 80124]

Id:	80126-A589
Version:	NA
Description:	6.12 ControlDTCSetting (85hex)

Id:	80126-A593
Version:	NA
Description:	The UDS-Service ControlDTCSetting (85hex) must be implemented according to [VW-ISO-TP] and [Q-LAH 80124]

Id:	80126-A590
Version:	NA
Description:	6.12.1 Request

Id:	80126-A591
Version:	NA
Description:	The request must be implemented with the following parameters

Id:	80126-A592
Version:	NA
Description:	Table 70 - ControlDTCSetting, DTCSettingType

Id:	80126-A594
Version:	NA
Description:	The UDS-Service DTCSettingType must be implemented according to [VW-ISO-TP] and [Q-LAH 80124]

Id:	80126-A944
Version:	NA
Description:	OBD-servers shall be accessd for ControlDTCSetting only by functional adressing mode.

Id:	80126-A597
Version:	NA

Description:	6.12.2 Negative Response

Id:	80126-A598
Version:	NA
Description:	The structure of the negative response must be implemented according to [Q-LAH 80124].

Id:	80126-A599
Version:	NA
Description:	The following NRCs must be implemented for the described error states:

Id:	80126-A600
Version:	NA
Description:	conditionsNotCorrect

Id:	80126-A797
Version:	NA
Description:	conditionsNotCorrect

Id:	80126-A798
Version:	NA
Description:	requestOutOfRange

Id:	80126-A601
Version:	NA
Description:	Section "Error general behavior" must also be observed.

Id:	80126-A602
Version:	NA
Description:	6.13 ResponsePending (78hex)

Id:	80126-A603
Version:	NA
Description:	The NRC=0x78 (ResponsePending, 78hex) must be implemented according to [VW-ISO-TP] and [Q-LAH 80124]

Id:	80126-A604

Version:	NA
Description:	If the time required to process a service exceeds the time P2CAN_Server, the server must respond with "ResponsePending". This response must be repeated until the service is fully processed. Upon complete execution of the service, the server must send a positive or negative (not equal to ResponsePending) response.

Id:	80126-A605
Version:	NA
Description:	6.14 Error general behavior

Id:	80126-A606
Version:	NA
Description:	A server must report an error situation by a negative response with the NRC defined for this situation.

Id:	80126-A607
Version:	NA
Description:	If an error occurs, a server must remain in the state prior to the error event and must wait for the next service of the client.

Id:	80126-A608
Version:	NA
Description:	Note: The status of the server must be predictable for the client at any time. This also applies to an NRC or a negative RoutineResult. In particular, the following applies:

Id:	80126-A609
Version:	NA
Description:	An automatic restart or other repair attempts of the server are not permissible in the event of an error.

Id:	80126-A610
Version:	NA
Description:	A server must only deliver the NRCs defined in these Performance Specifications and in the reference documents. Other NRCs are not permitted.

Id:	80126-A611

Version:	NA
Description:	The error situation must be reported with the error event. Signaling at a later time, for instance as a subsequent error, is not permissible.

Id:	80126-A612
Version:	NA
Description:	The client must always re-start an interrupted programming process from the very start, beginning with a pre-programming phase.

Id:	80126-A614
Version:	NA
Description:	The following NRCs have been defined for the following conditions: (80126-A615,-A799, -A800, -A901)

Id:	80126-A615
Version:	NA
Description:	ServiceNotSupported

Id:	80126-A799
Version:	NA
Description:	SubFunctionNotSupported

Id:	80126-A800
Version:	NA
Description:	SubFunctionNotSupportedInActiveDiagnosticSession

Id:	80126-A801
Version:	NA
Description:	ServiceSupportedInActiveDiagnosticSession

Id:	80126-A616
Version:	NA
Description:	For the error situations described in 80126-A615, -A799, -A800 and -A801, no response must be sent for a functionally addressed service (see requirmement 80124-5343).

Id:	80126-A617

Version:	NA
Description:	IncorrectMessageLengthOrInvalidFormat

Id:	80126-A946
Version:	NA
Description:	GeneralProgrammingFailure

Id:	80126-A618
Version:	NA
Description:	7 Application- und Diagnostic Management

Id:	80126-A619
Version:	NA
Description:	The requirements of [Q-LAH 80124] must be implemented.

Id:	80126-A620
Version:	NA
Description:	8 Special protocol requirements and timing

Id:	80126-A621
Version:	NA
Description:	Unless otherwise specified in this Section, the requirements in [Q-LAH 80124], [VW-ISO-TP] must be implemented.

Id:	80126-A622
Version:	NA
Description:	The programming of control units places other requirements on the time behavior of the diagnostic and transport protocol than other diagnostic applications. This results in requirements for the bootloader and also for the application that deviate from, or supplement, other specifications.

Id:	80126-A1164
Version:	NA
Description:	For CAN, the maximum value of 4095 bytes (each incl. 2 bytes for the UDS header) must not be exceeded. For pure Ethernet connections a maximum block length of 4 GByte applies (see Q LAH_80124-309).

Id:	80126-A623

Version:	NA
Description:	8.1 UDS-Layer

Id:	80126-A1165
Version:	NA
Description:	8.1.1 General Requirements

Id:	80126-A627
Version:	NA
Description:	During execution of routines in the boot loader (erasing flash, calculating check value, etc.), the server must sustain communication with the client.

Id:	80126-A867
Version:	NA
Description:	Each server must transmit a block length of minimum 514 bytes in the response to a RequestDownload [VW-ISO-TP].

Id:	80126-A869
Version:	NA
Description:	The server must return and support the largest possible value for the respective hardware when responding to a RequestDownload-Request.

Id:	80126-A632
Version:	NA
Description:	The client must adhere to the transmitted block length of the server exactly, as long as the end of the data block is not reached. The last TransferData request contains the remaining data. Padding with fill bytes does not occur in the client.

Id:	80126-A1166
Version:	NA
Description:	8.1.2 Timings

Id:	80126-A624
Version:	NA
Description:	Server:

Id:	80126-A625
Version:	NA

Description:	P2*CAN-Server must be transferred to the tester in the positive response of the service request DiagnosticSessionControl (ExtendedSession), according to the maximum time required to revert to the boot loader. A deviation from the default value of 5 seconds must be agreed with the purchaser's department responsible for flashing.

Id:	80126-A626
Version:	NA
Description:	P2*CAN-Server must be coded with the default value of 5 s in the positive response of the service request DiagnosticSessionControl (ProgrammingSession) even if the maximum required time for executing routines in the boot loader will exceed this value.

Id:	80126-A628
Version:	NA
Description:	P2CAN-Server has the value 50 ms in the ProgrammingSession. This value must be transmitted to the tester in the positive response of the service request DiagnosticSessionControl (ProgrammingSession).

Id:	80126-A631
Version:	NA
Description:	Client:

Id:	80126-A633
Version:	NA
Description:	The P2, P2* timing parameters transmitted by the server in the positive response of the service request DiagnosticSessionControl must be guaranteed during the respective active diagnostic session.

Id:	80126-A1118
Version:	NA
Description:	It needs to be changed in the PDX the value of comparam CP_RC78 Completion Timeout e.g. the service CheckProgrammingDependencies. Therefor DiagnServi_RoutiContrCheckProgrDepen e.g. must be changed to the value 1800000000 us. = 30 min.

Id:	80126-A635
Version:	NA
Description:	8.2 Networklayer

Id:	80126-A1020
Version:	NA
Description:	8.2.1 CAN-BUS

Id:	80126-A636
Version:	NA
Description:	The following conditions must be fulfilled for control units on a CAN bus. The values to be set relate to parameters of the CAN transport protocol according to [VW-ISO-TP].

Id:	80126-A868
Version:	NA
Description:	The server must support each block length of minimal 255 bytes to the server provided by the maximum value, even if it returns in response to the Request Download a higher value.

Id:	80126-A637
Version:	NA
Description:	Server: Receiving messages

Id:	80126-A638
Version:	NA
Description:	N_Br inferior or equal to 1 ms Sending the FlowControlFrame within a maximum of 1 ms after receipt of a FirstFrame or ConsecutiveFrame.

Id:	80126-A639
Version:	NA
Description:	SeparationTime (STmin) = 0x00 Fastest possible receipt of ll CAN messages by the control unit

Id:	80126-A640
Version:	NA
Description:	Block size (BS) = 0x0F It must be possible for the control unit to set not only the indicated default value but also every other value including 0x00 and, thus, the control unit must provide for all mechanisms necessary for this.

Id:	80126-A641
Version:	NA

Description:	The value to be set for BS by the boot loader can be specified by the client via diagnosis by means of WriteDataByIdentifier (see Section "WriteDataByIdentifier – BSBootloader").

Id:	80126-A642
Version:	NA
Description:	The server must support the reading and writing of the DataIdentifiers "0x0410-Bootloader TP Blocksize" in each ExtendedDiagnosticSession, both in the application and in the bootloader.

Id:	80126-A643
Version:	NA
Description:	With the return to the DefaultDiagnosticSession, the value must be reset to the default value.

Id:	80126-A644
Version:	NA
Description:	Figure 80126-A645 illustrates the relationships:

Id:	80126-A645
Version:	NA
Description:	Figure 14 – Influence on the TP block size of the boot loader

Id:	80126-A646
Version:	NA
Description:	The server must use the value for block size (BS) set by the client in the boot loader. If there is no value specified by the client, the default value must be set.

Id:	80126-A647
Version:	NA
Description:	The value for block size (BS) in the boot loader remains valid up to the return to the DefaultDiagnosticSession of the application or the boot loader and then switches to the default value.

Id:	80126-A648
Version:	NA
Description:	Server: Sending messages

Id:	80126-A649

Version:	NA
Description:	Definition: N_Cs inferior or equal 1 ms Depending on the transmitted value STmin. When STmin = 0x00, the distance between two Consecutive frames or between Flow Control Frame and a ConsecutiveFrame smaller than 100us.

Id:	80126-A650
Version:	NA
Description:	Client: Receiving messages

Id:	80126-A651
Version:	NA
Description:	SeparationTime (STmin) = 0x00 Fastest possible receipt of all CAN messages by the tester

Id:	80126-A652
Version:	NA
Description:	Block size (BS) = 0x0F It must be possible for the client to set not only the indicated default value but also every other value including 0x00 and, thus, the client must provide for all mechanisms necessary for this.

Id:	80126-A653
Version:	NA
Description:	N_Br inferior or equal to 1 ms Sending the FlowControlFrame within a maximum of 1 ms after receipt of a FirstFrame or ConsecutiveFrame.

Id:	80126-A654
Version:	NA
Description:	Client: Sending messages

Id:	80126-A655
Version:	NA
Description:	N_Cs inferiore or equal to 1 ms It must be possible to adapt the actual sending interval to the infrastructure of the vehicle or the performance of the target data bus.

Id:	80126-A980
Version:	NA
Description:	8.2.2 FlexRay

Id:	80126-A981
Version:	NA
Description:	The bootloader of FlexRay ECUs must receive data on all diagnostic channels (e.g. the ISO tester PDUs and the flash PDUs have to be read).

Id:	80126-A1009
Version:	NA
Description:	A single diagnostic request is allowed to to be sent only on a single diagnostic channel (request can exclusively be sent via ISO tester PDUs or via Flash PDUs). A distribution of a single request over ISO tester PDUs and flash PDUs is not allowed.

Id:	80126-A1010
Version:	NA
Description:	The switch algorithm in the gateway for useage of the Flash channel is described in ISO-TP Annex.

Id:	80126-A1158
Version:	NA
Description:	Each server must response to a Request Download with a minimum block length of at least 514 bytes and transmit accordingly [VW ISO-TP]. For FlexRay, the maximum value of 4095 bytes (each incl. 2 bytes for the UDS header) must not be exceeded.

Id:	80126-A1159
Version:	NA
Description:	The server must support every block length with a minimum of 255 bytes to the maximum value provided by the server, even if it returns a higher value in response to the RequestDownload.

Id:	80126-A1101
Version:	NA
Description:	8.2.3 Flashing over Ethernet

Id:	80126-A1102
Version:	NA
Description:	To ensure a faultless flash sequence, The existing TCP / IP connection to the tester by switching from application to bootloader and vice versa may not interrupted at any time.

Id:	80126-A1108
Version:	NA
Description:	To ensure a faultless flash sequence, all ComParams which were set by the Tester must be preserved while the mode changes from application to bootloader and vice versa.

Id:	80126-A1103
Version:	NA
Description:	If in error case a switch from the bootloader back into the application is not possible, the ECU shall support the DoIP requirements of ISO13400-2 and the VW-specific additions to the ISO13400 after KL15 / KL30 cycle.

Id:	80126-A1104
Version:	NA
Description:	If in error cases a switch into the bootloader from application is not possible, the existing TCP/-IP connection must not be interrupted.

Id:	80126-A1105
Version:	NA
Description:	It must be ensured that the connection-specific parameters of the TCP / IP stack shared by the application as well as the boot loader.

Id:	80126-A882
Version:	NA
Description:	8.3 Performance requirements for flashing

Id:	80126-A883
Version:	NA
Description:	This Section describes all performance requirements that a control unit must fulfill with regard to flashing. In addition, examples of optimization measures for shortening the total flash times must be presented.

Id:	80126-A901
Version:	NA
Description:	The total flash time of a control unit depends on various factors, such as: • Program runtimes in the tester • Send rate of the tester • Transfer rate of the bus system • Flash-erasing speed of the processor • Flash-writing speed of the processor • Checksum check in the control unit

Id:	80126-A902
Version:	NA
Description:	All offboard-side factors will be further developed step by step in order to reach the maximum performance here. All onboard factors must also be optimized. Therefore, the following requirements apply.

Id:	80126-A1338
Version:	NA
Description:	The binary data volume per logical block is limited to 64 MBytes. A limitation of 256 MBytes applies to the entire data volume. Deviations must be agreed together with the P and Customer Service departments, and shall be documented in the BT-LAH..

Id:	80126-A884
Version:	NA
Description:	As part of the control unit development, attention must be paid that the total flash time is kept as short as possible.

Id:	80126-A1156
Version:	NA
Description:	In a new development of an ECU the time for flash programming need to improve by 20% to its predecessor (basis for comparison: with the same amount of data - more data allows more time). Deviations must be agreed with the responsible ECU department.

Id:	80126-A1141
Version:	NA
Description:	The programming of a server (including all necessary internal server steps, such as deletion, processing, writing, post-evaluation, post-run, and updating of identification data) using diagnostic tools released in the Group must not take longer than the following times, irrespective of the data volume to be flashed: Data volume Maximum flashing time < 16 MB 5 minutes; >= 16 MB < 1 6.01 · ln(data volume) - 11.5 minutes >= 1 GB 30 minutes

Id:	80126-A1200
Version:	NA
Description:	Deviations and tolerances shall be agreed upon with the P [/EN: production] and KD [/EN: customer service] departments.

Id:	80126-A1201
Version:	NA
Description:	The requirements of 80126-A1141 apply per physical ECU (Host-ECU), including all the virtual systems subordinate to the physical ECU. A virtual system of the diagnostic class DK2FV, DK3V, DK4V or software cluster must not prevent the requirement A1141 from being fulfilled for its host.

Id:	80126-A903
Version:	NA
Description:	At a minimum, the following onboard optimizations for shortening the flash time must be evaluated and implemented in agreement with the purchaser.

Id:	80126-A898
Version:	NA
Description:	In order to minimize the handshaking between the tester and control unit, the control unit must not send any unnecessary ResponsePending (e.g., as a default response to any service). ResponsePending is only allowed on a switch between the application and boot loader (DiagnosticSessionControl 10hex 0x02 / 10hex 0x01, ECUReset 11hex 0x01) and on execution of routines (RoutineControl). Further use must be justified in advance and requires the approval of the department responsible for the "Flashing" function.

Id:	80126-A906
Version:	NA
Description:	NRC 0x21 must not be used by the control unit during the flash sequence. Exceptions: Requests with a different diagnostic protocol are rejected with NRC 0x21 (see Q-LAH_80124-300).

Id:	80126-A899
Version:	NA
Description:	The data rate of the control unit must not fall below the maximum data rates for flash-erasing and flash-writing indicated in the processor specification by more than 20%

Id:	80126-A908
Version:	NA
Description:	Note: The time between the last ConsecutiveFrame of the TransferData command and the positive response of the control unit serves as the basis for evaluating the flash writing times.

Id:	80126-A904
Version:	NA
Description:	The checksum check must be implemented using current time-optimized methods (e.g., lookup tables).

Id:	80126-A907
Version:	NA
Description:	For the UDS block size to be transferred the requirement VW80126-A869 applies.

Id:	80126-A894
Version:	NA
Description:	8.3.1 Multi-processor systems

Id:	80126-A885
Version:	NA
Description:	For multi-processor systems, the communication between the individual processors must be optimized and operated during flashing in such a way that the writing of data in the slave processor is not slowed down by the internal communication between the processors (Data rate of internal bus >= Data rate of flash writing in the slave processor).

Id:	80126-A905
Version:	NA
Description:	For multi-processor systems, the checksum check must be implemented in such a way that checksums of a secondary processor are also computed on this processor and only the result is sent to the main processor (i.e., the data used to compute the checksum must not be sent over the internal communication bus for the computation).

Id:	80126-A1121
Version:	NA
Description:	In multiprocessor implementations inter-process communication must be possible during the update programming.

Id:	80126-A1132
Version:	NA
Description:	Especially for multi-processor systems, the requirement is observed 80126-A982.

Id:	80126-A1157
Version:	NA
Description:	In a multiprocessor system the flash programming operates via a single control device ID.

Id:	80126-A887
Version:	NA
Description:	8.3.2 Partial programming

Id:	80126-A889
Version:	NA
Description:	In the case of partial programming, an intelligent control system in the programming tool ensures that only those logical blocks that have changed compared to the current version are programmed. For this purpose, the client evaluates "0xF1AB-VW Logical Software Block Version" of the blocks contained in the server and compares them with the blocks in the flash container. The comparison is made based on the OWN-IDENTS in the ECU-MEM Flash container (see [Q-LAH 80128]).

Id:	80126-A890
Version:	NA
Description:	The tester then flashes only the identified differences to the control unit, i.g., all blocks in which the version information (0xF1AB-VW Logical Software Block Version) do not match between the control unit and ECU-MEM Flash container are programmed.

Id:	80126-A892
Version:	NA
Description:	Figure 80126-A891 represents the partial programming sequence for a control unit with 3 logical blocks. • Block 1 is not updated because it already exists in the same version in the control unit. • Block 2 is flashed because it is currently invalid in the control unit. • Block 3 is flashed because the version in the flash container is more current than that in the control unit.

Id:	80126-A891
Version:	NA
Description:	Figure 15 - Workflow of partial programming

Id:	80126-A888

Version:	NA
Description:	During software development, attention must be paid that different software modules are generated that can then be flashed as individual logical blocks.

Id:	80126-A956
Version:	NA
Description:	The Data Identifier "0xF189-VW Application Software Version Number" is composed of a unique combination of individual VWLogicalSoftwareBlockVersions.

Id:	80126-A957
Version:	NA
Description:	Figure 16 -Example for the Uniqueness of VWLogicalSoftwareBlockVerison in relation to VWSoftwareVersionNumber

Id:	80126-A909
Version:	NA
Description:	Logical blocks must be developed in the following cases: - Separation between stable basic software (standard SW or modules) and variable application portions - Separation between application and data (see 80126-A43) - Functional separation of different software modules - Characteristic ranges, graphics, or fonts - Flash time limitations for logical blocks (see 80126-A716)

Id:	80126-A893
Version:	NA
Description:	The corresponding OWN-Idents for each logical block must be created in the flash container. See [Q-LAH 80128].

Id:	80126-A1052
Version:	NA
Description:	For further validation may be required by the responsible ECU department that in addition to the VWLogicalSoftwareBlockVersion the signature/checksum for this logical block need also to be checked (see also Routine Control - Verify_partial_software_checksum). If this occurs, it is necessary to document this in the ECU component specification (BT-LAH).

Id:	80126-A1057
Version:	NA
Description:	Server which have not implemented the RoutineControl - Verify_partial_software_checksum, have to acknowledge such a request with a NRC but don't in-

	terrupt the current programming session. (Request has to be discarded after NRC).

Id:	80126-A1058
Version:	NA
Description:	The following requirements relate to Servers that have implemented the RoutineControl Verify_partial_software_checksum.

Id:	80126-A1054
Version:	NA
Description:	Server which have implemented the RoutineControl Verify_partial_software_checksum have to support partial flashing also without a request of RoutineControl - Verify_partial_software_checksum.

Id:	80126-A1055
Version:	NA
Description:	The optional execution and evaluation of routine Control - Verify_partial software checksum is controlled only by the client (eg Flash Job.).

Id:	80126-A1056
Version:	NA
Description:	The routine Control - Verify Partial Software checksum is not intended as an additional safeguard in the server, but as a help for the client, which logical blocks need to be updated.

Id:	80126-A1100
Version:	NA
Description:	If the routine status Routine Control - Verify_partial_software Checksum 0x00 (Checksum is identical) is returned, then a triggered update for this logical block must not be rejected by the ECU.

Id:	80126-A1053
Version:	NA
Description:	A checksum/signature must be checked with RoutineControl - Verify_partial software checksum if the comparison between the logical block in the ODX-flash-container and logical block in the server recognizes the identical WLogicalSoftwareBlockVersion.

Id:	80126-A949

Version:	NA
Description:	8.3.3 EarlyAcknowledge

Id:	80126-A896
Version:	NA
Description:	Explanation: EarlyAcknowledge During transfer of TransferData services during the flash sequence, the positive response to this service is already sent directly after receipt of the last ConsecutiveFrame, thus, before the data have been physically written to the flash memory. The Client then begins immediately to transfer the next TransferData. With the positive response to the second TransferData the successful transfer and writing of the first TransferData is acknowledged. This mechanism shortens the transfer of data from the Client to Server.

Id:	80126-A1167
Version:	NA
Description:	The Early Acknowledge procedure is also known as "Pipelined Programming".

Id:	80126-A955
Version:	NA
Description:	Example Workflow

Id:	80126-A950
Version:	NA
Description:	The mechanism Early Acknowledge has to be implemented by all servers in contradiction to [Q-LAH 80124] for the TransferData service.

Id:	80126-A1113
Version:	NA
Description:	If the server detects the transfer or writing of erroneous data the flash operation is cancelled as fast as possible by transmission of NRC 0x72 (GeneralProgrammingFailure) using the Service "36hex TransferData".

Id:	80126-A1119
Version:	NA
Description:	If one of the following errors is detected in the transmission of the service "36hex TransferData" the error has to be recognized and rejected immediately with the NRC: - Service Request contains no Data - Inital BSN (BlockSequenceNumber) - discontinuous BSN (BlockSequenceNumber) All further program-

	ming errors have to be detected with the consecutive RequestService "36hex-TransferData" and acknowledged with NRC.

Id:	80126-A1122
Version:	NA
Description:	If the Service request contains no Data the NRC 0x31 (RequestOutOfRange) must be transmitted.

Id:	80126-A1123
Version:	NA
Description:	If the initial Sequence Number does not start with 1 the NRC 0x73 (WrongBlock-SequenceCounter) has to be sent.

Id:	80126-A1124
Version:	NA
Description:	If wrong Sequence Number is received the NRC 0x73 (WrongBlockSequence-Counter) has to be sent.

Id:	80126-A1059
Version:	NA
Description:	Implementation example for a flash drive with 64 Kbits FPG and a block size CB = 4K: Not every 64 Kbytes of Flash data written to flash, but collected / only held in a RAM Buffer_1. Once the RAM is full Buffer_1 (4 KB), the data stored in the flash. In parallel with the RAM Buffer_1 a second RAM Buffer_2 shall be provided which receives the messages while the flash controller writes the data of RAM_Buffer_1 into Flash. If the RAM_Buffer_2 is completely filled with new messages before the flash controller can store the data in RAM_Buffer_1, so the ECU has to send a ResponsePending message until the RAM_Buffer_1 was emptied and can be used again. This procedure repeates between RAM_-Buffer_1 and RAM_Buffer_2 up until all the data was written to the flash.

Id:	80126-A951
Version:	NA
Description:	Servers that can not implement the mechanism EarlyAcknowledge need to coordinate this with the responsible ECU department and document this in the ECU component specification (BT-LAH).

Id:	80126-A1168
Version:	NA

Description:	8.3.4 Pipelined Verification

Id:	80126-A1169
Version:	NA
Description:	Pipelined Verification is a process for the Flash time reduction by parallelized checksum / hash value calculation and verification.

Id:	80126-A1170
Version:	NA
Description:	The use of Pipelined Verification is optional.

Id:	80126-A1171
Version:	NA
Description:	If pipelined Verification is used, the calculation of the checksum or hash value has do be done on the already written data in the flash memory. (only check on the final result in persistent memory). The checksum / hash calculation performed on data in the receive buffer (RAM) is explicitely not permitted.

Id:	80126-A983
Version:	NA
Description:	8.3.5 Parallel Flashing

Id:	80126-A984
Version:	NA
Description:	To shorten the vehicle guidance and vehicle update times, the mechanism of parallel flash process is introduced.

Id:	80126-A1060
Version:	NA
Description:	Through the parallel flashing of multiple servers the flashing-time compared to the sequential flashing can be significantly shortened. To optimize the parallel flashing, certain deviations from standard flash sequence are necessary.

Id:	80126-A1061
Version:	NA
Description:	All servers must also support the revised procedure for parallel flashing.

Id:	80126-A1062

Version:	NA
Description:	When using parallel flashing, the sequences Pre-Programming, Programming-Execution and Post Programming shall be redistributed.

Id:	80126-A1063
Version:	NA
Description:	For parallel flashing the pre-sequence is called once for the entire vehicle . After finishing the pre-sequence, all the Programming Executions called specific for each server. This Programming Executions can be in nested order (parallel and sequential). If all Programming-Executions have been processed, the post-programming sequence is called once for the entire vehicle at the end. As a consequence, this shifts some services of the pre-sequence into the Programming-Executions.

Id:	80126-A1093
Version:	NA
Description:	The differences between the single flashing and the parallel flashing are very small, so that the processes, only the changes from the single flash were highlighted. The color green represents new and the color red for the omission at this point.

Id:	80126-A1064
Version:	NA
Description:	8.3.5.1 Pre-Programming during Parallel Flashing

Id:	80126-A1069
Version:	NA
Description:	The pre-programming sequence is transmitted with functional adressing mode when parallel flashing.

Id:	80126-A1070
Version:	NA
Description:	The query of the programming prerequisites for a single server is in the pre-programming sequence not possible when parallel flashing.

Id:	80126-A1067
Version:	NA
Description:	Figure 18 - Pre-Programming when Parallel Flashing

Id:	80126-A1065
Version:	NA
Description:	8.3.5.2 Programming-Execution during Parallel Flashing

Id:	80126-A1071
Version:	NA
Description:	In Programming Execution at parallels flashing the services are sent to the server by physical addressing mode.

Id:	80126-A1073
Version:	NA
Description:	The request for programming preconditions is now part of the programming execution for parallel flashing.

Id:	80126-A1074
Version:	NA
Description:	As a consequence the check of the programming pre-condition takes place a a point in time at which the communication by the Communication Control has be already deactivated (disablenon-diagnostic communication).

Id:	80126-A1075
Version:	NA
Description:	The Programming Executions can be seen as individual modules that can occur parallel, sequential, or in any mixed form of sequential-parallel. The nesting of parallel / sequential lasts on external dependencies such as master-slave dependencies or Clamp 15 control units or other dependencies.

Id:	80126-A1080
Version:	NA
Description:	It has to be noticed, that parallel-flashing of a DK2F ECU also required functional services for switching (Routing Activation) of the Master-ECU for the DK2F Slave. (see Q-LAH 80127). These functional Services must not lead to a cancellation of the flash-programming in other ECUs.

Id:	80126-A1076
Version:	NA
Description:	After programming and ECUReset the server would enter the application mode and start its communcation and error logging functions which may affect additional serves to be flashed. In order not to negatively affect the flash process of

	additional servers the requests ControlDTCSetting Off and StopCommunication is sent via functional adressing mode.

Id:	80126-A1117
Version:	NA
Description:	The multiple reception of the functional ControlDTCSetting OFF, without any ControlDTCSetting ON in between, must be accepted by each controller. The ControlDTCSetting OFF must remain active.

Id:	80126-A1114
Version:	NA
Description:	The OBD relevant ECUs should accept a ControlDTCSetting OFF only via functional adressing mode. This has to be respected in the parallel flash session for all ECUs (OBD and non-OBD).

Id:	80126-A1077
Version:	NA
Description:	The Sequence ControlDTCSetting OFF and StopCommunication after an ECUReset may be omitted also in the Programming-Execution.

Id:	80126-A1072
Version:	NA
Description:	Figure 19a - ProgrammingExecution Part 1 when parallel flashing

Id:	80126-A1094
Version:	NA
Description:	Figure 19b - ProgrammingExecution Part 2 when parallel flashing

Id:	80126-A1066
Version:	NA
Description:	8.3.5.3 Post-Programming when Parallel Flashing

Id:	80126-A1078
Version:	NA
Description:	In the post-programming sequence when parallel flashing all the services are sent with functional addressing mode.

Id:	80126-A1079
Version:	NA

Description:	Figure 20 - Post-programming when Parallel Flashing

Id:	80126-A987
Version:	NA
Description:	8.4 Optional Bootloader-Update

Id:	80126-A990
Version:	NA
Description:	8.4.1 Requirements

Id:	80126-A991
Version:	NA
Description:	For a Bootloader-Update the partial programming concept must be used.

Id:	80126-A992
Version:	NA
Description:	The Flash container includes for the complete software. That means: In the first data block the bootloader update or -downgrade is located and in the following data blocks all application software components and dataset records.

Id:	80126-A1088
Version:	NA
Description:	Normally, the bootloader update has to be performend in one stage. Both, the bootloader and the application and dataset areas have to be updated together in a single flash sequence. Two-staged updates are permitted only in exceptional cases and shall be documented in the ECU component specification (BT-LAH). For FDS ECUs the PublicKey (key exchange) has to be exchanged with a Bootloader-update, so that the update of the application and dataset record areas must be carried out in a second flash operation.

Id:	80126-A996
Version:	NA
Description:	The naming of the flash container has to be done in accordance with [Q-LAH 80128]. They differ not much from conventional flash containers.

Id:	80126-A995
Version:	NA
Description:	Any interruption of the erase operation for updating the boot loader is to make sure that the server can still recover and be re-programmed.

Id:	80126-A993
Version:	NA
Description:	The HW-logistical data remains unchanged with a bootloader update (see also "Logistical data").

Id:	80126-A1087
Version:	NA
Description:	For control units which support the Bootloader update, the logical block of the bootloader is allowed to contain in "0xF1AB-VW Logical Software Block Version" the old logic software block index as well as "----" (0x2D2D2D2D) after an inconsistent update.

Id:	80126-A989
Version:	NA
Description:	8.4.2 Sequence of a 1 staged bootloader exchange (standard case)

Id:	80126-A994
Version:	NA
Description:	The Bootloader-Update /-Downgrade takes place when the Bootloader-Version in the OWN-IDENT information whithin the Bootloader-Block in the Flashcontainer differs from the "0xF1AB-VW Logical Software Block Version" in the first logical block.

Id:	80126-A997
Version:	NA
Description:	The old bootloader (BL) must be backuped by a copying process to a dedicated memory area. This area can be in the application-, dataset record or a second boot loader area if no additional Bootloader-Updater (BLU) is used to overwrite the old bootloader. In that case (BL updated by BLU) the original BL must be erased and the "0xF1DF-ECU Programming Information" has to be updated (Inconsistency Indication), when the EraseMemory-Routine has to be started to clear a memory area and the entry address of the server jumps to the previously saved backup of the bootloader or, alternatively if BLU concept is used, to the received and stored Bootloader-Updater.

Id:	80126-A1099
Version:	NA
Description:	In contrast to the Requirement 80126-A871, the sequence of logical blocks must strictly adhered during the bootloader update.

Id:	80126-A998
Version:	NA
Description:	It follows the transmission of the new boot loader in the first logical block of the server. After successful CheckMemory Test, the entry point is changed to the new programmed bootloader, which is original value (boot block). The "0xF1DF ECU Programming Information" remains unchanged.

Id:	80126-A999
Version:	NA
Description:	After that, the transfer of further data blocks follows. At least, this is only one datablock - the one which was used to save the old bootloader. Please note that all datablocks which were used to save the original bootlader have to coded in their OWN-IDENT numbers in such way that after the bootloader-Update / Downgrade a new programming of this datablocks takes place.

Id:	80126-A1084
Version:	NA
Description:	The end of the procedure takes place after successful compatibility / consistency check (checkProgrammingDependencies). The Inconsitency Indication is cleared (ECUProgrammingInformation).

Id:	80126-A1000
Version:	NA
Description:	As an alternative of copying the old bootloader in the application or data areas, is the reservation of a second bootloader area to store the new boot loader. Example: Bootloader Block 0 and Block 1 bootloader with 16 KB. Application and data set blocks of 128 KB. Backing up the old bootloader in application or data set blocks is omitted here. There the new boot loader in the previously unused free bootloader block is only flashed, set the entry point to this bootloader block and after successful compatibility / KCC (checkProgrammingDependencies) deleted the Inkonsistenzanzeige (ECUProgrammingInformation).

Id:	80126-A1001
Version:	NA
Description:	After saving the old bootloader and transmission of the new bootloader, no reset of the server is performed by the client.

Id:	80126-A1002
Version:	NA

Description:	If it is required that the new bootloader shall be used for the transmission of the application an dataset blocks an internal reset can be triggered internally (eg. by the CheckMemory-Routine). For this case the server must send a ResponsePending (NRC 0x78) after a CheckMemory-Request of the Bootloader-Block, then do the ECUReset and continue the data transmission for application an dataset blocks with a positive Response of the CheckMemory Request.

Id:	80126-A1085
Version:	NA
Description:	If it is required that the new bootloader shall be used for the transmission of the application an dataset blocks an internal reset can be triggered internally (eg. by the CheckMemory-Routine). For this case the server must send a ResponsePending (NRC 0x78) after a CheckMemory-Request of the Bootloader-Block, then do the ECUReset and continue the data transmission for application an dataset blocks with a positive Response of the CheckMemory Request.

Id:	80126-A1095
Version:	NA
Description:	When the CheckMemory request ist going to be repeated without any other Request (i.e. ereaseMemory, RequestDownloader, TransferDate, RequestTransferExit, ECUReset, etc.) the positive Response to CheckMemory must be repeated. But if there was any other Request the ECU must send the required NRC because it is a sequence error).

Id:	80126-A1003
Version:	NA
Description:	Figure 21 - Example of a Workflow for Bootloader-Update

Id:	80126-A1142
Version:	NA
Description:	8.4.3 Sequence of a 2-staged bootloader exchange (exceptional case)

Id:	80126-A1143
Version:	NA
Description:	A two-staged bootloader update shall only be used if the new bootloader can only be programmed via a "tojan" because the one-staged concept is not supported by the ECU or there were major incompatibilities which would brick the ECU when using the 1-staged concept. This could be for instance the introduction of new signature keys or changes on the memory layout that forbid a later downgrade.

Id:	80126-A1144
Version:	NA
Description:	The following things have to be noticed when using a 2-staged Bootloader-Update:

Id:	80126-A1145
Version:	NA
Description:	Two separate Flashcontainers have to be provided. One for the Bootloader-Exchange and a second fo the new application-SW.

Id:	80126-A1146
Version:	NA
Description:	In version42 a specific order has to prescribed, that it is unmistakable clear that the Bootloader-Container must be flashed first and the application-SW container second.

Id:	80126-A1147
Version:	NA
Description:	The update programming must been executed two times in the specific order specified by version42.

Id:	80126-A1148
Version:	NA
Description:	The bootloader-Update is successful if the CheckProgrammingDependencies-Routine and the ECUReset (hardReset) were positively acknowledged by the Server. The positive Response upon the ECUReset (hardReset) must be sent from the Bootloader. The Server remains afterwards in the Bootloader and the Inconsitency indication of the application-SW is signaled with ECUProgrammingInformation (0xF1DF = 0x44). The Server behaves ab this point in the same way as after a cancellation of flashprogramming.

Id:	80126-A1149
Version:	NA
Description:	After the successfull bootloader-update the server must update the "0xF189 - VW Software Version Number" and return the new VW software version number for an explicit bootloader software.

Id:	80126-A1150

Version:	NA
Description:	A single Bootloader-SW without a valid Application-SW must return DataIdentifier "0xF189 - VW Software Version Number" the versions [B001] to [B999] directly from the Bootloader.

Id:	80126-A1151
Version:	NA
Description:	8.4.3.1 Requirements of a 2-stage bootloader update with incompatibility

Id:	80126-A1152
Version:	NA
Description:	When doing a bootloader exchange with incompatibilities (means "old" application-SW is not functional with the "new" bootloader) the following requirements have to be followed additionally:

Id:	80126-A1153
Version:	NA
Description:	After a successful update of the new bootloader the VW Spare Part Number (0xF187) must be changed. For example: from 8W0907063A (old) to 8W0907063B (new)

Id:	80126-A1154
Version:	NA
Description:	The new VW Spare part number (0xF187) must also a get a changed calculation algorithm for the SecurityAccessTypes 0x11/0x12 to avoid programming of an "old" Application-SW in combination with a "new" bootloader and vice versa.

Id:	80126-A1011
Version:	NA
Description:	8.5 Optional 1MBaud-Baudrateswitching

Id:	80126-A1011a
Version:	NA
Description:	This chapter describes the option to increase the CAN-Baudrate with "87hex-LinkControl" to 1MBaud for the flashing.

Id:	80126-A1023
Version:	NA

Description:	The 1MBaud programming option can be used in the ECU's standalone operation or inside the car with all other ECUs on the bus if all of them support the 1MBaud rate switch.

Id:	80126-A1024
Version:	NA
Description:	The switch-to- 1MBaud feature can only be used by servers which support this feature by themselves (e.g. Motor-ECU)

Id:	80126-A1012
Version:	NA
Description:	8.5.1 Requirements

Id:	80126-A1013
Version:	NA
Description:	The switch of the CAN Baudrate has to be performed according to [Q-LAH 80124]

Id:	80126-A1014
Version:	NA
Description:	The switch of the CAN Baudrate is performed by the client (e.g the tester) in the ExtendedDiagnosticSession (0x03).

Id:	80126-A1015
Version:	NA
Description:	If a baudrate switch is not possible, the request has to be denied with NRC 0x22 (ConditionsNotCorrect).

Id:	80126-A1019
Version:	NA
Description:	During all NonDefaultSessions the new baudrate is effective. Only by a change into the DefaultSession (0x01) the Default-Baudrate becomes active.

Id:	80126-A1016
Version:	NA
Description:	8.5.2 Sequence

Id:	80126-A1017

Version:	NA
Description:	In the first step it is checked if the baudrate requested by the client is supported by the server. The request to the server is done by the client with Service 0x87hex and Subfunction 0x01 (VerifyBaudrateTransitionWithFixedBaudrate). The parameter BaudrateIdentifier contains the requested Baudrate (i.e. 0x14h for CAN with 1 MBaud). The SuppressPositiveResponseBit has to be set to 0 (Null) for this. The positive answer of the server to client does not cause a baudrate switch because it is only the information if it would be possible to switch to.

Id:	80126-A1018
Version:	NA
Description:	The second step is only possible if the first step was directly ahead and the server acknowledged with a positive answer. Then, the client requests the baudrate switch with the service 0x87hex and the Subfunction 0x83 and Subfunction 0x03 with set SuppressPositiveResponseBit. The Baudrateidentifier must get the previously requested value (e.g. 0x13 for 1 MBaud). A positive answer from the server is not necessary and the switch to the new baudrate is performed as quick as possible to the new value. The switching time must not exeed P2ServerFunctional (50ms). The Client switches its baudrate after the minumtime P2ClientFunctional (150ms).

Id:	80126-A656
Version:	NA
Description:	9 Referenced documents

Id:	80126-A657
Version:	NA
Description:	The following documents cited in this standard are necessary for its application. In this Section, terminological inconsistencies may occur as the original titles are used.

Id:	80126-A661
Version:	NA
Description:	[VW-ISO-TP] ISO Transportprotokoll - ISO 15765-2 Networklayer

Id:	80126-A662
Version:	NA
Description:	[Q-LAH 80124] Q-LAH 80124 - , LAH.DUM.909.G Unified Diagnostic Services Protocol and OBD Services

Id:	80126-A663
Version:	NA
Description:	[Q-LAH 80125] Q-LAH 80125 - LAH.DUM.909.H Identification of Electronic Vehicle Systems

Id:	80126-A1343
Version:	NA
Description:	[Q-LAH 80127] Q-LAH 80127, LAH.DUM.909.B Diagnose verteilter Systeme

Id:	80126-A666
Version:	NA
Description:	[Q-LAH 80128] VW80128 - Teil 1+2 -Author's Guideline for the Creation of ODX Data Sets Q-LAH 80128 - Teil 3 - V4.0 Specification for Flashcontainer ODX-Flash PDX-Flash

Id:	80126-A1191
Version:	NA
Description:	[Q-LAH ORU] OnlineRemoteUpdate Feature Performance Specification,LAH.-DUM.035.E

Id:	80126-A712
Version:	NA
Description:	[HIS SECMOD] HIS Security Module Specification, Thomas Miehling et. al.

Id:	80126-A669
Version:	NA
Description:	[FDS] Flashdatensicherheit für UDS Steuergeräte, AH DUM.000.AD.1 To be implemented only in consultation with the purchaser.

Id:	80126-A1320
Version:	NA
Description:	[VW 80180-2] VW 80180-2 Use of Cryptography in Vehicles Implementation of Protection Target Classes for Sensitive Data and Cryptographic Protective Measures

Id:	80126-A1321
Version:	NA

Description:	[NM-CAN] Network Management High Group Performance Specification, LAH.-DUM.857.AK

Id:	80126-A1325
Version:	NA
Description:	[NM-FlexRay] FlexRay Group Performance Specification, LAH.DUM.907.S

Id:	80126-A928
Version:	NA
Description:	[Diagglossar] DiagnoseglossarLAH.DUM.909.D List of All Diagnostic-Specific Terms – continuously updated - Continuously improved

Id:	80126-A935
Version:	NA
Description:	The versions of documents indicated here have been used for the document comparison. If agreed within the project, newer versions can also be used. The use of older versions is strongly discouraged because this can result in contradictions and information loss, which will make it difficult to comply with the requirements set forth in this document.

Id:	80126-A670
Version:	NA
Description:	9.1 Relation of Q-LAH 80126

Id:	80126-A671
Version:	NA
Description:	Figure 22 - Documents and Standards

Id:	80126-A718
Version:	NA
Description:	10 Appendix A – Security Access

Id:	80126-A781
Version:	NA
Description:	10.1 Scope

Id:	80126-A782
Version:	NA

Description:	This Appendix is aimed at developers of boot loaders for electronic control units and developers of programming devices of the purchaser.

Id:	80126-A783
Version:	NA
Description:	It describes the function and coding of the algorithm used to authorize a client by means of "SecurityAccess" for server programming.

Id:	80126-A929
Version:	NA
Description:	Note: It is not necessary to implement a mini-instruction set interpreter in the control unit in order to implement the SecurityAccess algorithm. It is sufficient to map the chosen algorithm through an efficient implementation.

Id:	80126-A719
Version:	NA
Description:	10.2 Requirements

Id:	80126-A720
Version:	NA
Description:	10.2.1 Mini-instruction set

Id:	80126-A721
Version:	NA
Description:	The specifications for randomizing the seed and key (hereinafter referred to as "algorithm" or "SA2") are defined by the supplier.

Id:	80126-A722
Version:	NA
Description:	The required mini-instruction set is limited to a few basic computer operations.

Id:	80126-A723
Version:	NA
Description:	It is necessary to disclose the algorithm to the programming device. This occurs in the ODX flash container as an instruction that can be read and executed by the programming devices.

Id:	80126-A724

Version:	NA
Description:	10.2.2 Arithmetic instructions

Id:	80126-A725
Version:	NA
Description:	The following generally applies to the arithmetic instructions listed below:

Id:	80126-A726
Version:	NA
Description:	- Seed and key each consist of 32 bits

Id:	80126-A727
Version:	NA
Description:	- The seed is a random value that is computed by the server.

Id:	80126-A728
Version:	NA
Description:	- All comutation is binary.

Id:	80126-A729
Version:	NA
Description:	- On start the operand is loaded with the seed.

Id:	80126-A730
Version:	NA
Description:	- The individual instructions carry out logical or arithmetic operations with the operand and output, if applicable, an additional value.

Id:	80126-A731
Version:	NA
Description:	- The operand is the result of the previous computation in each case.

Id:	80126-A732
Version:	NA
Description:	- If the operation exceeds 32 bits, the carry bit is set.

Id:	80126-A733

Version:	NA
Description:	- At the end, the operand contains the key.

Id:	80126-A734
Version:	NA

Id:	80126-A735
Version:	NA

Id:	80126-A736
Version:	NA

Id:	80126-A737
Version:	NA

Id:	80126-A738
Version:	NA

Id:	80126-A739
Version:	NA
Description:	TabLe 73 - Calculation or XOR function

Id:	80126-A740
Version:	NA
Description:	10.2.3 Sequence instructions

Id:	80126-A741
Version:	NA
Description:	In addition to the arithmetic instructions, additional instructions are required for the computation sequence.

Id:	80126-A742
Version:	NA
Description:	For I equals "value" to 1 Indicates how often an instruction or a series of instructions must be run through. "Value" = Number of repetitions (1 to 255). Special case: I = 0 This case was not intended and must not occur. If this case occurs nevertheless, due to an error in the script or a miscalculation, the BLF must ensure that the control unit does not become unusable (total crash). Next termi-

	nates the loop. It must be run through the exact number of times as defined by the number of operations.

Id:	80126-A743
Version:	NA
Description:	Jump, if Carry null [BCC "value"] Results in a branch in the sequence that is determined by the preceding computation. The "value" variable contains the number of memory locations (bytes) to be skipped. This assumes that the program counter is at the entry address of the next instruction. "Value" has a size of 1 byte and is always positive (1 to 255).

Id:	80126-A744
Version:	NA
Description:	Always jump [BRA "value"] Can be used to skip certain sequence sections. The "value" variable contains the number of memory locations (bytes) to be skipped. This assumes that the program counter is at the entry address of the next instruction. "Value" has a size of 1 byte and is always positive (1 to 255).

Id:	80126-A745
Version:	NA
Description:	Completed indicates that the operand can be output as a key.

Id:	80126-A746
Version:	NA
Description:	10.2.4 Coding of the instruction set

Id:	80126-A747
Version:	NA
Description:	Table VW80126-A748 below defines the hexadecimal numbers for the instructions. The coding was done in such a way that the arithmetic instructions contain an even number of bits and the sequence instructions contain an odd number of bits.

Id:	80126-A748
Version:	NA

Id:	80126-A749
Version:	NA

Description:	10.2.5 Further requirements

Id:	80126-A750
Version:	NA
Description:	The use of a trivial Seed and Key algorithm (SA2 script) or examples from the purchaser's Performance Specifications is not permitted.

Id:	80126-A751
Version:	NA
Description:	The algorithm must contain at least one "For/Next loop" with a minimum of five repetitions. At least one "EOR", "RSR", or "RSL", one "ADD" or "SUB", and one "BCC" must be executed within the loop.

Id:	80126-A1128
Version:	NA
Description:	Threading of multiple loops is not allowed. Usage of the same loop-end (next, opcode 0x48) from multiple loops is also not allowed.

Id:	80126-A752
Version:	NA
Description:	The "BCC" instruction must only be used directly after an instruction that changes the carry bit (RSL, RSR, ADD, or SUB).

Id:	80126-A1129
Version:	NA
Description:	If jump calls BCC and BRA are used within loops, there jump destination must be inside the loop. It must be noted that only operators (opcodes) are skipped.

Id:	80126-A1131
Version:	NA
Description:	The computing period for the SA2 algorithm must not exeed 500ms.

Id:	80126-A753
Version:	NA
Description:	The server must send a quasi-random 32-bit number to the client for SecurityAccess

Id:	80126-A754

Version:	NA
Description:	Die Verwendung von berechneten Zahlen aus Vorgängerwerten oder Zahlen aus Listen ist unzulässig.

Id:	80126-r
Version:	NA
Description:	The Seed and Key algorithm implemented in the boot loader must exhibit the behavior specified in the SA2 script of the associated data container for the entire value range.

Id:	80126-A756
Version:	NA
Description:	The interpreter in the programming device for the mini-instruction set must be set up to be resistant to faulty algorithms.

Id:	80126-A757
Version:	NA
Description:	The "invalid mini-instruction" error must be detected.

Id:	80126-A758
Version:	NA
Description:	The "invalid jump" error ("BRA" or "BCC" jumps to an illegal point inside or outside the script) must be detected.

Id:	80126-A759
Version:	NA
Description:	The "missing instruction argument" error must be detected.

Id:	80126-A760
Version:	NA
Description:	If the boot loader detects an error in the algorithm, it must report the "invalid key" error to the programming device.

Id:	80126-A761
Version:	NA
Description:	If the programming device detects an error in the algorithm, it must cancel the programming no later than just before the "Send key" instruction and display an "Error in data container" message.

Id:	80126-A762
Version:	NA
Description:	For control units with incompatible hardware versions, different algorithms must be used to prevent the introduction of an incompatible software.

Id:	80126-A763
Version:	NA
Description:	For identical devices that are not compatible with earlier versions, due to changes in bootloader versions or EEPROM layouts, different algorithms must be used.

Id:	80126-A764
Version:	NA
Description:	The SA2 script must not change during the life cycle of a hardware part number (supplier part number), so that the programming of compatible software is not prevented on the due to different algorithms.

Id:	80126-A765
Version:	NA
Description:	10.2.6 Examples

Id:	80126-A766
Version:	NA
Description:	10.2.6.1 Simple example

Id:	80126-A767
Version:	NA
Description:	A possible sequence is shown as an example.

Id:	80126-A768
Version:	NA
Description:	Script: 0x68,0x05,0x81,0x4A,0x05,0x87,0x0A,0x22,0x12,0x89,0x49,0x4C

Id:	80126-A769
Version:	NA

Id:	80126-A770

Version:	NA
Description:	Figure 23 - Sequence for the calculation of the key according to example in 80126-A769

Id:	80126-A771
Version:	NA
Description:	The following "random values", together with the example algorithm from 80126-A769 and Figure 80126-A770 may be used for purposes of testing the interpreter for the Seed and Key algorithm:

Id:	80126-A772
Version:	NA
Description:	Calculation with Examplealgorithmn from Table 80126-A769 and Seed 0xAC13491B:

Id:	80126-A773
Version:	NA

Id:	80126-A774
Version:	NA
Description:	10.2.6.2 More comprehensive example

Id:	80126-A775
Version:	NA
Description:	Script = "0x68, 0x06, 0x82, 0x84, 0x92, 0x12, 0x02, 0x73, 0x4A, 0x01, 0x81, 0x87, 0x12, 0x20, 0x01, 0x07, 0x82, 0x93, 0x05, 0x08, 0x18, 0x29, 0x49, 0x81, 0x4C"

Id:	80126-A776
Version:	NA

Id:	80126-A777
Version:	NA
Description:	The following "random values", together with the example algorithm from Table 6 and Figure VW80126-A779 may be used for purposes of testing the interpreter for the Seed and Key algorithm

Id:	80126-A778

Version:	NA
Description:	Seed: 0xEF29D38C Key: 0x445BEAE4 (see Calculation in Table 80126-A780) Seed: 0xAC13491B Key: 0x2658BF04 Seed: 0x198F23CE Key: 0x0961909F Seed: 0xFA9E0138 Key: 0x57E2E51C Seed: 0x27B3EA04 Key: 0x1B9673BA

Id:	80126-A779
Version:	NA
Description:	Figure 24 - Program Flow for the calculation of a key according the example from table 80126-A776

Id:	80126-A780
Version:	NA
Description:	Computation with example algorithm from Table 80126-A776 and Seed 0xEF29D38C: Table 77 - Computation Example

Id:	80126-A792
Version:	NA
Description:	11 Appendix B – Programming states

Id:	80126-A916
Version:	NA
Description:	Figure VW80126-A793 shows the states within the control unit as they occur during programming. It also shows the dependency of the diagnostic services during the restart following programming.

Id:	80126-A793
Version:	NA
Description:	Figure 25 - Programming States

Id:	80126-A875
Version:	NA
Description:	12 Appendix C – Identification data during programming

Id:	80126-A876
Version:	NA
Description:	By way of example, this Section describes the coding of a control unit for the following cases: • Delivery without driving program (only boot loader present) •

	Canceled flash operation for 2nd block • Flashing successful • Canceled flash operation for 3rd block

Id:	80126-A878
Version:	NA
Description:	The following control unit structure is assumed for this example: • 3 logical blocks with the following distribution: - Boot loader - Application - Data • Boot-loader not replaceable • No flash data security

Id:	80126-A917
Version:	NA
Description:	Example:

Id:	80126-A879
Version:	NA
Description:	The coding in the as-received condition without driving program is shown to start. In this case, the control unit still has no SW version (0xF189) and no Volkswagen part number (0xF187). For this reason, the VWECUHWNumber is output as the VWSparePartNumber (0xF187).

Id:	80126-A880
Version:	NA
Description:	The following Table depicts the same control unit after a canceled flash operation in the third logical block.

Id:	80126-A881
Version:	NA
Description:	The following Table presents the logistical data of the control unit after a successful flash operation with Version 0020.

Id:	80126-A930
Version:	NA
Description:	The following Table presents the logistical data of the control unit after a cancelled flash operation in the 3rd logical block when flashing the SW 0030. In this case, the control unit continues to output the VWSoftwareVersionNumber 0020 (of the last valid programmed software).. Table 78 - Identificationdata, Examples

Id:	80126-A1089
Version:	NA

Description:	13 Appendix D - Example-Trace

Id:	80126-A1090
Version:	NA
Description:	The trace in 80126-A1091 exemplifies a flash operation of a control device with two logic blocks (boot loader and application). First, a driver is transferred, then the data of the application software to be flashed.

Id:	80126-A1091
Version:	NA

Id:	80126-A1216
Version:	NA
Description:	An introduction to the subject of downgrade protection can be found in section 3.5.6 "Downgrade protection."

Id:	80126-A1217
Version:	NA
Description:	Downgrade protection is ensured with the signing of all logical blocks of a flash container and is therefore conditional on flash data security (FDS) or a similar mechanism. This makes it impossible to change version numbers in logical blocks without the necessary authorization.

Id:	80126-A1227
Version:	NA
Description:	Resetting of downgrade protection is conditional on protection of vehicle diagnostics (SFD), level 1.

Id:	80126-A1220
Version:	NA
Description:	This section describes how the boot loader and the flash memory are safeguarded from downgrading by insertion of a new version number, the downgrade protection version number ("RFS version number" or "RfsV" for short), for each logical block, parallel to the existing values, such as time stamp of the last flashing process (programming date DID 0xF15B), etc.

Id:	80126-A1221
Version:	NA

Description:	The DID 0x0189-VW_logical_block_downgrade_protection_versions must be implemented as per [Q-LAH 80125].

Id:	80126-A1226
Version:	NA
Description:	The RFS version numbers must be interpreted as four-digit decimal numbers and compared accordingly.

Id:	80126-A1223
Version:	NA
Description:	If one of the logical blocks in the server has an RFS version number outside the valid value range, then the application software is not executable and is evaluated as inconsistent.

Id:	80126-A1224
Version:	NA
Description:	The RFS version number must be present in each logical block in the server or flash container.

Id:	80126-A1225
Version:	NA
Description:	This also applies to logical blocks in which the boot loader is located.

Id:	80126-A1229
Version:	NA
Description:	In order to ensure downgrade protection, the server must store the currently programmed RFS version number, RfsVcurrent, and the minimum supported RFS version number, RfsVminimum, for each logical block. The application may only be executed if the current RFS version number is greater than or equal to the minimum RFS version number for each logical block in the server. When the application is started successfully after a flashing process, the current RFS version number is adopted as the new minimum RFS version number, so that downgrading to a previous RFS version number is no longer directly possible (see section 14.3 "Flashing process" for the precise behavior). However, if it is necessary to flash to a previous RFS version number, then this is possible in exceptional cases via SFD Level 1 Extended User (see section 14.4 "Resetting the downgrade protection").

Id:	80126-A1230

Version:	NA
Description:	Each of the RFS version numbers is assigned to one logical block in the server, and they are independent of each other.

Id:	80126-A1231
Version:	NA
Description:	For each logical block b of the server, an RFS version number that the logical block on the server currently possesses must be stored persistently and authentically as per [VW 80180-2] and designated as the RfsVcurrent(b) field.

Id:	80126-A1232
Version:	NA
Description:	For each logical block b on the server, a minimal flashable RFS version number must be stored persistently and authentically as per [VW 80180-2] and designated as the RfsVminimum(b) field.

Id:	80126-A1233
Version:	NA
Description:	The RfsVminimum RFS version numbers of the logical blocks in the server must never be decremented or reset (apart from in exceptional cases by a diagnostic routine after unlocking SFD level 1 in an extended role; see section 14.4 "Resetting the downgrade protection").

Id:	80126-A1234
Version:	NA
Description:	Example representation of the RFS version numbers for all logical blocks of the server

Id:	80126-A1235
Version:	NA
Description:	Downgrade protection must not be circumvented, even if the server supports other means of updating software (e.g., debugger flashing) apart from downgrade protection and UDS flashing.

Id:	80126-A1236
Version:	NA
Description:	The RFS version numbers stored in the server may only be changed by means of the mechanisms described in this document.

Id:	80126-A1237
Version:	NA
Description:	The RfsVcurrent RFS version numbers may only be updated by the server after a successfulflashing process of a logical block in the UDS server of the boot loader.

Id:	80126-A1238
Version:	NA
Description:	The RfsVminimum RFS version numbers may only be updated by the server after a successful application software start-up.

Id:	80126-A1239
Version:	NA
Description:	Even a boot loader update must not lead to this data being overwritten.

Id:	80126-A1240
Version:	NA
Description:	The RfsVcurrent(b) field may only adopt the value of the actual RFS version number of the logical block b in the server, to which this field is assigned.

Id:	80126-A1241
Version:	NA
Description:	On delivery, the RfsVcurrent and RfsVminimum fields must correspond to the downgrade protection version numbers of the respective logical blocks of the software located on the server. The default value for RfsVcurrent and RfsVminimum is "0001".

Id:	80126-A1329
Version:	NA
Description:	The initial value "0001" means that the server supports RFS, but this has yet to be activated.

Id:	80126-A1242
Version:	NA
Description:	The RFS version numbers of a new software version (RfsVLB) must be greater than or equal to the RFS version numbers (RfsVminimum) in the server.

Id:	80126-A1246

Version:	NA
Description:	Analogously to DataIdentifier "0xF1AB-VW Logical Software Block Version," the RfsVminimum(b) and RfsVcurrent(b) RFS version numbers have DataIdentifier "0x0189-VW_logical_block_downgrade_protection_versions" available for each logical blockb in the server (see [Q-LAH 80125]).

Id:	80126-A1249
Version:	NA
Description:	In order to prevent incorrect or obsolete software from being flashed onto the server and making it impermissible for the application to be started by the boot loader, the deletion and programming with a pseudo flash driver must be enabled within an ECUProgrammingSession (see [Q-LAH 80124]) before a flashing process. This means that flashing is not possible without a valid pseudo flash driver with valid data. The pseudo flash driver contains all logical block IDs and associated downgrade protection (RFS) version numbers, which are intended to be flashed, so it can be determined in advance whether the flashing process can be completed successfully.

Id:	80126-A1250
Version:	NA
Description:	Location of the pseudo flash driver in the flash container (example)

Id:	80126-A1251
Version:	NA
Description:	The pseudo flash driver must be signed. The signature of the pseudo flash driver uses the same key pair of the same FDS ID as that described in section 3.5.4 "Integrity/authenticity check (checkMemory)".

Id:	80126-A1252
Version:	NA
Description:	The signature is located as per [Q-LAH 80128-3] in the flash container of the pseudo flash driver.

Id:	80126-A1253
Version:	NA
Description:	The pseudo flash driver is handled in the same way as an ordinary logical block in the flash container, i.e., it can also be compressed and encrypted.

Id:	80126-A1254

Version:	NA
Description:	All logical blocks of a flash container must be referenced in the pseudo flash driver with the respective logical block ID (BID) and RFS version number.

Id:	80126-A1255
Version:	NA
Description:	The logical blocks announced in the pseudo flash driver can contain a superset of the logical blocks actually flashed in the flashing process.

Id:	80126-A1256
Version:	NA
Description:	It is recommended to keep software and calibration independent of each other in separate flash containers.

Id:	80126-A1257
Version:	NA
Description:	For the purpose of subsequent processing and verification of data during the flashing process, the pseudo flash driver must be stored in volatile memory in the server.

Id:	80126-A1258
Version:	NA
Description:	If flashing is interrupted, the pseudo flash driver must be discarded.

Id:	80126-A1259
Version:	NA
Description:	Flashing is interrupted if the consistency bit of a logical block in the server is not reset to the "consistent" state during the flashing process. This can be the case, for example, in the event of errors in the flashing process or if the power supply is interrupted.

Id:	80126-A1261
Version:	NA
Description:	The pseudo flash driver must contain the following fields: (LIST)

Id:	80126-A1340
Version:	NA

Description:	For the numerical elements within the pseudo flash driver, the most significant byte must be stored first every time (“Network Byte Order” resp. “Big-Endian-Format”).

Id:	80126-A1264
Version:	NA
Description:	The pseudo flash driver begins with an identification text, the "pseudo flash driver identification" or "PTI" for short, which consists of two parts: the two-byte length specification PTIL and the following PTID text with a length of PTIL.

Id:	80126-A1265
Version:	NA
Description:	The PTI field serves the purpose of identifying the pseudo flash driver and indicating the different data structures that result from the different versions of the implementation.

Id:	80126-A1266
Version:	NA
Description:	The PTID field is ASCII-coded, whereby only displayable characters (space 32 through to tilde 126) are allowed.

Id:	80126-A1267
Version:	NA
Description:	The text of the "pseudo flash driver identification" PTID is located in the pseudo flash driver from byte 2 to byte PTIL + 1.

Id:	80126-A1270
Version:	NA
Description:	The pseudo flash driver contains an "RFS block version table" or "BVT" for short, for an arbitrary number of entries, "BVTA", which each define a logical block ID "BID" and associated RFS version number "RfsVPFT" that is intended to be flashed for this block.

Id:	80126-A1271
Version:	NA
Description:	The "RFS block version table" is checked by the boot loader to prevent a flashing process from being performed unintentionally with obsolete RFS version numbers.

Id:	80126-A1272
Version:	NA
Description:	The BVTA field has a value range of 1 to 65 535.

Id:	80126-A1273
Version:	NA
Description:	The "RFS block version table" is located in the pseudo flash driver in the area from byte (PTIL + 4) to byte (PTIL + BVTA * (4 + BIDLEN) + 3).

Id:	80126-A1275
Version:	NA
Description:	The data of a pseudo flash driver is considered valid if the following validation criteria are met

Id:	80126-A1276
Version:	NA
Description:	Validation criterion 1: All fields of the pseudo flash driver must lie within the valid value ranges described in sections 3.1.1 and 3.1.2.

Id:	80126-A1277
Version:	NA
Description:	Validation criterion 2: The pseudo flash driver must contain at least one logical block ID with associated RFS version number in the BVT.

Id:	80126-A1278
Version:	NA
Description:	Validation criterion 3: A particular block ID may only occur once in the block version table of the pseudo flash driver.

Id:	80126-A1279
Version:	NA
Description:	Validation criterion 4: Each block ID in the block version table of the pseudo flash driver must have a corresponding block ID in the server software.

Id:	80126-A1280
Version:	NA
Description:	Validation criterion 5: Each of the RFS version numbers of the logical blocks, which are stored in the pseudo flash driver, must be greater than or equal to the

	minimum RFS version number of the corresponding logical blocks, RfsVminimum, stored on the server.

Id:	80126-A1281
Version:	NA
Description:	Validation criterion 6: In order to ensure conformity with this Performance Specification, the PTID field must correspond to the text "RFSV1.0" and have a length, "PTIL", of 7 bytes.

Id:	80126-A1282
Version:	NA
Description:	The signature of the pseudo flash driver must be valid to prevent the pseudo flash driver from being discarded. This functionality is not changed by RFS.

Id:	80126-A1312
Version:	NA
Description:	The following figure illustrates the extensions that must be incorporated into the flashing process by downgrade protection and that are described in this section.

Id:	80126-A1286
Version:	NA
Description:	(A) If a previously transmitted pseudo flash driver does not meet validation criteria 1 through 4 and 6 described in section 14.2.2 "Validation criteria," then the CheckMemory routine must acknowledge with response "invaldRfsDriver" (0x80), and the PFT must be discarded.

Id:	80126-A1287
Version:	NA
Description:	(A) If a previously transmitted pseudo flash driver does not meet validation criterion 5 described in section 14.2.2 "Validation criteria," then the CheckMemory routine must acknowledge with response "invaldRfsVersion" (0x81), and the PFT must be discarded.

Id:	80126-A1289
Version:	NA
Description:	(B) If a valid pseudo flash driver has yet to be received in an ECUProgrammingSession, the UDS server in the boot loader must reject every deletion request (EraseMemory) and acknowledge with response "missingRfsDriver" (0x80).

Id:	80126-A1290
Version:	NA
Description:	(C) If a valid pseudo flash driver has yet to be received in an ECUProgrammingSession, the UDS server in the boot loader must reject every programming request (RequestDownload) and acknowledge with negative response code "uploadDownloadNotAccepted" (0x70).

Id:	80126-A1291
Version:	NA
Description:	(B) If a deletion request (EraseMemory) is received after the receipt of a valid pseudo flash driver, but the logical block to be deleted in the server is not a logical block that has been announced in the pseudo flash driver, then the boot loader must reject the deletion request (EraseMemory) and acknowledge with response "invalidRfsBlockID" (0x81).

Id:	80126-A1292
Version:	NA
Description:	(C) If a programming request (RequestDownload) is received after the receipt of a valid pseudo flash driver, but the specified logical block of the server is not a logical block that has been announced in the pseudo flash driver, then the boot loader must reject the programming request (RequestDownload) and acknowledge with negative response code "uploadDownloadNotAccepted" (0x70).

Id:	80126-A1293
Version:	NA
Description:	(C) Following completion of the programming process (RequestDownload) of a logical block b on the server, the RFS version number, RfsVLB, of this block must be copied to field RfsVcurrent(b).

Id:	80126-A1294
Version:	NA
Description:	(C) If a logical block without an RFS version number is flashed onto the server, this must be detected by the boot loader, and the RfsVcurrent field must be set to a value of "invalid" (0000).

Id:	80126-A1295
Version:	NA
Description:	(D) If a logical block has been programmed, whose RFS version number does not match the RFS version number announced for this logical block

	in the server, then the CheckMemory() routine must return a response of "invalidRfsVersion" (0x81) and continue to evaluate the status of the logical block in the server as inconsistent (0xF1DF programming status = inconsistent).

Id:	80126-A1297
Version:	NA
Description:	If the pseudo flash driver is not available to compare this with the RFS version number, then the CheckProgrammingDependencies() routine must return a response of "missingRfsDriver" (0x05).

Id:	80126-A1298
Version:	NA
Description:	(E) If the RFS version number of a logical block b in the server, RfsVcurrent(b), is less than the value RfsVminimum(b) or a logical block has been programmed, whose RFS version number does not match the RFS version number announced for this logical block in the server, then the CheckProgrammingDependencies() routine must return a response of "invalidRfsVersion" (0x06).

Id:	80126-A1299
Version:	NA
Description:	(E) In addition to the criteria specified in section 3.5.5 "Compatibility/consistency check" and section 6.7.4 "0xFF01-Check Programming Dependencies," the software from routine "CheckProgrammingDependencies()" must only be evaluated as consistent (0xF1DF programming status = consistent) if: During a flashing process, all or a subset of the logical blocks announced in the PFT have been successfully flashed with exactly the same RFS version number as was announced in the PFT, and · The RfsVcurrent(b) values are greater than or equal to the RfsVminimum(b) values of each logical block b located in the server. 80126-A1300

Id:	80126-A1302
Version:	NA
Description:	The server must implement the "0x065E-Clear_downgrade_protection_data" routine in order for resetting of the RFS to be possible.

Id:	80126-A1328
Version:	NA
Description:	If resetting of the downgrade protection is not possible in the ECU, this must be agreed upon with Production and After-Sales Service.

Id:	80126-A1303
Version:	NA
Description:	The "0x065E-Clear_downgrade_protection_data" routine can only be executed after unlocking SFD level 1 in an extended role as per [SFD].

Id:	80126-A1304
Version:	NA
Description:	The "0x065E-Clear_downgrade_protection_data" routine sets the RfsVminimum(b) field for each logical block of the server to the first valid value "0001".

Id:	80126-A1305
Version:	NA
Description:	Resetting of the downgrade protection may only be performed with the "0x065E-Clear_downgrade_protection_data" routine.

Id:	80126-A1307
Version:	NA
Description:	Each incrementation of the software RFS version number must be made in consultation with the purchaser. The department responsible for flashing must be informed.

Id:	80126-A1308
Version:	NA
Description:	The RFS version number must be incremented in consultation with the purchaser in the case of security-relevant changes to the software.

Id:	80126-A1309
Version:	NA
Description:	The supplier of 3 -party software is responsible for its RFS versioning, reliable detection, and the fixing of security vulnerabilities. This must be contractually regulated elsewhere.

Id:	80126-A672
Version:	NA
Description:	14 History

Id:	80126-A674

Version:	NA

Id:	80126-A675
Version:	NA
Description:	Erweiterungen in Version 2.1 zu Version 2.0:

Id:	80126-A1081
Version:	NA

Id:	80126-A1161
Version:	NA
Description:	Erweiterungen in Version 2.2 zu Version 2.1:

Id:	80126-A1162
Version:	NA

Id:	80126-A1341
Version:	NA

Id:	80126-A1342
Version:	NA