

# Flash Bootloader OEM

## Technical Reference

Vector UDS (SLP3) - Download Sequence Specification

Version 5.2

Authors	visach, virjijn
Status	Released

## Document Information

### History

Author	Date	Version	Remarks
visach	2023-01-05	5.0	Update to support UserManual 3
visach	2024-02-28	5.1	Update to recent template, fix smaller issues
virjijn	2024-10-22	5.2	Described download to inactive partition

### Reference Documents

No.	Source	Title	Version
[1]	Vector	Flash Bootloader User Manual	3.0.0
[2]	ISO	14229 Road Vehicles – Unified diagnostic services (UDS) Part 1: Specification and Requirements	2005/2013/...



#### Caution

We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector's release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

**Contents**

**1 Introduction..... 5**

**2 Download Concept ..... 6**

2.1 Overview..... 6

2.1.1 Pre-Programming Step ..... 6

2.1.2 Programming Step ..... 7

2.1.3 Post-Programming Step..... 10

2.2 Detailed Description of Diagnostic Services ..... 10

2.2.1 Diagnostic Session Control ..... 11

2.2.2 ECU Reset..... 11

2.2.3 Read Data By Identifier ..... 12

2.2.4 Security Access..... 12

2.2.4.1 Request Seed ..... 13

2.2.4.2 Send Key ..... 13

2.2.5 Communication Control..... 14

2.2.6 Write Data By Identifier ..... 14

2.2.7 Routine Control..... 15

2.2.7.1 Check Routine ..... 15

2.2.7.2 Check Programming Preconditions..... 15

2.2.7.3 Erase Memory ..... 16

2.2.7.4 Check Programming Dependencies..... 17

2.2.7.5 Force Boot Mode ..... 17

2.2.7.6 Perform Backup ..... 17

2.2.7.7 Activate Software..... 18

2.2.7.8 Perform Rollback ..... 18

2.2.8 Request Download..... 19

2.2.9 Transfer Data ..... 19

2.2.10 Request Transfer Exit..... 19

2.2.11 Tester Present..... 20

2.2.12 Control DTC Setting..... 20

2.2.13 Negative Response Codes..... 20

**3 Glossary and Abbreviations ..... 21**

3.1 Glossary ..... 21

3.2 Abbreviations ..... 21

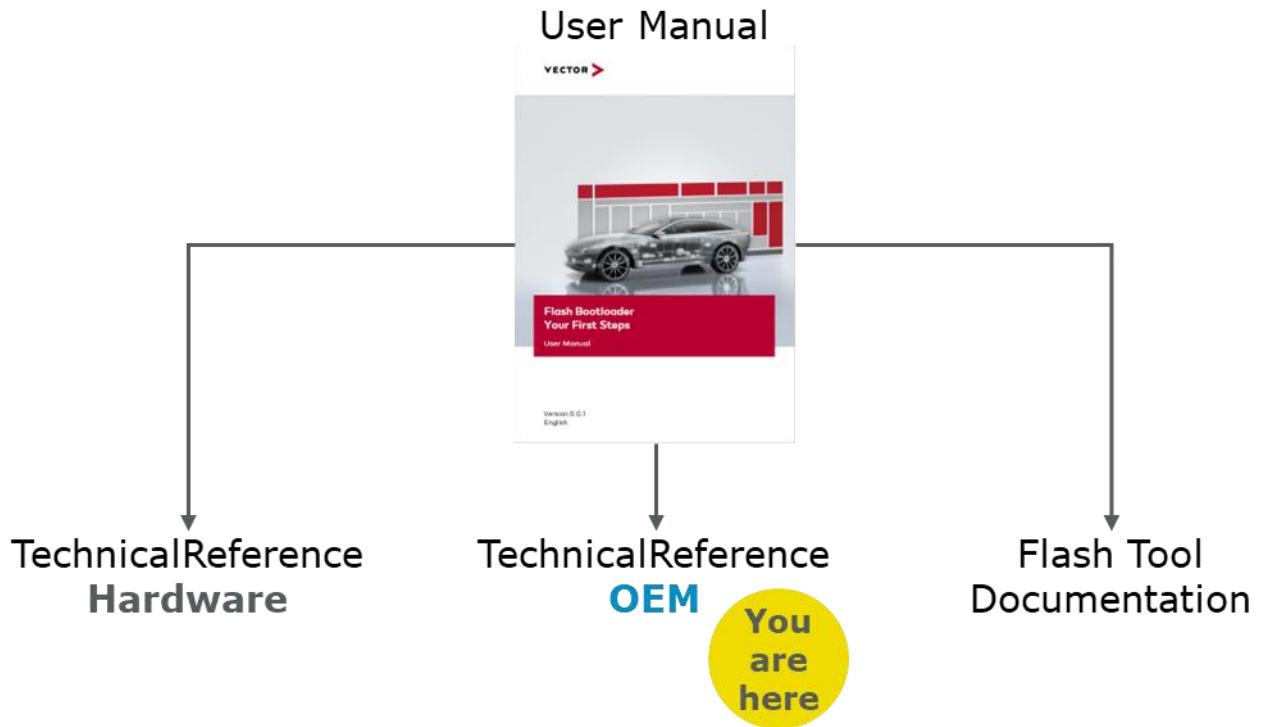
**4 Contact..... 22**

**Illustrations**

Figure 2-1	Pre-Programming Step .....	7
Figure 2-2	Programming Step.....	8
Figure 2-3	Data Transmission .....	9
Figure 2-4	Post-Programming Step .....	10

# 1 Introduction

This document covers the OEM-specific particularities of the flash bootloader. It complements the explanations started in the user manual with OEM-specific details. All references there are resumed here in this document again and explained in detail.



## 2 Download Concept

The Vector UDS (SLP3) download sequence is a download sequence based on various examples made by HIS and ISO 14229. It is an OEM independent interpretation of a download sequence with some flexibility.

### 2.1 Overview

The flash tool (Diagnostic Tester) uses the UDS protocol to communicate with the ECU. This tester can be located inside the vehicle or used as an external tester. The flash bootloader processes the UDS services sent by the tester and supports the diagnostic services needed to update an ECU.

The download process can be divided into three basic steps:

- ▶ **Pre-Programming Step:** Preparation of the vehicle network and target ECU for flashing. If an ECU is updated, this part of the flashing sequence is received and handled by the application software.
- ▶ **Programming Step:** Actual update sequence of the ECU. This part of the update sequence is received by the flash bootloader of the updated ECU only.
- ▶ **Post-Programming Step:** This step sets the vehicle network back to normal operation. This step addresses all ECUs in the network and is executed by the updated application software after a successful flashing sequence.



#### Note

The detailed sequences below show functional messages (broadcasts for all ECUs) in grey and physical messages (messages dedicated to one ECU) in red. Steps which are detailed later on are blue.

#### 2.1.1 Pre-Programming Step

The Pre-Programming Step prepares the vehicle's network for flashing. This includes disabling normal communication to increase the available bandwidth, stop logging of DTCs and a check to ensure the ECU, which should be flashed, is in a state, which allows flashing.

The step "Routine Control – Check Programming Preconditions" is optional and can be deactivated in both bootloader and vFlash.

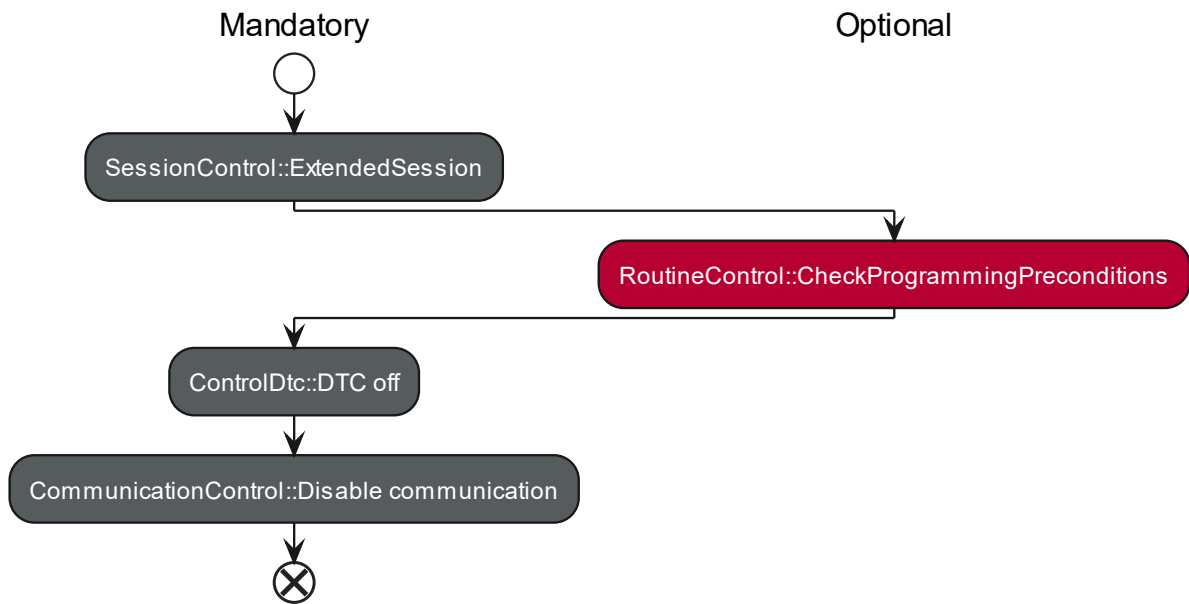


Figure 2-1 Pre-Programming Step

### 2.1.2 Programming Step

The Programming Step performs the actual software download to the ECU. The other ECUs in the network must be kept in extended session by the tester while the ECU is flashed. This is achieved by sending Tester Present messages (\$3E) functionally addressed to all ECUs in the network.

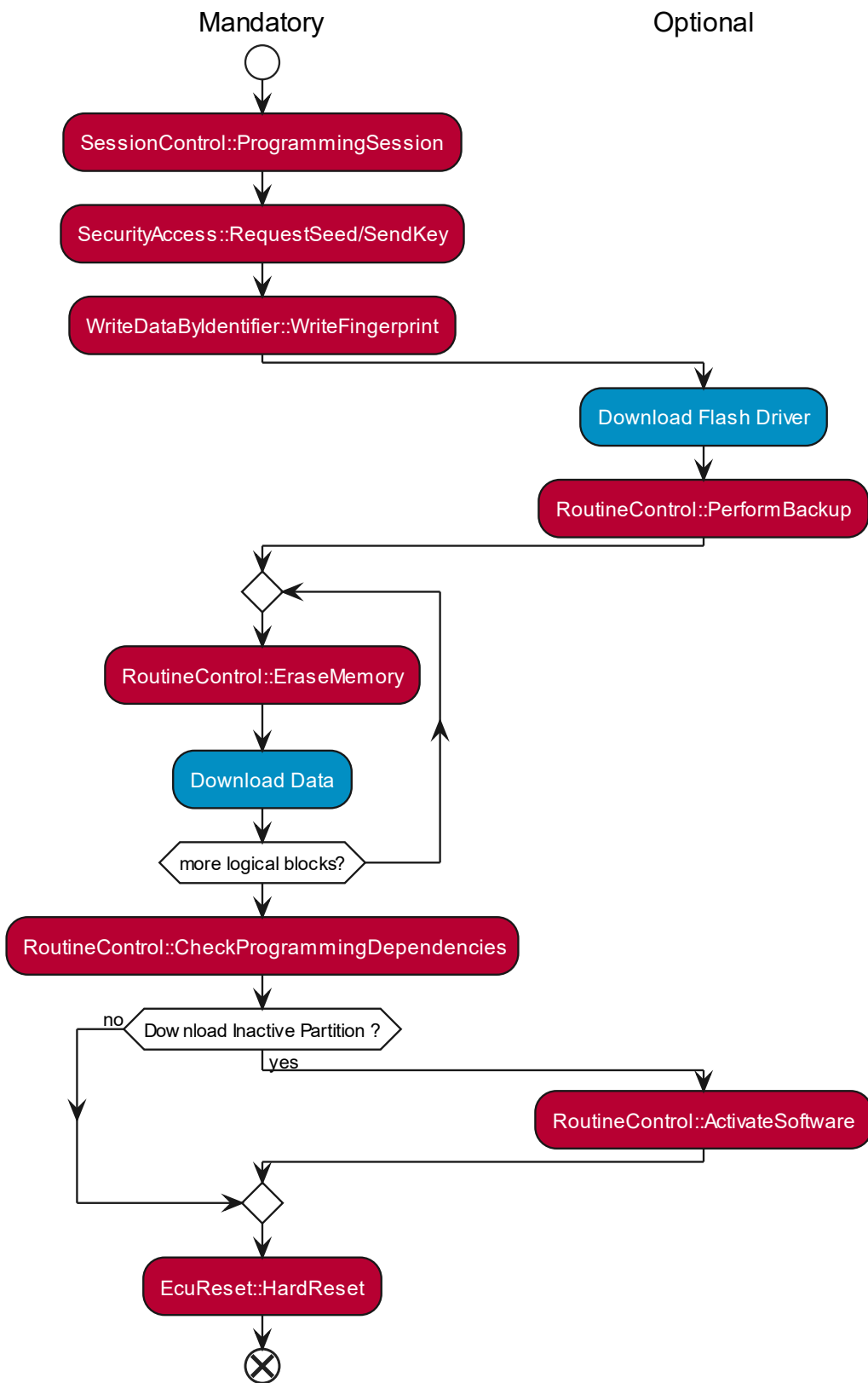


Figure 2-2 Programming Step

The flash driver download is optional. Depending on the configuration, the bootloader can fall back to a flash driver stored in the ECU or always use the stored flash driver. A repetition of the flash driver download (after the download data block) is possible as well.

The download of flash driver and data to be written to flash use the same download sequence:

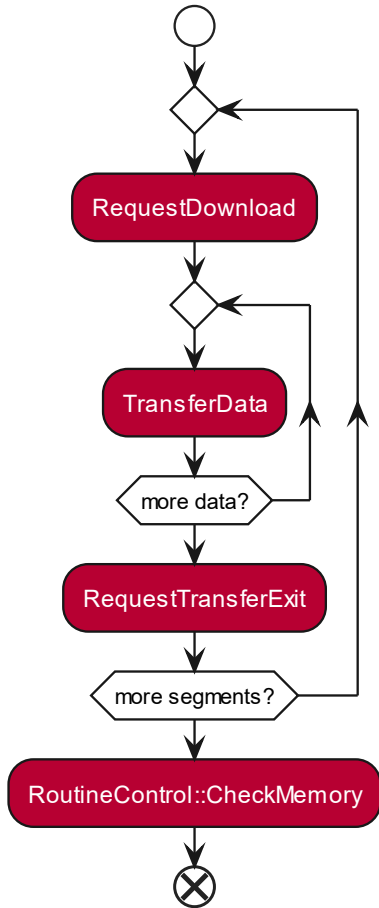


Figure 2-3 Data Transmission

Routine Control – Erase Memory and the Data Transmission sequence must be executed for each logical block. Depending on the bootloader configuration, one or more iterations of the services Request Download, Transfer Data and Request Transfer Exit are allowed for every logical block.

Routine Control – Perform Backup is optional. It is used to copy data from the active partition to the backup partition before erase.

Routine Control – Activate Software is optional. It is used to copy data from the inactive partition to the active partition or swap the two partitions if the hardware supports it.

### 2.1.3 Post-Programming Step

The Post-Programming Step is used to reactivate the normal network behavior after flashing one ECU.

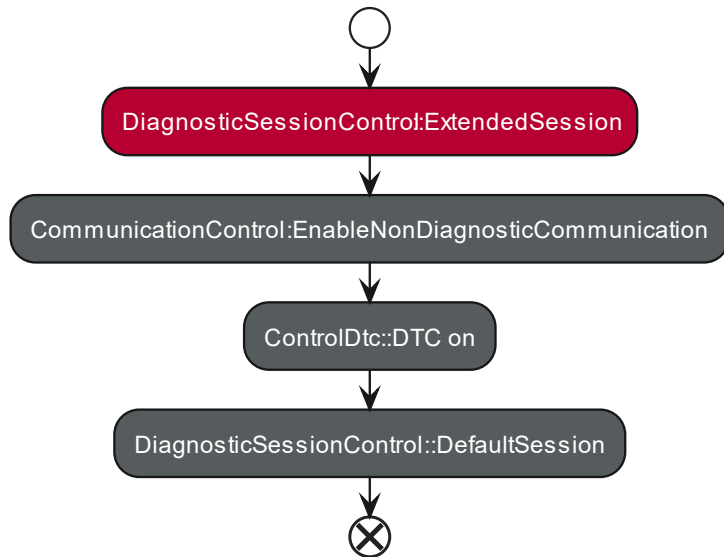


Figure 2-4 Post-Programming Step

## 2.2 Detailed Description of Diagnostic Services

The flash bootloader supports several diagnostic services which are relevant for the download sequence. A list and more detailed description of the services can be found in [2].



#### Note

The service descriptions in [2] are generic definitions of UDS. Please note that this bootloader package specifies the usage and interpretation of some parameters if they are not specified in UDS in detail.

## 2.2.1 Diagnostic Session Control

This service is used to select the different diagnostic sessions in the ECU.

Diagnostic Session Control \$10 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$10	Request Service ID
2	\$xx	Diagnostic Session Type ▶ \$01 Default Session ▶ \$02 Programming Session ▶ \$03 Extended Session
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$50	Response Service ID
2	\$xx	Diagnostic Session Type (see byte #2 of request message)
3..4	\$xxxx	P2 Server Max [ms]
5..6	\$xxxx	P2* Server Max [10ms]

## 2.2.2 ECU Reset

ECU Reset \$11 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$11	Request Service ID
2	\$01	Reset Type – Hard Reset
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$51	Response Service ID
2	\$01	Reset Type – Hard Reset

### 2.2.3 Read Data By Identifier

Read Data By Identifier \$22 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$22	Request Service ID
2..3	\$xxxx	Data Identifier #1
...	\$xxxx	Data Identifier #n
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$62	Response Service ID
2..3	\$xxxx	Data Identifier #1
4..x	\$xx	Requested data of DID #1
(x+1)..(x+2)	\$xxxx	Data Identifier #n
(x+3)...	\$xx	Requested data of DID #n

Several DIDs can be read with one request as long as they fit into the diagnostic buffer.



**Note**

This service is not necessary for the download sequence, but it can be used to obtain useful information from the ECU. The DIDs can be implemented in user callouts and are not defined by basic software parts of the delivery.

### 2.2.4 Security Access

This service is used to restrict access to certain services in the ECU, e.g. all services which write data to the ECU.



**Caution**

Please note that Vector UDS (SLP3) bootloaders include a demonstration seed/key algorithm. This algorithm does not provide a secure access restriction in production ECUs and should be replaced by a secure algorithm.

### 2.2.4.1 Request Seed

Security Access; Request Seed \$27 \$SL – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$27	Request Service ID
2	\$SL	Seed Level
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$67	Response Service ID
2	\$SL	Seed Level
3..x	\$xx	Seed Value

The security level (Request Seed subfunction) is configurable. It must be configured to the same value in bootloader and vFlash. vFlash supports security levels \$01 - \$13.

### 2.2.4.2 Send Key

Security Access; Send Key \$27 \$(SL+1) – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$27	Request Service ID
2	\$(SL+1)	Send Key. Subfunction is calculated from seed level: \$(SL + 1)
3..x	\$xx	Key calculated by tester
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$27	Response Service ID
2	\$(SL+1)	Send Key (SL+1)

The Send Key sub-function is the incremented Request Seed sub-function (SK = SL + 1).

### 2.2.5 Communication Control

Transmission and/or reception of messages can be controlled using this service, e.g. to provide additional bandwidth during reprogramming. The service is mainly used by the application software and intended for the ECUs in the vehicle, which are not flashed. The bootloader includes an implementation of this service, which responds to the requests, but does nothing else, because the bootloader supports only diagnostic communication in most cases.

Communication Control \$28 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$28	Request Service ID
2	\$xx	Control Type <ul style="list-style-type: none"> <li>▶ \$00: Enable Rx and Tx</li> <li>▶ \$01: Enable Rx and disable Tx</li> <li>▶ \$03: Disable Rx and Tx</li> </ul>
3	\$01	Communication Type – Normal Communication

Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$68	Response Service ID
2	\$xx	Control Type <ul style="list-style-type: none"> <li>▶ \$00: Enable Rx and Tx</li> <li>▶ \$01: Enable Rx and disable Tx</li> <li>▶ \$03: Disable Rx and Tx</li> </ul>

One of the two possible sub-functions \$01 and \$03 is used in the pre-programming sequence. The variant is configurable in bootloader and vFlash.

The post-programming sequence always expects sub-function \$00.

### 2.2.6 Write Data By Identifier

This service can be used to transmit additional information to the ECU.

Write Data By Identifier \$2E – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$2E	Request Service ID
2..3	\$xxxx	Data Identifier
4..x	\$xx	Data Record

Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$6E	Response Service ID
2..3	\$xxxx	Data Identifier

The default configuration of the bootloader expects a fingerprint to be written before anything is changed on the ECU. This fingerprint can be used to identify the tester and date of the last update. DID and length of the request can be adapted in vFlash and bootloader.

## 2.2.7 Routine Control

This service is used to start a routine, stop a routine or request routine results. The Vector UDS (SLP3) bootloader executes all routines synchronously and uses the start routine sub-function only.

### 2.2.7.1 Check Routine

This routine is used to ensure the integrity (checksum calculation, signature verification) and to check the download's authenticity (signature verification).

Routine Control; Check Routine \$31 \$01 \$02 \$02 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$31	Request Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$0202	Routine Identifier – Calculate Checksum
5..x	\$xx	Optional parameter Routine Control Option Record. These bytes can contain the checksum or signature. If this parameter is left empty, the checksum or signature is part of the downloaded data (usually appended right behind the last data byte) and the bootloader must extract the comparison data used in this service from there (embedded checksum/signature).
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$71	Response Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$0202	Routine Identifier – Calculate Checksum
5	\$xx	Routine Status Record: \$00 indicates a successful checksum/signature verification. Different values indicate an error, e.g. non-matching checksum.

### 2.2.7.2 Check Programming Preconditions

This service is used to check if the ECU is in a state, which allows the transition to the bootloader. It is executed in application software context while the application software is still running. In case the ECU is in a state, which doesn't allow to go to the bootloader (e.g. engine ECU and engine is running), this service has to deny the transition to bootloader.

The bootloader itself includes a reduced implementation, which is only used in case the application software is not valid.

The service is optional and can be deactivated on the ECU and in vFlash.

Routine Control; Check Programming Preconditions \$31 \$01 \$02 \$03 Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$31	Request Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$0203	Routine Identifier – Check Programming Preconditions
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$71	Response Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$0203	Routine Identifier – Check Programming Preconditions
5..x	\$xx	Routine Status Record – Contains the preconditions that are not fulfilled.

### 2.2.7.3 Erase Memory

Routine Control; Erase Memory \$31 \$01 \$FF \$00 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$31	Request Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$FF00	Routine Identifier – Erase Memory
5	\$mn	ALFI – supported values BCD coded between \$11 and \$44
6..(6+n-1)	\$xx	Memory address (n Bytes)
(6+n).. ((6+n)+m-1)	\$xx	Erase Length (m Bytes)
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$71	Response Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$FF00	Routine Identifier – Erase Memory
5	\$xx	Routine Status Record: \$00 indicates a successful erase operation, any different value indicates, which kind of error occurred.

### 2.2.7.4 Check Programming Dependencies

Routine Control; Check Programming Dependencies \$31 \$01 \$FF \$01 Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$31	Request Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$FF01	Routine Identifier – Check Programming Dependencies
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$71	Response Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$FF01	Routine Identifier – Check Programming Dependencies
5	\$xx	Routine Status Record: \$00 indicates a successful dependency check, a different value indicates, which kind of error occurred.

### 2.2.7.5 Force Boot Mode

This service is optional and is available if “Stay In Boot” is enabled in the configuration tool. The message contents can be changed to any service, which doesn’t change the ECU’s state.

Routine Control; Force Boot Mode \$31 \$01 \$F5 \$18 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$31	Request Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$F518	Routine Identifier – Stay In Boot
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$71	Response Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$F518	Routine Identifier – Stay In Boot

### 2.2.7.6 Perform Backup

This routine is used to copy data from the active partition to the backup partition. This service can be enabled by configuring the “Backup Routine Id” parameter in the configuration tool.

Routine Control; Perform Backup \$31 \$01 \$xx \$xx – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$31	Request Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$xxxx	Routine Identifier –Perform Backup Routine

Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$71	Response Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$xxxx	Routine Identifier – Perform Backup Routine
5	\$yy	Routine Status Record – 00 indicates that the backup is done successfully, and 01 indicates that the routine failed.

### 2.2.7.7 Activate Software

This routine is used to copy data from the inactive partition to the active partition or swap the two partitions if the hardware supports it. This service can be enabled by configuring the “Activate Software Routine Id” parameter in the configuration tool.

Routine Control; Activate Software \$31 \$01 \$xx \$xx – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$31	Request Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$xxxx	Routine Identifier – Activate Software Routine

Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$71	Response Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$xxxx	Routine Identifier – Activate Software Routine
5	\$yy	Routine Status Record – 00 indicates that the activation of software is done successfully, and 01 indicates that the routine failed.

### 2.2.7.8 Perform Rollback

This routine is used to switch back to a previous software version located in the backup/inactive partition, which can be copied/swapped to the active partition depending on the hardware support. This service can be enabled by configuring the “Rollback Routine Id” parameter in the configuration tool.

Routine Control; Perform Rollback \$31 \$01 \$xx \$xx – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$31	Request Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$xxxx	Routine Identifier – Perform Rollback Routine

Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$71	Response Service ID
2	\$01	Routine Control Type – Start Routine
3..4	\$xxxx	Routine Identifier – Perform Rollback Routine

5	\$yy	Routine Status Record – 00 indicates that the rollback is done successfully, and 01 indicates that the routine failed.
---	------	------------------------------------------------------------------------------------------------------------------------

### 2.2.8 Request Download

Request Download \$34 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$34	Request Service ID
2	\$xx	DFI – Determines if a download is compressed or encrypted.
3	\$mn	ALFI – supported values BCD coded between \$11 and \$44
4..(4+n-1)	\$xx	Memory Address. Length n Bytes
(4+n-1).. ((4+n)+m-1)	\$xx	Memory Size. Length m Bytes
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$74	Response Service ID
2	\$n0	Length Format Identifier – n (4 Bits) determines the length of the maxNumberOfBlockLength parameter.
3..(3+n-1)	\$xx	maxNumberOfBlockLength. Contains the maximum number of bytes which can be transmitted with each transfer data service.

### 2.2.9 Transfer Data

Transfer Data \$36 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$36	Request Service ID
2	\$xx	Block sequence counter. This value starts at \$01 and is incremented with each following transfer data service.
3..x	\$xx	Transfer Data Parameter Record. Contains the downloaded data.
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$76	Response Service ID
2	\$xx	Block sequence counter

### 2.2.10 Request Transfer Exit

Request Transfer Exit \$37 – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$37	Request Service ID
Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$77	Response Service ID

### 2.2.11 Tester Present

Tester Present \$3E – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$3E	Request Service ID
2	\$00	Zero Sub-function

Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$7E	Response Service ID
2	\$00	Zero Sub-function

### 2.2.12 Control DTC Setting

Diagnostic Service – Request Message Format		
Data Byte#	Data Value	Parameter Description
1	\$85	Request Service ID
2	\$xx	Control DTC Setting Type: \$01 = on, \$02 = off
3..5	\$FFFFFF	Control DTC Option Record. Deactivate all DTCs (optional parameter, can be deactivated in configuration tool and vFlash)

Positive Response Message Format		
Data Byte#	Data Value	Parameter Description
1	\$C5	Response Service ID
2	\$xx	Control DTC Setting Type

### 2.2.13 Negative Response Codes

The Vector UDS (SLP3) bootloader implements NRCs and NRC priorities mostly according to the standard priorities defined in [2]. However, there are deviations, which are either caused by size optimizations or an established test process.

## 3 Glossary and Abbreviations

### 3.1 Glossary

Term	Description
NV-Memory	Non-volatile memory: Memory which is used to store small amounts of data which are not included in the download images and created during the bootloader's runtime.

### 3.2 Abbreviations

Abbreviation	Description
ALFI	Address/Length Format Identifier
BCD	Binary Coded Digit
CAN	Controller Area Network
DID	Data Identifier
DFI	Data Format Identifier
FBL	Flash Bootloader
ECU	Electronic Control Unit
HIS	Herstellerinitiative Software
ISO	International Organization for Standardization
LIN	Local Interconnect Network
UDS	Unified Diagnostic Services

## 4 Contact

Visit our website for more information on

- > News
- > Products
- > Demo software
- > Support
- > Training data
- > Addresses

**[www.vector.com](http://www.vector.com)**