Elektrobit

# EB tresos Bootloader for Essentials documentation

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

## Technical support

https://www.elektrobit.com/support

## Legal disclaimer

Confidential and proprietary information

# Table of Contents

# 1. Overview of EB tresos Bootloader for Essentials documentation

Welcome to the EB tresos Bootloader for Essentials (BL for Essentials) product documentation.

This document provides:

- Chapter 2, “BL for Essentials release notes“: release notes for the BL for Essentials modules
- Chapter 3, “BL for Essentials user guide“: containing background information and instructions
- Chapter 4, “BL for Essentials module references“: information about configuration parameters and the application programming interface

# 2. BL for Essentials release notes

## 2.1. Overview

This chapter provides the BL for Essentials specific release notes.

## 2.2. Scope of the release

### 2.2.1. Configuration tool

Your release of EB tresos Bootloader for Essentials is compatible with the release of the EB tresos Studio configuration tool:

- EB tresos Studio: 26.2.0 b191017-0938

### 2.2.2. EB tresos Bootloader for Essentials modules

The following table lists modules which are part of BL for Essentials release.

| Module name | Module version | Supplier |
|---|---|---|
| APP | 3.6.8 | Elektrobit Automotive GmbH |
| BM | 1.3.0 | Elektrobit Automotive GmbH |
| BlPduR | 0.13.0 | Elektrobit Automotive GmbH |
| BundleBoot | 0.1.13 | Elektrobit Automotive GmbH |
| BundleBootOEMInd | 0.0.15 | Elektrobit Automotive GmbH |
| Prog | 2.17.0 | Elektrobit Automotive GmbH |
| ProgOEMInd | 1.3.8 | Elektrobit Automotive GmbH |
| SA | 1.6.3 | Elektrobit Automotive GmbH |
| Uds | 3.9.2 | Elektrobit Automotive GmbH |

Table 2.1. Modules specified by OEM specification

# 2.3. Module release notes

## 2.3.1. APP module release notes

- Module version: 3.6.8.BL-3.X_B280905
- Supplier: Elektrobit Automotive GmbH

### 2.3.1.1. Change log

This chapter lists the changes between different versions.

**Module version 3.6.8**
2019-12-09

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.6.7**
2018-10-25

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.6.6**
2018-07-20

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.6.5**
2017-04-03

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.6.4**
2016-12-15

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.6.3**

2016-10-10

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.6.2**

2016-08-12

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.6.1**

2016-05-30

**Module version 3.6.0**

2016-03-29

**Module version 3.5.6**

2016-01-14

**Module version 3.5.5**

2015-10-26

- OSCAPP-352: Renamed file APP_Boot_VCC.c for APP_Boot.c in order to make it generic for all the OEMs.

**Module version 3.5.4**

2015-07-20

- OSCAPP-312: [BOOT_VCC_JLR] Add the PROG_SwitchApplicationModeInd callback to allow the integrator to do some actions before jumping to SBL.

**Module version 3.5.3**

2015-04-29

- OSCAPP-295: Remove EB_Init from APP_Init

**Module version 3.5.2**
2015-01-08

- Internal changes

**Module version 3.5.1**
2014-12-15

- OSCAPP-274: [BOOT_VCC_JLR] Add PROG_GetSBLInfo to allow customer to do some check before jumping in SBL.

**Module version 3.5.0**
2014-11-05

- OSCAPP-263: [BOOT_VCC_JLR] Add new Prog PROG_CustomerInit callback into APP_Boot_VCC.c
- OSCAPP-266: A TRESOS field list was created to allow to choose NvM MODE into APP. This field is only available when Dummy_Application is true. OSCAPP-262: - [PSA_CAN_LS] DummyCode in DEM_-StoreDtcCbk is updated. - [ALL] DummyCode in UDS_ClearDiagnosticInformation is updated.

**Module version 3.4.0**
2014-09-25

- OSCAPP-254: [PSA_CAN_HS_IS] DummyCode updated in APP_NmMonDefaultInd and APP_ComIf-BusStatusInd.
- OSCAPP-252: [BOOT_VCC_JLR] Add interfaces with ZF Framework

**Module version 3.3.8**
2014-09-15

- OSCAPP-243: [BOOT_VCC_JLR] A new integration part is generated in APP_TpTxConf if the network use is FR. It shall not be removed by customer.
- OSCAPP-245: [BOOT_VCC_JLR] Add two callbacks to manage the AlredyErase flag use to know if the ECU is already erase for the first reprogramming (when deliver to VCC)
- OSCAPP-244: [PSA_CAN_HS_IS] DummyCode in ComIf_ClearAllDTC and UDS_ClearDiagnosticInformation function updated.
- OSCAPP-247: [BOOT_VCC_JLR] To allow the correct asynchronous data retrieving in FR stack the buffer pass to TP_SetMsgData is now a global variable

**Module version 3.3.7**
2014-08-18

- OSCAPP-234: [BOOT_PSA] Callback implementation improved to better match implementation rules
- OSCAPP-235: [ALL] New UDS_SecurityCheck callback to check if ECU is locked (NRC33)
- OSCAPP-239: [BOOT_PSA] New callback PROG_ReprogStepCbk implemented
- OSCAPP-238: [ALL] Callback APP_GetUdsDataBufferInd have a new argument to notify the result of the UDS treatment

**Module version 3.3.6**
2014-07-09

- OSCAPP-227: [PSA_CAN_HS] Update about IE: callback IE_GetIeModeFromNVM replaced by IE_SetIeModeFromNVM API

**Module version 3.3.5**
2014-06-24

- OSCAPP-221: [PSA_CAN_HS] Improved to integrate new module IE

**Module version 3.3.4**
2014-06-06

- OSCAPP-220: [PSA_CAN_LS/PSA_CAN_HS] DummyCode in APP_UdsSessionStatusInd updated

**Module version 3.3.3**
2014-05-23

- OSCAPP-217: [PSA_CAN_HS_IS] Add callback APP_GetDiagnosticConditions
- OSCAPP-218: [RSA_CAN_HS] Simplify the Bootflag read management

**Module version 3.3.2**
2014-05-05

- OSCAPP-209: To be fully compliant with PSA HS LS specifications, APP_ComTxTimeout callback is used in PSA_CAN_HS_IS also.

- OSCAPP-210: [RSA_CAN_HS] Prototype of DEM_StoreDtcCbk changed (type of input parameter)
- OSCAPP-211: [VCC_JLR] Bug fix: Correction of the two following issue - The DSC 01 request shall not check the application validity for VCC. Only for JLR (when sleep management is enabled) - APP_ReprogReqManage shall be called in APP_Manage for VCC_JLR variant
- OSCAPP-211: [RSA_CAN_HS] Dummy code in APP_NmMonDefaultInd corrected

**Module version 3.3.1**
2014-04-10

- OSCAPP-206: APP_GetUdsDataBufferInd callback prototype added. This callback allows to indicate (and update if necessary) the data buffer status.
- OSCAPP-199: [RSA_CAN_HS] APP integration test code updated for MUTE and ABSENT DTC including DEM

**Module version 3.3.0**
2014-03-26

- OSCAPP-197: [BOOT_EB] Add APP_CalculateCrc function and plugins parameter Crc_Buffer_Size for CRC Algorithm
- OSCAPP-198: [BOOT_PSA_CAN_LS] Add initialization of eBootFromAppli in default case
- OSCAPP-200: [ALL] All src and includes files moved to generated/templates folder

**Module version 3.2.21**
2014-03-05

- OSCAPP-193: [BOOT_VCC_JLR]: Add callbacks to manage timeout error of CompleteAndCompatible callout

**Module version 3.2.20**
2014-02-27

- OSCAPP-184 : CDTCS (Inter Memo Def 0x85) could be integrated to APP
- OSCAPP-188: [BOOT_PSA_CAN_LS] Refresh Watchdog callback
- OSCAPP-189: Improvement: [OM_TYPE3] Update OM Callback APP_OmModifyInd When BSI life phase is received. LOM (LNI STACK) is set to NORMAL or SLEEP.

### Module version 3.2.19

2014-02-18

- OSCAPP-182: Improvement: in case DEM is present it is possible to add the noMedAck (0xDF0000) in the DEM directly
- OSCAPP-181: Fixed known issue: [PSA]: JDD_Flush API is called when UDS clear DTC is received (instead of JDD_Init and JDD_Start)

### Module version 3.2.18

2014-02-14

- OSCAPP-176: [BOOT_PSA_CAN_LS]: Bug fix: Switching APP to BOOT is done with UDS SA1 (SA1 response in BOOT)

### Module version 3.2.17

2014-01-24

- OSCAPP-157: [BOOT_PSA_CAN_LS] Remove APP_TpTxConf implementation code as not used anymore
- OSCAPP-157: [BOOT] Remove unused implementation in APP_ReprogReqManage (APP_Boot_XXX.c)
- OSCAPP-159: New CAN_GetBaudRateIdx callback in case of multi baudrate configuration
- OSCAPP-168: Update APP_UdsSessionStatusInd callback adding new arguments (new session, old session and the reason of the changing session)
- OSCAPP-169: [BOOT_JLR] Update APP with new feature to manage the bootloader JLR sleep management

### Module version 3.2.16

2013-12-17

- OSCAPP-152: Fixed known issue: [PSA_CAN_LS] redeclaration variables

### Module version 3.2.15

2013-11-28

- OSCAPP-146: Fixed known issue: [PSA], Generate correctly the DEM callback for PSA_CAN_LS variants
- OSCAPP-148: Improvement: [DEM PSA], Automatically generate the DEM_Manage into APP_Manage

- OSCAPP-150: Fixed known issue: [PSA], TELE unlocking status check missing in integration test code
- OSCAPP-149: Fixed known issue: [PSA_CAN_LS], Management of GHD is not correct in case defect added is ABSENT_BSI (integration test code)
- OSCAPP-147: Improvement: [JLR], Adding variant JLR for bootloader management.

**Module version 3.2.14**
2013-11-14

- OSCAPP-144: Fixed known issue: [PSA], ComIf_NoMedAcqIndication have now two argument (DTCcode and Header value) to allow customer to remove the MED from memory

**Module version 3.2.13**
2013-10-24

- OSCAPP-137: Allow generation of boot source files in PSA_CAN_LS
- OSCAPP-138: Fixed known issue: include mechanism changed (EBCLG_Prj.h included instead of EBLIN_Prj.h if CLG is present)
- OSCAPP-141: Fixed known issue: [PSA_CAN_HS_IS], Add COM_SendFrame of version frame from EB_-Init to APP_Init

**Module version 3.2.12**
2013-10-03

- OSCAPP-134: Fixed known issue: DEM_StartOperatingCycle is now called for OM Type3

#### 2.3.1.2. New features

- Add new tresos field list to allow to choose NvM MODE. This field is only available when Dummy_Application is true.

  Description:

  New feature available since version 3.4.0 In all variant, it is now possible to configure NvM Mode. There are two modes : the asynchronous mode and the synchronous mode. The NvM mode influence DEM/GHD features. DTC erasing shall be completed befor diagnostic response transmission and NM reactivation.

  Date

2014-10-30

#### 2.3.1.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

#### 2.3.1.4. Deviations

This module is not part of the AUTOSAR specification.

#### 2.3.1.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- For this module no limitations are known.

#### 2.3.1.6. Open-source software

osc_App module does not use open-source software.

### 2.3.2. BM module release notes

- Module version: 1.3.0.BL-3.X_B280905
- Supplier: Elektrobit Automotive GmbH

#### 2.3.2.1. Change log

This chapter lists the changes between different versions.

**Module version 1.3.0**
2019-11-29

- Implemented secure boot feature for profile 10.
- Implemented to support FlexRay protocol in BL 3.9.1 bootloader version.
- Implemented "Dual Memory Bank" feature.
- Changed the name of the API to perform a reset and go to sleep in order to be in line with new API design naming

**Module version 1.1.6**
2019-03-26

- Implemented support to crypto ASR 4.3 stack via the Demo_CSM_Wrapper.
- Implemented secure boot feature for profile31.
- Implemented the support of Demo_CSM wrapper for Crypto ASR 4.3.

**Module version 1.1.5**
2018-10-25

- Improved Time-out at startup.

**Module version 1.1.4**
2018-07-23

- Implemented secure boot feature for profile50.

**Module version 1.1.3**
2018-06-14

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.1.2**
2018-03-22

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.1.1**
2017-12-18

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.1.0**
2017-10-26

- Implemented Authenticated Boot feature for profile50.

**Module version 1.0.9**
2017-10-16

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.8**
2017-07-03

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.7**
2017-05-11

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.6**
2017-04-03

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.5**
2016-12-16

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.4**
2016-10-10

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.3**
2016-08-12

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.2**
2016-05-30

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.1**
2016-03-29

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.0**
2016-02-23

- Internal module improvement. This module version update does not affect module functionality

**Module version 0.0.3**
2015-11-27

- OSCBM-64 : Bug Fix : Get the diagnostic tester source address if programming is requested by application. Otherwise, set it to EB_ALL_TESTER_ADDRESS

**Module version 0.0.2**
2015-07-15

- OSCBM-47 : Bug Fix : to avoid problem after long loss of synchronization All slot mode shall be set each time we go in BootMode

**Module version 0.0.1**
2015-04-29

- Module creation

#### 2.3.2.2. New features

- No new features have been added since the last release.

#### 2.3.2.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

#### 2.3.2.4. Deviations

This module is not part of the AUTOSAR specification.

#### 2.3.2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- For this module no limitations are known.

#### 2.3.2.6. Open-source software

osc_BM module does not use open-source software.

### 2.3.3. BlPduR module release notes

- Module version: 0.13.0.BL-3.X_B280905
- Supplier: Elektrobit Automotive GmbH

#### 2.3.3.1. Change log

This chapter lists the changes between different versions.

**Module version 0.13.0**
2019-06-12

**Module version 0.12.255**
2099-11-29

**Module version 0.12.0**
2019-11-29

- Implemented the Queued Requests feature

**Module version 0.11.0**
2019-11-07

- OSCBLPDUR-156: Fixed known issue: No NRC=13(IMLOIF) response for TesterPresent with Suppress Response Bit
- Implemented streaming feature for Implementation-10 and Implementation-11

**Module version 0.10.0**
2019-09-10

- Implemented support to communicate over FlexRay protocol

**Module version 0.9.0**
2019-07-24

- OSCBLPDUR-137: Fixed known issue: Prevent from going into sleep mode in case of long treatment of frame.
- Implemented Standard CAN ID reception which can bypass tester filtering

**Module version 0.8.0**
2019-03-24

- Added a function in order to check an online ethernet connection for profile20

**Module version 0.7.0**
2019-03-22

- Improved the handling of functional tester present with physical requests
- OSCBLPDUR-89: Fixed known issue: which leads to not releasing the connection on transmission confirmation when the transmission is done without reception first
- OSCBLPDUR-102: Fixed known issue: return incorrect values for BS and STmin when reading them through RDBI

**Module version 0.6.0**
2018-10-25

- Implemented Tester Filtering in the start of request reception

**Module version 0.5.0**
2018-07-20

- Internal module improvement. This module version update does not affect module functionality
- Corrected the request reception simulation at startup
- Implemented the intialization and de-activation of Com Stack in the integration code instead of doing it in plugin.

**Module version 0.4.1**
2018-06-14

**Module version 0.4.0**
2018-03-23

- OSCBLPDUR-80: Fixed known issue: which leads to the impossibility to activate the TP Change parameter feature
- OSCBLPDUR-70: Fixed known issue: which leads to timeout with MultipleBuffers
- Improved functional Pdu handling for several connection

**Module version 0.3.0**
2017-12-18

- Implemented Dynamic reconfiguration of TP parameters

**Module version 0.2.1**
2017-10-26

- Internal module improvement. This module version update does not affect module functionality

**Module version 0.2.0**
2017-10-13

- Internal module improvement. This module version update does not affect module functionality

**Module version 0.1.0**
2017-09-11

- OSCBLPDUR-44: Fixed known issue: which leads to a non response after reset to an ECU Reset and Programming Session request when multiple identifiers were used

**Module version 0.0.6**
2017-08-02

- Internal module improvement. This module version update does not affect module functionality

**Module version 0.0.5**
2017-07-03

- OSCBLPDUR-35: Fixed known issue: which leads to corruption of received data when multiple receive buffers were used

**Module version 0.0.4**
2017-06-12

- Internal module improvement. This module version update does not affect module functionality

**Module version 0.0.3**
2017-06-07

- OSCBLPDUR-26: Fixed known issue: that stop physical segmentation when functional request are received at the same time
- Corrected Multiple Identifier feature

- Improved Lin management to allow full LIN feature
- OSCBLPDUR-31: Fixed known issue: which led the ECU to go in exception on a reception of a frame with suppress positive response bit

**Module version 0.0.2**
2017-05-11

- OSCBLPDUR-17: Fixed known issue: that lock connection when message with Suppress Positive Response Bit set is received
- Added management of Ethernet communication stack

**Module version 0.0.1**
2017-04-03

- Module creation
- Added management of multiple buffer
- Added management of multiple Identifier
- Added management of LIN slave routing

### 2.3.3.2. New features

### 2.3.3.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

### 2.3.3.4. Deviations

This module is not part of the AUTOSAR specification.

### 2.3.3.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

### 2.3.3.6. Open-source software

osc_BlPduR module does not use open-source software.

## 2.3.4. Prog module release notes

- Module version: 2.17.0.BL-3.X_B280905
- Supplier: Elektrobit Automotive GmbH

### 2.3.4.1. Change log

This chapter lists the changes between different versions.

**Module version 2.17.0**
2019-12-06

- Implemented check to ensure Fr Network synchronization before transmission of response for profile 11 and improvement in the handling of multiple TP connections.
- OSCPROG-1918: Fixed known issue: Wrong response is sent when maximum reprogramming counter is reached for DAG.
- OSCPROG-1948: Fixed known issue: Wrong return code on download verification.
- Implemented HSM Software Update.
- OSCPROG-1872: Fixed known issue: Checksum generation errors are not detected if Authenticated/Secured Boot is enabled.
- OSCPROG-1931: Fixed known issue: Wrong NRC in case of a Transfer Data request with no data.
- OSCPROG-1930: Fixed known issue: Corruption flag of the flash driver is not correctly updated after a signature length check failure.
- OSCPROG-1929: Fixed known issue: Transition from Compare Key to Request Download is not rejected.
- OSCPROG-1917: Fixed known issue: [OemInd] Response to a DSC01 request is not sent after reset.
- OSCPROG-1906: Fixed known issue: [OemInd] The callback PROG_CustomSetCrcCompareSuccess() is not called if CRC verification is wrong.
- Implemented the activationt of the anti-scanning feature for implementation 10.
- OSCPROG-1983: Fixed known issue: [GM] Response to a DSC01 request is not sent after reset
- OSCPROG-1961: Fixed known issue: Sleep Timer is deactivated once a request is received while in default session

- Changed the name of the API which performs a reset and go to sleep to be in line with new API design naming
- OSCPROG-1950: Fixed known issue: The bootloader does not respond after the reset to an open programming session request received by the application
- Implemented the feature "Dual Memory Bank"
- OSCPROG-1901: Fixed known issue: NRC78 response is not sent for an EcuReset (0x11)
- Implemented the Check Memory status feature
- OSCPROG-1773: Fixed known issue: Receive size of data during TransferData is corrupted

**Module version 2.14.0**
2019-07-23

- Implemented a preprocessor check to unselect the Cry_LN.h header file when the Tresos parameter "Use_CSM_ASR430_DemoWrapper" is set to "TRUE"
- OSCPROG-1847: Fixed known issue: SecurityAccess subfunction parameter is not read correctly.
- Management of two successive Check Memory request for profile 10 and 11
- Implemented the feature to perform Session change without SecurityLevel Reset across the transitions between Non-default sessions for implementation 31
- OSCPROG-1743: Fixed known issue: Bootloader returns a wrong memory check result because it does not check the FileSize value for implementation 31
- OSCPROG-1896: Fixed known issue: Correct wrong answer sent when second Check Memory request is correct for implementation 10
- Added preInit waiting function for profile20
- OSCPROG-1869: Fixed known issue: Programming can continue even if the writing of the fingerprint failed for implementation 31
- Implemented the replacement of the SBI flag variable by customer callbacks
- Implemented the feature of write public key for profile 10
- OSCPROG-1865: Fixed known issue: NRC78 response is not sent for a DiagnosticSessionControl(defaultSession) request with the SuppressPositiveResponse bit set
- OSCPROG-1829: Fixed known issue: Flash writing operation is not done correctly when signature verification is performed on received data

**Module version 2.13.0**
2019-05-16

- Implemented the feature "ECU Software Structure"

- Implemented the feature "Download verification using the verification block table"
- Implemented the feature "Secure boot using Mac"
- OSCPROG-1772: Fixed known issue: Sleep Timer Management in Default Session
- OSCPROG-1765: Fixed known issue: [OEMInd] Continuous pending response when two successive WriteFingerPrint requests are received and the first is wrong
- OSCPROG-1821: Fixed known issue: [VWAG/OEMInd] ECU does not return to Lock state when GetSeed is not allowed (NRC_37)
- Implemented a guard function to ensure that only two successive check memory are allowed for profile 10

**Module version 2.12.1**
2019-04-03

**Module version 2.12.0**
2019-03-26

- OSCPROG-1646: Fixed known issue: Read finger print service returns a wrong response
- OSCPROG-1715: Fixed known issue: Information of two bytes block identifier is not correctly retrieved when feature "Signature Verification with address and length from Request Download" is activated
- OSCPROG-1561: Fixed known issue: Wrong PEC value is stored when RequestTransferExit request is received before all transfer data requests are finished
- Updated "check programming dependencies" feature by using callbacks in this for GM.
- OSCPROGFCA-72: Fixed known issue: Continuous NRC78 on request Check memory 0xF000.
- OSCPROG-1712: Fixed known issue: In programming session in case of security access is unlocked a new programming session request resets the security level.
- Implemented support of fingerprint for the RAM segments and check to avoid erase sector by sector for RAM segements
- OSCPROG-1445: Fixed known issue: Erase routine rejected after asynchronous fingerprint writing.
- OSCPROG-1731: Fixed known issue: Compile error if routine Verify_partial_software_checksum is not configured.
- OSCPROG-1703: Fixed known issue: Activate feature "Programming Counter" for FCA Atlantis High
- OSCPROG-1211: Fixed known issue: The transition from diagnostic default session to default session causes the ECU to reset
- OSCPROG-1753: Fixed known issue: CheckProgram routine fails when block identifier configuration doesn't start from 0

- OSCPROG-1711: Fixed known issue: Activate feature "Erase Check" for FCA Atlantis High
- Implemented a returned status in API PROG_CustomIncrementProgCounter
- OSCPROG-1548: Fixed known issue: Continuous pending response on reception of an invalid signature
- Implemented the feature "Protected Calibration" for GM bootloader
- Implemented the support of Demo_CSM wrapper for Crupto ASR 4.3.
- OSCPROG-1758: Fixed known issue: Invalid Response length for "Adjust ISO 15765-2 BS and STmin Parameter" DID writing for Daimler and Volkswagen
- OSCPROG-1767: Fixed known issue: SPREC parameters are present in DSC02 although SPREC_IN_-RESPONSE is deactivated
- Implemented secure boot feature for profile31.

**Module version 2.11.0**
2018-10-25

- Implemented the All Custom Memory Access
- Modified programming counter which shall not be incremented if the blocks have already been completely erased
- Added implementation variant 31
- Added of new structure in order to use two private key for GM signature calculate
- Removing program failed function and adding the behavior of profile20 to TD failed and RTE failed entry functions instead
- Implemented Preliminary Erasing feature
- Added call of blpdur for the tester filtering feature
- Modified the response to the request $F0 $F0 in the case the application is invalid/revoked by adding the information of invalid calibration [GM]
- Added Compression on variant 30
- Added download by logical block only feature for profile 40
- Added PROG_Dsc02Cbk callback implementation for profile 40
- Fixed compilation error when sleep management is off
- Implemented the feature "Compressed Flash Driver"
- Implemented the feature "Resumable reprogramming"
- Fixed compilation error when macros are defined in C files
- Implemented the feature "Block header reading"
- Implemented the feature "OEMInd: support asynchronous memory access"

- Implemented the feature "Signature on Compressed data"
- Implemented the feature "Signature Verification with address and length from Request Download"

**Module version 2.10.0**
2018-07-24

**Module version 2.9.0**
2018-07-20

- Corrected bad behavior on signature check start, update and finish as well as CRC computation [GM]
- Internal module improvement. This module version update does not affect module functionality
- OSCPROG-1388: Fixed known issue: Correct bad behaviour when receiving successive CheckMemory requests
- Implemented the Impl40/Impl50 public key management for Impl20.
- Added of callback to notify once there is a synchronous memory access.
- Implemented the application checksum computation and write in non-volatile memory for Secure Boot feature
- OSCPROG-1591: Fixed known issue: Correct compilation error when "Diagnostic Reprogramming response" parameter is not set
- Corrected bad behavior when response to "open programming session" request is sent by application
- Corrected bad behavior on the sending of positive response for the Transfer Data in the last buffer when Multiple buffers is enabled (no more Rx buffers available)

**Module version 2.8.2**
2018-06-14

**Module version 2.8.1**
2018-06-11

- Implemented SecurityAcces in Application feature.
- Implemented Flash Erase All feature.
- OSCPROG-1251: Fixed known issue: The minimum length of a TransferData request is not correctly checked.

- OSCPROG-1255: Fixed known issue: The calculation of the end address in RequestDownload and Erase request can overflow.
- Corrected bad behaviour on Check Memory when receiving Request Download and no Erase requests
- Corrected integration issue between PROG and Flash modules when Prog is configured to Synchronous and Flash returns Busy state.

**Module version 2.8.0**
2018-03-23

- OSCPROG-1451: Fixed known issue : Compression format 0xA is not supported, compression algorithm is now configurable
- Implemented Data Decryption feature.
- OSCPROG-1448: Fixed known issue : Only first segment of asynchronous memory is erased
- Improved CRC calculation form OEMInd by adding the possibility to deactivate the calculation
- Improved Request download feature by adding new configuration parameter (Maximum RequestDownload Per Block) to size the storing of downloaded area for every logical block
- Improved FingerPrint reading and writing feature.
- Improved Erase feature : NRC78 can be transmitted by configuration before software invalidation on the Erase request reception.
- Improved Erase feature : Modification of configuration parameter Erase Check Type
- Corrected VerifyPartialSoftwareChecksum routine with Signature feature
- Implemented Downloading Flash Driver feature.
- OSCPROGDAG-74: Fixed known issue: [DAG] Randomly incorrect response to SecurityAccess request.

**Module version 2.7.1**
2018-02-06

- OSCPROG-1390: Fixed known issue : [OEMInd] Missing return value in PROG_CustomSetCrcCompareSuccess

**Module version 2.7.0**
2017-12-18

- Removed Csm_Init call (moved to BM plugin)
- OSCPROG-1384: Fixed known issue : Multiple buffer processing can be interrupted by new incoming requests

- Implemented Dynamic reconfiguration of TP parameters feature.
- OSCPROG-1357: Fixed known issue : The Verify_partial_software_checksum routine is rejected
- Implemented Support of signature check for Impl40 and Impl60
- OSCPROG-1424: Fixed known issue : Undefined API when only external flash is used

**Module version 2.6.1**
2017-10-26

- Internal module improvement. This module version update does not affect module functionality

**Module version 2.6.0**
2017-10-16

- OSCPROG-1277: Fixed known issue : Give the user the possibility to address block on two bytes
- OSCPROG-1345: Fixed known issue : When compression is activated, downloading a non compressed data doesn't work
- Implemented Several external memory with a different access modes feature
- Implemented Signature verification on the fly feature

**Module version 2.5.0**
2017-09-08

- Updated Multiple buffer feature by managing the last TDaccording to Daimler clarification
- OSCPROG-1269: Fixed known issue : Fix compare key design
- OSCPROG-1315: Fixed known issue : Jump from application to bootloader through ProgrammingRequest (DSC02) does not work
- Updated decommpression feature in order to add slicing
- OSCPROG-1224: Fixed known issue : Block 0 is supported by bootloader when no bootloader partition is defined
- Improved response after reset feature by storing and retrieving connection context when the ECU shall reset
- OSCPROG-1214: Fixed known issue : Verification of partial software checksum failure

**Module version 2.4.1**
2017-08-02

- OSCPROG-1222: Fixed known issue : The key NBID is not updated in NVM if a more recent one is received
- OSCPROG-1223: Fixed known issue : Correct Response to Request 0xFF00 depending of type errors occured during check of erase memory parameters

**Module version 2.4.0**
2017-07-11

- Implementated LZSS compression feature

**Module version 2.3.3**
2017-06-16

- Internal module improvement. This module version update does not affect module functionality

**Module version 2.3.2**
2017-06-12

- Internal module improvement. This module version update does not affect module functionality

**Module version 2.3.1**
2017-05-11

- OSCPROG-1145: Fixed known issue : Positive Response was not sent after Ecu Reset whereas option was activated
- OSCPROG-1149: Fixed known issue : Correct Response to Request 0xFF00 when errors occured during check of erase memory parameters
- Added callbacks to get symmetrical/asymmetrical cryptography public keys

**Module version 2.3.0**
2017-04-03

- Improved sleep management by activating the feature for Impl40 and Impl50
- Improved decommpression feature by managing correctly the status returned by decompression library
- Implementated Implementation 60 features
- Implementated Multiple receive buffer feature

- Improved EcuId feature: Added possibility to use a callback to get ECUID
- Improved EraseMemory and CheckMemory responses in Impl50
- Improved management of ECCCheck for all memory access functions
- OSCPROG-982: Fixed known issue: Wrong management in case of asynchronous memory use
- Removed unused API
- Updated CRC feature by adding the possibility to compute CRC on a particular Block Id
- Implementated watchdog deactivation for Impl50
- OSCPROG-1020: Fixed known issue: The NRC sequence is not correctly checked for an RTE request
- Improved DiagnosticSessionControl feature : Manage DiagnosticSessionControl request with subservice ExtendedSession and reset security level when entering ExtendedSession
- Updated fingerprint feature to handle target with pages superior to 8 bytes
- Implementated Programming counter for logical blocks feature
- Implementated Partial software checksum feature
- OSCPROG-1093: Fixed known issue : Erasing by blockID could return NRC_72 if PROG software module is configured in asynchronous mode
- OSCPROG-1020: Fixed known issue: The NRC sequence is not correctly checked for a TD request
- Implementated Routine CheckProgrammingDependencies feature for Impl20
- OSCPROG-1120: Fixed known issue : When logical block feature is activated, the RoutineControl EraseMemory 0xFF00 will respond with the wrong answer (fields routineInfo and routineStatusRecord)
- OSCPROG-1122: Fixed known issue : When CRC32 feature is activated, the CRC value is extracted at wrong index from the RC 0x0202
- OSCPROG-1121: Fixed known issue : When Coherency check feature is activated, the RoutineControl EraseMemory 0xFF01 will respond with the wrong answer (fields routineInfo and routineStatusRecord)
- OSCPROG-1141: Fixed known issue : The RoutineControl EraseMemory 0xFF00 responds with the wrong answer (routineInfo and routineStatusRecord fields)

**Module version 2.2.0**
2016-12-16

- Changed header calculation for GM by including AppSwInfo header in a region
- Improve Erase by block id feature by adding new callback PROG_InvalidateSection_BlockID
- Improved demcompression feature by substracting compression header size to have the real size of compressed data
- OSCPROG-888: Fixed known issue: Fix issue on CheckProgrammingRequest when using Flash_Ext

- Improved robustness for decompression errors management
- OSCPROG-939: Fixed known issue: Fix issue on CheckProgrammingRequest when using Flash_Ext (by RANGE)
- OSCPROG-914: Fixed known issue: Erasing is performed even if the memory is already erased
- Improved erase feature by calling of setting the erase status upon reception of the Request download.
- Improved DSC response
- OSCPROG-953: Fixed known issue: Routine CheckProgrammingDependencies is rejected for address range on several segments
- Improved design of PROG module
- Improved CRC calculation by providing the information of which segment is invalidated when using the callback PROG_CustomSetAppValidity
- Implementated coherency check feature
- OSCPROG-955: Fixed known issue: LZMA/ARLE decompression can lead to an infinite loop
- Improved calculation of the Programming Status
- Improved CRC range calculation

**Module version 2.1.1**
2016-10-10

- OSCPROG-854: Fixed known issue: Add a verification on open programming session request's callback
- Implementated LZMA decompression feature
- Improved CRC calculation allowing asynchronous management upon receiving the Request Transfer exist
- Updated data access error management

**Module version 2.1.0**
2016-08-12

- Implemented Programming pre-condition check
- Implemented Reset cause and DSC/ER response management
- Implemented Management of CheckMemory routine for CRC verification
- OSCPROG-731: Fixed known issue: Unexpected behavior when reading DID F0F0 and no application is present
- Implementated Logical block management and Download by logical address
- OSCPROG-794: Fixed known issue: Missing initialization variable in PROG_CheckDecompHeaderStatus

- OSCPROG-794: Fixed known issue: Tresos error when Sleep_Management_Type is Off and ProgGM is present
- OSCPROG-758: Fixed known issue: GM application verification fails when external Flash is used
- Internal module improvement. This module version update does not affect module functionality
- OSCPROG-780: Fixed known issue: Unexpected behavior when DFI from RequestDownload request is not aligned with datatype from TransferData request
- Improved GM BootInfo Block management by adding memmap section
- Corrected data overwriting issue in decompression buffer (ARLE decompression)
- OSCPROG-697: Fixed known issue: SBA DID reading shall not contain Datatype field
- Added APIs to get access to DID F0F3/F0F6

**Module version 2.0.0**
2016-05-31

- OSCPROG-663: Fixed known issue: Correction of a bug where the TxConf where not handle correctly in case of diag response received from functional addressing
- Improved DiagnosticSessionControl feature by transmitting NRC78 at init for DSC02 response
- OSCPROG-681: Fixed known issue: Correction of a bug where negative response is not sent in case of failure during TD reception
- Implemented ARLE decompression support
- Implemented Write Fingerprint management
- Implemented Anti-scaning management
- Implemented FCA Autocontrol management
- Implemented Programming status management
- Added F180 DID Service Management

**Module version 1.0.0**
2016-02-23

- OSCPROG-555: Fixed known issue : Fix erase routine response size
- OSCPROG-631: Fixed known issue: The new API called by state machine PROG_Erase_NRC78 has been added in the product

**Module version 0.1.3**
2016-01-26

- Internal module improvement. This module version update does not affect module functionality

**Module version 0.1.2**
2016-01-14

- OSCPROG-585: Fixed known issue : Correct decompression issue where in case of high compression ratio the length can be superior to u16 max value
- OSCPROG-591: Fixed known issue: The sleep timer is now correctly reloaded when a Diag frame is received
- OSCPROG-590: Fixed known issue: Correction of a bug where the TxConf where not handle correctly in case of diag response received from functional addressing
- Implemented management of flashPage and skippage in product for implementation 2
- Improved external Flash memory access management
- OSCPROG-601: Fixed known issue: The management for reception of the same TD several times is now correctly handled
- OSCPROG-607: Fixed known issue: The compression is now correctly handle in the error case of a unfinished segmentation
- Improved watchdog feature by calling it before starting the SBL software

**Module version 0.1.1**
2015-11-18

- Improved External flash feature by adding the management of an offset in the external Flash addressing
- OSCPROG-462: Fixed known issue : Remove patch done for management of different P2 and P2* by session
- Improved DiagnosticSessionControl feature by adding the possibility to respond to DSC 02 received in application from the bootloader

**Module version 0.1.0**
2015-10-30

- Improved support of external Flash for PSI reading
- Added SBA management at bootloader startup
- Added cryptography management with verification of signer info and signature.
- Updated PSI management to be able to wirte it twice without erasing the Flash Memory
- OSCPROG-462: Removed Fixing of known issue : P2 and P2* values handled per sessions.

### Module version 0.0.3

2015-09-08

- OSCPROG-462: Fixed known issue : P2 and P2* value to return from DCS response are now manage by the PROG layer to allow returning a different value depending on the session required
- OSCPROG-444: Fixed known issue : S3 timeout is now correctly reloaded when NetworkStatus BOOT is received
- Improved JLR implementation
- Added Page buffer management to allow reception of TD not aligned on a Flash page size

### Module version 0.0.2

2015-07-20

- OSCPROG-404: Fixed known issue: The BlockSequenceCounter of TD request is now correctly returned and is not incremented if the same TD is received two time in a row
- OSCPROG-406: Fixed known issue: When a writing error happened after the full reception of a TD, a NRC_72 is now correctly sent
- OSCPROG-429: Fixed known issue: Prog shall call EB_AllSlots() when network is synchronized
- OSCPROG-421: Fixed known issue: Prog shall go to ERASE_FINISH state in case of erase error
- OSCPROG-435: Fixed known issue: Wrong NRC was transmitted in CheckProgDependencies routine
- OSCPROG-415: Fixed known issue: Interruption are disabled twice in PROG_SwitchApplicationMode
- OSCPROG-430: Fixed known issue: Fix mistake in Design
- OSCPROG-407: Fixed known issue: When RTE is received without a full TD reception, positive response shall not be sent.
- OSCPROG-446: Fixed known issue: Check on argument shall be added in PROG_EraseMemory and PROG_ActiveSBL
- OSCPROG-465: Fixed known issue: TD positive response shall only be requested if no error where detected during the TD
- OSCPROG-471: Fixed known issue: Add "volatile" to some const to avoid that the compiler optimize the code and change the needed behaviour
- OSCPROG-470: Fixed known issue: Correction of the length check in PROG_GetSegmentByAddress API
- OSCPROG-477: Fixed known issue: Correction of the compression management
- Implemented asynchronous management of Autocontrol for Implementation1/2
- Added callback to notify upper layer that the ECU will switch to Application
- Corrected SBL header management to avoid issue with compiler the two variable for SBL header has been set into the same structure

**Module version 0.0.1**
2015-04-29

- Module creation

### 2.3.4.2. New features

- Dual Memory Bank

  Description:

  The bootloader allows the reprogramming of a second memory bank (inactive memory bank) while software is running from the first one (active memory bank).

  Date

  2019-11-19
- HSM Software Update

  Description:

  The bootloader allows the reprogramming of HSM module.

  Date

  2019-08-30
- Allow session transitions without the SecurityLevel reset

  Description:

  The bootloader can allow the non-default session transitions without SecurityLevel reset for Impl31

  Date

  2019-07-09
- SBI flag Callbacks

  Description:

  The bootloader calls customer callbacks for set/get of the SBI flag which be located by the customer.

  Date

  2019-07-01
- Protected Calibration

Description:

The bootloader do not revoke protected calibrations on application erasing and do a compatibility check of the protected calibrations modules on application RTE.

Date

2019-03-08

- Programming counter

  Description:

  The bootloader now blocks the programming after a configurable counter lock value for Impl31

  Date

  2018-11-16

- Add Address and Length to the data to perform the signature verification

  Description:

  The Bootloader can verify the signature by including address and length extracted from RD to the data to verify.

  Date

  2018-10-23

- Signature on compressed data

  Description:

  The Bootloader can verify the signature on data passed through Transfer Data requests.

  Date

  2018-10-23

- Block header reading

  Description:

  The Bootloader can read a block header and use it to validate the block content.

  Date

  2018-10-22

- Resumable reprogramming

Description:

The Bootloader, in case of download interrupt, can restart the download from the beginning of the current memory segment.

Date

2018-10-08

- Compressed Flash Driver

  Description:

  The Flash Driver is compressed in bootloader binary and is decompressed in RAM after a successful Security Access.

  Date

  2018-10-08

- Security Access in Application

  Description:

  The bootloader can accept a Securitylevel change that was triggered by a SecurityAccess in Application.

  Date

  2018-04-04

- Flash Erase All

  Description:

  The bootloader can erase all the Flash memory with only one request (without parameters).

  Date

  2018-04-04

- Downloading Flash Driver

  Description:

  The bootloader can download the flash routines using UDS requests.

  Date

  2018-03-13

- Data Decryption

Description:

The bootloader can provide a callback for Data Decryption.

Date

2018-02-07

- Signature computation on the fly

  Description:

  The bootloader can manage the verification of cryptographics signature during the software download or after the software download for impl50

  Date

  2017-10-06

- Several external memory with a different access modes

  Description:

  The bootloader can manage several memories (1 internal, 1 external, 1 RAM) with a different access mode for each one

  Date

  2017-10-06

- Security access with crypto callbacks

  Description:

  The bootloader get callbacks for symmetrical cryptography key and for asymmetrical cryptography public key

  Date

  2017-04-28

- Partial software checksum

  Description:

  The bootloader now can receive and handle the partial software checksum request for Impl40

  Date

  2017-02-21

- Programming counter

  Description:

  The bootloader now blocks the programming after a configurable counter lock value for Impl40 and Impl50

  Date

  2017-02-20

- Multiple buffer management

  Description:

  The bootloader now handles multiple buffers, which allow to receive transfer data requests while writting the previous ones received.

  Date

  2017-01-07

- Add Implementation 60 features

  Description:

  The bootloader is now able to run with features for implementation 60

  Date

  2017-01-05

- Coherency check

  Description:

  The bootloader now handles the coherency check of the blocks existing in the ECU's flash memory.

  Date

  2016-11-30

- Logical block management and Download by logical address

  Description:

  The bootloader now handles logical blocks and the download by logical address.

  Date

  2016-06-27

- CheckMemory routine design

Description:

Management of the CheckMemory routine for CRC verification. Used for the validation of the downloaded application/calibration.

Date

2016-06-07

- Reset cause and DSC/ER response management

  Description:

  A way of knowing the reset cause is needed as well as if a response should be sent after reset and to witch service.

  Date

  2016-06-06

- Programming pre-condition check

  Description:

  Before starting the programming the pre-conditions are checked.

  Date

  2016-06-02

- Add F180 DID Service Management

  Description:

  The bootloader now handle the Boot Information (Number of Modules, Boot Id, Part Number, DLS)

  Date

  2016-05-26

- Add ARLE decompression support

  Description:

  The bootloader now handle ARLE decompression using Autosar decompression module

  Date

  2016-04-18

- Programming status management

Description:

Programming status available at the end of each programming cycle.

Date

2016-03-02

- FCA Autocontrol management

  Description:

  Autocontrol of the programmed application/calibration validity (for FCA).

  Date

  2016-03-02

- Add of Anti-scaning management

  Description:

  In the case of an unlock request received by the ECU a key is needed. If this key is wrong a retry is accepted. If the key received by the ECU is wrong the second time also, the ECU will be locked for a configurable period of time.

  Date

  2016-03-02

- Write Fingerprint management

  Description:

  Writing of programmed application, calibration or bootloader fingerprint in ECU memory.

  Date

  2016-03-02

- Add management of flashPage and skippage in product for implementation 2

  Description:

  The bootloader now handle sending the two feature flashPage and skippage for the implementation 2

  Date

  2015-12-15

- Add the possibility to respond to DSC 02 received in application from the bootloader

Description:

The bootloader now handle sending the response from a reprogramming request received from the application layer

Date

2015-11-17

- Add the management of an offset in the external Flash addressing

  Description:

  The bootloader is now able to manage the external Flash that shall be addressed by a tool in with an address offset.

  Date

  2015-11-13

- Added new feature for full cryptography management

  Description:

  The bootloader now manage the validation of the signed header (signer + signature) and the hash calculation at the of the reprogramming sequence

  Date

  2015-09-28

- Added new feature for SBA check at bootloader startup

  Description:

  The bootloader now manage the SBA ticket validation at startup

  Date

  2015-09-17

### 2.3.4.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

#### 2.3.4.4. Deviations

This module is not part of the AUTOSAR specification.

#### 2.3.4.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- The already erase feature is currently not available

  Description:

  The feature already erase that allow not to erase the memory if it is already erase is not available.

  Rationale:

  The memory erasing will be done in any case even if it is already erased

  Requirements:

  SwAD-ARCH-0218-1

#### 2.3.4.6. Open-source software

osc_PROG module does not use open-source software.

### 2.3.5. ProgOEMInd module release notes

- Module version: 1.3.8.BL-3.X_B280905
- Supplier: Elektrobit Automotive GmbH

#### 2.3.5.1. Change log

This chapter lists the changes between different versions.

**Module version 1.3.8**
2019-12-09

- New feature: Ethernet Sync after reset

**Module version 1.3.7**
2019-09-26

- OSCPROGOEMIND-81: Fixed known issue: [OEMInd] Unexpected call to callback PROG_CustomSetDownloadVerificationSuccess() when it returns PROG_E_NOT_OK

**Module version 1.3.6**
2019-07-23

- OSCPROG-1821: Fixed known issue: [OEMInd] ECU does not return to Lock state when GetSeed is not allowed (NRC_37)
- OSCPROGOEMIND-73: Fixed known issue: [OEMInd] Continuous pending response when some transitions are requested

**Module version 1.3.5**
2019-03-22

**Module version 1.3.4**
2018-10-25

- Internal module improvement. This module version update does not affect module functionality
- OSCPROG-1388: Fixed known issue: ECU is in infinite Response pending when receiving Check Memory Request after Check Programming dependencie
- OSCPROG-1388: Fixed known issue: Correct bad behaviour when receiving successive CheckMemory requests
- Internal module improvement. This module version update does not affect module functionality
- Internal module improvement. This module version update does not affect module functionality

**Module version 1.3.3**
2018-06-14

**Module version 1.3.2**
2018-06-11

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.3.1

2018-03-22

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.3.0

2017-12-18

- New feature: Add signature check support for CheckMemory

### Module version 1.2.1

2017-10-26

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.2.0

2017-10-13

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.1.0

2017-09-08

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.0.2

2017-05-11

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.0.1

2017-04-03

- Improved sequence management

**Module version 1.0.0**
2016-11-24

- Implemented Hsm for OEM Independent

### 2.3.5.2. New features

- Add Hsm for OEM Independent

### 2.3.5.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

### 2.3.5.4. Deviations

This module is not part of the AUTOSAR specification.

### 2.3.5.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- For this module no limitations are known.

### 2.3.5.6. Open-source software

osc_Prog_OEMInd module does not use open-source software.

## 2.3.6. SA module release notes

- Module version: 1.6.3.BL-3.X_B280905

- Supplier: Elektrobit Automotive GmbH

### 2.3.6.1. Change log

This chapter lists the changes between different versions.

**Module version 1.6.3**
2019-09-25

**Module version 1.6.2**
2019-09-25

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.6.1**
2019-07-17

- Implemented the feature "Standard OEM security algorithm" for profile 10
- Internal module improvement. This module version update does not affect module functionality

**Module version 1.6.0**
2019-04-01

**Module version 1.5.0**
2019-03-21

- Implemented Static Seed management
- Implemented support to crypto ASR 4.3 stack via the Demo_CSM_Wrapper

**Module version 1.4.4**
2018-10-25

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.4.3**
2018-07-24

- Added publick key exponent support

**Module version 1.4.2**
2018-07-20

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.4.1**
2018-06-14

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.4.0**
2018-03-22

- Improved callback using to allow SA to use its own public key

**Module version 1.3.0**
2017-12-18

- Removed Csm initialization (managed by the BM plugin)

**Module version 1.2.1**
2017-10-26

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.2.0**
2017-10-13

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.1.1

2017-08-02

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.1.0

2017-07-11

- OSCSA-182: Fixed known issues: No response to two consecutive seed requests
- OSCSA-183: Fixed known issues: No response to seed request after a failed unlocking attempt

### Module version 1.0.7

2017-06-12

- OSCSA-173: Fixed known issues: Correct bug SecurityAccess anti-scanning timer cannot be set to value higher than 65s

### Module version 1.0.6

2017-05-11

- New feature: Added crypto support with random seed generation and signature verification

### Module version 1.0.5

2017-04-03

- Internal module improvement. This module version update does not affect module functionality

### Module version 1.0.4

2016-12-16

- Improved tresos parameter for antiscanning now only available if antiscanning is enable
- Correct Seed generation in order to take into account Seed Size

### Module version 1.0.3

2016-10-10

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.2**
2016-08-12

- Internal module improvement. This module version update does not affect module functionality

**Module version 1.0.1**
2016-05-31

- New feature: Adding of Anti-scaning management

**Module version 1.0.0**
2016-03-29

- Internal module improvement. This module version update does not affect module functionality

**Module version 0.0.3**
2016-01-26

- Internal module improvement. This module version update does not affect module functionality

**Module version 0.0.2**
2015-07-16

- OSCSA-71: Fixed known issues: Correct bug in get decompress data management where the decompress status where not correctly send back
- Improvement: remove input decompress buffer and use diag one instead

**Module version 0.0.1**
2015-04-29

- module creation

### 2.3.6.2. New features

- Added static seed support

  Description:

In the case of consecutive GetSeed requests without CompareKey request (the first generated seed was not used), the response to the second GetSeed will contain the seed generated on the first seed request.

Date

2019-02-26

- Added crypto support

  Description:

  Added random seed generation and signature verification

  Date

  2017-05-05

- Adding of Anti-scaning management

  Description:

  In the case of an unlock request received by the ECU a key is needed. If this key is wrong a retry is accepted. If the key received by the ECU is wrong the second time also, the ECU will be locked for a configurable period of time.

  Date

  2016-05-02

#### 2.3.6.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

#### 2.3.6.4. Deviations

This module is not part of the AUTOSAR specification.

#### 2.3.6.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- For this module no limitations are known.

### 2.3.6.6. Open-source software

osc_SA module does not use open-source software.

## 2.3.7. Uds module release notes

- Module version: 3.9.2.BL-3.X_B280905
- Supplier: Elektrobit Automotive GmbH

### 2.3.7.1. Change log

This chapter lists the changes between different versions.

**Module version 3.9.2**
2019-11-29

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.9.1**
2019-07-24

- Added the callback to notify positive response to be sent.

**Module version 3.9.0**
2019-03-21

- OSCUDS-419 Fix known issue: [ISO] Minimum size of the RequestDownload is not verified
- Added the callback to manage supplier services.
- Improved handling of response pending using hardware timer(external) with the aid of configuration parameter

**Module version 3.8.4**
2018-10-25

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.8.3**
2018-06-15

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.8.2**
2018-03-22

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.8.1**
2017-12-18

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.8.0**
2017-10-16

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.7.0**
2017-09-11

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.6.0**
2017-04-03

- Improved response pending management to be usable with BlPduR module and to be manage separatly from the basic scheduling
- OSCUDS-381: S3 timeout is no more reloaded at each UDS request.
- OSCUDS-383: Fixed know issue: Minimal length for WDBI / IOCBI is now 4

**Module version 3.5.5**
2016-12-16

- Updated S3 timer management adding a new API to stop this timer.

**Module version 3.5.4**
2016-10-10

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.5.3**
2016-08-12

- OSCUDS-328: Fixed known issue: UDS compilation error fix if Security check is activated an no ServiceDID or RoutineControl configured.

**Module version 3.5.2**
2016-05-30

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.5.1**
2016-03-29

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.5.0**
2015-11-18

- Improved of the NRC_78 timings management. It is now handled per sessions / only ISO variant

**Module version 3.4.3**
2015-07-15

- OSCUDS-286: [ISO] No NRC_7E or NRC_7F negative response to functionaly adressed request (ISO 14229-1:2013(E))

**Module version 3.4.2**
2015-04-29

- Improved of the NRC_78 timings management. It is now handled per sessions / only ISO variant

**Module version 3.4.1**
2015-04-07

- OSCUDS-259: [RSA_MIXED] Useless suppressPosRspMsgIndicationBit management removed
- OSCUDS-248: Fixed known issue: NRC_78 supported for DSC service

**Module version 3.4.0**
2015-03-06

- OSCUDS-233: [PSA_CAN_LS/RSA_MIXED] Fixed known issue: Security check callback (UDS_SecurityCheck) missing for RoutineControl service (NRC33)
- OSCUDS-261: Improvement: Security check feature managed by UDS

**Module version 3.3.1**
2014-11-06

- OSCUDS-239: Fixed known issue: HIS_STMT_Max metric error fixed (UDS_FilteringCfg2).

**Module version 3.3.0**
2014-10-01

- OSCUDS-237: Fixed known issue: Tresos fields "UDS_MAX_DID_MULTI_RDBI" range had been updated to prevent compiler warning (range = [0 ; 1000]).

**Module version 3.2.0**
2014-08-18

- OSCUDS-233: Fixed known issue: Security check callback (UDS_SecurityCheck) missing for some services (NRC33)

**Module version 3.1.4**
2014-06-05

- OSCUDS-220: Added Specific NRC definition in UDS_Types.h
- OSCUDS-229: [PSA_CAN_LS] Fixed known issue: Unpredictable NRC returned on RoutineControl request too short

**Module version 3.1.3**
2014-04-29

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.1.2**
2014-04-24

- OSCUDS-207: [PSA_CAN_LS] Fixed known issue: Management of suppressPosRspMsgIndicationBit if response pending (NRC78) is activated
- OSCUDS-142: [PSA_CAN_LS] Add of a new API UDS_ForceRoutineControlStatus to force RoutineControl Status
- OSCUDS-220: Added Multi DID for RDBI UDS service

**Module version 3.1.1**
2014-02-13

- Internal module improvement. This module version update does not affect module functionality

**Module version 3.1.0**
2014-01-27

- OSCUDS-203: [ISO] Fixed known issue: Management of suppressPosRspMsgIndicationBit if response pending (NRC78) is activated
- OSCUDS-205: Update APP_UdsSessionStatusInd and UDS_SessionStatusInd callbacks adding new arguments (new session, old session and the reason of the changing session)

**Module version 3.0.2**
2013-11-12

- OSCUDS-193: Fixed known issue: Number max of DID allowed must be configurable in Tresos

**Module version 3.0.1**
2013-10-02

- OSCUDS-190: Add of a new API to get the current running session (UDS_GetCurrentSession)

**Module version 3.0.0**
2013-06-26

- OSCUDS-173: Variants management: The management of module multi-variants has been improved for ISO, PSA_CAN_LS and RSA_MIXED variants. Only the ordered variant is now visible.
- OSCUDS-178: Fixed known issue: Code may create warning if no OBD service configured.
- OSCUDS-180: Design document has been updated.
- OSCUDS-181: Fixed known issue: undefined reference to _UDS_LongRequestResponseInd, only for PSA_CAN_LS and RSA_MIXED variants.
- OSCUDS-183: Improvement: The variant name appears clearly into the header of all source files.

### 2.3.7.2. New features

- [ISO]New handling of NRC_78 timings

  Description:

  The timings for P2/P2* are now handled per sessions. This impact a rework on the UDS plugin design. Introduction of a generic adjust value for P2/P2* as well. This is substracted from P2/P2* timings configured on sessions.

  Date

  2015-10-21

- [ALL]Added security check management in UDS

  Description:

  Security check is now managed by UDS. The security level is configurable for each services configured in Service, Service_DID or Routine_Control. Careful: a callback to get the current security level shall be provided to UDS. Please refer to Userguide for details.

  Date

  2015-02-11

- [ALL]Added Multi DID for RDBI UDS service

Description:

Multi DID for RDBI UDS service is now available for all variants. Careful: *puwLen is not the received length but the current buffer size consumed. Please refer to Integration Manual for details.

Date

2014-04-17

- [PSA_CAN_LS] Added new API to force the RoutineControl status for specific RoutineIdentifier (UDS_-ForceRoutineControlStatus)

  Description:

  API can be used by application if you need to reset the status or RoutineControl after reset or session switching for example

  Date

  2014-04-07

- Added new API to get the current running session (UDS_GetCurrentSession)

  Description:

  Added new API to get the current running session (UDS_GetCurrentSession)

  Date

  2013-10-02

### 2.3.7.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

### 2.3.7.4. Deviations

This module is not part of the AUTOSAR specification.

### 2.3.7.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- Usage of NRC_78 on RDBI request when MULIT-DID is activated

  Description:

  If MULTI-DID is activated and some RDBI requests use NRC_78, some DID informations will be missing in UDS answer. For example: * if DID#2 uses NRC_78 * tester send the request RDBI DID#1 DID#2 DID#3 The ECU will take into account only the request for DID#1 and DID#2; DID#3 request is lost.

### 2.3.7.6. Open-source software

osc_UDS module does not use open-source software.

# 3. BL for Essentials user guide

## 3.1. Overview

This chapter provides user guide for the BL for Essentials modules.

## 3.2. EB Tresos Bootloader user's guide for OEMInd

### 3.2.1. Introduction

This document present the different integration requirements of the EB tresos bootloader product

Three types of requirements will be described:

- Interface requirements between Bootloader and integration code
- Configuration requirement
- General performance requirement

### 3.2.2. Integration steps

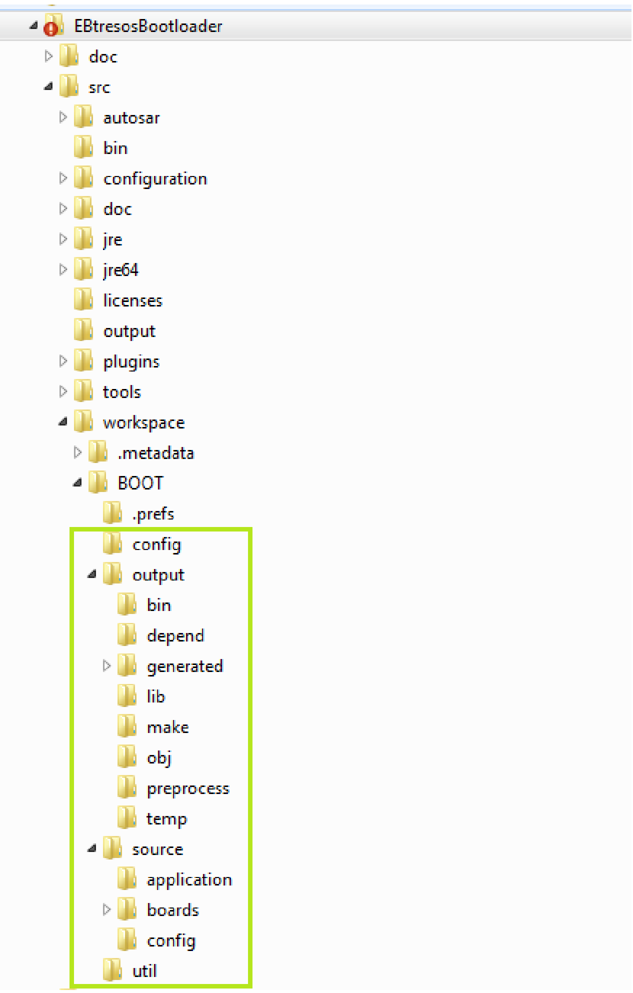
In order to perform the Bootloader integration, it is adviced to follow the following steps:

1. Read EB tresos Bootloader Documentation:
    - Bootloader implementation matrixes:
        - identify for every OEM requirements if the requirement is fully/partially or not implemented in EB tresos Bootloader software module
        - allow integrator identifying the remaining work to do to have a full OEM compliant Bootloader
    - EB tresos Bootloader Generic Documentation: Bootloader environment documentation (Compiler,Base,Make, platform,demo)
    - EB tresos Bootloader for 'OEM' documentation (This document): Public Api definition, user guide
2. Run the delivered Demo software on evaluation board or adapt it to make it runs on project specific board (e.g real ECU)

3. Build the bootloader for your ECU/project:

   - Replace or update, according to project/hardware needs, code and configuration from demo.
   - Update Linkerfile according to your memory mapping
   - Update Modules configuration (see advice after)
   - Implement callbacks present in template directory (Project/hardware dependent)
   - Check in implementation matrix requirements that are not implemented in Bootloader software modules (Project/Hardware specific) and implement them.

Here is some information about integration workspace folder structure:



```
EBtresosBootloader
  doc
  src
    autosar
    bin
    configuration
    doc
    jre
    jre64
    licenses
    output
    plugins
    tools
    workspace
      .metadata
      BOOT
        .prefs
        config
        output
          bin
          depend
          generated
          lib
          make
          obj
          preprocess
          temp
        source
          application
          boards
          config
        util
```

Figure 3.1.

**config**: plugins configurations file (see advice after)

**output**: output files of the build process:

- **bin**: Bootloader binary file
- **generated**: all files generated by Tresos (shall not be modified)
  - **include**: generated include files
  - **make**: generated makefile using for Bootloader software build
  - **src**: generated source files
  - **templates**: generated template files that shall be completed by integrator. The files from this folder are not compiled and shall be copied in application folder before being modified and compiled.

**source**: integration source code (shall be modified, completed by integrator)

- **application**: integration specific files that shall be implemented by integrator.
- **board**: hardware specific Api, startup code and linker files (shall be modified, completed by integrator)
- **config**: Autosar Compiler_Cfg.h and MemMap.h (shall be modified, completed by integrator)

**util**: workspace build scripts

Table 3.1.

Here is some advice for configuration management:

- Communication stack modules: Configure according to OEM requirements (use EcuExtract if provided by the OEM)
- UDS:
    - Update with additional DIDs for WDBI and RDBI service
    - Update with additional routines
    - Add supplier specific session (if needed)
- BM: use demo configuration
- CSM/CRY/CPL/CAL (if ordered): use demo configuration
- SA (if required): use demo configuration
- EB/BlPduR: use demo configuration and update it if additional Pdus are required
- APP: use demo configuration
- FLASH: use demo configuration
    - Only adapt sector configuration if ECU micro-controller memory size is different from delivered micro-controller configuration.
- PROG:
    - Update project specific parameters in General container
    - Configure Blocks according to application mapping, if OEM required logical block addressing for erasing/writting
    - Configure Segments according to application mapping

## 3.2.3. General overview

Important note: In order to run qualification process on a target, a startup code is required. This startup code is a sample created by Elektrobit or directly received from the hardware manufacturer. This is for EB internal testing purposes only. The startup code provided within Elektrobit release (startup assembly code and C implementation in board.c/.h) is not tested nor qualified for series production use. The qualification statement does not refer to the startup code. The user shall not be entitled to use the startup code in connection with any series production. It shall be replaced by an operating system or a project and ECU specific startup code, which is qualified for mass production.

### 3.2.3.1. Layer overview

The following pictures show you the EB tresos Bootloader architecture.

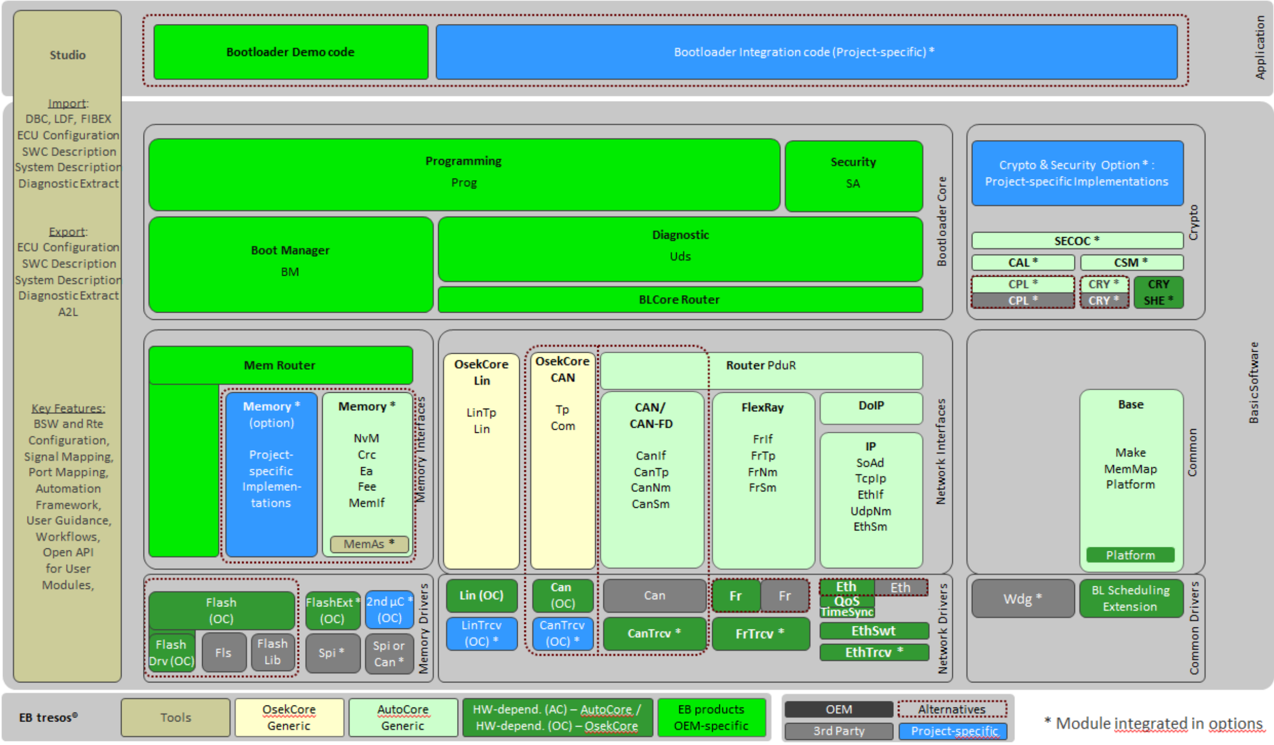


Figure 3.2. Bootloader architecture

The following software modules constitute the EB tresos Bootloader:

- BM: BootManager that is used at startup to route the software to bootloader or application software
- BLCore Router: Software router between all the communication stack and the bootloader core. Depending on the communication stack this router can be two different module:
  - If the communication stack is a OsekCore Can based the module BLCore Router will be managed by module EB
  - If the communication stack is an Autosar based (any network) the module BLCore Router will be managed by module BlPduR
- UDS : UDS ISO compliant for diagnostic service management
- PROG : Bootloader core software divided in two parts:
  - PROG<OEM> containing the bootloader state machine
  - PROG containing the programming features
- SA (optional) : Manage Security Access UDS request and the LZSS decompression algorithm
- FLASH : Flash driver called by the bootloader to write erase and read from Flash memory

The communication stack (CAN, FlexRay, Ethernet) shall be configured by the integrator except if the configuration package had been purchase.

If Autosar communication stack is used, configuration can be partially done by importing an ECU extract.

### 3.2.3.2. plugin files structure

Each plugin is composed with two kinds of files:

- Core files
- Generated files

The core files don't contain any configuration parameters, they are stored in the base folder "src" for the .c files and "include" for header files. These files don't have to be modified by the integrator.

The generated files are template source files completed by the EB tresos Studio tool, taking into account the module configuration. They are stored in "generate/src" for the .c files and "generate/include" for header files. These files don't have to be modified by the integrator.

The generated files to compile or include after generation are in the workspace of the EB tresos Studio of the project

NOTE: Some plugins doesn't contained any core files (e.g. EB) all the files are generated from EB tresos Studio.

The template files are template source files which have to be moved from the "generated/templates" folder to the "src/application" folder in the workspace of the EB tresos Studio of the project. They are stored in "generate/template" for the .c files and header files. These files can be modified by the integrator.

## 3.2.4. Response Pending scheduling

This chapter defines integration to STM(System) Timer Intialisation and interrupt configuration.

### 3.2.4.1. STM Timer ISR

The STM timer needs to be initialised and Interrupt generation on Timer Compare is enabled. Compare register to be loaded with value which corresponds to 1 millisecond based on the MCU clock.

If stm timer interrupt shall be used to schedule the NRC78 sending, the API UDS_ResponsePending_TimCntManage shall be used in timer ISR and parameter TIMER_RESPONSE_PENDING_CHECK shall be checked(enabled) in uds configuration(tresos). The use of the timer interrupt ensure to not miss sending of NRC 78 at P2 and P2 star time.

## 3.2.5. Can Integration

This chapter defines the integration constraints when using the CAN driver.

### 3.2.5.1. CAN RX/TX interrupt priority levels

Part of the services provided by the bootloader software is triggered upon the external events, such as the reception of certain CAN messages and the transmission of certain CAN messages. At the integration phase, if the CAN driver uses the interrupts for sending/receiving CAN messages, user shall configure CAN TX interrupt with a higher priority than that of CAN RX interrupt. So if both CAN TX interrupt flag and CAN RX interrupt flag are set, the ISR of CAN TX will be served first.

## 3.2.6. Configuration

This chapter defines configuration parameters that are advised for the correct behaviour of the bootloader.

### 3.2.6.1. Notes

Configuration parameters that are not described in this document shall be adapt according to the project needs, they are not mandatory and their use is depending of the project integration. You can refer to the parameter description in EB tresos Studio.

Some software modules contains OEM dedicated configuration parameters, these parameters won't be available if you are not integrating a Bootloader for this OEM. Please ignore these parameters in such case.

Some configuration parameters can be automatically disabled depending of your configuration/integration and appear "grayed" in EB tresos configuration. Their value will be unused so please ignore these parameters in such case.

### 3.2.6.2. Prog Configuration

#### 3.2.6.2.1. General

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0071 |

| | |
|---|---|
| Version: | 1 |
| Description: | The integrator shall ensure that "Max_Bytes_in_TD" field is filled with a value multiple of "Min_Value_To_Flash" field value + 2. |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0079 |
| Version: | 1 |
| Description: | The integrator shall ensure that if the Demo_CSM wrapper shall be used for integrating crypto stack ASR version 4.3:<br>▶ The Use_CSM_ASR430_DemoWrapper checkbox is ticked/enabled<br>▶ Crypto stack ASR version 4.3, demo_csmwrapper.c and demo_csmwrapper.h files are integrated in the Bootloader<br>▶ Demo_SA_RNDcallback_403 and Crypto_MainFunction functions are called from the scheduler as background tasks |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0123 |
| Version: | 1 |
| Description: | The integrator shall ensure that if Dual Memory Bank feature shall be used:<br>▶ The "Dual Memory Bank Used" checkbox is ticked/enabled |

#### 3.2.6.2.1.1. EraseState Callbacks

##### 3.2.6.2.1.1.1. PROG_CustomIsFirstProgramming Callback

This callback is called on reception of Erase routine

It allows skipping the erasing if the ECU Flash has never been programmed and so is already fully erased. It allows saving time during ECU first download.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0114 |
| Version: | 2 |
| Description: | The integrator shall implement in PROG_CustomIsFirstProgramming callback software providing information if Flash was never programmed before and that erase shall be skipped. The callback should return PROG_TRUE if the Flash was never programmed and PROG_FALSE otherwise. |

#### 3.2.6.2.1.1.2. PROG_CustomDownloadNotification Callback

This callback is called on reception of RequestDownload routine

It allows the update of the first programming flag to point that the ECU Flash has already been programmed.

Optionally, other treatments related to the starting of the Flash memory programming can be done by the integrator in this callback.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0159 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomDownloadNotification callback software by updating the information that Flash was already programmed before. This indicates that, for any other further programming, the Flash memory should be erased. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0020 |
| Version: | 1 |
| Description: | The integrator shall ensure that "Transmit_Response_Before_Reset" field is set. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0022 |
| Version: | 1 |
| Description: | The integrator shall ensure that "Transmit_Nrc78_On_SecurityAccess" field is set if the Security Access seed/key generation process is long (more than P2 time). |

#### 3.2.6.2.2. Memory

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0070 |
| Version: | 2 |
| Description: | The integrator shall configure all memory used for the project. Only 1 memory of each type (FLASH, FLASH_EXT, RAM) can be configured. |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0077 |
| Version: | 1 |
| Description: | For every memory, the integrator shall configure the memory type: Internal Flash (FLASH), external Flash (FLASH_EXT) or RAM memory (RAM). |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0073 |
| Version: | 1 |
| Description: | For every memory, the integrator shall configure the Memory Mode parameter depending if the Flash driver support synchronous or asynchronous interface call. Synchronous means that when the Flash Api is called, it will returns only when the request operation is performed. Asynchronous means that when the Flash Api is called, it returns before performing the requested operation, Prog module will later call periodically a "GetJobStatus" Api to be informed when the operation is finished. In case synchronous mode is used Prog module will request the erasing of Flash sector per sector to avoid a too long block time is Flash call. |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0074 |
| Version: | 1 |
| Description: | For every memory, the integrator shall ensure that "Minimum value to write" field is set to the minimum size that shall be write for the memory (Flash page size). |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0075 |
| Version: | 1 |
| Description: | For every memory, the integrator shall configure the address offset to be used when accessing the memory. It's used to convert the logical address get from the diagnostic request to the physical address of the memory. |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0076 |
| Version: | 1 |
| Description: | For every memory, the integrator shall configure the erase state value of the memory (0x00 or 0xFF depending of the memory architecture). |

#### 3.2.6.2.3. Block and Segments

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0072 |
| Version: | 1 |
| Description: | The integrator shall ensure that the configuration of the memory areas manipulated by the flashloader is consistent against protected areas (hardware protection key, any other sections that shall not be changed) |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0078 |
| Version: | 1 |

| | |
|---|---|
| Description: | The integrator shall ensure to map every segment to the correct memory. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0120 |
| Version: | 1 |
| Description: | The integrator shall ensure that at least one Aplication segment is configured as follow:<br>▶ Memory_Type set to FLASH<br>▶ Access_Type set to READ_WRITE<br>▶ Reprog_Start_Address is the start of Application area<br>▶ Reprog_End_Address is the end of Application area<br>▶ Erase_Start_Address is equal to Reprog_Start_Address<br>▶ Erase_End_Address is equal to Reprog_End_Address<br>▶ Partition Type is equal to PROG_APPLICATION_PARTITION<br>NOTE: Additional application or calibration segment can be configured. |

### 3.2.6.2.4. Oem independent specific parameters

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0100 |
| Version: | 1 |
| Description: | The integrator shall configure Erasing Mode parameter according to the expected erasing request. The following values are possible:<br>▶ All: No information are provided in Erase request, on receception of the erase request, all the memory segments will be erased.<br>▶ Address: The Erase request contains the address range to be erased, only 1 segment can be erased with this request. Note that in this case the checksum will be computed on a single segment.<br>▶ LogicalBlock: The Erase request contains the logical block to be erased. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0101 |
| Version: | 1 |
| Description: | The integrator shall configure the "Erase request ALFI Enable" parameter to indicate if the Erase request contains the UDS ALFI field. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0102 |

| | |
|---|---|
| Version: | 1 |
| Description: | The integrator shall configure the "Application validity" parameter to indicate if Elektrobit algorithm or customer specific algorithm shall be used. If customer specific, the integrator shall implement the PROG_InvalidateSection and PROG_CustomSetCrcCompareSuccess callbacks to manage the application validity status. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0103 |
| Version: | 1 |
| Description: | The integrator shall configure the "CRC algorithm" parameter to indicate which checksum shall be used to verify the software download. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0104 |
| Version: | 1 |
| Description: | The integrator shall configure the "FingerPrint Enable" parameter to indicate if a fingerprint shall be managed for the software download. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0105 |
| Version: | 1 |
| Description: | If FingerPrint is enable, the integrator shall configure the "Size_Of_FingerPrint" parameter to indicate the expected FingerPrint size. |

#### 3.2.6.2.5. Download verification

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-VERIF-BOOTLOADER-0001 |
| Version: | 1 |
| Description: | The integrator shall configure (parameter "Maximum RequestDownload Per Block") the maximum number of RequestDownload request that can be received, by the Bootloader, for a single logical block. This value will be used to size the RAM structure storing the downloaded memory area that shall be used to perform the signature/CRC verification. |

### 3.2.6.3. BM Configuration

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0010 |
| Version: | 1 |

| | |
|---|---|
| Description: | The integrator shall ensure that "BM_TIMEOUT_CHECK " field is NOT set. |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0011 |
| Version: | 1 |
| Description: | The integrator shall ensure that "BM_SOURCE_ADDRESS_CHECK" field is NOT set. |

### 3.2.6.4. SA Configuration

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0070 |
| Version: | 1 |
| Description: | The integrator shall ensure that the parameter "Security_Access_Seed_Length" is set to 4 |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0071 |
| Version: | 1 |
| Description: | The integrator shall ensure that the parameter "Security_Access_Key_Length" is set to 4 |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-BOOTLOADER-0072 |
| Version: | 1 |
| Description: | If the Static Seed (response with precedent seed in the case of successive GetSeed requests without a respective key received) is required for the project, the integrator shall ensure that "Enable_Static_Seed" field is set. |

### 3.2.6.5. UDS Configuration

#### 3.2.6.5.1. General

The parameter "External Response Pending Manage Call" allow the integrator to call the scheduling function UDS_ResponsePending_Manage by itself outside of the rest of the stack scheduling (e.g. in a task with higher priority). This API manage the timer for P2 and P2*.

If stm timer interrupt shall be used to schedule the NRC78 sending, the API UDS_ResponsePending_TimCntManage shall be used instead and parameter TIMER_RESPONSE_PENDING_CHECK shall be checked(enabled) in uds configuration(tresos). The use of the timer interrupt ensure to not miss sending of NRC 78 at P2 and P2 star time.

Note: This API shall be called at the configured period of UDS_MANAGE_PERIOD

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0104 |
| Version: | 1 |
| Description: | The integrator shall ensure that "Standard" field is set to "ISO" |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0105 |
| Version: | 1 |
| Description: | The integrator shall ensure that "UDS_MANAGE_PERIOD " field is set to the call period of the UDS_Manage Api |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0100 |
| Version: | 1 |
| Description: | The integrator shall ensure that "SecurityCheck" field is set and that the "SecurityFunction" is set to "PROG_GetSecurityLevel" |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0106 |
| Version: | 1 |
| Description: | The integrator shall ensure that "RC_NRC_IMPLEMENTATION " field is set to 0x31 |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0107 |
| Version: | 1 |
| Description: | The integrator shall ensure that "DID_NRC_IMPLEMENTATION " field is set to 0x31 |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0101 |
| Version: | 1 |
| Description: | The integrator shall ensure that "RESPONSE_PENDING" field is set |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0102 |
| Version: | 1 |
| Description: | The integrator shall ensure that "SPREC_IN_RESPONSE" field is set |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0103 |

| | |
|---|---|
| Version: | 1 |
| Description: | The integrator shall ensure that "RELOAD_TSTOPDIAG" field is set |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0108 |
| Version: | 1 |
| Description: | If for integration reason the NRC78 response pending shall be send before the end of the configured P2/P2star time, the integrator can configure the P2/P2star adjust parameter. In this case the UDS module will trigg the NRC78 transmission at a time equal to (P2/P2star - P2/P2star adjust) |

#### 3.2.6.5.2. Session

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0120 |
| Version: | 1 |
| Description: | The integrator shall configure every of following session: DEFAULT, PROGRAMMING, EXTENDED, SUPPLIER, OTHER_01, OTHER_02, OTHER_03, OTHER_04 |

#### 3.2.6.5.3. Service

The integrator shall configure in Service configuration window all service (except ReadDataByIdentifier, WriteDataByIdentifier and RoutineControl) that shall be supported by the ECU

#### 3.2.6.5.4. Service_DID

The integrator shall configure in Service_DID configuration window all DIDs that shall be supported by the ECU

#### 3.2.6.5.5. Routines

The integrator shall configure in Routine_Control configuration window all routines that shall be supported by the ECU

### 3.2.6.6. BlPduR Configuration

This configuration is only available if the project is based on a ASR communication stack

#### 3.2.6.6.1. Connection configuration

A connection define the reception PDUs and transmission PDU that shall be used to communication with a diagnostic tester. In order to ensure Bootloader communication, at least one connection shall be defined. If the Bootloader shall be able to communicate with several testers, several connections shall be defined.

For every connection, the following parameters shall be configured:

| | |
|---|---|
| Id: | OSC-INTMAN--BLPDUR-BOOTLOADER-0001 |
| Version: | 1 |
| Description: | The integrator shall configure the TxPDU (TxPdu Reference) to be used for response transmission. It shall reference a valid ECUC PDU. |

| | |
|---|---|
| Id: | OSC-INTMAN--BLPDUR-BOOTLOADER-0002 |
| Version: | 1 |
| Description: | The integrator shall configure the TxPDU confirmation identifier (TxPdu Identifier) to be used by PduR module to confirm the transmission. The TxPdu Identifiers shall be unique and consecutives. |

| | |
|---|---|
| Id: | OSC-INTMAN--BLPDUR-BOOTLOADER-0003 |
| Version: | 1 |
| Description: | The integrator shall configure the Tester Address associated to this connection. |

| | |
|---|---|
| Id: | OSC-INTMAN--BLPDUR-BOOTLOADER-0004 |
| Version: | 1 |
| Description: | The integrator shall indicate if this connection is associated to a LIN communication (Lin Connection) . |

| | |
|---|---|
| Id: | OSC-INTMAN--BLPDUR-BOOTLOADER-0005 |
| Version: | 1 |
| Description: | The integrator shall indicate if this connection shall re-use a functional RxPduId already defined in another connection (Share Functional Id). |

| | |
|---|---|
| Id: | OSC-INTMAN--BLPDUR-BOOTLOADER-0006 |
| Version: | 1 |
| Description: | If "Share Functional Id" is enabled, the integrator shall configured the shared Pdu Reference. |

| | |
|---|---|
| Id: | OSC-INTMAN--BLPDUR-BOOTLOADER-0007 |

| | |
|---|---|
| Version: | 1 |
| Description: | The integrator shall configure all the RxPDU to be used for request reception. It shall reference a valid ECUC PDU, specify if the Pdu is functionalor physical and the associated Pdu Identifier to be used by PduR module. The RxPdu Identifiers shall be unique and consecutives. |

| | |
|---|---|
| Id: | OSC-INTMAN--BLPDUR-BOOTLOADER-0008 |
| Version: | 1 |
| Description: | A connection shall contain only one functional RxPdu or one reference functional RxPdu. |

### 3.2.6.6.2. Multiple Receive Buffer Configuration

The Multiple Receive buffer allows the bootloader to receive a Transfer Data request while it writes the previous one received.

This feature allows to improve downloading performance but will greatly increase the RAM size usage.

In case the feature is enabled, the maximum number of buffer allowed is 4.

**Warning: This feature shall be used only if Hardware/MCAL is able to perform a Flash write in less than a CAN frame duration (Depending of the Hardware/MCAL a flash page writting can be superior to CAN frame duration or the writting of a single Flash page is not possible).**

**Otherwise there is a risk to lose CAN frame reception.**

**Warning: This feature can not be used if the feature Queued Requests is enabled.**

#### 3.2.6.6.2.1. Behavior on reception of a Transfer Data request

**Be aware that if the Multiple Buffer feature is activated, behavior will no more be compliant with ISO-14229. Bootloader will always acknowledge the first received Transfer Data. Answers sent by the bootloader after the reception of Transfer Data requests will always concern the Transfer Data request previously received.**

This behavior can be explained by the fact that bootloader needs to send quickly an answer to tester tool to trigger the transmission from tester tool of the next Transfer Data request.

This answer is done:

- Before the current Transfer Data request is processed.
- Regardless the true state of the Transfer Data request currently processed.

#### 3.2.6.6.3. Multiple Identifier Configuration

The Multiple Identifier feature is only available for Can network. It allows the bootloader select a group ID that contains one or multiple connections. The received Can frame configured will then be filtered depending on the current group activated

In order to select a group ID, an "Architecture Frame" (which identifier is configured), shall be sent at startup before a delay (which is also configured). It will contain the information of group ID to select.

Anyway, the "Architecture Frame" is not taken into account at each restart of the ECU. So it is not possible to select a group ID each time that ECU restarts.

The "Architecture Frame" will:

- be taken into account each time that the previous connection context is not needed.
- not be taken into account each time that the previous connection context is needed. It means that the previous group ID and the previous active connection before the restart will be retrieved.

Examples where previous connection context is not retrieved:

- if the feature "Send Response After Reset" is deactivated, then each request which makes the ECU restart will be responded before the restart. After restart, as no frame need to be transmitted before the timeout P2, no group ID and active connection will be set and the "Architecture Message" will be able to select one.
- if the feature "Send Response After Reset" is activated, and positive response bit is present then each request which makes the ECU restart will not be responded even after the restart. After restart, as no frame need to be transmitted before the timeout P2, no group ID and active connection will be set and the "Architecture Message" will be able to select one.

Example where previous connection context is retrieved:

- if the feature "Send Response After Reset" is activated, then each request which makes the ECU restart will be responded after the restart. After restart, as frames need to be transmitted on the same group ID and same active connection before the timeout P2, group ID and active connection will be retrieved and the "Architecture Message" will not be able to select different one.

The multiple identifier configuration can be used following two differents configurations

- EXTERNAL_NOTIFICATION: The group ID will be retrieve at startup by the bootloader by calling a callback
- CAN_NOTIFICATION: The Bootloader wait at startup the reception of the CAN frame that has one byte containing the architecture ID of a group ID

#### 3.2.6.6.3.1. External Notification Configuration

With this configuration the customer shall only ensure to correctly fill the BlPduR_GetGroupIdVal callback

#### 3.2.6.6.3.2. Can Notification Configuration

With this configuration the customer shall ensure to configure correctly the following element

In MultipleIdentifier container:

- Multiple Identifier Timeout: value of the timeout after which if the CAN frame has not been received the default group ID will be used
- ID group PDU reference: Reference to the PDU that contain the Architecture ID
- ID group PDU Id: Pdu Id that shall be used by PduR to call notify the PDU reception
- ID group Byte Number: Byte where the Architecture information is stored in the PDU (0 is LSB, 7 is MSB)

In IDGroup container:

- Default ID group: One default group ID shall be selected
- Architecture Id: In every group ID the architecture value shall be defined

### 3.2.6.7. Crypto Configuration

If signature verification is used for the project the following configuration shall be performed (required configuration is for RSA-PSS 2048 bits with SHA256) .

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0001 |
| Version: | 1 |
| Description: | The integrator shall configure in CRY module a CryRsaSsaPssVerify configuration with the following configuration:<br>- CryRsaSsaPssVerifyUseTimeSlices (Use time slicing for RSASSA-PSS signature verification) set to true<br>- CryRsaSsaPssVerifyNumberOfTimeSlices (Number of RsaSsaPss time slices) set to 10<br>- CryRsaSsaPssVerifyUseCbk (Use configured callback function which returns maximum number of time slices) set to false<br>- CryRsaSsaPssVerifyImmediateRestartEnabled (Enable the cancelation of an ongoing calculation regardless of the configuration ID) set to true |

| | |
|---|---|
| | ▶ CryRsaSsaPssVerifyHashCfgRef (Hash configuration) set to CsmHashConfig_0<br>▶ CryRsaSsaPssVerifyKeyLength (Key Length) set to 256<br>▶ CryRsaSsaPssVerifySaltLength (Salt Length) set to 0<br>▶ CryRsaSsaPssVerifyB64Encoded (Base64 Encoded) set to false<br>▶ CryRsaSsaPssVerifyUseBarrett (Barrett reduction) set to false<br>▶ CryRsaSsaPssVerifySupportRestart (Enable the cancelation of ongoing requests) set to true |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0002 |
| Version: | 1 |
| Description: | The integrator shall configure in CRY module the following general parameters:<br>▶ CrySHAOneAndTwoImplementation (Implementation variant) set to CRY_SHAONEANDTWO_INTERRUPTABLE<br>▶ CryInterruptableLN (Interruptable LN operations) set to true |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0003 |
| Version: | 1 |
| Description: | The integrator shall configure in CRY module a CrySHA2 configuration with the following configuration:<br>▶ CrySHA2ImmediateRestartEnabled ( Enable the cancelation of an ongoing calculation regardless of the configuration ID) set to true<br>▶ CrySHA2Type (Prime) set to CRY_SHA_256<br>▶ CrySHA2IterationsPerMain (Number of iterations per MainFunction) set to 1<br>▶ CrySHA2SupportRestart (Enable the cancelation of ongoing requests) set to true |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0004 |
| Version: | 1 |
| Description: | The integrator shall configure in CSM module a CsmSignatureVerify configuration with the following configuration:<br>▶ CsmCallbackSignatureVerify set to "PROG_CsmNotification"<br>▶ CsmSignatureVerifyMaxKeySize set to 524<br>▶ CsmSignatureVerifyInitConfiguration set to CryRsaSsaPssVerifyConfig_0 (configuration done in OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0001) |

| | |
|---|---|
| | ▶ CsmSignatureVerifyEnableRteInterface (Enable Rte Interface) set to false<br>▶ CsmSignatureVerifyEnableRestart (Enable the cancelation of ongoing requests) set to true<br>▶ CsmSignatureVerifyUsePriorities (Csm priorities handling) set to true |

| | |
|---|---|
| Id: | OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0005 |
| Version: | 1 |
| Description: | The integrator shall configure a CsmHash configuration (CsmHashConfig_0) with the following configuration:<br>▶ CsmCallbackHash set to Cry_RsaSsaPssVerifyCallback<br>▶ CsmHashInitConfiguration set to CrySHA2Config_0 (configuration done in OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0001)<br>▶ CsmHashPrimitiveName set to SHA2<br>▶ CsmHashEnableRteInterface (Enable Rte Interface) set to false<br>▶ CsmHashEnableRestart (Enable the cancelation of ongoing requests) set to true<br>▶ CsmHashUsePriorities (Csm priorities handling) set to true |

### 3.2.6.8. Ethernet Configuration

This information is only suitable if the project is using the EB ethernet communication stack on ACG-8 or newer versions.

#### 3.2.6.8.1. MemMap related

From versions ACG-8 or newer, two new buffers have been added in ethernet driver: Eth_BufferRamRx and Eth_BufferRamTx. These two needs to be aligned on 8 bytes in memory to be handled correctly. To do so, a specific MemMap definition should be added, for each buffer, in project MemMap's configuration file (MemMap.xdm). The absence of these two new definitions could prevent the ethernet driver from any communication possibilities. Example of MemMapAddressingMode:

```
Eth_BufferRamRx:

    #if (defined EthBufferRx_ETH_BUFFER_RX_START_SEC_Common)
        #if (!defined ETH_MEMMAP_BUFFERRX_PRAGMA_EXECUTED)
      #define ETH_MEMMAP_BUFFERRX_PRAGMA_EXECUTED
```

```
  #if defined _GREENHILLS_C_PPC_
    #pragma alignvar (8)
  #elif defined _DIABDATA_C_PPC_
      #pragma section ETH_BUFFER ".ETH_BUFFER_SEG" ".ETH_BUFFER_SEG"
      #pragma use_section ETH_BUFFER Eth_BufferRamRx
  #elif defined _TASKING_C_TRICORE_
      #pragma align 8
  #endif
  #endif
    #undef EthBufferRx_ETH_BUFFER_RX_START_SEC_Common
    #undef MEMMAP_ERROR
#elif (defined EthBufferRx_ETH_BUFFER_RX_STOP_SEC_Common)
    #ifdef _GREENHILLS_C_PPC_
  #pragma ghs section bss=default
#elif defined _TASKING_C_TRICORE_
  #pragma align 0
#endif
    #undef EthBufferRx_ETH_BUFFER_RX_STOP_SEC_Common
    #undef MEMMAP_ERROR
```

### 3.2.7. Callbacks

The callbacks are pieces of code that shall be implemented by integrator.

The callbacks are divided in different types:

- The bootloader diagnostic service callbacks.
- The hardware related callbacks.
- The BootManager callbacks.
- The Security callbacks.
- The Programming callbacks.
- The Communication callbacks.

The callback implementation execution time shall be as short as possible in order not to block other Bootloader tasks.

| | |
|---|---|
| Id: | OSC-INTMAN-CBK-0001 |
| Version: | 1 |
| Description: | The integrator shall ensure that the callback execution doesn't block the processing of Bootloader MainFunction/Manage cyclic task (e.g BlPduR_Manage) |

### 3.2.7.1. Programming sequences

The following sequences show the common download sequence and indicate for every diagnostic request if a callback is called. These callbacks shall be implemented by integrator and help him identifying when the callback is called during the programming sequence.

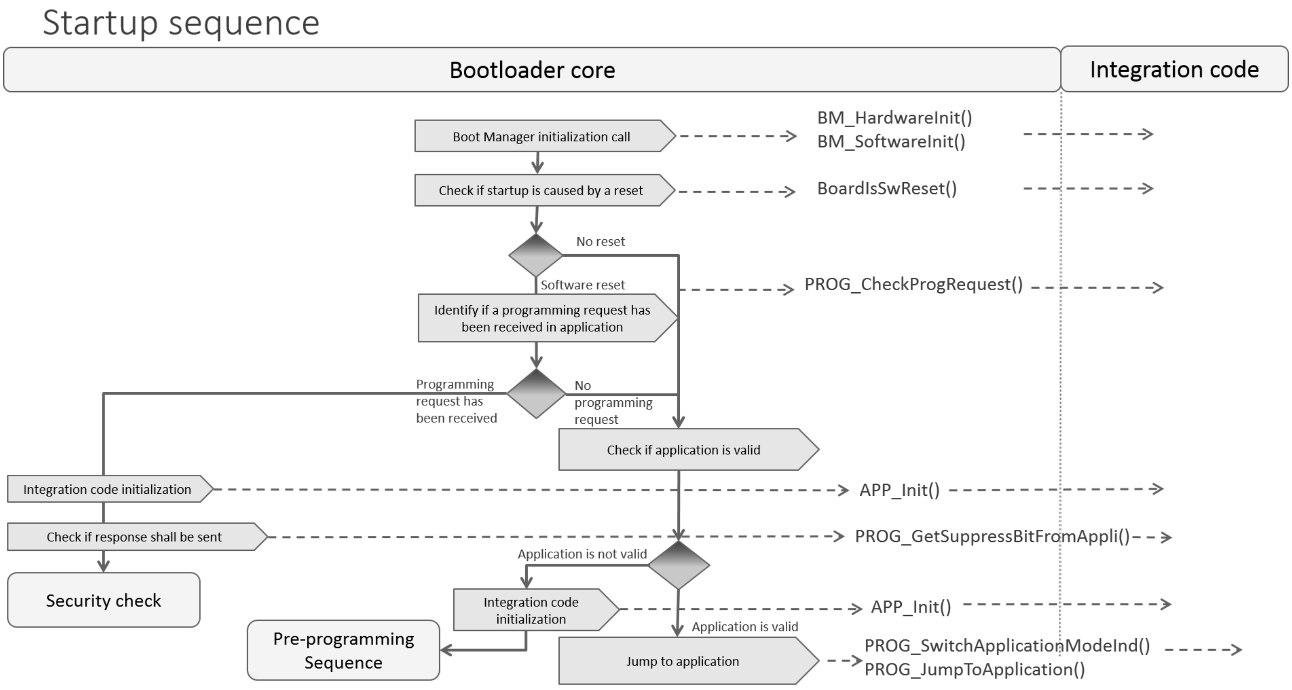

Startup sequence
Bootloader core
Integration code
Boot Manager initialization call
BM_HardwareInit()
BM_SoftwareInit()
Check if startup is caused by a reset
BoardIsSwReset()
No reset
Software reset
Identify if a programming request has been received in application
PROG_CheckProgRequest()
Programming request has been received
No programming request
Check if application is valid
Integration code initialization
APP_Init()
Check if response shall be sent
PROG_GetSuppressBitFromAppli()
Security check
Application is not valid
Integration code initialization
APP_Init()
Application is valid
Pre-programming Sequence
Jump to application
PROG_SwitchApplicationModeInd()
PROG_JumpToApplication()

Figure 3.3. Startup sequence

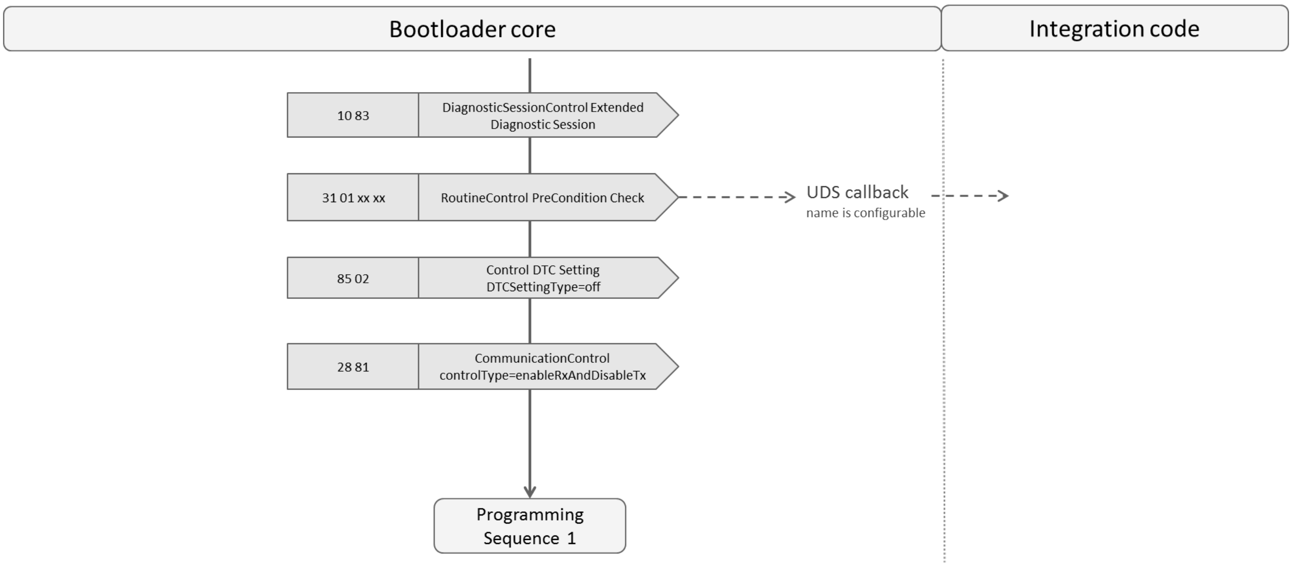

Bootloader core
Integration code
10 83
DiagnosticSessionControl Extended Diagnostic Session
31 01 xx xx
RoutineControl PreCondition Check
UDS callback
name is configurable
85 02
Control DTC Setting DTCSettingType=off
28 81
CommunicationControl controlType=enableRxAndDisableTx
Programming Sequence 1

Figure 3.4. Pre-Programming sequence

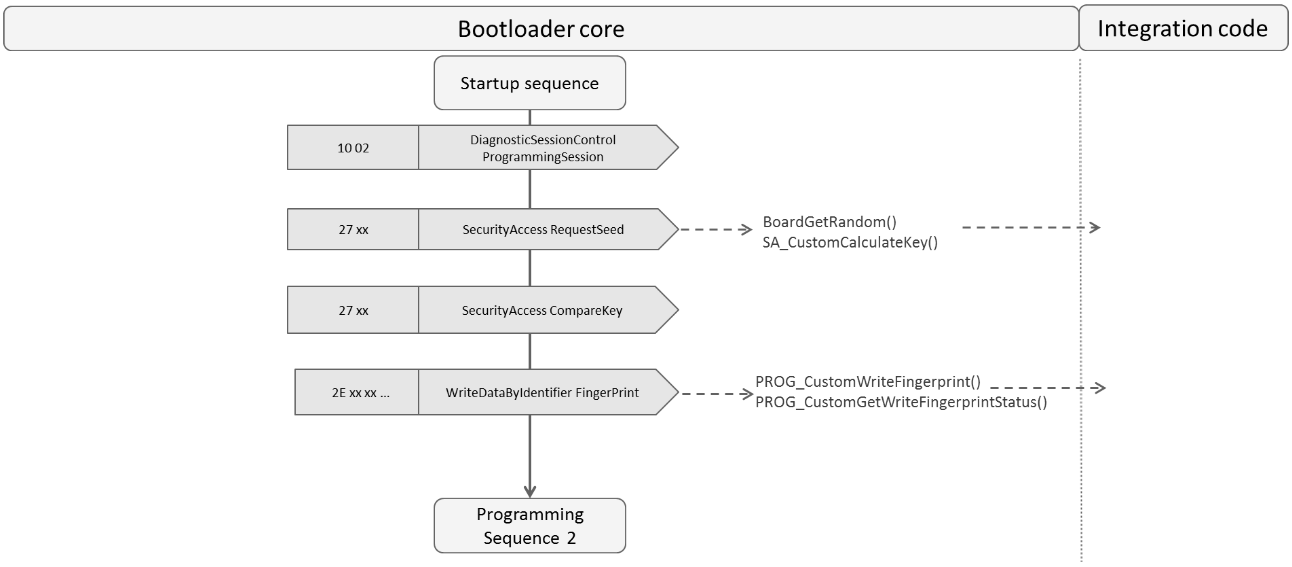

Bootloader core
Integration code
Startup sequence
10 02
DiagnosticSessionControl ProgrammingSession
27 xx
SecurityAccess RequestSeed
BoardGetRandom()
SA_CustomCalculateKey()
27 xx
SecurityAccess CompareKey
2E xx xx ...
WriteDataByIdentifier FingerPrint
PROG_CustomWriteFingerprint()
PROG_CustomGetWriteFingerprintStatus()
Programming Sequence 2

Figure 3.5. Reprogramming sequence

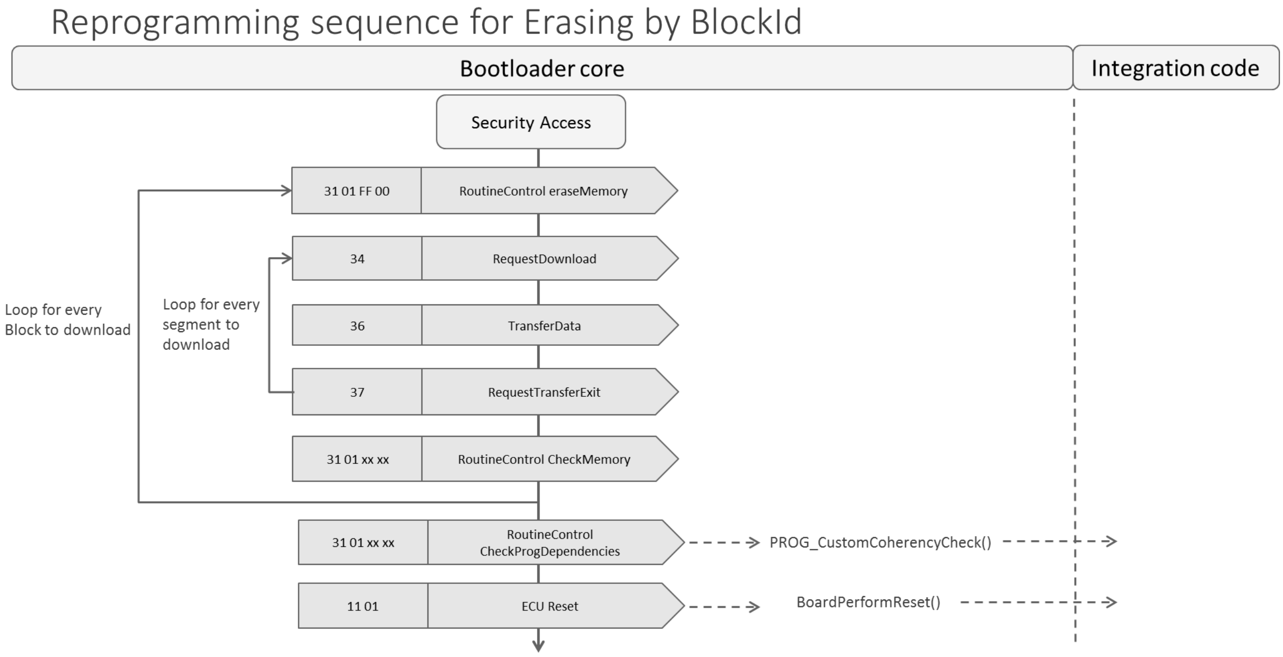

Reprogramming sequence for Erasing by BlockId
Bootloader core
Integration code
Security Access
31 01 FF 00
RoutineControl eraseMemory
34
RequestDownload
36
TransferData
37
RequestTransferExit
31 01 xx xx
RoutineControl CheckMemory
Loop for every Block to download
Loop for every segment to download
31 01 xx xx
RoutineControl CheckProgDependencies
PROG_CustomCoherencyCheck()
11 01
ECU Reset
BoardPerformReset()

Figure 3.6. Reprogramming sequence

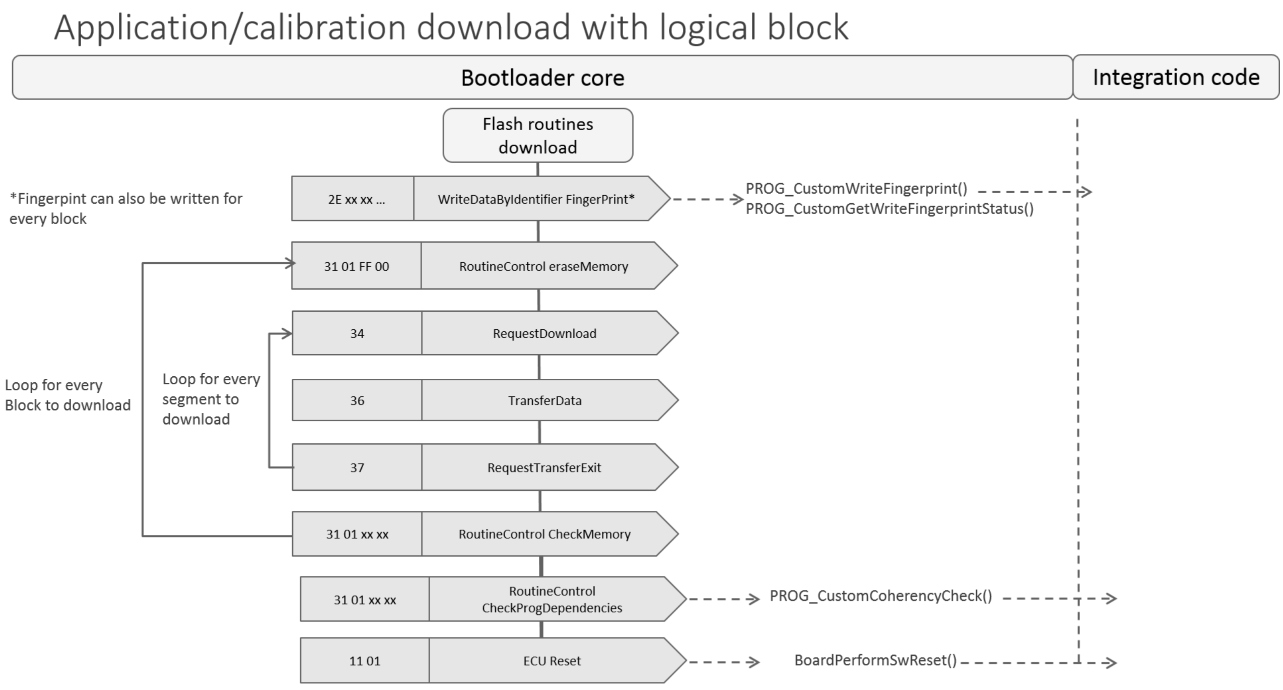

Application/calibration download with logical block
Bootloader core
Integration code
Flash routines download
*Fingerpint can also be written for every block
2E xx xx ...
WriteDataByIdentifier FingerPrint*
PROG_CustomWriteFingerprint()
PROG_CustomGetWriteFingerprintStatus()
31 01 FF 00
RoutineControl eraseMemory
34
RequestDownload
Loop for every Block to download
Loop for every segment to download
36
TransferData
37
RequestTransferExit
31 01 xx xx
RoutineControl CheckMemory
31 01 xx xx
RoutineControl CheckProgDependencies
PROG_CustomCoherencyCheck()
11 01
ECU Reset
BoardPerformSwReset()

Figure 3.7. Reprogramming sequence

## 3.2.7.2. Bootloader diagnostic Callbacks

Bootloader diagnostic callbacks are generated by the UDS module, based on customer configuration, for all not required service of the bootloader (e.g. specific DID (RDBI services) for the project).

Mandatory services are described in the OEM UDS configuration.

These callbacks shall be configured and fully managed by the integrator of the bootloader.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0001 |
| Version: | 1 |
| Description: | The integrator shall configure and fill all callbacks out of scope of the main bootloader purpose. |

### 3.2.7.2.1. UDS callbacks

All UDS callbacks shall have the following prototype: tUdsStatus *CallbackName*(u16 *puwLen, u8 *aubUdsData), with

tUdsStatus: status of the diagnostic service processing (UDS_ACK: positive response can be sent / UDS_NRC_*ErrorCode*) puwLen: Request length(IN) / Response length(OUT)

aubUdsData: Request data(IN) / Response data(OUT)

**Important Note**: For service RDBI *puwLen is not the received length but the current buffer size consumed. This length is useful for customer in case of MULTI DID to check if response exceeds TP buffer size (NRC shall be returned in this case)

One implementation example of RDBI callback:

```
tUdsStatus UDS_RDBI_ECUSerialNumberDataIdentifier_f18c(u16 *puwLen, u8 *aubUdsData)
{
    tUdsStatus ubUdsStatus;
    u16 ubExpectedCallSize;
    u16 ubIdx;

    /* Here calculate the expected return size */
    /* including SID (1) DID (2) + data       */
    ubExpectedCallSize = 0x38;

    if ((*puwLen + ubExpectedCallSize) > TP_DIAG_MSG_DATA_MAX)
    {
        ubUdsStatus = UDS_NRC_XX;
    }
    else
    {
        ubUdsStatus = UDS_ACK;

        /* Here fill the Data buffer from the third byte */
        for(ubIdx = 3U; ubIdx < ubExpectedCallSize; ubIdx++)
        {
            aubUdsData[ubIdx]  = 0xEBU;
        }

        /* Return the expected size to the UDS layer */
        *puwLen = ubExpectedCallSize;
    }

    return ubUdsStatus;
}
```

### 3.2.7.2.2. Indication callbacks

Additional indications are provided by the bootloader at some important point of its processing (e.g. Segmented message reception). It is the integrator choice to fulfil them or not depending on the project need.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0030 |
| Version: | 1 |
| Description: | The integrator shall ensure that the following callbacks are implemented:<br>▶ UDS_CustomPositiveAnswerInd(): This callback is called to give possibility to the user to execute an action before the positive answer transmission.<br>▶ APP_GetUdsDataBufferInd(): This callback is called when UDS reponse is available but not yet transmitted. The buffer can be updated if necessary.<br>▶ APP_UdsSessionStatusInd(): Notification for diagnostic session transition.<br>▶ APP_TpRxInd(): This callback is called when a message reception is completed, successfully or not.<br>▶ APP_TpTxConf(): This callback is called when a message transmission is completed, successfully or not.<br>▶ UDS_P2AboutToExpireInd(): Notification just before the P2/P2_STAR timeout |

### 3.2.7.3. Hardware Related Callbacks

These callbacks are used by the bootloader to get information regarding hardware register or perform some hardware control.

They are managed in the board.c file as an example but can moved and reworked by the integrator.

For more details please refer to the Bootloader_Generic_documentation.pdf file.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOT-HWCBK-0010 |
| Version: | 1 |
| Description: | The integrator shall ensure that the following callbacks are compiled and implemented:<br>▶ BoardSetSleepState()<br>▶ BoardPerformSwReset()<br>▶ BoardIsSwReset()<br>▶ BoardEnableInterrupts()<br>▶ BoardDisableInterrupts()<br>▶ BoardGetRandom() |

BoardDisableInterrupts and BoardEnableInterrupts APIs can be called by any layer of the bootloader to disable and enable the interruption.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOT-HWCBK-0011 |
| Version: | 1 |
| Description: | The integrator shall ensure that the API BoardDisableInterrupts is implemented and allow disabling the interruption. |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOT-HWCBK-0012 |
| Version: | 1 |
| Description: | The integrator shall ensure that the API BoardEnableInterrupts is implemented and allow enabling the interruption. Note: For Bootloader software that does not use the interruption this API shall stays empty. |

The BoardIsSwRese API is called by the PROG layer to know if a software reset has been done and if it shall be evaluate if a programming request has been done.

It allows Bootloader identifying if the Bootloader start is caused by a power-on or by a software reset (e.g by application when receiving a programming event)

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOT-HWCBK-0016 |
| Version: | 2 |
| Description: | The integrator shall ensure that the API BoardIsSwRese is implemented and provide the cause of the previous reset. |

The BoardSetSleepState API is called by the PROG layer to set the microcontroler in a sleep state

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOT-HWCBK-0013 |
| Version: | 2 |
| Description: | The integrator shall ensure that the API BoardSetSleepState is implemented and allow setting the ECU in sleep state. |

The BoardPerformSwReset API is called by the PROG layer to perform the reset the microcontroler.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOT-HWCBK-0014 |
| Version: | 2 |

| | |
|---|---|
| Description: | The integrator shall ensure that the API BoardPerformSwReset is implemented and allows performing a software reset. |

The BoardGetRandom API is called by the SA layer to get a pseudo random value for seed use.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOT-HWCBK-0015 |
| Version: | 1 |
| Description: | The integrator shall ensure that the API BoardGetRandom is implemented and return a 32bit pseudo random value. |

### 3.2.7.4. BM Callbacks

The BootManager has its own callback. They will be the first function call at the ECU startup and allow integrator to initialize its Hardware with BM_HardwareInit and its software with BM_SoftwareInit

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0020 |
| Version: | 1 |
| Description: | The integrator shall ensure that the API BM_HardwareInit is implemented and if necessary contains some specific hardware initialization. |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0021 |
| Version: | 1 |
| Description: | The integrator shall ensure that the API BM_SoftwareInit is implemented and if necessary contains some specific software initialization. |

### 3.2.7.5. BM_CustomDualBankInit Callback

This callback is called at Bootloader start

It is used to configure and initialize the dual memory bank.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0202 |
| Version: | 1 |
| Description: | The integrator shall implement in BM_CustomDualBankInit callback the configuration of the hardware for the use of dual memory banks. Also, this callback shall initialize all the needed data for the dual memory banks usage. |

### 3.2.7.6. SA Callbacks

#### 3.2.7.6.1. SA_CustomCalculateKey Callback

This callback is called on reception of SecurityAccess service to perfom computation of the security key based on a random seed.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTSACBK-0001 |
| Version: | 1 |
| Description: | In SA_CustomCalculateKey callback, the integrator shall implement the key computation based on the provided random value. Computation shall be done using the algorithm required for the project. |

### 3.2.7.7. SA Antiscanning Callbacks

#### 3.2.7.7.1. SA_CustomStoreAsRetryCnt Callback

This callback is called on reception of SecurityAccess service in case anti-scanning feature is activated.

It allows integrator storing in non-volatile memory the retry counter value used for anti-scanning feature.

Bootloader uses callback SA_CustomRestoreAsRetryCnt at startup to get the stored value.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTSACBK-0002 |
| Version: | 1 |
| Description: | In SA_CustomStoreAsRetryCnt callback, the integrator shall implement the storage in non-volatile memory of the provided counter value. |

#### 3.2.7.7.2. SA_CustomRestoreAsRetryCnt Callback

This callback is called at Bootloader startup (if anti-scanning feature is activated) to get the retry counter value from non-volatile memory.

Bootloader uses callback SA_CustomStoreAsRetryCnt to store the counter value.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTSACBK-0003 |
| Version: | 1 |
| Description: | In SA_CustomRestoreAsRetryCnt callback, the integrator shall implement the get from non-volatile memory of the counter value. It shall be ensure that if value has never been written, the return value is 0. |

### 3.2.7.8. Programming related callbacks

Callbacks from the following sections shall be implemented during integration task

#### 3.2.7.8.1. PROG_CheckProgRequest Callback

This callback is called at Bootloader startup to know if a programing request has been received in Application.

If a programing request has been received in Application, Bootloader won't check application validity and stay in Bootloader mode and wait programming event from the network.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0100 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CheckProgRequest callback software allowing getting information from application if a programming request has been received (e.g: read a flag from noinit RAM shared between Bootloader and Application). PROG_BOOT_REPROG value shall be returned if a programming request has been received. PROG_BOOT_NO_REPROG shall be returned if no programming request has been received |

#### 3.2.7.8.2. PROG_JumpToApplication Callback

This callback is called at Bootloader startup if application is valid/coherent and shall be executed (no programming request received from application).

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0101 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_JumpToApplication callback software allowing jumping to application start address. |

#### 3.2.7.8.3. PROG_isValidApplication Callback

This callback is called when application validity check is required

This can e.g happen at Bootloader startup to identify if Bootloader mode or application mode shall be processed.

Callback definition can be found in module reference chapter

| Id: | OSC-INTMAN-BOOTCBK-0104 |
|---|---|
| Version: | 1 |
| Description: | The integrator shall implement in PROG_isValidApplication callback software peforming a check of the full software (application and calibration) to identify if application software is in a state where it can be started (valid and coherent). This that e.g (integration dependent) been done by checking that validity flag of every block/segment that are part the application software and check that different block/segment are all coherent with e.g version check. Return TRUE if application is valid, return FALSE if application is not valid. |

#### 3.2.7.8.4. PROG_InvalidateSection_BlockID Callback

This callback is called on Erase routine reception

Bootloader notify that an erasing will be performed on a logical block and allow integrator performing actions before an erasing (e.g invalidate application status)

Integration software is responsible to manage the application validity status. This callback can be used by integrator to manage the validity status (shall invalidate the block status with this callback).

If the feature Preliminary Erasing is activated, when the Erase request is received with the max block identifier, it shall perform the correspondent action (e.g. invalidate application status) in all segments.

Integrator has also the possibility to reject the erasing by returning a PROG_E_NOT_OK value (PROG_E_OK shall be returned if accepted)

Callback definition can be found in module reference chapter

| Id: | OSC-INTMAN-BOOTCBK-0107 |
|---|---|
| Version: | 1 |
| Description: | The integrator shall implement in PROG_InvalidateSection_BlockID callback software performing operation that can be required by integration software before erasing and invalidating the logical block that will be erased. PROG_E_OK shall be returned if erasing is allowed, PROG_E_NOT_OK in other case. |

### 3.2.7.8.5. PROG_SwitchApplicationModeInd Callback

This callback is called before Bootloader perform a jump to application

Bootloader notify that a jump to application will be performed and allow integrator performing actions before the jump

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0108 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_SwitchApplicationModeInd callback software performing operation that can be required before jumping to application. |

### 3.2.7.8.6. PROG_GetSuppressBitFromAppli Callback

This callback is called when Bootloader shall send a response to a request that has been received in application

When ECU is running in application mode and receive a programming request, depending on configuration, the response can be sent by the Bootloader after reseting the ECU. In this cas the Bootloader needs to know the UDS suppressPositiveResponse bit value from the request to identify if a response shall be sent or not.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0113 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_GetSuppressBitFromAppli callback software getting from appplication information if the suppressPositiveResponse bit was set in the received request (e.g: read a flag from noinit RAM shared between Bootloader and Application) |

### 3.2.7.8.7. Fingerprint Callbacks

#### 3.2.7.8.7.1. PROG_CustomWriteFingerprint Callback

This callback is called on reception of WriteDataByIdentifier service for Fingerprint DID

It allows integrator peforming the Fingerprint data validity check and peform its storing in non-volatile memory.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0116 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomWriteFingerprint callback software checking the validity of FingerPrint data and performing the writting in non-volatile memory of the Fingerpint data (pubRamBuffer points on the dataIdentifier field of the WriteDataByIdentifier, allowing integrator identifying the fingerprint using the DID identifier value) request Asynchronous management can be implemented, in this PROG_E_BUSY value is returned and further call to PROG_CustomGetWriteFingerprintStatus will allow Bootloader to get writting status. |

#### 3.2.7.8.7.2. PROG_CustomGetWriteFingerprintStatus Callback

This callback is called after PROG_CustomWriteFingerprint returns PROG_E_BUSY, this callback is called periodically until getting a status different from PROG_E_BUSY

It allows integrator implementing writting of the fingerprint.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0117 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomGetWriteFingerprintStatus callback software providing status of the fingerprint writting |

### 3.2.7.8.8. Get Crypto Keys Callbacks

#### 3.2.7.8.8.1. PROG_CustomGetAsymPublicKey Callback

This callback is called in order to get the cryptographic asymetrical public key from user defined location.

PROG_CustomGetAsymPublicKey has the following parameters:

- The pointer to the public key's modulus array
- The pointer to the public key's exponent

It allows integrator to fetch the asymetrical public key and give it to the bootloader.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0142 |

| | |
|---|---|
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomGetAsymPublicKey callback the fetching of the asymetrical public key. |

#### 3.2.7.8.8.2. PROG_CustomGetSymStaticKey Callback

This callback is called in order to get the cryptographic symetrical static key from user defined location.

It allows integrator to fetch the symetrical static key and give it to the bootloader.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0143 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomGetSymStaticKey callback the fetching of the symetrical static key. |

#### 3.2.7.8.9. PROG_CustomDecryptData Callback

In order to use this callback the Tresos parameter "Enable_Decryption" shall be set to "TRUE".

This callback is called on reception of a Transfer Data (and before decompression if activated)

It allows the integrator to implement the decryption algorithm.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0161 |
| Version: | 1 |
| Description: | The integrator shall implement PROG_CustomDecryptData callback copying decrypted data at the same location than the encrypted one. |

#### 3.2.7.8.10. MemoryAccessNotification Callback

#### 3.2.7.8.10.1. PROG_CustomMemoryAccessNotification Callback

This callback is called after a successful memory data access.

It allows the customers to place their routines.

However the callback is called only on synchronous memory access.

PROG_CustomMemoryAccessNotification has the following parameters:

- Memory type (e.g. RAM, Flash or Flash Ext).
- Operation type (e.g. Read, Write or Erase).
- Start address of the memory.
- Data length.
- Data buffer.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0162 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomMemoryAccessNotification callback the procedure desired like an subsystem update. |

#### 3.2.7.8.11. Custom Memory Access Callbacks

##### 3.2.7.8.11.1. PROG_CustomMemoryErase Callback

This callback is called upon receipt of an erasing request.

It allows the customer to implement his/her own erase routine.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0170 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomMemoryErase callback the desired erase routine. The callback should return one of the following macros PROG_E_OK, PROG_E_NOT_OK or PROG_E_BUSY, according to its memory access status. It is recommended to implement it asynchronously using the callback PROG_CustomMemGetJobStatus if it is a slow operation. |

##### 3.2.7.8.11.2. PROG_CustomMemoryWrite Callback

This callback is called upon receipt of an writing request.

It allows the customer to implement his/her own write routine.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0171 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomMemoryWrite callback the desired write routine. The callback should return one of the following macros PROG_E_OK, PROG_E_NOT_OK or PROG_E_BUSY, according to its memory access status. It is recommended to implement it asynchronously using the callback PROG_CustomMemGetJobStatus, if it is a slow operation. |

#### 3.2.7.8.11.3. PROG_CustomMemoryRead Callback

This callback is called upon receipt of an reading request.

It allows the customer to implement his/her own reading routine.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0172 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomMemoryRead callback the desired read routine. The callback should return one of the following macros PROG_E_OK, PROG_E_NOT_OK or PROG_E_BUSY, according to its memory access status. It is recommended to implement it asynchronously using the callback PROG_CustomMemGetJobStatus, if it is a slow operation. |

#### 3.2.7.8.11.4. PROG_CustomMemGetJobStatus Callback

This callback is called after an asynchronous memory access operation.

It allows the customer to get the status of the memory job.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0173 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomMemGetJobStatus callback the routine to get the memory job status. After a custom memory access this call- |

| | |
|---|---|
| | back shall be called periodically until getting a status different from PROG_E_-BUSY. |

### 3.2.7.8.11.5. PROG_CustomGetNextSectorAddr Callback

This callback is called after an synchronous memory access operation.

It allows the customer to get the address of the next memory sector.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0174 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomGetNextSectorAddr callback the routine to get the next sector memory address. |

### 3.2.7.8.12. Dual Memory Bank Callbacks

### 3.2.7.8.12.1. PROG_CustomCalcInactiveBankWriteAddr Callback

This callback is called whenever an erase or write to the inactive memory bank is needed.

It allows the integrator to perform the calculation of the address in the inactive bank where the erase/write will be done.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0203 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomCalcInactiveBankWriteAddr callback the calculation of the erase or write address in the inactive bank based on the offset between banks and the current active memory bank. |

### 3.2.7.8.12.2. PROG_CustomCalcInactiveBankReadAddr Callback

This callback is called whenever a read from the inactive memory bank is needed.

It allows the integrator to perform the calculation of the address in the inactive bank from where the read be done.

Callback definition can be found in module reference chapter

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0204 |
| Version: | 1 |
| Description: | The integrator shall implement in PROG_CustomCalcInactiveBankReadAddr callback the calculation of the read address in the inactive bank based on the offset between banks and the current active memory bank. |

### 3.2.7.9. BlPduR related callbacks

Callbacks from the following sections shall be implemented during integration task

#### 3.2.7.9.1. BlPduR_GetRxPduId Callback

This callback is called each time the ECU reset and need to retrieve connection context in order to respond to a request received before this reset (received in application or bootloader mode).

It allows Bootloader to retrieve the Rx Pdu Id that shall be used to send a response to a diagnostic request received before the reset.

This value is completly project dependent and it is up to the integrator to define its own way to find the correct Rx Pdu ID

The value shall be get either from a previous call (before reset) to callback BlPduR_StoreRxPduId or shall be get from application.

Callback definition can be found in module reference chapter.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0145 |
| Version: | 1 |
| Description: | The integrator shall implement in callback BlPduR_GetRxPduId software getting the Rx Pdu Id to be used to respond to the request received before the reset. |

#### 3.2.7.9.2. BlPduR_StoreRxPduId Callback

This callback is called each time the Bootloader will reset and need to store connection context in order to retrieve it later.

It allows Bootloader or application to identify the Tx Pdu Id to be used to send the response after a reset.

Callback definition can be found in module reference chapter.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0146 |
| Version: | 1 |
| Description: | The integrator shall implement in callback BlPduR_StoreRxPduId the storage of the Rx Pdu Id to be used by ECU in order to send a response after a reset. |

#### 3.2.7.9.3. BlPduR_Custom_Com_Init Callback

This callback is called inside BlPduR_Init1 function, it initializes all the modules of Communication stack

It allows Bootloader to intialize all the modules of the communication stack

Initialization of communication stack is project dependent and it is up to the integrator to define which Communication protocol stack to be initialized based on OEM and their requirement (e.g. CAN or FR or Eth or Lin)

This function will return void

Callback definition can be found in module reference chapter.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0168 |
| Version: | 2 |
| Description: | The Integrator will add the initialization of all the modules in the communication stack and it is up to the integrator to define which Communication protocol stack to be initialized based on OEM and their requirement (e.g. CAN or FR or Eth or Lin). The integrator shall ensure that the PDUID of the BlPdur connection is initialized at the startup if it is not a software reset. |

#### 3.2.7.9.4. BlPduR_Custom_Com_Deactivate Callback

This callback is called when Bootloader wants to disable the communication or switch the state machine to NO communication mode

It allows Bootloader to Deactive or switch to communication mode

Deactivation/Disabling of communication is project dependent and it is up to the integrator to define which Communication protocol stack to be Disabled based on OEM and their requirement (e.g. CAN or FR or Eth or Lin)

This function will return void

Callback definition can be found in module reference chapter.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0169 |

| | |
|---|---|
| Version: | 1 |
| Description: | The Integrator will add disabling the communication or switch the state machine to NO communication mode code and it is up to the integrator to define which Communication protocol stack to be disabled based on OEM and their requirement (e.g. CAN or FR or Eth or Lin) |

## 3.2.7.10. Communication related callbacks

Callbacks from the following sections shall be implemented during integration task

### 3.2.7.10.1. BlPduR_GetGroupIdVal Callback

This callback is called at ECU initialization if the feature multiple identifier is configured to External Notification

It allows Bootloader retrieving the Group Id that shall be used by the ECU. Bootloader will only accept identifiers belonging to this group, other won't be responded.

This value is completly project dependent and it is up to the integrator to define its own way to find the correct group ID (e.g. from an I/O value, from a NVM data store,...)

In case the value returned is out of range the default configured group ID will be selected

Callback definition can be found in module reference chapter.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTCBK-0138 |
| Version: | 1 |
| Description: | If the feature multiple identifier is configured to External Notification the callback BlPduR_GetGroupIdVal shall be filled to return the group ID to be used |

## 3.2.7.11. State machine guard callbacks

Callbacks from PROG_Guard.c shall be implemented during integration task as defined in section "Programming sequence adaptation" from Bootloader_OEMInd_specification document.

## 3.2.7.12. Response callbacks

Callbacks from PROG_Responses.c shall be implemented during integration task as defined in section "Response management" from Bootloader_OEMInd_specification document.

## 3.2.8. Downloadable Flash driver Feature

Gives the possibility to download flash routines to the RAM.

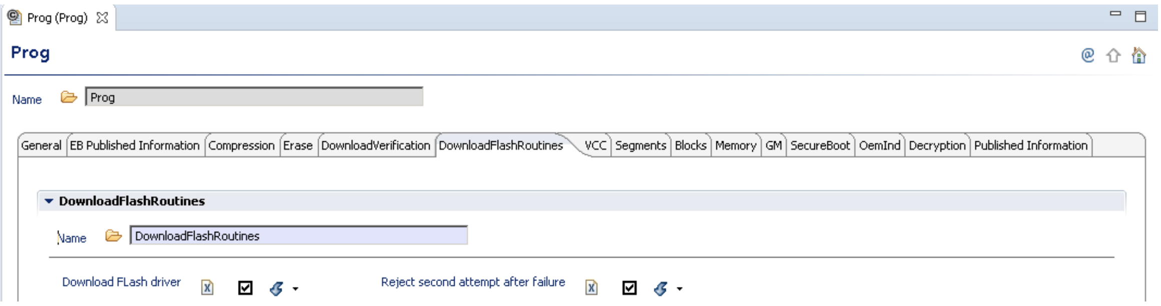
The Flash routines are the functions responsible to write/erase the flash, so the functions are required to be executed in RAM to modify the FLASH.

### 3.2.8.1. Activation of the feature

The following steps describe the feature activation.

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0170 |
| Version: | 1 |
| Description: | If downloading of the flash routines is required for the project the following shall be performed:<br>▶ Activate the feature in PROG module by setting "Download FLash driver" field . |

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0171 |
| Version: | 1 |
| Description: | If the rejection of any new attempt of the flash routines after a failed attempt is required for the project the following shall be performed:<br>▶ Activate the feature in PROG module by setting "Download FLash driver" field . |



Name: Prog

Name: DownloadFlashRoutines

Download FLash driver ☑

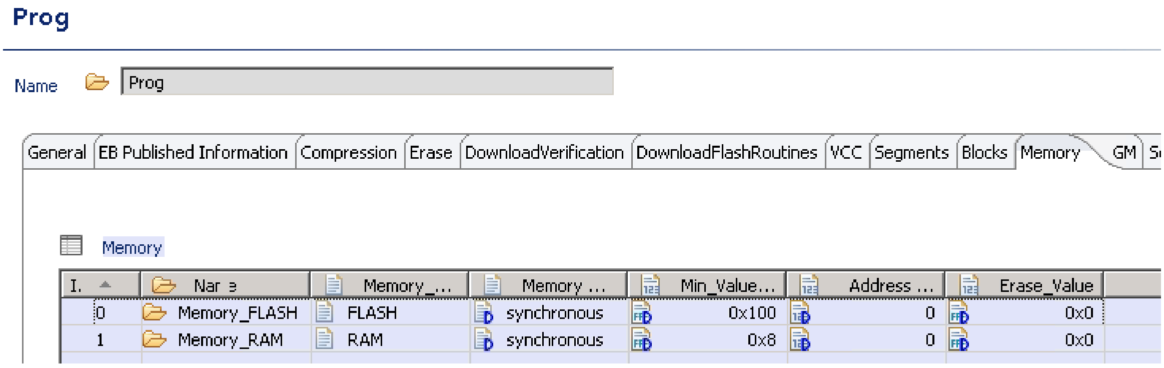
Reject second attempt after failure ☑

Figure 3.8. Activation of the feature Downloadable Flash driver

### 3.2.8.2. Configuration of the feature

The following steps describe the feature configuration.

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0172 |
| Version: | 1 |
| Description: | If not already existing,add a new Ram memory in the "Memory" tab. |



Prog

Name: Prog

Memory

| I. | Nar e | Memory_... | Memory ... | Min_Value... | Address ... | Erase_Value |
|---|---|---|---|---|---|---|
| 0 | Memory_FLASH | FLASH | synchronous | 0x100 | 0 | 0x0 |
| 1 | Memory_RAM | RAM | synchronous | 0x8 | 0 | 0x0 |

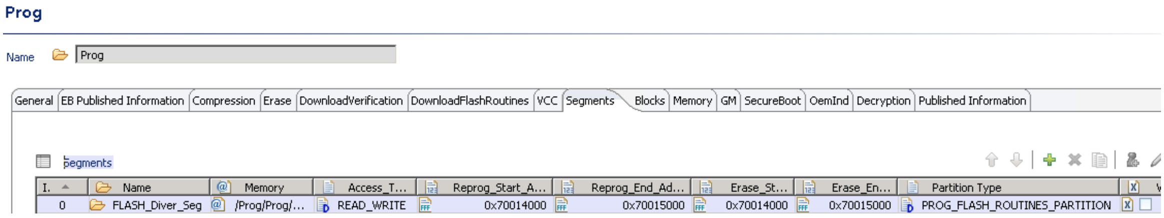
Figure 3.9. Adding new RAM memory

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0173 |
| Version: | 1 |
| Description: | Add a new (and unique) segment for the flash driver routines, and put the addresses where these routines are mapped. |

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0174 |
| Version: | 1 |
| Description: | Set the partition type of the segment to "PROG_FLASH_ROUTINES_-PARTITION". |

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0175 |
| Version: | 1 |
| Description: | Map the segment to the RAM memory create previously. |

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0176 |
| Version: | 1 |
| Description: | Deactivate the validity check for this segment, or the generation will fail. |



Prog

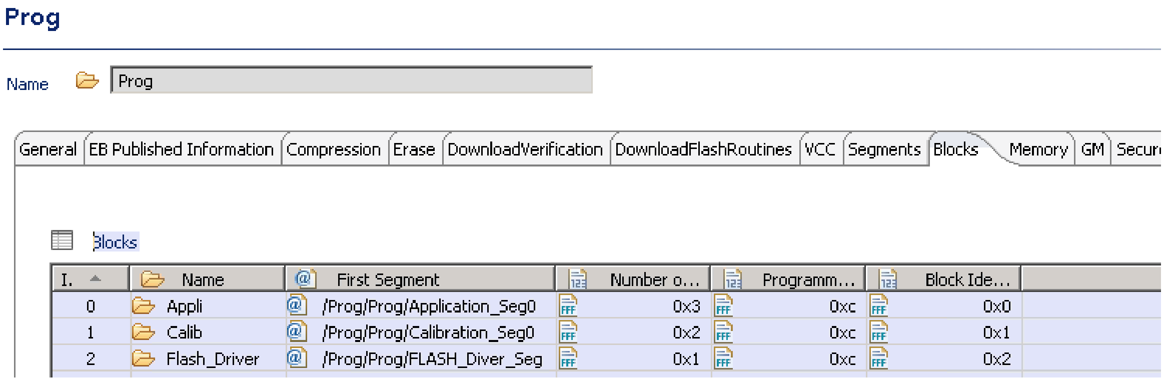
Name: Prog

Segments

| I. | Name | Memory | Access_T... | Reprog_Start_A... | Reprog_End_Ad... | Erase_St... | Erase_En... | Partition Type | ☒ | V |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | FLASH_Diver_Seg | /Prog/Prog/... | READ_WRITE | 0x70014000 | 0x70015000 | 0x70014000 | 0x70015000 | PROG_FLASH_ROUTINES_PARTITION | ☒ | □ |

Figure 3.10. Adding new segment for flash routines

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0177 |
| Version: | 1 |
| Description: | Add a new block for the flash driver routines and set the "First segment" field to the created segment in the previous step. |



Prog

Name: Prog

Blocks

| I. | Name | First Segment | Number o... | Programm... | Block Ide... |
|---|---|---|---|---|---|
| 0 | Appli | /Prog/Prog/Application_Seg0 | 0x3 | 0xc | 0x0 |
| 1 | Calib | /Prog/Prog/Calibration_Seg0 | 0x2 | 0xc | 0x1 |
| 2 | Flash_Driver | /Prog/Prog/FLASH_Diver_Seg | 0x1 | 0xc | 0x2 |

Figure 3.11. Adding new block for flash routines

### 3.2.8.3. Integration of the feature

#### 3.2.8.3.1. Linker file update

The following steps shall be done in the linker file in order to correctly map the Flash routines to the RAM:

| | |
|---|---|
| Id: | OSC-INTMAN-FLSDOWN-0178 |
| Version: | 1 |
| Description: | Add a linker file section in RAM to map all the code from MemMap section "FLASH_FLS_START_SEC_CODE". |

#### 3.2.8.3.2. Available APIs

The following APIs are available for the integrator :

- PROG_DrvDown_IsFlashRoutinesPresent: Returns the value of m_ubFlashRoutinesPresent, that represents the presence of the flash routines in RAM.

  PROG_TRUE Flash routines are present in RAM.

PROG_FALSE Flash routines are not present in RAM.

### 3.2.9. Dual Memory Bank Feature

The dual memory bank feature allows the download of a new application in a different memory bank (inactive) while the bootloader and/or the application is running on the current memory bank (active).

The activation of the feature is done by checking the "Dual Memory Bank Used" checkbox in the "General" tab under Tresos Studio.

When the feature is activated, the EB tresos Bootloader will always download the new application/calibration in the inactive memory bank.

A swap request needs to be done by the tester. The implementation of the banks swap shall be done in integration code.

The activation of the freshly programmed memory bank is done after the ECU reset.

After the banks swap, the software execution is done from the active bank. This means that the EB tresos Bootloader needs to be flashed in both banks to assure the startup and the application validity checks.

Following integration requirements apply when using Dual Memory Bank feature.

The following callbacks shall be implemented by integrator when feature is used:

- BM_CustomDualBankInit: Shall implement the activation of the dual bank capability of the hardware and the initialization of the needed data for the memory banks manipulation.
- PROG_CustomCalcInactiveBankWriteAddr: Shall implement the calculation of the addresses to be erased or written in the inactive memory bank.
- PROG_CustomCalcInactiveBankReadAddr: Shall implement the calculation of the addresses to be read from the inactive memory bank.

| | |
|---|---|
| Id: | OSC-INTMAN-DUALBANK-0001 |
| Version: | 1 |
| Description: | If dual memory bank feature is required, the integrator shall program the EB tresos Bootloader in both active and inactive memory banks. |

| | |
|---|---|
| Id: | OSC-INTMAN-DUALBANK-0002 |
| Version: | 1 |
| Description: | If dual memory bank feature is required, the integrator shall implement and manage the swap request (UDS service configuration and handler). |

| | |
|---|---|
| Id: | OSC-INTMAN-DUALBANK-0003 |
| Version: | 1 |
| Description: | If dual memory bank feature is required, the integrator shall assure that the memory banks configurations are identical in order to have similar peformances of the software on both banks. |

## 3.2.10. Mandatory API to be called by the system

These API have to be called by the system scheduler at some point to allow the correct behaviour of the bootloader

The API BM_Startup is the entry point of the BootManager and shall be the first API to be called by the scheduler

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0040 |
| Version: | 1 |
| Description: | The integrator shall ensure that BM_Startup is the very first API called once startup is done. |

The API BM_Manage shall be cyclically called with the period configured in EB Tresos Studio, if the BM_-TIMEOUT_CHECK in BM configuration is set to ON

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0041 |
| Version: | 1 |
| Description: | The integrator shall ensure that BM_Manage is called cyclically with the correct period configured in EB tresos Studio, i.e. BM_CYCLE_CALL = 1 if Bootloader CAN and BM_CYCLE_CALL = 5 if Bootloader FlexRay (recommended values). |

The APIs EB_Manage/BlPduR_Manage is the main scheduler of the bootloader software. It calls every other layer Manage fonction. It shall be cyclically called to the period configured in EB tresos Studio.

Depending of your project, EB tresos Bootloader is provided with EB or BlPduR plugin, integration code shall only call the Api of the provided plugin.

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0042 |
| Version: | 1 |
| Description: | The integrator shall ensure that EB_Manage/BlPduR_Manage is called cyclically with the correct period configured in EB tresos Studio, i.e. MANAGE_PERIOD = 1 if Bootloader CAN and MANAGE_PERIOD = 5 if Bootloader FlexRay (recommended values). |

| | |
|---|---|
| Provides coverage to: (Id-Version \| Variants) | SwAD-ARCH-0110-1 |

The API PROG_BckdManage is the scheduler of the bootloader state machine. It shall be called as fast as possible to reduce all bootloader treatment time (e.g. in background while loop)

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0043 |
| Version: | 1 |
| Description: | The integrator shall ensure that PROG_BckdManage is called as fast as possible to reduce all bootloader treatment time. |

## 3.2.11. General Bootloader Performance

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0081 |
| Version: | 1 |
| Description: | If Autosar Flexray stack is used, the integrator shall ensure that Flexray main-functions (i.e. FrIf_MainFunction, FrTp_MainFunction and FrSM_MainFunction) are called every 5ms. |
| Provides coverage to: (Id-Version \| Variants) | SwAD-ARCH-0109-1 |

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0082 |
| Version: | 1 |
| Description: | If Autosar cryptographic libraries are used, the integrator shall ensure that Csm_MainFunction is called continously in background task. |
| Provides coverage to: (Id-Version \| Variants) | SwAD-ARCH-0109-1 |

Note: Integrator shall ensure that FrTp_MainFunction can not be interrupted by FrIf_JobListExec processing that is called under interruption.

## 3.2.12. Exception handling

| | |
|---|---|
| Id: | OSC-INTMAN-BOOTLOADER-0090 |
| Version: | 1 |

| | |
|---|---|
| Description: | The integrator shall ensure the exception handling and defines/implements the action(s) to perform in case an exception happens. |

## 3.2.13. Glossary

You can find in the following table a definition of the different naming used with the Bootloader documentation

| **Term** | **Definition** |
|---|---|
| Application | Runnable code of ECU software that can be updated by a Bootloader |
| AUTOSAR | Automotive Open System Architecture<br>A consortium of OEMs, Tier1's and semiconductor vendors that work on standardization of an automotive software architecture. |
| Bootloader | Permanent software located in Flash memory allowing updating application or calibration software of an ECU. It provides communication with an external Tester<br>**Also called**: Flashloader. |
| Calibration | Configuration code of an ECU software that can be updated by a Bootloader |
| CRY | Cryptographic primitives implementation (used by the CSM)<br>This is an AUTOSAR module |
| CSM | Cryptograpic Service Manager<br>This is an AUTOSAR module |
| Download verification | Operation realized after a download to verify that the data present in memory match the expected downloaded data. This is usually done using a CRC, Checksum, Signature or Hash.<br>**Also called**: Check programming dependencies, check memory, message digest |
| Flash sector | Smallest amount of flash memory that can be erased in one pass. Size depends on flash technology used. |
| Flash page | Smallest amount of flash memory that can be programmed in one pass. Size depends on flash technology used. |
| Segment | Continuous address range within a logical block.<br>Static segment: static address and size present in configuration defining the memory section that can be erased or programmed. |

| Term | Definition |
|---|---|
| | Dynamic segment: dynamic address and size present in RAM defining the memory section that have been programmed. |
| HSM | Hardware Security Module |
| Logical block | Smallest amount of flash memory that can be individually reprogrammed. Size depends on technology (flash sector, flash page) and user settings. |
| Memory Erased check | Verification if memory is already erased, before trying to erase it. It allow sparing time, if the memory is already erased.<br>**Also called**: blank check. |
| PDU | Protocol Data Unit<br>Any piece of information exchanged between two or more communicating entities |
| PduR | PDU Router<br>This is an AUTOSAR module |
| SecOC | Secure OnBoard Communication<br>This is an AUTOSAR module |
| SHE | Secure Hardware Extension<br>A hardware security extension specified by the HIS consortium. |
| Software integrity | Feature ensuring the software integrity before executing it.<br>**Also called**: Software authentication. |
| Software acceptance check | Feature verifying the validity of the received application/calibration software to be updated<br>**Also called**: Application/Calibration signature check. |
| Software coherency check | Feature verifying the coherency of the application/calibrations<br>**Also called**: Application coherency check, consistency check, application validity, Check Programming dependencies. |
| Streaming | Allow writting data in memory on reception of every consecutive frame of a TransferData request (No wait of the reception of the full request before starting the write). This improve the download performance. |

Table 3.2. Bootloader definitions

## 3.3. Annex: list of requirements

Find below the complete list of userguide requirements. User shall check each requirement for integration.

| Id | Description | CheckList |
|---|---|---|
| OSC-INTMAN-BOOT-LOADER-0071 | The integrator shall ensure that Max_Bytes_in_TD field is filled with a value multiple of Min_Value_To_Flash field value + 2. | |
| OSC-INTMAN-BOOT-LOADER-0079 | The integrator shall ensure that if the Demo_CSM wrapper shall be used for integrating crypto stack ASR version 4.3: The Use_CSM_-ASR430_DemoWrapper checkbox is ticked/enabled Crypto stack ASR version 4.3, demo_csmwrapper.c and demo_csmwrapper.h files are integrated in the Bootloader Demo_SA_RNDcallback_403 and Crypto_MainFunction functions are called from the scheduler as background tasks | |
| OSC-INTMAN-BOOT-LOADER-0123 | The integrator shall ensure that if Dual Memory Bank feature shall be used: The Dual Memory Bank Used checkbox is ticked/enabled | |
| OSC-INTMAN-BOOTCBK-0114 | The integrator shall implement in PROG_CustomIsFirstProgramming callback software providing information if Flash was never programmed before and that erase shall be skipped. The callback should return PROG_TRUE if the Flash was never programmed and PROG_FALSE otherwise. | |
| OSC-INTMAN-BOOTCBK-0159 | The integrator shall implement in PROG_CustomDownloadNotification callback software by updating the information that Flash was already programmed before. This indicates that, for any other further | |

| Id | Description | CheckList |
|---|---|---|
| | programming, the Flash memory should be erased. | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0020 | The integrator shall ensure that Transmit_Response_Before_Reset field is set. | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0022 | The integrator shall ensure that Transmit_Nrc78_On_SecurityAccess field is set if the Security Access seed/key generation process is long (more than P2 time). | |
| OSC-INTMAN-BOOT-LOADER-0070 | The integrator shall configure all memory used for the project. Only 1 memory of each type (FLASH, FLASH_EXT, RAM) can be configured. | |
| OSC-INTMAN-BOOT-LOADER-0077 | For every memory, the integrator shall configure the memory type: Internal Flash (FLASH), external Flash (FLASH_EXT) or RAM memory (RAM). | |
| OSC-INTMAN-BOOT-LOADER-0073 | For every memory, the integrator shall configure the Memory Mode parameter depending if the Flash driver support synchronous or asynchronous interface call. Synchronous means that when the Flash Api is called, it will returns only when the request operation is performed. Asynchronous means that when the Flash Api is called, it returns before performing the requested operation, Prog module will later call periodically a GetJobStatus Api to be informed when the operation is finished. In case synchronous mode is used Prog module will request the erasing of Flash sector per sector to | |

| Id | Description | CheckList |
|---|---|---|
| | avoid a too long block time is Flash call. | |
| OSC-INTMAN-BOOT-LOADER-0074 | For every memory, the integrator shall ensure that Minimum value to write field is set to the minimum size that shall be write for the memory (Flash page size). | |
| OSC-INTMAN-BOOT-LOADER-0075 | For every memory, the integrator shall configure the address offset to be used when accessing the memory. It's used to convert the logical address get from the diagnostic request to the physical address of the memory. | |
| OSC-INTMAN-BOOT-LOADER-0076 | For every memory, the integrator shall configure the erase state value of the memory (0x00 or 0xFF depending of the memory architecture). | |
| OSC-INTMAN-BOOT-LOADER-0072 | The integrator shall ensure that the configuration of the memory areas manipulated by the flashloader is consistent against protected areas (hardware protection key, any other sections that shall not be changed) | |
| OSC-INTMAN-BOOT-LOADER-0078 | The integrator shall ensure to map every segment to the correct memory. | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0120 | The integrator shall ensure that at least one Aplication segment is configured as follow: Memory_Type set to FLASH Access_Type set to READ_WRITE Reprog_Start_Address is the start of Application area Reprog_End_Address is the end of Application area Erase_Start_Address is equal to Reprog_Start_Address | |

| Id | Description | CheckList |
|---|---|---|
| | Erase_End_Address is equal to Reprog_End_Address Partition Type is equal to PROG_APPLICATION_PARTITION NOTE: Additional application or calibration segment can be configured. | |
| OSC-INTMAN-OEMIND-BOOTLOADER-0100 | The integrator shall configure Erasing Mode parameter according to the expected erasing request. The following values are possible: All: No information are provided in Erase request, on receception of the erase request, all the memory segments will be erased. Address: The Erase request contains the address range to be erased, only 1 segment can be erased with this request. Note that in this case the checksum will be computed on a single segment. LogicalBlock: The Erase request contains the logical block to be erased. | |
| OSC-INTMAN-OEMIND-BOOTLOADER-0101 | The integrator shall configure the Erase request ALFI Enable parameter to indicate if the Erase request contains the UDS ALFI field. | |
| OSC-INTMAN-OEMIND-BOOTLOADER-0102 | The integrator shall configure the Application validity parameter to indicate if Elektrobit algorithm or customer specific algorithm shall be used. If customer specific, the integrator shall implement the PROG_-InvalidateSection and PROG_CustomSetCrcCompareSuccess callbacks to manage the application validity status. | |
| OSC-INTMAN-OEMIND-BOOTLOADER-0103 | The integrator shall configure the CRC algorithm parameter to indicate which checksum shall be | |

| Id | Description | CheckList |
|---|---|---|
| | used to verify the software download. | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0104 | The integrator shall configure the FingerPrint Enable parameter to indicate if a fingerprint shall be managed for the software download. | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0105 | If FingerPrint is enable, the integrator shall configure the Size_Of_-FingerPrint parameter to indicate the expected FingerPrint size. | |
| OSC-INTMAN-OEMIND-VERIF-BOOTLOADER-0001 | The integrator shall configure (parameter Maximum RequestDownload Per Block) the maximum number of RequestDownload request that can be received, by the Bootloader, for a single logical block. This value will be used to size the RAM structure storing the downloaded memory area that shall be used to perform the signature/CRC verification. | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0010 | The integrator shall ensure that BM_TIMEOUT_CHECK field is NOT set. | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0011 | The integrator shall ensure that BM_SOURCE_ADDRESS_CHECK field is NOT set. | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0070 | The integrator shall ensure that the parameter Security_Access_Seed_Length is set to 4 | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0071 | The integrator shall ensure that the parameter Security_Access_Key_-Length is set to 4 | |
| OSC-INTMAN-OEMIND-BOOT-LOADER-0072 | If the Static Seed (response with precedent seed in the case of successive GetSeed requests without a respective key received) is required for the project, the integra- | |

| Id | Description | CheckList |
|---|---|---|
| | tor shall ensure that Enable_Static_Seed field is set. | |
| OSC-INTMAN-BOOT-LOADER-0104 | The integrator shall ensure that Standard field is set to ISO | |
| OSC-INTMAN-BOOT-LOADER-0105 | The integrator shall ensure that UDS_MANAGE_PERIOD field is set to the call period of the UDS_-Manage Api | |
| OSC-INTMAN-BOOT-LOADER-0100 | The integrator shall ensure that SecurityCheck field is set and that the SecurityFunction is set to PROG_GetSecurityLevel | |
| OSC-INTMAN-BOOT-LOADER-0106 | The integrator shall ensure that RC_NRC_IMPLEMENTATION field is set to 0x31 | |
| OSC-INTMAN-BOOT-LOADER-0107 | The integrator shall ensure that DID_NRC_IMPLEMENTATION field is set to 0x31 | |
| OSC-INTMAN-BOOT-LOADER-0101 | The integrator shall ensure that RESPONSE_PENDING field is set | |
| OSC-INTMAN-BOOT-LOADER-0102 | The integrator shall ensure that SPREC_IN_RESPONSE field is set | |
| OSC-INTMAN-BOOT-LOADER-0103 | The integrator shall ensure that RELOAD_TSTOPDIAG field is set | |
| OSC-INTMAN-BOOT-LOADER-0108 | If for integration reason the NRC78 response pending shall be send before the end of the configured P2/P2star time, the integrator can configure the P2/P2star adjust parameter. In this case the UDS module will trigg the NRC78 transmission at a time equal to (P2/P2star - P2/P2star adjust) | |
| OSC-INTMAN-BOOT-LOADER-0120 | The integrator shall configure every of following session: DEFAULT, PROGRAMMING, EXTENDED, SUPPLIER, OTHER_01, | |

| Id | Description | CheckList |
|---|---|---|
| | OTHER_02, OTHER_03, OTHER_04 | |
| OSC-INTMAN--BLPDUR-BOOTLOADER-0001 | The integrator shall configure the TxPDU (TxPdu Reference) to be used for response transmission. It shall reference a valid ECUC PDU. | |
| OSC-INTMAN--BLPDUR-BOOTLOADER-0002 | The integrator shall configure the TxPDU confirmation identifier (TxPdu Identifier) to be used by PduR module to confirm the transmission. The TxPdu Identifiers shall be unique and consecutives. | |
| OSC-INTMAN--BLPDUR-BOOTLOADER-0003 | The integrator shall configure the Tester Address associated to this connection. | |
| OSC-INTMAN--BLPDUR-BOOTLOADER-0004 | The integrator shall indicate if this connection is associated to a LIN communication (Lin Connection) . | |
| OSC-INTMAN--BLPDUR-BOOTLOADER-0005 | The integrator shall indicate if this connection shall re-use a functional RxPduId already defined in another connection (Share Functional Id). | |
| OSC-INTMAN--BLPDUR-BOOTLOADER-0006 | If Share Functional Id is enabled, the integrator shall configured the shared Pdu Reference. | |
| OSC-INTMAN--BLPDUR-BOOTLOADER-0007 | The integrator shall configure all the RxPDU to be used for request reception. It shall reference a valid ECUC PDU, specify if the Pdu is functionalor physical and the associated Pdu Identifier to be used by PduR module. The RxPdu Identifiers shall be unique and consecutives. | |
| OSC-INTMAN--BLPDUR-BOOTLOADER-0008 | A connection shall contain only one functional RxPdu or one reference functional RxPdu. | |

| Id | Description | CheckList |
|---|---|---|
| OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0001 | The integrator shall configure in CRY module a CryRsaSsaPssVerify configuration with the following configuration: CryRsaSsaPssVerifyUseTimeSlices (Use time slicing for RSASSA-PSS signature verification) set to true CryRsaSsaPssVerifyNumberOfTimeSlices (Number of RsaSsaPss time slices) set to 10 CryRsaSsaPssVerifyUseCbk (Use configured callback function which returns maximum number of time slices) set to false CryRsaSsaPssVerifyImmediateRestartEnabled (Enable the cancelation of an ongoing calculation regardless of the configuration ID) set to true CryRsaSsaPssVerifyHashCfgRef (Hash configuration) set to CsmHashConfig_0 CryRsaSsaPssVerifyKeyLength (Key Length) set to 256 CryRsaSsaPssVerifySaltLength (Salt Length) set to 0 CryRsaSsaPssVerifyB64Encoded (Base64 Encoded) set to false CryRsaSsaPssVerifyUseBarrett (Barrett reduction) set to false CryRsaSsaPssVerifySupportRestart (Enable the cancelation of ongoing requests) set to true | |
| OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0002 | The integrator shall configure in CRY module the following general parameters: CrySHAOneAndTwoImplementation (Implementation variant) set to CRY_SHAONEANDTWO_INTERRUPTABLE CryInterruptableLN (Interruptable LN operations) set to true | |

| Id | Description | CheckList |
|---|---|---|
| OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0003 | The integrator shall configure in CRY module a CrySHA2 configuration with the following configuration: CrySHA2ImmediateRestartEnabled ( Enable the cancelation of an ongoing calculation regardless of the configuration ID) set to true CrySHA2Type (Prime) set to CRY_SHA_256 CrySHA2IterationsPerMain (Number of iterations per MainFunction) set to 1 CrySHA2SupportRestart (Enable the cancelation of ongoing requests) set to true | |
| OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0004 | The integrator shall configure in CSM module a CsmSignatureVerify configuration with the following configuration: CsmCallbackSignatureVerify set to PROG_CsmNotification CsmSignatureVerifyMaxKeySize set to 524 CsmSignatureVerifyInitConfiguration set to CryRsaSsaPssVerifyConfig_0 (configuration done in OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0001) CsmSignatureVerifyEnableRteInterface (Enable Rte Interface) set to false CsmSignatureVerifyEnableRestart (Enable the cancelation of ongoing requests) set to true CsmSignatureVerifyUsePriorities (Csm priorities handling) set to true | |
| OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0005 | The integrator shall configure a CsmHash configuration (CsmHashConfig_0) with the following configuration: CsmCallbackHash set to Cry_RsaSsaPssVerifyCallback CsmHashInitConfig- | |

| Id | Description | CheckList |
|---|---|---|
| | uration set to CrySHA2Config_0 (configuration done in OSC-INTMAN-OEMIND-CRY-BOOTLOADER-0001) CsmHashPrimitiveName set to SHA2 CsmHashEnableRteInterface (Enable Rte Interface) set to false CsmHashEnableRestart (Enable the cancelation of ongoing requests) set to true CsmHashUsePriorities (Csm priorities handling) set to true | |
| OSC-INTMAN-CBK-0001 | The integrator shall ensure that the callback execution doesn't block the processing of Bootloader MainFunction/Manage cyclic task (e.g BlPduR_Manage) | |
| OSC-INTMAN-BOOTLOADER-0001 | The integrator shall configure and fill all callbacks out of scope of the main bootloader purpose. | |
| OSC-INTMAN-BOOTLOADER-0030 | The integrator shall ensure that the following callbacks are implemented: UDS_CustomPositiveAnswerInd(): This callback is called to give possibility to the user to execute an action before the positive answer transmission. APP_GetUdsDataBufferInd(): This callback is called when UDS reponse is available but not yet transmitted. The buffer can be updated if necessary. APP_UdsSessionStatusInd(): Notification for diagnostic session transition. APP_TpRxInd(): This callback is called when a message reception is completed, successfully or not. APP_TpTxConf(): This callback is called when a message transmission is completed, successfully or not. | |

| Id | Description | CheckList |
|---|---|---|
| | UDS_P2AboutToExpireInd(): Notification just before the P2/P2_STAR timeout | |
| OSC-INTMAN-BOOT-HW-CBK-0010 | The integrator shall ensure that the following callbacks are compiled and implemented: BoardSetSleepState() BoardPerformSwReset() BoardIsSwReset() BoardEnableInterrupts() BoardDisableInterrupts() BoardGetRandom() | |
| OSC-INTMAN-BOOT-HW-CBK-0011 | The integrator shall ensure that the API BoardDisableInterrupts is implemented and allow disabling the interruption. | |
| OSC-INTMAN-BOOT-HW-CBK-0012 | The integrator shall ensure that the API BoardEnableInterrupts is implemented and allow enabling the interruption. Note: For Bootloader software that does not use the interruption this API shall stays empty. | |
| OSC-INTMAN-BOOT-HW-CBK-0016 | The integrator shall ensure that the API BoardIsSwRese is implemented and provide the cause of the previous reset. | |
| OSC-INTMAN-BOOT-HW-CBK-0013 | The integrator shall ensure that the API BoardSetSleepState is implemented and allow setting the ECU in sleep state. | |
| OSC-INTMAN-BOOT-HW-CBK-0014 | The integrator shall ensure that the API BoardPerformSwReset is implemented and allows performing a software reset. | |
| OSC-INTMAN-BOOT-HW-CBK-0015 | The integrator shall ensure that the API BoardGetRandom is implemented and return a 32bit pseudo random value. | |

| Id | Description | CheckList |
|---|---|---|
| OSC-INTMAN-BOOT-LOADER-0020 | The integrator shall ensure that the API BM_HardwareInit is implemented and if necessary contains some specific hardware initialization. | |
| OSC-INTMAN-BOOT-LOADER-0021 | The integrator shall ensure that the API BM_SoftwareInit is implemented and if necessary contains some specific software initialization. | |
| OSC-INTMAN-BOOTCBK-0202 | The integrator shall implement in BM_CustomDualBankInit callback the configuration of the hardware for the use of dual memory banks. Also, this callback shall initialize all the needed data for the dual memory banks usage. | |
| OSC-INTMAN-BOOTSACBK-0001 | In SA_CustomCalculateKey callback, the integrator shall implement the key computation based on the provided random value. Computation shall be done using the algorithm required for the project. | |
| OSC-INTMAN-BOOTSACBK-0002 | In SA_CustomStoreAsRetryCnt callback, the integrator shall implement the storage in non-volatile memory of the provided counter value. | |
| OSC-INTMAN-BOOTSACBK-0003 | In SA_CustomRestoreAsRetryCnt callback, the integrator shall implement the get from non-volatile memory of the counter value. It shall be ensure that if value has never been written, the return value is 0. | |
| OSC-INTMAN-BOOTCBK-0100 | The integrator shall implement in PROG_CheckProgRequest callback software allowing getting information from application if a pro- | |

| Id | Description | CheckList |
|---|---|---|
| | gramming request has been received (e.g: read a flag from noinit RAM shared between Bootloader and Application). PROG_BOOT_REPROG value shall be returned if a programming request has been received. PROG_BOOT_NO_REPROG shall be returned if no programming request has been received | |
| OSC-INTMAN-BOOTCBK-0101 | The integrator shall implement in PROG_JumpToApplication callback software allowing jumping to application start address. | |
| OSC-INTMAN-BOOTCBK-0104 | The integrator shall implement in PROG_isValidApplication callback software peforming a check of the full software (application and calibration) to identify if application software is in a state where it can be started (valid and coherent). This that e.g (integration dependent) been done by checking that validity flag of every block/segment that are part the application software and check that different block/segment are all coherent with e.g version check. Return TRUE if application is valid, return FALSE if application is not valid. | |
| OSC-INTMAN-BOOTCBK-0107 | The integrator shall implement in PROG_InvalidateSection_BlockID callback software performing operation that can be required by integration software before erasing and invalidating the logical block that will be erased. PROG_E_OK shall be returned if erasing is allowed, PROG_E_NOT_OK in other case. | |

| Id | Description | CheckList |
|---|---|---|
| OSC-INTMAN-BOOTCBK-0108 | The integrator shall implement in PROG_SwitchApplicationModeInd callback software performing operation that can be required before jumping to application. | |
| OSC-INTMAN-BOOTCBK-0113 | The integrator shall implement in PROG_GetSuppressBitFromAppli callback software getting from appplication information if the suppressPositiveResponse bit was set in the received request (e.g: read a flag from noinit RAM shared between Bootloader and Application) | |
| OSC-INTMAN-BOOTCBK-0116 | The integrator shall implement in PROG_CustomWriteFingerprint callback software checking the validity of FingerPrint data and performing the writting in non-volatile memory of the Fingerpint data (pubRamBuffer points on the dataIdentifier field of the WriteDataByIdentifier, allowing integrator identifying the fingerprint using the DID identifier value) request Asynchronous management can be implemented, in this PROG_E_BUSY value is returned and further call to PROG_CustomGetWriteFingerprintStatus will allow Bootloader to get writting status. | |
| OSC-INTMAN-BOOTCBK-0117 | The integrator shall implement in PROG_CustomGetWriteFingerprintStatus callback software providing status of the fingerprint writting | |
| OSC-INTMAN-BOOTCBK-0142 | The integrator shall implement in PROG_CustomGetAsymPublicKey callback the fetching of the asymetrical public key. | |

| Id | Description | CheckList |
|---|---|---|
| OSC-INTMAN-BOOTCBK-0143 | The integrator shall implement in PROG_CustomGetSymStaticKey callback the fetching of the symetrical static key. | |
| OSC-INTMAN-BOOTCBK-0161 | The integrator shall implement PROG_CustomDecryptData callback copying decrypted data at the same location than the encrypted one. | |
| OSC-INTMAN-BOOTCBK-0162 | The integrator shall implement in PROG_CustomMemoryAccessNotification callback the procedure desired like an subsystem update. | |
| OSC-INTMAN-BOOTCBK-0170 | The integrator shall implement in PROG_CustomMemoryErase callback the desired erase routine. The callback should return one of the following macros PROG_E_OK, PROG_E_NOT_OK or PROG_E_BUSY, according to its memory access status. It is recommended to implement it asynchronously using the callback PROG_CustomMemGetJobStatus if it is a slow operation. | |
| OSC-INTMAN-BOOTCBK-0171 | The integrator shall implement in PROG_CustomMemoryWrite callback the desired write routine. The callback should return one of the following macros PROG_E_OK, PROG_E_NOT_OK or PROG_E_BUSY, according to its memory access status. It is recommended to implement it asynchronously using the callback PROG_CustomMemGetJobStatus, if it is a slow operation. | |
| OSC-INTMAN-BOOTCBK-0172 | The integrator shall implement in PROG_CustomMemoryRead call- | |

| Id | Description | CheckList |
|---|---|---|
| | back the desired read routine. The callback should return one of the following macros PROG_E_OK, PROG_E_NOT_OK or PROG_E_BUSY, according to its memory access status. It is recommended to implement it asynchronously using the callback PROG_CustomMemGetJobStatus, if it is a slow operation. | |
| OSC-INTMAN-BOOTCBK-0173 | The integrator shall implement in PROG_CustomMemGetJobStatus callback the routine to get the memory job status. After a custom memory access this callback shall be called periodically until getting a status different from PROG_E_BUSY. | |
| OSC-INTMAN-BOOTCBK-0174 | The integrator shall implement in PROG_CustomGetNextSectorAddr callback the routine to get the next sector memory address. | |
| OSC-INTMAN-BOOTCBK-0203 | The integrator shall implement in PROG_CustomCalcInactiveBankWriteAddr callback the calculation of the erase or write address in the inactive bank based on the offset between banks and the current active memory bank. | |
| OSC-INTMAN-BOOTCBK-0204 | The integrator shall implement in PROG_CustomCalcInactiveBankReadAddr callback the calculation of the read address in the inactive bank based on the offset between banks and the current active memory bank. | |
| OSC-INTMAN-BOOTCBK-0145 | The integrator shall implement in callback BlPduR_GetRxPduId software getting the Rx Pdu Id to be | |

| Id | Description | CheckList |
|---|---|---|
|  | used to respond to the request received before the reset. |  |
| OSC-INTMAN-BOOTCBK-0146 | The integrator shall implement in callback BlPduR_StoreRxPduId the storage of the Rx Pdu Id to be used by ECU in order to send a response after a reset. |  |
| OSC-INTMAN-BOOTCBK-0168 | The Integrator will add the initialization of all the modules in the communication stack and it is up to the integrator to define which Communication protocol stack to be initialized based on OEM and their requirement (e.g. CAN or FR or Eth or Lin). The integrator shall ensure that the PDUID of the BlPdur connection is initialized at the start-up if it is not a software reset. |  |
| OSC-INTMAN-BOOTCBK-0169 | The Integrator will add disabling the communication or switch the state machine to NO communication mode code and it is up to the integrator to define which Communication protocol stack to be disabled based on OEM and their requirement (e.g. CAN or FR or Eth or Lin) |  |
| OSC-INTMAN-BOOTCBK-0138 | If the feature multiple identifier is configured to External Notification the callback BlPduR_GetGroupIdVal shall be filled to return the group ID to be used |  |
| OSC-INTMAN-FLSDOWN-0170 | If downloading of the flash routines is required for the project the following shall be performed: Activate the feature in PROG module by setting Download FLash driver field . Note: Please note that smaller the decompression slice is, big- |  |

| Id | Description | CheckList |
|---|---|---|
| | ger is the time needed for the overall decompression process. | |
| OSC-INTMAN-FLSDOWN-0171 | If the rejection of any new attempt of the flash routines after a failed attempt is required for the project the following shall be performed: Activate the feature in PROG module by setting Download FLash driver field . Note: Please note that smaller the decompression slice is, bigger is the time needed for the overall decompression process. | |
| OSC-INTMAN-FLSDOWN-0172 | If not already existing,add a new Ram memory in the Memory tab. | |
| OSC-INTMAN-FLSDOWN-0173 | Add a new (and unique) segment for the flash driver routines, and put the addresses where these routines are mapped. | |
| OSC-INTMAN-FLSDOWN-0174 | Set the partition type of the segment to PROG_FLASH_-ROUTINES_PARTITION. | |
| OSC-INTMAN-FLSDOWN-0175 | Map the segment to the RAM memory create previously. | |
| OSC-INTMAN-FLSDOWN-0176 | Deactivate the validity check for this segment, or the generation will fail. | |
| OSC-INTMAN-FLSDOWN-0177 | Add a new block for the flash driver routines and set the First segment field to the created segment in the previous step. | |
| OSC-INTMAN-FLSDOWN-0178 | Add a linker file section in RAM to map all the code from MemMap section FLASH_FLS_START_-SEC_CODE. | |
| OSC-INTMAN-DUALBANK-0001 | If dual memory bank feature is required, the integrator shall program the EB tresos Bootloader in both active and inactive memory banks. | |

| Id | Description | CheckList |
|---|---|---|
| OSC-INTMAN-DUALBANK-0002 | If dual memory bank feature is required, the integrator shall implement and manage the swap request (UDS service configuration and handler). | |
| OSC-INTMAN-DUALBANK-0003 | If dual memory bank feature is required, the integrator shall assure that the memory banks configurations are identical in order to have similar peformances of the software on both banks. | |
| OSC-INTMAN-BOOT-LOADER-0040 | The integrator shall ensure that BM_Startup is the very first API called once startup is done. | |
| OSC-INTMAN-BOOT-LOADER-0041 | The integrator shall ensure that BM_Manage is called cyclically with the correct period configured in EB tresos Studio, i.e. BM_CYCLE_CALL = 1 if Bootloader CAN and BM_CYCLE_CALL = 5 if Bootloader FlexRay (recommended values). | |
| OSC-INTMAN-BOOT-LOADER-0042 | The integrator shall ensure that EB_Manage/BlPduR_Manage is called cyclically with the correct period configured in EB tresos Studio, i.e. MANAGE_PERIOD = 1 if Bootloader CAN and MANAGE_PERIOD = 5 if Bootloader FlexRay (recommended values). | |
| OSC-INTMAN-BOOT-LOADER-0043 | The integrator shall ensure that PROG_BckdManage is called as fast as possible to reduce all bootloader treatment time. | |
| OSC-INTMAN-BOOT-LOADER-0081 | If Autosar Flexray stack is used, the integrator shall ensure that Flexray mainfunctions (i.e. FrIf_-MainFunction, FrTp_MainFunc- | |

| Id | Description | CheckList |
|---|---|---|
| | tion and FrSM_MainFunction) are called every 5ms. | |
| OSC-INTMAN-BOOT-LOADER-0082 | If Autosar cryptographic libraries are used, the integrator shall ensure that Csm_MainFunction is called continously in background task. | |
| OSC-INTMAN-BOOT-LOADER-0090 | The integrator shall ensure the exception handling and defines/implements the action(s) to perform in case an exception happens. | |

Table 3.3. List of requirements

# 4. BL for Essentials module references

## 4.1. Overview

This chapter provides module references for the BL for Essentials product modules. These include a detailed description of all configuration parameters. Furthermore this chapter lists the application programming interface with all data types, constants and functions.

The content of the sections is sorted alphabetically according the EB tresos AutoCore Generic module names.

For further information on the functional behavior of these modules, refer to the chapter BL for Essentials user's guide.

## 4.2. APP

### 4.2.1. Configuration parameters

| Containers included | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| General | 1..1 | This container describes the general properties of the node. |
| Bootloader_Options | 1..1 | This container contains the options for bootloader mode. |
| CommonPublishedInformation | 1..1 | **Label:** Common Published Information<br>Common container, aggregated by all modules. It contains published information about vendor and versions. |
| PublishedInformation | 1..1 | **Label:** EB Published Information<br>Additional published parameters not covered by CommonPublishedInformation container. |

#### 4.2.1.1. General

### 4.2.1.2. Bootloader_Options

### 4.2.1.3. CommonPublishedInformation

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ArMajorVersion | 1..1 |
| ArMinorVersion | 1..1 |
| ArPatchVersion | 1..1 |
| SwMajorVersion | 1..1 |
| SwMinorVersion | 1..1 |
| SwPatchVersion | 1..1 |
| ModuleId | 1..1 |
| VendorId | 1..1 |
| Release | 1..1 |

| Parameter Name | ArMajorVersion |
|---|---|
| **Label** | AUTOSAR Major Version |
| **Description** | Major version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| Parameter Name | ArMinorVersion |
|---|---|
| **Label** | AUTOSAR Minor Version |
| **Description** | Minor version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |

| | |
|---|---|
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ArPatchVersion** |
|---|---|
| **Label** | AUTOSAR Patch Version |
| **Description** | Patch level version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMajorVersion** |
|---|---|
| **Label** | Software Major Version |
| **Description** | Major version number of the vendor specific implementation of the module. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 3 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMinorVersion** |
|---|---|
| **Label** | Software Minor Version |
| **Description** | Minor version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 6 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwPatchVersion** |
|---|---|
| **Label** | Software Patch Version |

| | |
|---|---|
| **Description** | Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 8 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ModuleId** |
|---|---|
| **Label** | Numeric Module ID |
| **Description** | Module ID of this module from Module List |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **VendorId** |
|---|---|
| **Label** | Vendor ID |
| **Description** | Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 1 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **Release** |
|---|---|
| **Label** | Release Information |
| **Multiplicity** | 1..1 |
| **Type** | STRING_LABEL |
| **Default value** | |
| **Configuration class** | **PublishedInformation:** |

| | |
|---|---|
| **Origin** | Elektrobit Automotive GmbH |

### 4.2.1.4. PublishedInformation

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| PbcfgMSupport | 1..1 |

| **Parameter Name** | **PbcfgMSupport** |
|---|---|
| **Label** | PbcfgM support |
| **Description** | Specifies whether or not the APP can use the PbcfgM module for post-build support. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

## 4.2.2. Application programming interface (API)

### 4.2.2.1. Functions

#### 4.2.2.1.1. APP_CalculateCrc

| | | |
|---|---|---|
| **Purpose** | Implement CRC calculation. | |
| **Synopsis** | `u16 APP_CalculateCrc ( tFlashAddress uFlashAddress , u32 ulLen );` | |
| **Parameters (in)** | `uFlashAddress` | flash data address |
| | `ulLen` | flash data length |
| **Return Value** | CRC value | |

| | |
|---|---|
| **Description** | This function allows to calculate CRC from an custom algorithm<br>Only available for EB BOOTLOADER |

### 4.2.2.1.2. APP_Init

| | |
|---|---|
| **Purpose** | Initialize layer. |
| **Synopsis** | `void APP_Init ( void );` |
| **Description** | This function initializes all layers of the complete STACK. EB_Init is called in this API. It shall be called only once at ECU startup. |

### 4.2.2.1.3. APP_Manage

| | |
|---|---|
| **Purpose** | Regular tick of the layer. |
| **Synopsis** | `void APP_Manage ( void );` |
| **Description** | Ensure cyclic tasks of the layer. |

### 4.2.2.1.4. APP_ReadBootFlag

| | | |
|---|---|---|
| **Purpose** | Get Boot flag from NVM. | |
| **Synopsis** | `tFlashData APP_ReadBootFlag ( void );` | |
| **Return Value** | Boot flag from NVM | |
| | `DOWNLOAD_REQUESTED` | Download is requested from application |
| | `GO_IN_BOOT` | Request to switch to Bootloader |
| | `GO_IN_APP` | Request to switch to Application |
| | `STAY_IN_BOOT` | Request to stay in Bootloader |
| **Description** | This function allows to get Boot flag. It is called at start up to either:<br>▶ switch to Bootloader<br>▶ switch to Application<br>▶ download new binary<br>Only available for PSA BOOTLOADER | |

#### 4.2.2.1.5. APP_ReprogReqManage

| | |
|---|---|
| **Purpose** | Handle switching from Appli to BOOT. |
| **Synopsis** | `void APP_ReprogReqManage ( void );` |
| **Description** | This function is called in APP_Manage (scheduler). If switching from Appli to Boot is allowed, UDS response requested in Appli is now sent.<br>Only available for EB, RSA_CAN_HS and PSA BOOTLOADER |

#### 4.2.2.1.6. APP_WriteBootFlag

| | | |
|---|---|---|
| **Purpose** | Write Boot flag into NVM. | |
| **Synopsis** | `tFlashStatus APP_WriteBootFlag ( tFlashData aubData );` | |
| **Parameters (in)** | `aubData` | Boot flag value (DOWNLOAD_REQUESTED/GO_IN_BOOT/GO_IN_APP) |
| **Return Value** | Flash status | |
| | `FLASH_NO_ERROR` | no error in flash |
| | `FLASH_ACCESS_ERROR` | protection error in flash (access refused) |
| **Description** | This function allows to set Boot flag:<br>▶ at the end of downloading sequence (switch from Boot to Appli).<br>▶ in application if a downloading is requested.<br>Only available for PSA BOOTLOADER | |

#### 4.2.2.1.7. PROG_WriteBootFlag

| | | |
|---|---|---|
| **Purpose** | Write Boot flag into NVM. | |
| **Synopsis** | `void PROG_WriteBootFlag ( u32 ulBootram );` | |
| **Parameters (in)** | `ulBootram` | boot flag value (DOWNLOAD_REQUESTED/GO_IN_BOOT/GO_IN_APP) |
| **Description** | This function is called by PROG to write a new value of boot flag in Memory. The save in memory shall be done by the customer.<br>Only available for RSA BOOTLOADER | |

## 4.2.3. Integration notes

### 4.2.3.1. Exclusive areas

Exclusive areas information is not available for this module.

### 4.2.3.2. Production errors

Production errors information is not available for this module.

### 4.2.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

Memory mapping information is not available for this module.

### 4.2.3.4. Integration requirements

**WARNING** ⚠

**Integration requirements list is not exhaustive**

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the APP module.

# 4.3. BM

## 4.3.1. Configuration parameters

| Containers included | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| General | 1..1 | This container contains the general boot manager configuration element. |
| Security | 1..1 | |
| CommonPublishedInformation | 1..1 | **Label:** Common Published Information<br>Common container, aggregated by all modules. It contains published information about vendor and versions. |
| PublishedInformation | 1..1 | **Label:** EB Published Information<br>Additional published parameters not covered by CommonPublishedInformation container. |

### 4.3.1.1. General

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| BM_TIMEOUT_CHECK | 1..1 |
| BM_TIMEOUT_VALUE | 1..1 |
| BM_CALL_CYCLE | 1..1 |
| BM_FR_CYCLE_LENGTH | 1..1 |
| BM_SOURCE_ADDRESS_CHECK | 1..1 |

| Parameter Name | BM_TIMEOUT_CHECK |
|---|---|
| **Description** | This entry defines if a delay has to be waited before checking the application Validity.<br>Note:<br>▶ In case of CAN network: it allows to start a new reprog session even if the application is valid, by receiving a DSC02 during this delay.<br>▶ In case of FLEXRAY network: it is forced to true. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |
| **Origin** | EB |

| **Parameter Name** | **BM_TIMEOUT_VALUE** |
|---|---|
| **Description** | Define the delay value to wait before checking application validity<br>Note:<br>▶ In case of CAN network: it corresponds to the max time (in ms) waiting for DSC02 request.<br>▶ In case of FLEXRAY network: it corresponds to the max time (in FR cycle) waiting for FR synchronization or a valid NetworkStatus. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 20 |
| **Origin** | EB |

| **Parameter Name** | **BM_CALL_CYCLE** |
|---|---|
| **Description** | Define the periodicity of the call to BM_Manage.<br>Note:<br>▶ In case of CAN network: at each BM_manage a check is done to veriy if the DSC02 have been received.<br>This value shall be a multiple of BM_TIMEOUT_VALUE.<br>▶ In case of FLEXRAY network: a check is done to test the NetworkStatus (if the ECU is synchronized). |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 1 |
| **Origin** | EB |

| **Parameter Name** | **BM_FR_CYCLE_LENGTH** |
|---|---|
| **Description** | Define the FlexRay cycle length.<br>Note: has to be configured with the same value (in ms) than the FrIfGdCycle used parameter. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 5 |
| **Origin** | EB |

| **Parameter Name** | **BM_SOURCE_ADDRESS_CHECK** |
|---|---|

| | |
|---|---|
| **Description** | Enable or disable the management of diagnostic source filtering on a single address<br>Note:<br>▶ If no programming is requested by application, all tester requests shall be accepted. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

### 4.3.1.2. Security

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| BMCsmReferences | 1..1 | Contains references to Csm configuration. |
| SecureBoot | 1..1 | This container contains all configurations for Authenticated/Secure bootloader features. Configuration can be done only if Authenticated or Secure features are activated. |

### 4.3.1.3. BMCsmReferences

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| BMCsmChecksumConfigId | 1..1 |

| **Parameter Name** | **BMCsmChecksumConfigId** |
|---|---|
| **Label** | BMCsmChecksumConfigId |
| **Description** | Reference a *CsmHash* or a *CsmMacVerify*<br>Dependencies:<br>▶ Reference shall be valid |
| **Multiplicity** | 1..1 |
| **Type** | CHOICE-REFERENCE |

| Configuration class | **VariantPreCompile:** | VariantPreCompile |
|---|---|---|
| **Origin** | Elektrobit Automotive GmbH | |

### 4.3.1.4. SecureBoot

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| SECURE_AUTHENTICATED_BOOT | 1..1 |
| BOOT_VERIFICATION | 1..1 |
| CHECKSUM_LENGTH | 1..1 |
| BootCksStartAddress | 1..1 |
| BootCksRangeLength | 1..1 |

| **Parameter Name** | **SECURE_AUTHENTICATED_BOOT** |
|---|---|
| **Label** | Authenticated / Secure Boot |
| **Description** | Enable or disable the Authenticated or Secure Boot features. When the Authenticated Boot feature is enabled, Application and Bootloader checksum will be verified before any software execution by comparison to the previous one. When Secure Boot feature is enabled, Application and Bootloader checksum will be computed and verified before any software execution. These two features cannot be enabled at the same time.<br>▶ OFF : Neither Authenticated Boot nor Secure Boot feature is enabled.<br>▶ Authenticated : Authenticated Boot feature is enabled.<br>▶ Secure : Secure boot feature is enabled. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | OFF |
| **Range** | OFF<br>Authenticated<br>Secure |
| **Origin** | EB |

| **Parameter Name** | **BOOT_VERIFICATION** |
|---|---|

| | |
|---|---|
| **Label** | Bootloader verified/ Bootloader not verified |
| **Description** | Enable or disable the Bootloader verification at startup when Secure Boot or Authenticated Boot feature are enabled. If a HSM is used the Bootloader verification shall be disabled as ensured by the HSM.<br>▶ ON : Bootloader is verified at startup.<br>▶ OFF : Bootloader is not verified at startup. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | OFF |
| **Range** | OFF<br>ON |
| **Origin** | EB |

| **Parameter Name** | **CHECKSUM_LENGTH** |
|---|---|
| **Label** | Checksum length |
| **Description** | Size of the checksum for Authenticated / Secure Boot feature in Bytes |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 512 |
| **Origin** | EB |

| **Parameter Name** | **BootCksStartAddress** |
|---|---|
| **Label** | Bootloader checksum start address |
| **Description** | This value indicated from wich address the bootloader checksum shall be computed.<br>This value is 4 bytes long |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 00000000 |
| **Range** | <=4294967295 |
| **Origin** | EB |

| **Parameter Name** | **BootCksRangeLength** |
|---|---|
| **Label** | Bootloader checksum range length |

| | |
|---|---|
| **Description** | This value indicated the data length on which the Bootloader checksum shall be computed. It can be the full Bootloader software length or only a part of the bootloader software.<br>This value is 4 bytes long |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 00000000 |
| **Range** | <=4294967295 |
| **Origin** | EB |

### 4.3.1.5. CommonPublishedInformation

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ArMajorVersion | 1..1 |
| ArMinorVersion | 1..1 |
| ArPatchVersion | 1..1 |
| SwMajorVersion | 1..1 |
| SwMinorVersion | 1..1 |
| SwPatchVersion | 1..1 |
| ModuleId | 1..1 |
| VendorId | 1..1 |
| Release | 1..1 |

| **Parameter Name** | **ArMajorVersion** |
|---|---|
| **Label** | AUTOSAR Major Version |
| **Description** | Major version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| Parameter Name | ArMinorVersion |
|---|---|
| Label | AUTOSAR Minor Version |
| Description | Minor version number of AUTOSAR specification on which the appropriate implementation is based on. |
| Multiplicity | 1..1 |
| Type | INTEGER_LABEL |
| Default value | 0 |
| Configuration class | **PublishedInformation:** |
| Origin | Elektrobit Automotive GmbH |

| Parameter Name | ArPatchVersion |
|---|---|
| Label | AUTOSAR Patch Version |
| Description | Patch level version number of AUTOSAR specification on which the appropriate implementation is based on. |
| Multiplicity | 1..1 |
| Type | INTEGER_LABEL |
| Default value | 0 |
| Configuration class | **PublishedInformation:** |
| Origin | Elektrobit Automotive GmbH |

| Parameter Name | SwMajorVersion |
|---|---|
| Label | Software Major Version |
| Description | Major version number of the vendor specific implementation of the module. |
| Multiplicity | 1..1 |
| Type | INTEGER_LABEL |
| Default value | 1 |
| Configuration class | **PublishedInformation:** |
| Origin | Elektrobit Automotive GmbH |

| Parameter Name | SwMinorVersion |
|---|---|
| Label | Software Minor Version |
| Description | Minor version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| Multiplicity | 1..1 |

| | |
|---|---|
| **Type** | INTEGER_LABEL |
| **Default value** | 3 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwPatchVersion** |
|---|---|
| **Label** | Software Patch Version |
| **Description** | Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ModuleId** |
|---|---|
| **Label** | Numeric Module ID |
| **Description** | Module ID of this module from Module List |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **VendorId** |
|---|---|
| **Label** | Vendor ID |
| **Description** | Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 1 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **Release** |
|---|---|

| | |
|---|---|
| **Label** | Release Information |
| **Multiplicity** | 1..1 |
| **Type** | STRING_LABEL |
| **Default value** | |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

### 4.3.1.6. PublishedInformation

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| PbcfgMSupport | 1..1 |

| **Parameter Name** | **PbcfgMSupport** |
|---|---|
| **Label** | PbcfgM support |
| **Description** | Specifies whether or not the BM can use the PbcfgM module for post-build support. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

## 4.3.2. Application programming interface (API)

### 4.3.2.1. Functions

#### 4.3.2.1.1. BM_CsmNotification

| | |
|---|---|
| **Purpose** | CSM notification callback. |
| **Synopsis** | Std_ReturnType **BM_CsmNotification** ( Csm_ReturnType result ); |

| | | |
|---|---|---|
| **Parameters (in)** | `result` | Csm calculation result |
| **Return Value** | status | |
| | `E_OK` | |

### 4.3.2.1.2. BM_CustomBckOperation

| | |
|---|---|
| **Purpose** | Allow customer operation during long operation. |
| **Synopsis** | `void` **`BM_CustomBckOperation`** `( void );` |
| **Description** | This function is called during long operation (e.g. cryptography) allowing customer performing actions that cannot be stopped during a long time (e.g watchdog triggering). |

### 4.3.2.1.3. BM_CustomDualBankInit

| | |
|---|---|
| **Purpose** | Initialize the Dual Memory Bank. |
| **Synopsis** | `void` **`BM_CustomDualBankInit`** `( void );` |
| **Description** | This function is called when Bootloader starts to prepare the usage of the dual memory banks. |

### 4.3.2.1.4. BM_CustomGetComputedApplicationChecksum

| | | |
|---|---|---|
| **Purpose** | Get the last computed application checksum. | |
| **Synopsis** | `void` **`BM_CustomGetComputedApplicationChecksum`**<br>`( u8 * pubChecksum , u16 uwBlockIdentifier );` | |
| **Parameters (in)** | `uwBlockIdentifier` | Block Identifier |
| **Parameters (out)** | `pubChecksum` | address where the computed checksum shall be copied |
| **Description** | This function is called at Bootloader startup to verify the application checksum before allowing the start of application. This checksum shall have been computed by application on configured application data range. | |

### 4.3.2.1.5. BM_CustomGetExpectedApplicationChecksum

| | |
|---|---|
| **Purpose** | Get the expected application checksum. |

| | | |
|---|---|---|
| **Synopsis** | void **BM_CustomGetExpectedApplicationChecksum** ( u8 * pubChecksum , u16 uwBlockIdentifier ); | |
| **Parameters (in)** | uwBlockIdentifier | Block Identifier |
| **Parameters (out)** | pubChecksum | address where the checksum shall be copied |
| **Description** | This function is called at Bootloader startup to verify the application checksum before allowing the start of application. It gets from non-volatile memory the checksum that had been computed during the last application download. | |

#### 4.3.2.1.6. BM_CustomGetMacKey

| | |
|---|---|
| **Purpose** | Get the mac key used in Mac verification (should be the same key used in the generation). |
| **Synopsis** | void **BM_CustomGetMacKey** ( const u8 ** paubKeyData , u32 * pulKeyLength ); |
| **Description** | This function is called at the beginning of the Mac verification to get the key yo be used in the process |

#### 4.3.2.1.7. BM_CustomIsNormalStartup

| | | |
|---|---|---|
| **Purpose** | Request if ECU has started normally or not. | |
| **Synopsis** | tBMBoolean **BM_CustomIsNormalStartup** ( void ); | |
| **Return Value** | | |
| **Description** | This function is called at startup to know if normal or abnormal startup has been done. In case of abnormal startup ECU stay in Bootloader mode during a configured time before jumping to application (if valid). This time window allow the tester to send a programming session request to force the Boot mode | |

#### 4.3.2.1.8. BM_CustomSetInvalidAppli

| | | |
|---|---|---|
| **Purpose** | Notification that Application is invalid. | |
| **Synopsis** | void **BM_CustomSetInvalidAppli** ( u16 uwBlockIdentifier ); | |
| **Parameters (in)** | uwBlockIdentifier | Identifier of invalid Block |
| **Description** | This function is called when Application checksum verification failed and that Application cannot be started. This information shall be store in non-volatile memory. | |

#### 4.3.2.1.9. BM_CustomSetInvalidBoot

| | | |
|---|---|---|
| **Purpose** | Notification that Bootloader is invalid. | |
| **Synopsis** | `void BM_CustomSetInvalidBoot ( void );` | |
| **Description** | This function is called when Bootloader checksum verification failed and that Bootloader cannot be started. By this callback application shall be informed that Bootloader cannot be executed anymore. | |

#### 4.3.2.1.10. BM_DisableECCCheck

| | |
|---|---|
| **Purpose** | Callback that shall disable ECC if needed Callback is called: When Bootloader perform read access on Flash that can be unprogrammed and can cause ECC error. |
| **Synopsis** | `void BM_DisableECCCheck ( void );` |
| **Description** | Callback shall implement: If needed, disabling of ECC check Hardware specific) |

#### 4.3.2.1.11. BM_EnableECCCheck

| | |
|---|---|
| **Purpose** | Callback that shall enable ECC if needed Callback is called: After Bootloader has performed a read access on Flash that can be unprogrammed and can cause ECC error. |
| **Synopsis** | `void BM_EnableECCCheck ( void );` |
| **Description** | Callback shall implement: If needed, enabling of ECC check (Hardware specific) |

#### 4.3.2.1.12. BM_GetTesterAddress

| | | |
|---|---|---|
| **Purpose** | Get the diagnostic tester source address. | |
| **Synopsis** | `void BM_GetTesterAddress ( u8 * ubTesterAddress );` | |
| **Parameters (in,out)** | `ubTesterAddress` | pointer on tester address |
| **Description** | This function is called when programming is requested by application (eBootFromApli = BM_TRUE) and BM_SOURCE_ADDRESS_CHECK is set. This is a callback that get the tester address | |

#### 4.3.2.1.13. BM_HardwareInit

| | |
|---|---|
| **Purpose** | Hardware initialization. |

| | |
|---|---|
| **Synopsis** | `void BM_HardwareInit ( void );` |
| **Description** | This function is called at the very beginning of the Boot manager. It allows the application to do the minimum hardware initialization before the Boot manager start to check the application validity. |

#### 4.3.2.1.14. BM_Manage

| | |
|---|---|
| **Purpose** | Manage BM layer. |
| **Synopsis** | `void BM_Manage ( void );` |
| **Description** | This function will ensure the good behaviour of the startup timeout of 5ms requested before checking the application validity.<br>BM_Manage shall be called at the same period and after EB_Manage. |

#### 4.3.2.1.15. BM_SoftwareInit

| | |
|---|---|
| **Purpose** | Software RAM initialization. |
| **Synopsis** | `void BM_SoftwareInit ( void );` |
| **Description** | This function is called at the very beginning of the Boot manager. It allows the application to do the minimum software initialization before the Boot manager start to check the application validity. |

#### 4.3.2.1.16. BM_StartUp

| | |
|---|---|
| **Purpose** | Bootmanager startup. |
| **Synopsis** | `void BM_StartUp ( void );` |
| **Description** | Because of the presence of the bootloader, at the very beginning of the ECU startup shall be the first one called.<br>This function handles:<br>▶ The hardware initialization<br>▶ The check of the boot flag<br>▶ The validity of application<br>▶ The initialization of all EB layer if bootloader shall be started |

## 4.3.3. Integration notes

### 4.3.3.1. Exclusive areas

Exclusive areas information is not available for this module.

### 4.3.3.2. Production errors

Production errors information is not available for this module.

### 4.3.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

Memory mapping information is not available for this module.

### 4.3.3.4. Integration requirements

**WARNING** ⚠

**Integration requirements list is not exhaustive**

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the BM module.

# 4.4. BlPduR

## 4.4.1. Configuration parameters

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |

| Containers included | | |
|---|---|---|
| General | 1..1 | This container describes the general properties of the node. |
| MultipleIdentifier | 1..1 | |
| PduConnection | 0..n | Configuration of all the PduConnection |
| IDGroup | 0..4 | Configuration of all the connection ID Group |
| CommonPublishedInformation | 1..1 | **Label:** Common Published Information<br>Common container, aggregated by all modules. It contains published information about vendor and versions. |
| PublishedInformation | 1..1 | **Label:** EB Published Information<br>Additional published parameters not covered by CommonPublishedInformation container. |

### 4.4.1.1. General

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| MANAGE_PERIOD | 1..1 |
| Can_Protocol_Supported | 1..1 |
| FlexRay_Protocol_Supported | 1..1 |
| Eth_Protocol_Supported | 1..1 |
| Lin_Protocol_Supported | 1..1 |
| QueuedManagement | 1..1 |
| MultipleRxBuffer | 1..1 |
| RxPhysicalBufferSize | 1..1 |
| RxFunctionalBufferSize | 1..1 |
| RxBufferNum | 1..1 |
| Enable_DownloadData_Streaming | 1..1 |

| Parameter Name | MANAGE_PERIOD |
|---|---|
| **Description** | This entry allows to configure the period of the cyclic BlPduR task. |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | INTEGER |
| **Range** | >=1 |
| **Origin** | EB |

| **Parameter Name** | **Can_Protocol_Supported** |
|---|---|
| **Description** | This entry allows to specify if the CAN network shall be supported |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |
| **Origin** | EB |

| **Parameter Name** | **FlexRay_Protocol_Supported** |
|---|---|
| **Description** | This entry allows to specify if the FlexRay network shall be supported |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Eth_Protocol_Supported** |
|---|---|
| **Description** | This entry allows to specify if the Ethernet network shall be supported |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Lin_Protocol_Supported** |
|---|---|
| **Description** | This entry allows to specify if the LIN network shall be supported |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **QueuedManagement** |
|---|---|
| **Label** | Queued Management |
| **Description** | Specify if the Queued management feature is enable. |

| | |
|---|---|
| | This feature allows the bootloader to be able to receive a second physical request before finishing processing the response to the first one.<br>The Bootloader shall store the second request in a FIFO queue for later processing.<br>Queued management is used to reduce download time and latency caused by the gateways. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **MultipleRxBuffer** |
|---|---|
| **Label** | Multiple Receive Buffer |
| **Description** | Specify if the multiple receive buffer feature is enable<br>This feature allows supporting in parallel data reception and data flash writing.<br>Multiple receive buffers is used to improve global downloading time |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **RxPhysicalBufferSize** |
|---|---|
| **Label** | Rx Physical Buffer Size |
| **Description** | Define the size of the Rx physical buffer in bytes<br>If the multiple buffer is enabled, all physical buffers will have this same size |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 4095 |
| **Origin** | EB |

| **Parameter Name** | **RxFunctionalBufferSize** |
|---|---|
| **Label** | Rx Functional Buffer Size |
| **Description** | Define the size of the Rx functional buffer in bytes |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | INTEGER |
| **Default value** | 8 |
| **Origin** | EB |

| **Parameter Name** | **RxBufferNum** |
|---|---|
| **Label** | Rx Buffer Number |
| **Description** | Define the number of Rx buffer usable in Reception when multiple receive buffer feature or queued management feature is activated<br>Maximum 4 buffers are allowed |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 2 |
| **Range** | <=4<br>>=2 |
| **Origin** | EB |

| **Parameter Name** | **Enable_DownloadData_Streaming** |
|---|---|
| **Label** | DownloadData Streaming |
| **Description** | Tick this option to support Streaming of the data received in the UDS TransferData Request.<br>This feature shall improve the download performance. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

### 4.4.1.2. MultipleIdentifier

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| MultipleIdentifierGroup | 1..1 |
| MultipleIdentifierSelect-Timeout | 1..1 |
| IDGroupPDURef | 1..1 |

| Parameters included | |
|---|---|
| IDGroupPDUId | 1..1 |
| IDGroupByteNum | 1..1 |

| Parameter Name | MultipleIdentifierGroup |
|---|---|
| Label | Multiple Idenfier Group |
| Description | Specify if the Multiple Identifier Group feature is enable<br>This feature allows supporting several connection group and enable only one of them at initialization<br>The connection can be selected from different source<br>▶ OFF: The feature is disabled<br>▶ CAN_NOTIFICATION: A CAN frame reception will notify which Identifier Group shall be selected<br>▶ EXTERNAL_NOTIFICATION: A callback function will be called at startup to retrieve the Identifier Group |
| Multiplicity | 1..1 |
| Type | STRING |
| Default value | OFF |
| Range | OFF<br>CAN_NOTIFICATION<br>EXTERNAL_NOTIFICATION |
| Origin | EB |

| Parameter Name | MultipleIdentifierSelectTimeout |
|---|---|
| Label | Multiple Idenfier Timeout |
| Description | Specify The timeout (in ms) to select a default connection group<br>In case no connection has been chosen after this timeout the default connection group will be selected |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 500 |
| Origin | EB |

| Parameter Name | IDGroupPDURef |
|---|---|

| | |
|---|---|
| **Label** | ID group PDU reference |
| **Description** | Reference to the `PDU` that will contain the information on which Connection Group shall be selected<br>This PDU shall as well be reference in the CanIf module of the communication stack |
| **Multiplicity** | 1..1 |
| **Type** | REFERENCE |
| **Origin** | AUTOSAR_ECUC |

| **Parameter Name** | **IDGroupPDUId** |
|---|---|
| **Label** | ID group PDU Id |
| **Description** | Defines the Pdu ID that will be used by PduR |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Origin** | EB |

| **Parameter Name** | **IDGroupByteNum** |
|---|---|
| **Label** | ID group Byte Number |
| **Description** | Defines the index of the byte that contain information regarding the group ID within the data received<br>The values allowed are between 0 (LSB) and 7 (MSB) of the received data |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0<br><=7 |
| **Origin** | EB |

### 4.4.1.3. PduConnection

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |

| Containers included | | |
|---|---|---|
| RxPdu | 0..n | Configuration of all the RxPdus references |

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| TxPduRef | 1..1 |
| TxConfPduId | 1..1 |
| TesterAddress | 1..1 |
| LinConnection | 1..1 |
| ShareFunctionalId | 1..1 |
| SharedPduReference | 1..1 |

| Parameter Name | TxPduRef |
|---|---|
| Label | TxPdu Reference |
| Description | Reference to the Pdu in the EcucPduCollection configured for this Transmission Channel.<br>Through this reference, BlPduR can resolve the PduId used for Transmission in the PduR_BlPduRTransmit() API and defined by the PduR.<br>Through this reference, PduRouter can resolve the PduId used for Transmission by in the following APIs:<br>▶ BlPduR_CopyTxData()<br>▶ BlPduR_TpTxConfirmation()<br>and defined by the BlPduR in TxConfirmationPduId. |
| Multiplicity | 1..1 |
| Type | REFERENCE |
| Origin | AUTOSAR_ECUC |

| Parameter Name | TxConfPduId |
|---|---|
| Label | TxPdu Identifier |
| Description | This entry allows to configure the TxConfirmationPduId that shall be used by PduR to transmit Tx confirmation<br>In case of Lin connection this field is used to define the MsgIdx of the configured LTP message |
| Multiplicity | 1..1 |

| | |
|---|---|
| **Type** | INTEGER |
| **Default value** | 0 |
| **Origin** | EB |

| **Parameter Name** | **TesterAddress** |
|---|---|
| **Label** | Tester Address |
| **Description** | This defines the Tester Address used in source/target address of the current connection |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Origin** | EB |

| **Parameter Name** | **LinConnection** |
|---|---|
| **Label** | Lin Connection |
| **Description** | Defines if the connection is to managed Lin Pdu<br>If so the generation of this connection will be done differently |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **ShareFunctionalId** |
|---|---|
| **Label** | Share Functional Id |
| **Description** | Defines if this connection shall reuse an existing functional Id from another connection |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **SharedPduReference** |
|---|---|
| **Label** | Shared Pdu Reference |
| **Description** | Reference to the shared Pdu |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | REFERENCE |
| **Origin** | EB |

### 4.4.1.4. RxPdu

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Ref | 1..1 |
| Type | 1..1 |
| Id | 1..1 |

| **Parameter Name** | **Ref** |
|---|---|
| **Label** | RxPdu Reference |
| **Description** | Reference to `Pdu` from the `EcucPduCollection` configured for this Reception Channel.<br>Through this reference, the `PduRouter` can resolve the `PduId` used for Reception by `BlPduR` in the following APIs:<br>▶ `BlPduR_StartOfReception()`<br>▶ `BlPduR_CopyRxData()`<br>▶ `BlPduR_TpRxIndication()` |
| **Multiplicity** | 1..1 |
| **Type** | REFERENCE |
| **Origin** | AUTOSAR_ECUC |

| **Parameter Name** | **Type** |
|---|---|
| **Description** | This entry allows to specify which kind of RxPdu is used.<br>Please select between :<br>▶ PHYSICAL<br>▶ FUNCTIONAL |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | PHYSICAL |
| **Range** | PHYSICAL |

| | |
|---|---|
| | FUNCTIONAL |
| **Origin** | EB |

| **Parameter Name** | **Id** |
|---|---|
| **Description** | This entry allows to configure the RxPduId that shall be used by BlPduR to transmit diagnostic response<br>In case of Lin connection this field is used to define the MsgIdx of the configured LTP message |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Origin** | EB |

### 4.4.1.5. IDGroup

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| ConnectionReflist | 0..n | |

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Default | 1..1 |
| ArchitectureId | 1..1 |

| **Parameter Name** | **Default** |
|---|---|
| **Label** | Default ID group |
| **Description** | Specify if this IDGroup is the default one<br>Only one ID Group can be selected in the configuration |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **ArchitectureId** |
|---|---|
| **Label** | Architecture Id |

| | |
|---|---|
| **Description** | This defines the Architecture Identification for this group<br>This value will be used to find the associated ID to choose link to the received ID in the architecture frame |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 4 |
| **Range** | >=0<br><=255 |
| **Origin** | EB |

### 4.4.1.6. ConnectionReflist

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ConnectionRef | 1..1 |

| **Parameter Name** | **ConnectionRef** |
|---|---|
| **Label** | Connection reference |
| **Description** | Reference to the `connection` that will be part of the used connection group |
| **Multiplicity** | 1..1 |
| **Type** | REFERENCE |
| **Origin** | AUTOSAR_ECUC |

### 4.4.1.7. CommonPublishedInformation

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ArMajorVersion | 1..1 |
| ArMinorVersion | 1..1 |
| ArPatchVersion | 1..1 |
| SwMajorVersion | 1..1 |
| SwMinorVersion | 1..1 |

| **Parameters included** | |
|---|---|
| SwPatchVersion | 1..1 |
| ModuleId | 1..1 |
| VendorId | 1..1 |
| Release | 1..1 |

| **Parameter Name** | **ArMajorVersion** |
|---|---|
| **Label** | AUTOSAR Major Version |
| **Description** | Major version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ArMinorVersion** |
|---|---|
| **Label** | AUTOSAR Minor Version |
| **Description** | Minor version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ArPatchVersion** |
|---|---|
| **Label** | AUTOSAR Patch Version |
| **Description** | Patch level version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |

| | |
|---|---|
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMajorVersion** |
|---|---|
| **Label** | Software Major Version |
| **Description** | Major version number of the vendor specific implementation of the module. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMinorVersion** |
|---|---|
| **Label** | Software Minor Version |
| **Description** | Minor version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 13 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwPatchVersion** |
|---|---|
| **Label** | Software Patch Version |
| **Description** | Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ModuleId** |
|---|---|
| **Label** | Numeric Module ID |
| **Description** | Module ID of this module from Module List |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **VendorId** |
|---|---|
| **Label** | Vendor ID |
| **Description** | Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 1 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **Release** |
|---|---|
| **Label** | Release Information |
| **Multiplicity** | 1..1 |
| **Type** | STRING_LABEL |
| **Default value** | |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

### 4.4.1.8. PublishedInformation

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| PbcfgMSupport | 1..1 |

| **Parameter Name** | **PbcfgMSupport** |
|---|---|
| **Label** | PbcfgM support |
| **Description** | Specifies whether or not the BlPduR can use the PbcfgM module for post-build support. |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | BOOLEAN |
| **Default value** | false |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

## 4.4.2. Application programming interface (API)

### 4.4.2.1. Functions

#### 4.4.2.1.1. BlPduR_AllSlots

| | |
|---|---|
| **Purpose** | Activate the FlexRay mode AllSlots. |
| **Synopsis** | `void BlPduR_AllSlots ( void );` |
| **Description** | If FlexRay is used, this Api call FlexRay state manager to request him the AllSlots mode |

#### 4.4.2.1.2. BlPduR_CopyRxData

| | | |
|---|---|---|
| **Purpose** | This service is called by the receiving Tp module, through PduR, requesting a TP-buffer. | |
| **Synopsis** | `BufReq_ReturnType BlPduR_CopyRxData ( PduIdType RxPduId , const PduInfoType * PduInfoPtr , PduLengthType * BufferSizePtr );` | |
| **Parameters (in)** | `RxPduId` | Identification of the received I-PDU. |
| | `PduInfoPtr` | Pointer to the buffer (SduDataPtr) and its length (SduLength) containing the data to be copied by PDU Router module in case of gateway or upper layer module in case of reception. |
| **Parameters (out)** | `BufferSizePtr` | Available receive buffer after data has been copied. |
| **Return Value** | Result of buffer request | |
| | `BUFREQ_OK` | Buffer request accomplished successfully. |

| | | |
|---|---|---|
| | `BUFREQ_E_BUSY` | Temporarily no buffer available. It's up the requestor to retry request for a certain time. |
| | `BUFREQ_E_NOT_OK` | Buffer request not successful. Buffer cannot be accessed. |

#### 4.4.2.1.3. BlPduR_CopyTxData

| | | |
|---|---|---|
| **Purpose** | This service is called by the sending Tp module, through PduR, requesting a TP-buffer. | |
| **Synopsis** | `BufReq_ReturnType` **`BlPduR_CopyTxData`** `( PduIdType TxPduId , PduInfoType * PduInfoPtr , RetryInfoType * RetryInfoPtr , PduLengthType * AvailableDataPtr );` | |
| **Parameters (in)** | `TxPduId` | Identification of the transmitted I-PDU. |
| | `RetryInfoPtr` | This parameter is used to retransmit data because problems during the last service call. If the I-PDU is transmitted from a local module (e.g. DCM) the PDU router module will just forward the parameter value without check. If the I-PDU is gatewayed from another bus, the PDU Router module will make the following interpretation: If the transmitted TP I-PDU does not support the retry feature a NULL_PTR is provided. It indicates that the copied transmit data can be removed from the buffer after it has been copied. If the retry feature is used by the Tx I-PDU, RetryInfoPtr must point to a valid RetryInfoType element. If TpDataState indicates TP_CONFPENDING, the previously copied data must remain in the TP buffer to be available for error recovery. TP_DATACONF indicates that all data that have been copied so far are confirmed and can be removed from the TP buffer. Data copied by this API call are excluded and will be confirmed later. TP_DATARETRY indicates that this API call shall copy already copied data in order to recover from |

| | | |
|---|---|---|
| | | an error. In this case AvailableDataPtr specifies the offset of the first byte to be copied by the API call. |
| | `RTabLoTpConfigIdx` | Index referring to the configuration of the LoTp-Module. |
| **Parameters (out)** | `PduInfoPtr` | Provides destination buffer and the number of bytes to copy. In case of gateway the PDU Router module will copy, otherwise the source upper layer module will copy the data. If not enough transmit data is available, no data is copied. The transport protocol module will retry. A copy size of 0 can be used to indicate state changes in the retry parameter. |
| | `AvailableDataPtr` | Indicates the remaining number of bytes that are available in the PduR Tx buffer. AvailableDataPtr can be used by TP modules that support dynamic payload lengths (e.g. Iso FrTp) to determine the size of the following CFs. |
| **Return Value** | Result of buffer request | |
| | `BUFREQ_OK` | Data has been copied to the transmit buffer completely as requested. |
| | `BUFREQ_E_BUSY` | Request could not be fulfilled, because the required amount of Tx data is not available. TP layer might retry later on. No data has been copied. |
| | `BUFREQ_E_NOT_OK` | Data has not been copied. Request failed. |

#### 4.4.2.1.4. BlPduR_Custom_Com_Deactivate

| | |
|---|---|
| **Purpose** | Deactivate the Communication. |
| **Synopsis** | `void BlPduR_Custom_Com_Deactivate ( void );` |

#### 4.4.2.1.5. BlPduR_Custom_Com_Init

| | |
|---|---|
| **Purpose** | Initialization of modules of communication stack and initialization of the stored PDUID of the active connection. |

| | |
|---|---|
| **Synopsis** | `void BlPduR_Custom_Com_Init ( void );` |

### 4.4.2.1.6. BlPduR_GetGroupIdVal

| | |
|---|---|
| **Purpose** | This service is at system initialization by BlPduR module to retrieve Group Id use in the ECU. |
| **Synopsis** | `u8 BlPduR_GetGroupIdVal ( void );` |
| **Return Value** | value of the Group ID that shall be used in by the bootloader to run |
| **Description** | The Group ID shall be retrieve depending on the system architecture (e.g. from NVM, from a specific I/O,..) It shall then be returned. |

### 4.4.2.1.7. BlPduR_GetNextBuffer

| | | |
|---|---|---|
| **Purpose** | This service is called by the Prog module, to get the next buffer information to treat. | |
| **Synopsis** | `void BlPduR_GetNextBuffer ( u16 * puwLen , u8 ** paubData );` | |
| **Parameters (out)** | `puwLen` | Length of the data store in the provided buffer |
| | `paubData` | Address of the buffer to treat (Null pointer if no more buffer) |

### 4.4.2.1.8. BlPduR_GetRxPduId

| | | |
|---|---|---|
| **Purpose** | Get the Rx Pdu Identifier on which the response after reset shall be sent. | |
| **Synopsis** | `void BlPduR_GetRxPduId ( u8 * pubRxPduId );` | |
| **Parameters (out)** | `pubRxPduId` | Rx Pdu Identifier pointer |

### 4.4.2.1.9. BlPduR_GetTpParameter

| | | |
|---|---|---|
| **Purpose** | This API gets the value of a TP paremeter (STmin or BS). | |
| **Synopsis** | `u16 BlPduR_GetTpParameter ( tTpParameterId ubParameterId );` | |
| **Parameters (in)** | `ubParameterId` | parameter ID to get (BLPDUR_TP_STMIN or BLPDUR_TP_BS) |
| **Return Value** | u16 value of the requested TP parameter | |

#### 4.4.2.1.10. BlPduR_GroupIdFrameFilter

| | | |
|---|---|---|
| **Purpose** | This service is Called by Can module to filter receive frames. | |
| **Synopsis** | `boolean BlPduR_GroupIdFrameFilter ( Can_HwHandleType Hrh , Can_IdType CanId , uint8 CanDlc , const uint8 * CanSduPtr );` | |
| **Parameters (in)** | `Hrh` | Hardward object number |
| | `CanId` | Can ID of the received frame |
| | `CanDlc` | Length of the received frame |
| | `CanSduPtr` | Pointer to the data of the received frame |
| **Return Value** | | |
| **Description** | This service is used to know if the given CanId shall be accepted or rejected depending on the active group ID | |

#### 4.4.2.1.11. BlPduR_Init

| | |
|---|---|
| **Purpose** | Initialize all layers. |
| **Synopsis** | `void BlPduR_Init ( void );` |
| **Description** | This function call all the subInit function BlPduR_InitX It shall be called only once at ECU start-up. |

#### 4.4.2.1.12. BlPduR_Init1

| | |
|---|---|
| **Purpose** | Initialize the communication stack. |
| **Synopsis** | `void BlPduR_Init1 ( void );` |
| **Description** | This function initializes the communication stack (CAN or FR or Ethernet) by calling there *_Init function. |

#### 4.4.2.1.13. BlPduR_Init2

| | |
|---|---|
| **Purpose** | Initialize all specific bootloader layers. |
| **Synopsis** | `void BlPduR_Init2 ( void );` |
| **Description** | This function initializes the Bootloader specific layer (PROG, SA and FLASH) by calling there *_Init function. |

#### 4.4.2.1.14. BlPduR_IsNetworkSynchronized

| | | |
|---|---|---|
| **Purpose** | Check if FlexRay network is synchronized. | |
| **Synopsis** | `u8 BlPduR_IsNetworkSynchronized ( u8 * frCycle );` | |
| **Return Value** | synchronization status | |
| | `BLPDUR_TRUE` | Network is synchronized |
| | `BLPDUR_FALSE` | Network is not synchronized |
| **Description** | If FlexRay is used, this Api check if the Flexray network is synchronized and return the status and the FlexRay cycle. | |

#### 4.4.2.1.15. BlPduR_IsTcpConnectionReestablished

| | | |
|---|---|---|
| **Purpose** | Calls SoAd API that indicates an incoming TCP connection on a server socket. | |
| **Synopsis** | `tBlPduRBoolean BlPduR_IsTcpConnectionReestablished ( void );` | |
| **Return Value** | connection status | |
| | `BLPDUR_TRUE` | SoAd accepts the established connection |
| | `BLPDUR_FALSE` | SoAd refuses the established connection, TcpIp stack shall close the connection. |
| **Description** | This service is used to know if the given CanId shall be accepted or rejected depending on the active group ID | |

#### 4.4.2.1.16. BlPduR_LockBuffer

| | | |
|---|---|---|
| **Purpose** | This service is called by the Prog module, to lock the buffer receive for treatment. | |
| **Synopsis** | `void BlPduR_LockBuffer ( u8 * pubIsLastBuffer );` | |
| **Parameters (out)** | `pubIsLastBuffer` | Information if there is still some buffer available to lock |

#### 4.4.2.1.17. BlPduR_Manage

| | |
|---|---|
| **Purpose** | Periodical task of all layers. |
| **Synopsis** | `void BlPduR_Manage ( void );` |

| | |
|---|---|
| **Description** | Modules periodic functions are called in this function (ex: COM_Manage) BLPDUR_MANAGE_PERIOD is configured in Tresos BlPduR plugin. The reception of segmented frames are handled in this function. |

### 4.4.2.1.18. BlPduR_RetrieveConnectionInfo

| | |
|---|---|
| **Purpose** | Retrieve the active connection information. |
| **Synopsis** | `void BlPduR_RetrieveConnectionInfo ( void );` |

### 4.4.2.1.19. BlPduR_RxIndication

| | | |
|---|---|---|
| **Purpose** | This service is Called by CanIf through PduR to notify a Pdu reception. | |
| **Synopsis** | `void BlPduR_RxIndication ( PduIdType RxPduId , PduInfoType * PduInfoPtr );` | |
| **Parameters (in)** | `RxPduId` | Pdu Number received |
| | `PduInfoPtr` | Pointer to the Pdu Information |

### 4.4.2.1.20. BlPduR_SaveTesterAddress

| | |
|---|---|
| **Purpose** | This service is called by the PROG module to save tester address to be used later in tester filtering. |
| **Synopsis** | `void BlPduR_SaveTesterAddress ( void );` |

### 4.4.2.1.21. BlPduR_SendMsgData

| | | |
|---|---|---|
| **Purpose** | | |
| **Synopsis** | `tBlPduRStatus BlPduR_SendMsgData ( PduIdType PduId , u16 uwLen , u8 * paubData );` | |
| **Return Value** | | |

### 4.4.2.1.22. BlPduR_Send_TPFrame

| | |
|---|---|
| **Purpose** | Send A TP frame using the current Tester Address. |

| | | |
|---|---|---|
| **Synopsis** | void **BlPduR_Send_TPFrame** ( u16 uwLen , u8 * paubData ); | |
| **Parameters (in)** | uwLen | Message length |
| | paubUdsData | message data pointer |
| **Description** | Send A TP frame using the current Tester Address | |

### 4.4.2.1.23. BlPduR_SetTesterAddress

| | |
|---|---|
| **Purpose** | Store the tester address in global variable. |
| **Synopsis** | void **BlPduR_SetTesterAddress** ( u8 ubTesterAddress ); |
| **Description** | Provide to BlPduR the tester address that shall be accepted in reception. |

### 4.4.2.1.24. BlPduR_SimulateRxRequest

| | | |
|---|---|---|
| **Purpose** | Simulate the reception of a frame. | |
| **Synopsis** | void **BlPduR_SimulateRxRequest** ( u16 uwLen , u8 * paubUdsData , u8 ubWithResp ); | |
| **Parameters (in)** | uwLen | Message length |
| | paubUdsData | message data pointer |
| | ubWithResp | indicate if a response will be performed (TRUE/FALSE) |
| **Description** | Simulate the reception of a frame | |

### 4.4.2.1.25. BlPduR_StartOfReception

| | | |
|---|---|---|
| **Purpose** | This service is called by the receiving Tp module, through PduR, requesting a TP-buffer. | |
| **Synopsis** | BufReq_ReturnType **BlPduR_StartOfReception** ( PduIdType RxPduId , PduLengthType TpSduLength , PduLengthType * BufferSizePtr ); | |
| **Parameters (in)** | RxPduId | Identification of the received I-PDU. |
| | TpSduLength | Total length of the PDU to be received. |
| **Parameters (out)** | BufferSizePtr | Available receive buffer in the receiving module. This parameter will be used to |

| | | |
|---|---|---|
| | | compute the Block Size (BS) in the transport protocol module. |
| **Return Value** | Result of buffer request | |
| | `BUFREQ_OK` | Connection has been accepted. BufferSizePtr indicates the available receive buffer. |
| | `BUFREQ_E_BUSY` | Currently no buffer of the requested size is available. BufferSizePtr remains unchanged. Connection has been rejected. |
| | `BUFREQ_E_OVFL` | No Buffer of the required length can be provided. |
| | `BUFREQ_E_NOT_OK` | Connection has been rejected. BufferSizePtr remains unchanged. |

#### 4.4.2.1.26. BlPduR_StoreConnectionInfo

| | |
|---|---|
| **Purpose** | Store the active connection information. |
| **Synopsis** | `void` **`BlPduR_StoreConnectionInfo`** `( void );` |

#### 4.4.2.1.27. BlPduR_StoreRxPduId

| | | |
|---|---|---|
| **Purpose** | Store the Rx Pdu Identifier on which the request shall be responded after reset has been received. | |
| **Synopsis** | `void` **`BlPduR_StoreRxPduId`** `( u8 ubRxPduId );` | |
| **Parameters (in)** | `pubRxPduId` | Rx Pdu Identifier |

#### 4.4.2.1.28. BlPduR_TpChangeParameter

| | | |
|---|---|---|
| **Purpose** | This API changes the value of a TP paremeter (STmin or BS). | |
| **Synopsis** | `tBlPduRStatus` **`BlPduR_TpChangeParameter`** `( tTpParameterId ubParameterId , u16 uwParameterValue );` | |
| **Parameters (in)** | `ubParameterId` | parameter ID to change (BLPDUR_TP_STMIN or BLPDUR_TP_BS) |
| | `uwParameterValue` | new value to set |

| | | |
|---|---|---|
| **Return Value** | BlPduRStatus variable change status | |
| | `BLPDUR_E_OK` | Parameter is changed |
| | `BLPDUR_E_NOT_OK` | Parameter isn't changed |

### 4.4.2.1.29. BlPduR_TpRxIndication

| | | |
|---|---|---|
| **Purpose** | This service is called by the Tp module, through PduR, after an I-PDU has been received successfully or when an error occurred. It is also used to confirm cancellation of an I-PDU. | |
| **Synopsis** | `void **BlPduR_TpRxIndication** ( PduIdType RxPduId , NotifResultType Result );` | |
| **Parameters (in)** | `RxPduId` | Identification of the received I-PDU. |
| | `Result` | Result of the reception. |

### 4.4.2.1.30. BlPduR_TpTxConfirmation

| | | |
|---|---|---|
| **Purpose** | This service is called by the Tp module, through PduR, after the I-PDU has been transmitted on its network, the result will reveal if the transmission was successful or not. | |
| **Synopsis** | `void **BlPduR_TpTxConfirmation** ( PduIdType TxPduId , NotifResultType Result );` | |
| **Parameters (in)** | `TxPduId` | Identification of the transmitted I-PDU. |
| | `Result` | Result of the transmission of the I-PDU. |

### 4.4.2.1.31. BlPduR_UnlockBuffer

| | | |
|---|---|---|
| **Purpose** | This service is called by the Prog module, to unlock one or all buffer used. | |
| **Synopsis** | `void **BlPduR_UnlockBuffer** ( u8 ubBufferType );` | |
| **Parameters (in)** | `ubBufferType` | Buffer to be unlock (one or all) |

### 4.4.2.1.32. LIN_ComLossInd

| | |
|---|---|
| **Purpose** | This API. |

| | | |
|---|---|---|
| **Synopsis** | `void LIN_ComLossInd ( void );` | |

#### 4.4.2.1.33. LIN_StatusInd

| | | |
|---|---|---|
| **Purpose** | This API. | |
| **Synopsis** | `void LIN_StatusInd ( tLinFrameIdx uFrameIdx , tLinFrameStatus eStatus );` | |
| **Parameters (in)** | `uFrameIdx` | transmitted or received frame |
| | `eStatus` | LIN message status |

#### 4.4.2.1.34. LIN_WakeUpInd

| | |
|---|---|
| **Purpose** | This API. |
| **Synopsis** | `void LIN_WakeUpInd ( void );` |

#### 4.4.2.1.35. LTP_RxInd

| | | |
|---|---|---|
| **Purpose** | This API. | |
| **Synopsis** | `void LTP_RxInd ( u8 ebStatus );` | |
| **Parameters (in)** | `ebStatus` | LTP message status |

#### 4.4.2.1.36. LTP_TxConf

| | | |
|---|---|---|
| **Purpose** | This API gives transmission indication depending on the LTP module message status. | |
| **Synopsis** | `void LTP_TxConf ( u8 ebStatus );` | |
| **Parameters (in)** | `ebStatus` | LTP message status |

## 4.4.3. Integration notes

### 4.4.3.1. Exclusive areas

Exclusive areas information is not available for this module.

### 4.4.3.2. Production errors

Production errors information is not available for this module.

### 4.4.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

Memory mapping information is not available for this module.

### 4.4.3.4. Integration requirements

**WARNING** ⚠

**Integration requirements list is not exhaustive**
The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the BlPduR module.

# 4.5. Prog

## 4.5.1. Configuration parameters

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| CommonPublishedInformation | 1..1 | **Label:** Common Published Information<br>Common container, aggregated by all modules. It contains published information about vendor and versions. |
| PublishedInformation | 1..1 | **Label:** EB Published Information<br>Additional published parameters not covered by CommonPublishedInformation container. |
| General | 1..1 | This container contains the general proprieties of the node. |

| Containers included | | |
|---|---|---|
| DownloadVerification | 1..1 | This container contains the configuration element for download verification |
| DownloadFlashRoutines | 1..1 | This container contains the configuration element for downloading Flash routines |
| SBLVerificationBlockTable | 1..1 | This container contains the location data of the verification block table of the secondary bootloader |
| Startup | 1..1 | This container contains the general proprieties of the node |
| CompleteAndCompatibleBlock | 1..1 | |
| Segments | 0..n | This container contains the description of the Segments of available memory to reprogram and the kind of memory used.<br>PLEASE NOTE THAT:<br>- ONLY ONE SEGMENT SHALL BE OF DEFINED FOR UPDATER PARTITION<br>- UNIQUE SEGMENT FOR UPDATER PARTITION SHALL REFER TO BLOCK DEFINED WITH INDEX 0 |
| Blocks | 0..250 | This container contains the description of the blocks used to request the erasing.<br>PLEASE NOTE THAT BLOCK CONTAINING BOOTLOADER SEGMENT SHALL HAVE INDEX 0. |
| Memory | 0..n | This container contains the description of the memories used by the bootloader. |
| GM | 1..1 | This container contains the GM specific configuration |
| Security | 1..1 | |
| OemInd | 1..1 | This container contains the OemInd specific configuration |
| VAG | 1..1 | This container contains the Volkswagen specific configuration |
| Decryption | 1..1 | This container contains the Decryption specific configuration. |

### 4.5.1.1. CommonPublishedInformation

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ArMajorVersion | 1..1 |

| **Parameters included** | |
|---|---|
| ArMinorVersion | 1..1 |
| ArPatchVersion | 1..1 |
| SwMajorVersion | 1..1 |
| SwMinorVersion | 1..1 |
| SwPatchVersion | 1..1 |
| ModuleId | 1..1 |
| VendorId | 1..1 |
| Release | 1..1 |

| **Parameter Name** | **ArMajorVersion** |
|---|---|
| **Label** | AUTOSAR Major Version |
| **Description** | Major version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ArMinorVersion** |
|---|---|
| **Label** | AUTOSAR Minor Version |
| **Description** | Minor version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ArPatchVersion** |
|---|---|
| **Label** | AUTOSAR Patch Version |
| **Description** | Patch level version number of AUTOSAR specification on which the appropriate implementation is based on. |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMajorVersion** |
|---|---|
| **Label** | Software Major Version |
| **Description** | Major version number of the vendor specific implementation of the module. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 2 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMinorVersion** |
|---|---|
| **Label** | Software Minor Version |
| **Description** | Minor version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 17 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwPatchVersion** |
|---|---|
| **Label** | Software Patch Version |
| **Description** | Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| Parameter Name | ModuleId |
|---|---|
| Label | Numeric Module ID |
| Description | Module ID of this module from Module List |
| Multiplicity | 1..1 |
| Type | INTEGER_LABEL |
| Default value | 0 |
| Configuration class | **PublishedInformation:** |
| Origin | Elektrobit Automotive GmbH |

| Parameter Name | VendorId |
|---|---|
| Label | Vendor ID |
| Description | Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list |
| Multiplicity | 1..1 |
| Type | INTEGER_LABEL |
| Default value | 1 |
| Configuration class | **PublishedInformation:** |
| Origin | Elektrobit Automotive GmbH |

| Parameter Name | Release |
|---|---|
| Label | Release Information |
| Multiplicity | 1..1 |
| Type | STRING_LABEL |
| Default value | |
| Configuration class | **PublishedInformation:** |
| Origin | Elektrobit Automotive GmbH |

### 4.5.1.2. PublishedInformation

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| PbcfgMSupport | 1..1 |

| Parameter Name | PbcfgMSupport |
|---|---|

| | |
|---|---|
| **Label** | PbcfgM support |
| **Description** | Specifies whether or not the Prog can use the PbcfgM module for post-build support. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

### 4.5.1.3. General

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| ProgCalReferences | 1..1 | **Label:** ProgCalReferences<br>Contains references to Cal configuration identifiers. |

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| MANAGE_PERIOD | 1..1 |
| NO_SECURITYLEVEL_RESET_ON_SESSIONCHANGE | 1..1 |
| Enable_Compression | 1..1 |
| Compression_Algorithm | 1..1 |
| Decomp_Out_Buffer_size | 1..1 |
| Enable_Decompression_Slicing | 1..1 |
| Decomp_Slice_size | 1..1 |
| Data_Size_In_RD | 1..1 |
| Dsc_Prog_Response | 1..1 |
| Expected_Crc_Location | 1..1 |
| Network_Management | 1..1 |
| Auto_Control | 1..1 |

| Parameters included | |
|---|---|
| Use_CSM_ASR430_DemoWrapper | 1..1 |
| Tunable_Parameters | 1..1 |
| Dual_Memory_Bank_Used | 1..1 |
| Transmit_Nrc78_Before_EraseCheck | 1..1 |
| Transmit_Nrc78_On_Erase | 1..1 |
| PreliminaryErasing | 1..1 |
| MaxBlockID | 1..1 |
| Number_Of_Sector_To_Erase_Before_Sending_NRC78 | 1..1 |
| Erase_Check | 1..1 |
| Transmit_Response_Before_Reset | 1..1 |
| Check_Programming_PreConditions | 1..1 |
| ResetAfterDsc01InDefaultSession | 1..1 |
| Sleep_Management_Type | 1..1 |
| Sleep_Timeout | 1..1 |
| Max_Bytes_in_TD | 1..1 |
| FAR_POINTER_Definition | 1..1 |
| ResumableReprog | 1..1 |

| Parameter Name | MANAGE_PERIOD |
|---|---|
| Description | Specifies the period of the PROG manage task in ms.<br>This period must be multiple of EB periodical value in EB module configuration.<br>Range:[1ms ; 25ms] |
| Multiplicity | 1..1 |
| Type | INTEGER |

| | |
|---|---|
| **Default value** | 10 |
| **Range** | >=1<br><=25 |
| **Origin** | EB |

| **Parameter Name** | **NO_SECURITYLEVEL_RESET_ON_SESSIONCHANGE** |
|---|---|
| **Label** | No Security Level reset on Session change |
| **Description** | Enabling this parameter allows the Bootloader:<br>1. To have the security level unchanged(no security level reset) across the transitions between Non-Default sessions.<br>2. No need to unlock the same security level in the Bootloader, if the unlocking of Security (for the same Security level) is already done from the Application just before starting/jumping to the Bootloader.<br>Note: To have the second functionality enabled, the SecurityLevel variable shall be mapped to the shared Non volatile memory of Application and Bootloader. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Enable_Compression** |
|---|---|
| **Description** | Specify if the compression is enable. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Compression_Algorithm** |
|---|---|
| **Label** | Compression algorithm Id |
| **Description** | Compression algorithm id that shall be supported from dataFormatIdentifier field of RequestDownload service |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 1 |

| | |
|---|---|
| **Origin** | EB |

| **Parameter Name** | **Decomp_Out_Buffer_size** |
|---|---|
| **Description** | Size of the Output decompression buffer. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 4000 |
| **Range** | >=1000<br><=50000 |
| **Origin** | EB |

| **Parameter Name** | **Enable_Decompression_Slicing** |
|---|---|
| **Description** | Specify if the slicing decompression is enable. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Decomp_Slice_size** |
|---|---|
| **Description** | Size of the decompression input buffer slice. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 512 |
| **Range** | >=1<br><=1000 |
| **Origin** | EB |

| **Parameter Name** | **Data_Size_In_RD** |
|---|---|
| **Description** | This entry allows to specify if the data size passed in the RequestDownload service means the compressed or decompressed data size.<br>Please select between :<br>▶ compressed<br>▶ decompressed |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | STRING |
| **Default value** | compressed |
| **Range** | compressed<br>decompressed |
| **Origin** | EB |

| **Parameter Name** | **Dsc_Prog_Response** |
|---|---|
| **Label** | Diagnostic Reprogramming response |
| **Description** | Specify if the DSC 02 response shall be sent by the bootloader if the application receive a reprogramming request<br>▶ Case tick: The response will be sent by the bootloader<br>▶ Case untick: The response will not be sent by the bootloader |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Expected_Crc_Location** |
|---|---|
| **Label** | CRC Location: Request/Application |
| **Description** | Specify if the expected CRC should be get by calling a callback or if it is passed in the request.<br>▶ Case Application: The CRC is get by calling PROG_CustomGetExpectedCrc callback<br>▶ Case Request: The CRC is passed in the CheckMemory routine request. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | Request |
| **Range** | Request<br>Application |
| **Origin** | EB |

| **Parameter Name** | **Network_Management** |
|---|---|
| **Description** | Specify if the network management shall be supported or not.<br>This feature shall only be activated for VCC Bootloader on FlexRay. |

| | |
|---|---|
| | ▶ Case tick: Network management supported<br>▶ Case untick: Network management not supported |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Auto_Control** |
|---|---|
| **Description** | Specify if auto-control shall be done or not at the end of application download.<br>▶ Case tick: Auto-Control shall be done<br>▶ Case untick: Auto-Control shall not be done |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Use_CSM_ASR430_DemoWrapper** |
|---|---|
| **Label** | Use_CSM_ASR430_DemoWrapper |
| **Description** | To have the compatibility of BL modules with Cryto ASR 4.3.x the CSM wrapper in the demo is introduced.<br>Tick this option if you wish to make use of CSM wrapper to integrate ASR version 4.3.x Crypto modules. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Tunable_Parameters** |
|---|---|
| **Description** | This option allows to set the Segment configuration of the layer (in PROG_Cfg.c) in RAM.<br>An API called PROG_ParametersInit is also generated to initialize the parameters from the ROM.<br>This allows the customer to change dynamically the segment address values after the initialization. |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Dual_Memory_Bank_Used** |
|---|---|
| **Label** | Dual Memory Bank Used |
| **Description** | This option allows the usage of dual memory bank feature on the supported hardware. Please reffer to the user manual for details. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Transmit_Nrc78_Before_EraseCheck** |
|---|---|
| **Label** | NRC78 transmission before software invalidation |
| **Description** | Specify if an NRC78 response shall be systematically sent receiving the Erase request and before the routine processing (before software invalidation).<br>▶ Case untick: NRC78 will not be sent immediatly after a valid Erase request. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Transmit_Nrc78_On_Erase** |
|---|---|
| **Label** | NRC78 transmission after software invalidation |
| **Description** | Specify if an NRC78 response shall be sent before starting the memory erasing and after the software invalidation.<br>▶ Case untick: NRC78 will not be sent before the start of the memory erasing. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **PreliminaryErasing** |
|---|---|

| | |
|---|---|
| **Label** | Preliminary Erasing Enable |
| **Description** | Enable the Preliminary Erasing in EraseMemory request (Available only if erasing mode is by LogicalBlock) |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **MaxBlockID** |
|---|---|
| **Label** | Max Block ID |
| **Description** | The identifier used for the Preliminary Erasing can be defined in 1 or 2 bytes (e.g. 0xFF or 0xFFFF). |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | FF |
| **Range** | FF<br>FFFF |
| **Origin** | EB |

| **Parameter Name** | **Number_Of_Sector_To_Erase_Before_Sending_NRC78** |
|---|---|
| **Label** | Number of sector to erase before sending NRC78 |
| **Description** | Defines the number of sectors to erase before sending NRC78 to the tester<br>▶ If this value is set to "0" then the feature erase by sector is deactivated and the whole erase is performed<br>▶ If this value is greater than "0" then the erase by sector is enabled. In this case the NRC78 is sent each time the number of erased sectors reaches this value. It allows specifying a minimum time between two NRC78 |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0<br><=255 |
| **Origin** | EB |

| **Parameter Name** | **Erase_Check** |
|---|---|

| | |
|---|---|
| **Description** | Specify if software shall check if memory is already erased before doing an erase.<br>▶ Disabled: No check will be done and memory will be always erase on request.<br>▶ First Programming Check: Memory will not be erased if it has never been programmed.<br>▶ Memory Block Erased Check: If logical block is used for erasing, the memory block is not erased if the coresponding flag is set. One flag per memory block is used to know if the memory block is already erased. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | Disabled |
| **Range** | Disabled<br>First Programming Check<br>Memory Block Erased Check |
| **Origin** | EB |

| **Parameter Name** | **Transmit_Response_Before_Reset** |
|---|---|
| **Description** | Specify if response shall be sent before resetting software.<br>Case untick: No response transmitted before performing the reset<br>Case tick: Case tick: Response is transmitted before performing the reset |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Check_Programming_PreConditions** |
|---|---|
| **Description** | Specify if the programming pre-conditions should be checked.<br>If activated, Prog module will call callback PROG_CustCheck-ProgPrecond on reception of pre-conditions check routine and DiagnosticSessionControl(ProgrammingSession) request.<br>Project specific condition check shall be implemented in this callback. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |

| | |
|---|---|
| **Default value** | true |
| **Origin** | EB |

| **Parameter Name** | **ResetAfterDsc01InDefaultSession** |
|---|---|
| **Label** | Trigger Reset While Switching from default to Default Session |
| **Description** | Specify if ECU reset shall be triggered receiving a DefaultSession request while the ECU is already in Default Session.<br>▶ Case tick: Trigger reset from any session to default session.<br>▶ Case untick: No reset triggered if current session is already Default. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |
| **Origin** | EB |

| **Parameter Name** | **Sleep_Management_Type** |
|---|---|
| **Description** | Specify the sleep management type that shall be use<br>▶ Off: No sleep management managed by the bootloader<br>▶ Timeout: The bootloader will go into sleep mode on after a timeout without bus communication and wake up with network communication<br>▶ I/O: The bootloader will go into sleep mode on activation of an external I/O or switch (not supported yet) |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | Timeout |
| **Range** | Off<br>Timeout |
| **Origin** | EB |

| **Parameter Name** | **Sleep_Timeout** |
|---|---|
| **Description** | Specify the sleep timeout value in ms.<br>This timeout is started when the ECU enter in default session<br>After this timeout expired the ECU will go in sleep mode<br>This feature is only be supported when Sleep_Management type is "Timeout". |

| | |
|---|---|
| | Default value: 15000 (15 seconds) |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 15000 |
| **Origin** | EB |

| **Parameter Name** | **Max_Bytes_in_TD** |
|---|---|
| **Description** | Define the maxblock length for a TransferData If Osek stack is used, make sure this value was lower than buffer message value in TP module.<br>This value shall be lower than "Rx Physical Buffer Size" value. Please check BIP-duR . |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 4095 |
| **Origin** | EB |

| **Parameter Name** | **FAR_POINTER_Definition** |
|---|---|
| **Description** | Define the syntax for far pointer.<br>▶ **Example1:** `__far` will done `Data = (* (volatile __far *)<POINTER>)`<br>▶ **Example2:** (empty) will done `Data = (* (volatile *)<POINTER>)` |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | |
| **Origin** | EB |

| **Parameter Name** | **ResumableReprog** |
|---|---|
| **Label** | Resumable reprogramming |
| **Description** | Specify if the resumable reprogramming feature shall be used.<br>If activated, Prog module will store information allowing to resume an interrupted reprogramming. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |

| | |
|---|---|
| **Origin** | EB |

### 4.5.1.4. ProgCalReferences

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ProgCalDecompressConfigId | 1..1 |

| **Parameter Name** | **ProgCalDecompressConfigId** |
|---|---|
| **Label** | ProgCalDecompressConfigId |
| **Description** | Reference a *CalDecompress*<br>Dependencies:<br>▶ Reference shall be valid |
| **Multiplicity** | 1..1 |
| **Type** | CHOICE-REFERENCE |
| **Configuration class** | **PreCompile:** VariantPreCompile |
| **Origin** | Elektrobit Automotive GmbH |

### 4.5.1.5. DownloadVerification

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| VerificationOnTheFly | 1..1 |
| Checksum_Algo | 1..1 |
| SignatureVerificationOnFlashData | 1..1 |
| SignatureVerificationWithAddrLen | 1..1 |
| SignatureVerificationWithPhyAddr | 1..1 |
| AdditionalCRCComputation | 1..1 |
| CVN_Verification | 1..1 |

| Parameters included | |
|---|---|
| MaxNumberOfRD-PerBlock | 1..1 |
| Allow2MaxSuccessiveCheckMemoryRequests | 1..1 |

| **Parameter Name** | **VerificationOnTheFly** |
|---|---|
| **Label** | Verification on the fly |
| **Description** | Specify if the download verification shall be performed after data download or in parallel (on the fly) of the download.<br>Feature availability is OEM dependent. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Checksum_Algo** |
|---|---|
| **Label** | CRC algorithm: Signature / CRC32 Ethernet |
| **Description** | Specify which checksum algorithm shall be used<br>▶ Case Signature: A cryptographic signature verification will be done (only possible if cryptographic libraries are used). An additional CRC32 computation can be done by enabling "Additional CRC computation" parameter<br>▶ Case CRC32 Ethernet: Polynomial 0x04C11DB7 / Init value 0xFFFFFFFF / ReflectIn TRUE / ReflectOut TRUE / XOR on Output 0xFFFFFFFF |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | CRC32 Ethernet |
| **Range** | Signature<br>CRC32 Ethernet |
| **Origin** | EB |

| **Parameter Name** | **SignatureVerificationOnFlashData** |
|---|---|
| **Label** | Signature Verification on Flashed data |
| **Description** | If enabled: the signature will be verified by reading the Flash memory after the data have been written. If compression is used signature is so verified on decompressed data. |

| | |
|---|---|
| | If disabled: the signature will be verified on received data before writing to Flash memory. If compression is used signature is so verified on compressed data. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |
| **Origin** | EB |

| **Parameter Name** | **SignatureVerificationWithAddrLen** |
|---|---|
| **Label** | Signature Verification with segment address/length |
| **Description** | If enabled: the signature will be computed including the address and length of the segment.<br>If disabled: the signature will be computed including only the programmed data. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **SignatureVerificationWithPhyAddr** |
|---|---|
| **Label** | Signature Verification with physical address |
| **Description** | If enabled: When address shall be included in signature computation, the physical address will be used (i.e in case of external Flash, the configured offset will apply).<br>If disabled: When address shall be included in signature computation, the logical address (received address) will be used. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **AdditionalCRCComputation** |
|---|---|
| **Label** | Additional CRC Computation |
| **Description** | In case signature computation is performed, if enabled an additional CRC32 computation will be performed after signature verification on the reception of the CheckMemory routine.<br>Feature availability is OEM dependent. |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |
| **Origin** | EB |

| **Parameter Name** | **CVN_Verification** |
|---|---|
| **Label** | CVN Verification |
| **Description** | If enabled a calibration verification number will be performed on the reception of the verify partial software checksum routine. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **MaxNumberOfRDPerBlock** |
|---|---|
| **Label** | Maximum RequestDownload Per Block |
| **Description** | Define the maximum number of RequestDownload that Bootloader shall support for a single logical block.<br>It will impact the RAM memory consumption to store the downloaded memory area that shall be used for checksum computation. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 10 |
| **Origin** | EB |

| **Parameter Name** | **Allow2MaxSuccessiveCheckMemoryRequests** |
|---|---|
| **Label** | Allow 2 maximum successive Check Memory Requests |
| **Description** | If enabled: Only a second successive Check Memory Requests can be allowed by the Bootloader if the result of the first one was not sucessful.<br>If disabled: No successive Check Memory Requests will be allowed by the Bootloader. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |

| | |
|---|---|
| **Origin** | EB |

### 4.5.1.6. DownloadFlashRoutines

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Download_Flash_Routines | 1..1 |
| Decompress_Flash_Routines | 1..1 |
| Reject_RD_After_Corrupt_Flash_Routines | 1..1 |

| **Parameter Name** | **Download_Flash_Routines** |
|---|---|
| **Label** | Download FLash driver |
| **Description** | Specify if the flash routines will be downloaded to RAM via tester tool.<br>If activated, a RAM segment shall be configured. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Decompress_Flash_Routines** |
|---|---|
| **Label** | Decompress Flash driver |
| **Description** | Specify if the driver flash routines are compressed in the bootloader and need to be decompressed to RAM on security access unlock. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Reject_RD_After_Corrupt_Flash_Routines** |
|---|---|
| **Label** | Reject second attempt after failure |
| **Description** | if activated, after a failed check memory, a second attempt for downloading flash routines will be rejected. |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

### 4.5.1.7. SBLVerificationBlockTable

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Verification_Block_Table_Start_Address | 1..1 |
| Verification_Block_Table_Length | 1..1 |

| **Parameter Name** | **Verification_Block_Table_Start_Address** |
|---|---|
| **Description** | Start address of the verification block table of the secondary bootloader |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x7000DC00 |
| **Origin** | EB |

| **Parameter Name** | **Verification_Block_Table_Length** |
|---|---|
| **Description** | Length of the verification block table of the secondary bootloader before any processing (i.e. padding) |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x0000002C |
| **Origin** | EB |

### 4.5.1.8. Startup

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| PROG_Signature_High | 1..1 |

| Parameters included | |
|---|---|
| PROG_Signature_Low | 1..1 |
| PROG_Signature_Clear | 1..1 |

| Parameter Name | PROG_Signature_High |
|---|---|
| Description | MSB value set by the application to know if the reset is done from application.<br>the default value is recommended by VCC |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 0x50726f67 |
| Origin | EB |

| Parameter Name | PROG_Signature_Low |
|---|---|
| Description | LSB value set by the application to know if the reset is done from application.<br>the default value is recommended by VCC |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 0x5369676e |
| Origin | EB |

| Parameter Name | PROG_Signature_Clear |
|---|---|
| Description | LSB value set by the application to know if the reset is done from application.<br>the default value is recommended by VCC |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 0x00000000 |
| Origin | EB |

### 4.5.1.9. CompleteAndCompatibleBlock

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Block_Start_Addr | 1..1 |

| Parameters included | |
|---|---|
| Start_Complete_Compatible_Signature_High | 1..1 |
| Start_Complete_Compatible_Signature_Low | 1..1 |
| End_Complete_Compatible_Signature_High | 1..1 |
| End_Complete_Compatible_Signature_Low | 1..1 |
| CompleteCompatibleFunction_Timeout | 1..1 |

| Parameter Name | Block_Start_Addr |
|---|---|
| Description | First address of the Complete and compatible block. |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Origin | EB |

| Parameter Name | Start_Complete_Compatible_Signature_High |
|---|---|
| Description | MSB value set by the application at the start of the complete and compatible block<br>the default value is recommended by VCC |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 0x53746172 |
| Origin | EB |

| Parameter Name | Start_Complete_Compatible_Signature_Low |
|---|---|
| Description | LSB value set by the application at the start of the complete and compatible block<br>the default value is recommended by VCC |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 0x74536967 |
| Origin | EB |

| Parameter Name | End_Complete_Compatible_Signature_High |
|---|---|

| | |
|---|---|
| **Description** | MSB Value set by the application at the start of the complete and compatible block<br>the default value is recommended by VCC |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x456e6453 |
| **Origin** | EB |

| **Parameter Name** | **End_Complete_Compatible_Signature_Low** |
|---|---|
| **Description** | LSB Value set by the application at the start of the complete and compatible block<br>the default value is recommended by VCC |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x69676e61 |
| **Origin** | EB |

| **Parameter Name** | **CompleteCompatibleFunction_Timeout** |
|---|---|
| **Description** | Timeout value of the completecompatibleFunction<br>15ms by default |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 15 |
| **Origin** | EB |

### 4.5.1.10. Segments

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Memory | 1..1 |
| Access_Type | 1..1 |
| Reprog_Start_Address | 1..1 |
| Reprog_End_Address | 1..1 |
| Erase_Start_Address | 1..1 |
| Erase_End_Address | 1..1 |

| **Parameters included** | |
|---|---|
| Partition_Type | 1..1 |
| Protected_Partition_ID | 1..1 |
| HSM_PartitionID | 1..1 |
| HSM_RAM_Buffer | 1..1 |
| PROG_HSM_Timeout | 1..1 |
| ValidityCheck | 1..1 |

| **Parameter Name** | **Memory** |
|---|---|
| **Description** | Reference to the memory which contains the segment.<br>▶ **FLASH**<br>▶ **FLASH_EXT**<br>▶ **EEPROM**<br>▶ **RAM**<br>▶ **SCRATCHPAD** |
| **Multiplicity** | 1..1 |
| **Type** | REFERENCE |
| **Origin** | EB |

| **Parameter Name** | **Access_Type** |
|---|---|
| **Description** | Define Authorized memory access types to this segment<br>▶ **READ: Only allow Read memory access**<br>▶ **WRITE: Only allow Write and Erase memory access**<br>▶ **READ_WRITE: Allow Read, Write and Erase memory access** |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | READ_WRITE |
| **Range** | READ<br>WRITE<br>READ_WRITE |
| **Origin** | EB |

| **Parameter Name** | **Reprog_Start_Address** |
|---|---|

| | |
|---|---|
| **Description** | Start address of the segment in the memory.<br>Range: [0x00000000 ; 0xFFFFFFFF]<br>Coded on 32 bits |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Origin** | EB |

| **Parameter Name** | **Reprog_End_Address** |
|---|---|
| **Description** | End address of the segment in the memory.<br>Range: [0x00000000 ; 0xFFFFFFFF]<br>Coded on 32 bits |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Origin** | EB |

| **Parameter Name** | **Erase_Start_Address** |
|---|---|
| **Description** | Erasing start address of the segment in the memory.<br>Range: [0x00000000 ; 0xFFFFFFFF]<br>Coded on 32 bits |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Origin** | EB |

| **Parameter Name** | **Erase_End_Address** |
|---|---|
| **Description** | Erasing End address of the segment in the memory.<br>Range: [0x00000000 ; 0xFFFFFFFF]<br>Coded on 32 bits |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Origin** | EB |

| **Parameter Name** | **Partition_Type** |
|---|---|

| | |
|---|---|
| **Label** | Partition Type |
| **Description** | Define the partition type<br>▶ Application partition<br>▶ Calibration partition<br>▶ HSM partition<br>▶ Protected calibration partition - will not be erased<br>▶ Bootloader partition<br>▶ Flash Routines partition (here will be the erase/write routines of the flash driver, this partition shall be configured to RAM)<br>▶ Software Structure partition |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | PROG_APPLICATION_PARTITION |
| **Range** | PROG_FLASH_ROUTINES_PARTITION<br>PROG_APPLICATION_PARTITION<br>PROG_CALIBRATION_PARTITION<br>PROG_PROT_CALIBRATION_PARTITION<br>PROG_ESS_PARTITION<br>PROG_HSM_PARTITION |
| **Origin** | EB |

| **Parameter Name** | **Protected_Partition_ID** |
|---|---|
| **Label** | Protected Partition ID |
| **Description** | Define the partition ID for protected segment |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Range** | >=0x2 |
| **Origin** | EB |

| **Parameter Name** | **HSM_PartitionID** |
|---|---|
| **Label** | HSM Partition ID |
| **Description** | Define the BootLoader HSM Partition ID<br>The Partition ID needs to be defined |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 8 |
| **Range** | >=0x2 |
| **Origin** | EB |

| **Parameter Name** | **HSM_RAM_Buffer** |
|---|---|
| **Label** | HSM Buffer Size |
| **Description** | Define the BootLoader HSM Partition RAM buffer size |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 4000 |
| **Origin** | EB |

| **Parameter Name** | **PROG_HSM_Timeout** |
|---|---|
| **Label** | HSM TD Response Timeout |
| **Description** | Define the BootLoader HSM TD response Timeout in ms |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 10 |
| **Origin** | EB |

| **Parameter Name** | **ValidityCheck** |
|---|---|
| **Label** | Validity Check |
| **Description** | Check or not the status of this segment for the application validity check |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |
| **Origin** | EB |

### 4.5.1.11. Blocks

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |

| Containers included | | |
|---|---|---|
| SecureBoot | 1..1 | This container contains all configurations for Secure/Authenticated bootloader features. Configuration can be done only if Secure Boot feature or Authenticated Boot feature is activated under BM. |

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| First_Segment | 1..1 |
| Segment_Number | 1..1 |
| Block_Programming_Counter_Max | 1..1 |
| Block_Identifier | 1..1 |

| Parameter Name | First_Segment |
|---|---|
| **Description** | Reference to the first segment of the block |
| **Multiplicity** | 1..1 |
| **Type** | REFERENCE |
| **Origin** | EB |

| Parameter Name | Segment_Number |
|---|---|
| **Description** | Number of Segment in the block.<br>These represent the number of segment in the block based on the first configured segment<br>(segments of the same block shall be consecutive in segments configuration) |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Origin** | EB |

| Parameter Name | Block_Programming_Counter_Max |
|---|---|
| **Description** | Maximum number of allowed to program the block.<br>This represent the maximum number of times allowed to program the block<br>(If the counter is set to 0 then there will be no limit for programming) |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | INTEGER |
| **Origin** | EB |

| **Parameter Name** | **Block_Identifier** |
|---|---|
| **Description** | Identifier of the block. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Origin** | EB |

### 4.5.1.12. SecureBoot

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Verified_For_Secure_Boot | 1..1 |
| Blocker_for_Software_execution | 1..1 |
| Start_Address_Secure_Boot_Verification | 1..1 |
| Length_Secure_Boot_Verification | 1..1 |

| **Parameter Name** | **Verified_For_Secure_Boot** |
|---|---|
| **Label** | Verified in Secure Boot |
| **Description** | Specify if the Block shall be verified when Secure Boot is activated. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | Block won't be verified |
| **Range** | Block will be verified<br>Block won't be verified |
| **Origin** | EB |

| **Parameter Name** | **Blocker_for_Software_execution** |
|---|---|
| **Label** | Blocker for Software execution |

| | |
|---|---|
| **Description** | Specify if the Block will prevent the corresponding software to be executed in case of Secure Boot verification failure. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | Won't block software execution |
| **Range** | Will block software execution<br>Won't block software execution |
| **Origin** | EB |

| **Parameter Name** | **Start_Address_Secure_Boot_Verification** |
|---|---|
| **Label** | Start Address for the Secure Boot Verification |
| **Description** | Specify the Start Address of the area on which the Secure Boot verification will be done. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | <=4294967295 |
| **Origin** | EB |

| **Parameter Name** | **Length_Secure_Boot_Verification** |
|---|---|
| **Label** | Length of the Block area for the Secure Boot Verification |
| **Description** | Specify the length of the area on which the Secure Boot verification will be done. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | <=4294967295 |
| **Origin** | EB |

### 4.5.1.13. Memory

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Memory_Type | 1..1 |

| **Parameters included** | |
|---|---|
| Memory_Mode | 1..1 |
| Min_Value_To_Write | 1..1 |
| Addr_Offset | 1..1 |
| Erase_Value | 1..1 |

| **Parameter Name** | **Memory_Type** |
|---|---|
| **Label** | Memory Type |
| **Description** | Type of the memory.<br>▶ **FLASH**<br>▶ **FLASH_EXT**<br>▶ **RAM**<br>▶ **CUSTOM**<br>NOTE: Only one memory of each type can be defined! |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Range** | FLASH<br>FLASH_EXT<br>RAM<br>CUSTOM |
| **Origin** | EB |

| **Parameter Name** | **Memory_Mode** |
|---|---|
| **Label** | Memory Mode |
| **Description** | This entry allows to specify if the memory is access synchronously or asynchronously.<br>Please select between :<br>▶ asynchronous<br>▶ synchronous |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | synchronous |

| | |
|---|---|
| **Range** | asynchronous<br>synchronous |
| **Origin** | EB |

| **Parameter Name** | **Min_Value_To_Write** |
|---|---|
| **Label** | Minimum value to write |
| **Description** | Define the minimum size the memory driver could write at a time.<br>Range: [0x00000000 ; 0xFFFFFFFF].<br>Coded on 32 bits. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 8 |
| **Range** | >=8 |
| **Origin** | EB |

| **Parameter Name** | **Addr_Offset** |
|---|---|
| **Label** | Address Offset |
| **Description** | Define the start address offset for the memory.<br>In the case of the EXTERNAL FLASH and CUSTOM this offset is SUBSTRACTED from the received address.<br>In the case of the RAM and INTERNAL FLASH this offset is ADDED to the received address.<br>Range: [0x00000000 ; 0xFFFFFFFF].<br>Coded on 32 bits. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Origin** | EB |

| **Parameter Name** | **Erase_Value** |
|---|---|
| **Label** | Erase Value |
| **Description** | Define the value to set for each byte when the memory is erased |

| | |
|---|---|
| | Example: 0xFFu will affect the value 0xFF to every byte of the memory when erasure.<br>Range: [0x00 ; 0xFF].<br>Coded on 8 bits. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 255 |
| **Range** | >=0<br><=255 |
| **Origin** | EB |

### 4.5.1.14. GM

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| ProgCsmReferences | 1..1 | **Label:** ProgCsmReferences<br>Contains references to Csm configuration identifiers. |

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| PEC_Enable | 1..1 |
| MAX_PARTITION | 1..1 |
| MAX_REGION_ALLOWED | 1..1 |
| Security_Access_Seed_Length | 1..1 |
| ECU_ADDR | 1..1 |
| PSI_Programmed | 1..1 |
| PSI_Revoked | 1..1 |
| BCID | 1..1 |
| EcuId_Source | 1..1 |
| ECU_ID | 1..1 |
| Subject_Name | 1..1 |

| Parameters included | |
|---|---|
| ECU_Name | 1..1 |
| BOOT_Part_Number | 1..1 |
| BOOT_DLS | 1..1 |

| Parameter Name | PEC_Enable |
|---|---|
| Label | PEC Enable |
| Description | Define if the PEC is enable |
| Multiplicity | 1..1 |
| Type | BOOLEAN |
| Default value | true |
| Origin | EB |

| Parameter Name | MAX_PARTITION |
|---|---|
| Label | Maximum Partition |
| Description | Define the maximum partition supported by the ECU |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 2 |
| Origin | EB |

| Parameter Name | MAX_REGION_ALLOWED |
|---|---|
| Label | Maximum Number of regions |
| Description | Define the maximum number of region that is allowed in the software |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 2 |
| Range | >=2<br><255 |
| Origin | EB |

| Parameter Name | Security_Access_Seed_Length |
|---|---|
| Label | Length of the security seed |
| Description | Define the length of the seed to be able to return the correct size |
| Multiplicity | 1..1 |

| | |
|---|---|
| **Type** | INTEGER |
| **Default value** | 31 |
| **Origin** | EB |

| **Parameter Name** | **ECU_ADDR** |
|---|---|
| **Label** | Ecu Address |
| **Description** | Define the ECU address that will identify the ECU to then reprogramming tool |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0xEB |
| **Range** | >=0<br><255 |
| **Origin** | EB |

| **Parameter Name** | **PSI_Programmed** |
|---|---|
| **Label** | Programmed PSI Value |
| **Description** | Define PSI value when it is programmed<br>The value shall be enter as hexadecimal value without the 0x |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | A555A5AA5AAA5A55 |
| **Origin** | EB |

| **Parameter Name** | **PSI_Revoked** |
|---|---|
| **Label** | Revoked PSI Value |
| **Description** | Define PSI value when it is revoked<br>The value shall be enter as hexadecimal value without the 0x |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | 505245564F4B4544 |
| **Origin** | EB |

| **Parameter Name** | **BCID** |
|---|---|
| **Description** | This value shall contains the Bootloader Compatibility Identifier (BCID) |

| | |
|---|---|
| | This value is two bytes long |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 60395 |
| **Range** | <=65535 |
| **Origin** | EB |

| **Parameter Name** | **EcuId_Source** |
|---|---|
| **Label** | ECU ID Source |
| **Description** | This entry allows to specify if the ECU ID is coming from Tresos Configuration or from User Callback. If the User Callback option is used, the integrator shall implement the callback that will provide the ECU Id.<br>Please select between :<br>▶ Tresos Configuration<br>▶ User Callback |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | Tresos_Configuration |
| **Range** | Tresos_Configuration<br>User_Callback |
| **Origin** | EB |

| **Parameter Name** | **ECU_ID** |
|---|---|
| **Label** | ECU ID |
| **Description** | Define the ECU ID<br>The value shall be enter as hexadecimal value without the 0x<br>the parameter shall be 16 bytes long |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | 00000000000000000000000000000000 |
| **Origin** | EB |

| **Parameter Name** | **Subject_Name** |
|---|---|

| | |
|---|---|
| **Label** | ECU Subject Name |
| **Description** | Define the ECU subject name<br>The value shall be enter as hexadecimal value without the 0x<br>the parameter shall be 16 bytes long |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | 00112233445566778899001122334455 |
| **Origin** | EB |

| **Parameter Name** | **ECU_Name** |
|---|---|
| **Label** | ECU Name |
| **Description** | Define the ECU name<br>The name shall be set as a string with exactly 8 characters (space can be used) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | GM BOOT |
| **Origin** | EB |

| **Parameter Name** | **BOOT_Part_Number** |
|---|---|
| **Label** | BootLoader Part Number |
| **Description** | Define the BootLoader Part Number<br>The value shall be entered as hexadecimal value without the 0x<br>the parameter shall be 4 bytes long |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | 00112233 |
| **Origin** | EB |

| **Parameter Name** | **BOOT_DLS** |
|---|---|
| **Label** | BootLoader DLS |
| **Description** | Define the BootLoader DLS<br>The name shall be set as a string with exactly 2 characters (space cannot be used) |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | AB |
| **Origin** | EB |

### 4.5.1.15. ProgCsmReferences

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ProgCsmSignatureVerifyConfigId | 1..1 |
| ProgCsmHashConfigId | 1..1 |

| **Parameter Name** | **ProgCsmSignatureVerifyConfigId** | |
|---|---|---|
| **Label** | ProgCsmSignatureVerifyConfigId | |
| **Description** | Reference a *CsmSignatureVerify*<br>Dependencies:<br>▶ Reference shall be valid | |
| **Multiplicity** | 1..1 | |
| **Type** | CHOICE-REFERENCE | |
| **Configuration class** | **VariantPreCompile:** | VariantPreCompile |
| **Origin** | Elektrobit Automotive GmbH | |

| **Parameter Name** | **ProgCsmHashConfigId** | |
|---|---|---|
| **Label** | ProgCsmHashConfigId | |
| **Description** | Reference a *CsmHash*<br>Dependencies:<br>▶ Reference shall be valid | |
| **Multiplicity** | 1..1 | |
| **Type** | CHOICE-REFERENCE | |
| **Configuration class** | **VariantPreCompile:** | VariantPreCompile |
| **Origin** | Elektrobit Automotive GmbH | |

### 4.5.1.16. Security

| Containers included | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| ProgCsmReferences | 1..1 | **Label:** ProgCsmReferences<br>Contains references to Csm configuration identifiers. |
| NRC78_Transmission | 1..1 | **Label:** NRC78_Transmission<br>Contains configuration for the NRC78 transmission before SecurityAccess. |

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Secure_Checksum_computation | 1..1 |
| CHECKSUM_LENGTH | 1..1 |
| ProgCsmSecureConfigId | 1..1 |

| Parameter Name | Secure_Checksum_computation |
|---|---|
| **Label** | Checksum computation |
| **Description** | It allows the computation and writing of the checksum (Hash, MAC...) of logical blocks of Application used when Secure Boot or Authenticated Boot is activated. The verification of the checksum computed is activated when the parameter SECURE_AUTHENTICATED_BOOT is set either to Secure or to Authenticated in BM plugin. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| Parameter Name | CHECKSUM_LENGTH |
|---|---|
| **Label** | Checksum length |
| **Description** | Size of the checksum for Authenticated / Secure Boot feature in Bytes |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 16 |
| **Origin** | EB |

| **Parameter Name** | **ProgCsmSecureConfigId** |
|---|---|
| **Label** | ProgCsmSecureConfigId |
| **Description** | Reference a *CsmHash* or *CsmMacGenerate* for the Secure Boot checksum generation<br>Dependencies:<br>▶ Reference shall be valid |
| **Multiplicity** | 1..1 |
| **Type** | CHOICE-REFERENCE |
| **Configuration class** | **PreCompile:** VariantPreCompile |
| **Origin** | Elektrobit Automotive GmbH |

### 4.5.1.17. ProgCsmReferences

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ProgCsmSignatureVerifyConfigId | 1..1 |
| ProgCsmHashConfigId | 1..1 |

| **Parameter Name** | **ProgCsmSignatureVerifyConfigId** |
|---|---|
| **Label** | ProgCsmSignatureVerifyConfigId |
| **Description** | Reference a *CsmSignatureVerify*<br>Dependencies:<br>▶ Reference shall be valid |
| **Multiplicity** | 1..1 |
| **Type** | CHOICE-REFERENCE |
| **Configuration class** | **PreCompile:** VariantPreCompile |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ProgCsmHashConfigId** |
|---|---|
| **Label** | ProgCsmHashConfigId |
| **Description** | Reference a *CsmHash* |

| | |
|---|---|
| | Dependencies:<br>- Reference shall be valid |
| **Multiplicity** | 1..1 |
| **Type** | CHOICE-REFERENCE |
| **Configuration class** | **PreCompile:** VariantPreCompile |
| **Origin** | Elektrobit Automotive GmbH |

### 4.5.1.18. NRC78_Transmission

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Transmit_Nrc78_On_SecurityAccess | 1..1 |

| **Parameter Name** | **Transmit_Nrc78_On_SecurityAccess** |
|---|---|
| **Description** | Specify if an NRC78 response shall be sent before starting the Security Access.<br>- Case untick: NRC78 will not be sent |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |
| **Origin** | EB |

### 4.5.1.19. OemInd

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Erase_Mode | 1..1 |
| EraseALFI_Enable | 1..1 |
| CoherencyCheck_Enable | 1..1 |
| Application_Validity_Algo | 1..1 |
| Checksum_Algo | 1..1 |

| Parameters included | |
|---|---|
| Size_Of_FingerPrint | 1..1 |

| Parameter Name | Erase_Mode |
|---|---|
| Label | Erasing mode |
| Description | Specify how the erasing shall be performed.<br>▶ Case All: No information are provided in EraseMemory routine and all the configured segments will be erased on reception of the routine.<br>▶ Case Address: Erasing will be performed on the configured segment matching the address provided in EraseMemory routine<br>▶ Case LogicalBlock: Erasing will be performed on the configured segments associated (by configuration) to the logical block Id provided in EraseMemory routine |
| Multiplicity | 1..1 |
| Type | ENUMERATION |
| Default value | Address |
| Range | All<br>Address<br>LogicalBlock |
| Origin | EB |

| Parameter Name | EraseALFI_Enable |
|---|---|
| Label | Erase request ALFI Enable |
| Description | Enable the ALFI UDS field in EraseMemory request (Available only if erasing mode is by address) |
| Multiplicity | 1..1 |
| Type | BOOLEAN |
| Default value | false |
| Origin | EB |

| Parameter Name | CoherencyCheck_Enable |
|---|---|
| Label | Coherency check Enable |
| Description | Enable the coherency check (check programming dependencies). |
| Multiplicity | 1..1 |
| Type | BOOLEAN |

| | |
|---|---|
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Application_Validity_Algo** |
|---|---|
| **Label** | Application validity check and fingerprint: EB specific / Customer specific |
| **Description** | Specify if the Elektrobit algorithm shall be used for application validity management and fingerprint management or a customer specific one.<br>▶ Case EB: standard algorithm will be used<br>▶ Case Custom: Customer shall complete the PROG_IsValidApplication/PROG_CustomSetDownloadVerificationSuccess/PROG_InvalidateSection/PROG_EntryWriteFingerprint and managed the application status. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | EB |
| **Range** | EB<br>Custom |
| **Origin** | EB |

| **Parameter Name** | **Checksum_Algo** |
|---|---|
| **Label** | CRC algorithm: NO CRC / Signature / CRC16 CCITT/ CRC32 Ethernet / CRC32 International Standard 32-Bit CRC |
| **Description** | Specify which checksum algorithm shall be used<br>▶ Case NO CRC: No checksum needed.<br>▶ Case Signature: A cryptographic signature verification will be done (only possible if cryptographic libraries are used)<br>▶ Case CRC16 CCITT : Polynomial 0x1021 / Init value 0xFFFF / ReflectIn FALSE / ReflectOut FALSE / No XOR on Output<br>▶ Case CRC32 Ethernet: Polynomial 0x04C11DB7 / Init value 0xFFFFFFFF / ReflectIn TRUE / ReflectOut TRUE / XOR on Output 0xFFFFFFFF<br>▶ Case CRC32 International Standard 32-Bit CRC: Polynomial 0xEDB88320 / Init value 0xFFFFFFFF / ReflectIn FALSE / ReflectOut FALSE / XOR on Output 0xFFFFFFFF |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | CRC32 Ethernet |

| | |
|---|---|
| **Range** | NO CRC<br>Signature<br>CRC16<br>CRC32 Ethernet<br>CRC32 InternationalStandard |
| **Origin** | EB |

| **Parameter Name** | **Size_Of_FingerPrint** |
|---|---|
| **Description** | Define the size of the FingerPrint in Bytes.<br>The size can't be negative. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 16 |
| **Range** | >=0 |
| **Origin** | EB |

### 4.5.1.20. VAG

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Request_Download_Address_Mode | 1..1 |

| **Parameter Name** | **Request_Download_Address_Mode** |
|---|---|
| **Label** | Request Download Addressing Mode |
| **Description** | Specify the content of the request download start address<br>▶ Case Download by logical block: Only block ID will be sent in the request and the block cannot have more than one segment<br>▶ Case Download by logical block and segment: block ID and segment ID will be sent in the request and the block can have multiple segments |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | Download by logical block and segment |

| | |
|---|---|
| **Range** | Download by logical block<br>Download by logical block and segment |
| **Origin** | EB |

### 4.5.1.21. Decryption

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Enable_Decryption | 1..1 |

| **Parameter Name** | **Enable_Decryption** |
|---|---|
| **Description** | If enabled, a callback will be called on data reception allowing integration code to perform data decryption. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

## 4.5.2. Application programming interface (API)

### 4.5.2.1. Type definitions

#### 4.5.2.1.1. ptAPPL_START_ADDR

| | |
|---|---|
| **Purpose** | |
| **Type** | `void(*)(void)` |

#### 4.5.2.1.2. ptCompleteCompatibleCallOut

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Type** | `tProgCompleteStatus(*)(void)` |

### 4.5.2.1.3. ptSBL_StartUp_Code

| | |
|---|---|
| **Purpose** | |
| **Type** | `void(*)(u32 ulInfoSBL, u8 ubRxPduId)` |

### 4.5.2.1.4. tDataBlockType

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u32 ulStartAddress` | |
| | `u32 ulLength` | |
| | `u8 aubDigest` | |

### 4.5.2.1.5. tDataBufferType

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

### 4.5.2.1.6. tDataLength

| | |
|---|---|
| **Purpose** | |
| **Type** | `u32` |

### 4.5.2.1.7. tMultipleBuffReprogInfo

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u32 ulBufferedSizeOfData` | |
| | `u8 eResponsePending` | |
| | `u8 eBufferProcessing` | |

#### 4.5.2.1.8. tOperationType

| | |
|---|---|
| **Purpose** | |
| **Type** | u8 |

#### 4.5.2.1.9. tPageBuffer

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | u8 aubData | |
| | u16 uwOldDataLength | |
| | u16 uwNewDataLength | |

#### 4.5.2.1.10. tProgAccessType

| | |
|---|---|
| **Purpose** | |
| **Type** | u8 |

#### 4.5.2.1.11. tProgAddressType

| | |
|---|---|
| **Purpose** | |
| **Type** | u32 |

#### 4.5.2.1.12. tProgCompTimeoutStatus

| | |
|---|---|
| **Purpose** | |
| **Type** | u8 |

#### 4.5.2.1.13. tProgCompleteStatus

| | |
|---|---|
| **Purpose** | |
| **Type** | u32 |

#### 4.5.2.1.14. tProgDownloadType

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

#### 4.5.2.1.15. tProgMemIdx

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

#### 4.5.2.1.16. tProgMemMode

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

#### 4.5.2.1.17. tProgMemType

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

#### 4.5.2.1.18. tProgPECError

| | |
|---|---|
| **Purpose** | |
| **Type** | `u16` |

#### 4.5.2.1.19. tProgPartitionType

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

#### 4.5.2.1.20. tProgPsiValue

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Type** | `u8` |

### 4.5.2.1.21. tProgResCause

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

### 4.5.2.1.22. tProgSigBypass

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

### 4.5.2.1.23. tProgVerifAlgo

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

### 4.5.2.1.24. tProgVerificationInfo

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u32 ulAdd` | |
| | `u32 ulCnt` | |
| | `u32 ulVal` | |
| | `u16 uwVal` | |
| | `tProgVerifAlgo ubAlgo` | |
| | `u8 ubLogicalBlockId` | |

### 4.5.2.1.25. tRDParam

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u32 ulStartAddress` | |

| | | |
|---|---|---|
| | `u32 ulMemorySize` | |
| | `u16 uwBlockIdentifier` | |
| | `u8 ubSegmentId` | |
| | `u8 ubDataFormatId` | |

#### 4.5.2.1.26. tRegiontype

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u32 ulAddress` | |
| | `u32 ulSize` | |

#### 4.5.2.1.27. tReprogInfo

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u32 ulMemorySizeExpected` | |
| | `u32 ulReceivedSizeOfData` | |
| | `u32 ulTotalDecompDataWrite` | |
| | `u16 uwExpectedModuleId` | |
| | `u8 ubBlockSequenceCounter` | |
| | `u8 ubRDReceived` | |
| | `u8 ubCompRequired` | |

#### 4.5.2.1.28. tSegmentType

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u32 ulStartAddress` | |
| | `u32 ulSize` | |
| | `tRegiontype astRegion` | |
| | `u8 * pubRegionPtr` | |

| | | |
|---|---|---|
| | `u16 uwModuleId` | |
| | `u8 ubRegionNbr` | |
| | `u8 ubLogicalSegmentId` | |
| | `u8 ubLogicalBlockId` | |
| | `u8 ubSegmentId` | |
| | `u8 ubSegmentNbr` | |

#### 4.5.2.1.29. tWriteInfo

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u8 * pubDecompData` | |
| | `u32 ulAddressToWrite` | |
| | `u32 ulDataToBeWritten` | |
| | `u32 ulWrittenData` | |
| | `u32 ulWriteLength` | |
| | `u32 ulInDecompDataLength` | |
| | `u8 ubTDReceived` | |

#### 4.5.2.1.30. t_PROG_fctptr

| | |
|---|---|
| **Purpose** | |
| **Type** | `void *(*)(void)` |

#### 4.5.2.1.31. t_secondary_bootloader_interface

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u32 software_version` | |
| | `u8 referenceString` | |
| | `void * ptr_function` | |
| | `u32 ulSBLValidityFlagAddr` | |

#### 4.5.2.1.32. tpulGetAddress

| | |
|---|---|
| **Purpose** | |
| **Type** | `void *(*)(u8, u32)` |

#### 4.5.2.1.33. tpulVerifySectionCrc

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8(*)(void)` |

#### 4.5.2.1.34. tpulinvalidateSection

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8(*)(u32)` |

#### 4.5.2.1.35. tpulisValidApplication

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8(*)(void)` |

#### 4.5.2.1.36. tpulskipPage

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8(*)(u32 *)` |

### 4.5.2.2. Macro constants

#### 4.5.2.2.1. PROG_1ST_PROGRAMMING_ERASE_CHECK

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.2. PROG_APPLICATION_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.3. PROG_BLOCK_ERASE_CHECK

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.4. PROG_BOOTLOADER_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x05U |

#### 4.5.2.2.5. PROG_BOOT_DLS_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | 2U |

#### 4.5.2.2.6. PROG_BOOT_MAX_PROT_PARTITIONS

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"num:dectoint($NBR_PROT_CAL)"!]U |

#### 4.5.2.2.7. PROG_BOOT_MODULE_ID_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | 1U |

#### 4.5.2.2.8. PROG_BOOT_NB_MODULE_SIZE

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 1U |

#### 4.5.2.2.9. PROG_BOOT_NUMBER_OF_MODULES

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.10. PROG_BOOT_PART_NUMBER_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | 4U |

#### 4.5.2.2.11. PROG_BOOT_PRIMARY_MICRO_ID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x47U |

#### 4.5.2.2.12. PROG_BOOT_PROT_CALIB_NUMBER_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | 1U |

#### 4.5.2.2.13. PROG_BOOT_PROT_CALIB_PARTITION_ID_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | 1U |

#### 4.5.2.2.14. PROG_CALIBRATION_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.15. PROG_COMPLETECOMPATIBLE_END

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.16. PROG_COMPLETECOMPATIBLE_ERROR

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.17. PROG_COMPLETECOMPATIBLE_START

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.18. PROG_DIGEST_LENGTH

| | |
|---|---|
| **Purpose** | |
| **Value** | 32U |

#### 4.5.2.2.19. PROG_DISABLED_ERASE_CHECK

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.5.2.2.20. PROG_DOWNLOAD_BY_ADDR

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.21. PROG_DOWNLOAD_BY_LOGICAL_BLOCK

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 0x02U |

#### 4.5.2.2.22. PROG_DOWNLOAD_BY_LOGICAL_BLOCK_SEGMENT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.23. PROG_ECU_ID_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | 16U |

#### 4.5.2.2.24. PROG_ECU_NAME_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | 8U |

#### 4.5.2.2.25. PROG_ERR_APP_NBID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0016U |

#### 4.5.2.2.26. PROG_ERR_BCID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0010U |

#### 4.5.2.2.27. PROG_ERR_CAL_REGION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x001AU |

#### 4.5.2.2.28. PROG_ERR_CCID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0011U |

#### 4.5.2.2.29. PROG_ERR_CERT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0019U |

#### 4.5.2.2.30. PROG_ERR_COMPRESSION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x000BU |

#### 4.5.2.2.31. PROG_ERR_DATA_TYPE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x000AU |

#### 4.5.2.2.32. PROG_ERR_ECU_ID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0013U |

#### 4.5.2.2.33. PROG_ERR_ECU_NAME

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0012U |

#### 4.5.2.2.34. PROG_ERR_ERASE_CAL

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 0x0007U |

#### 4.5.2.2.35. PROG_ERR_ERASE_SW

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0004U |

#### 4.5.2.2.36. PROG_ERR_FLASH_WRITE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x000EU |

#### 4.5.2.2.37. PROG_ERR_GET_APP_INFO

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0005U |

#### 4.5.2.2.38. PROG_ERR_GET_CAL_INFO

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0008U |

#### 4.5.2.2.39. PROG_ERR_KEY_NBID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0018U |

#### 4.5.2.2.40. PROG_ERR_LENGTH_EXCEEDED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x000CU |

#### 4.5.2.2.41. PROG_ERR_MD

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x001BU |

#### 4.5.2.2.42. PROG_ERR_MODULE_ID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x000FU |

#### 4.5.2.2.43. PROG_ERR_MORE_DATA_EXPECTED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x000DU |

#### 4.5.2.2.44. PROG_ERR_MSG_OUT_OF_SEQUENCE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x001DU |

#### 4.5.2.2.45. PROG_ERR_PARTITION_ID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0001U |

#### 4.5.2.2.46. PROG_ERR_PER_DATA_TX_NOT_ALLOW

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0037U |

#### 4.5.2.2.47. PROG_ERR_PROTECTEDCAL_NOT_DEFINED

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 0x0035U |

#### 4.5.2.2.48. PROG_ERR_REVOKE_CAL

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0006U |

#### 4.5.2.2.49. PROG_ERR_REVOKE_SW

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0003U |

#### 4.5.2.2.50. PROG_ERR_ROOT_SIGNATURE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x001CU |

#### 4.5.2.2.51. PROG_ERR_SBA_CERT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x08U |

#### 4.5.2.2.52. PROG_ERR_SBA_ECU_ID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.53. PROG_ERR_SBA_ECU_NAME

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.54. PROG_ERR_SBA_SIGNATURE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x04U |

#### 4.5.2.2.55. PROG_ERR_SIGNATURE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0015U |

#### 4.5.2.2.56. PROG_ERR_SUBJECT_NAME

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0017U |

#### 4.5.2.2.57. PROG_ERR_SW_NOT_PRESENT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0002U |

#### 4.5.2.2.58. PROG_ERR_SW_REGION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0014U |

#### 4.5.2.2.59. PROG_ERR_UNDEFINED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0020U |

#### 4.5.2.2.60. PROG_ERR_UPDATE_PSI

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 0x0009U |

#### 4.5.2.2.61. PROG_ESS_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x07U |

#### 4.5.2.2.62. PROG_E_BUSY

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.63. PROG_E_BYPASS

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x04U |

#### 4.5.2.2.64. PROG_E_CHECK_FAILED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.65. PROG_E_COHCHK_CORRECT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.5.2.2.66. PROG_E_COHCHK_INCORRECT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.67. PROG_E_COHCHK_INCORRECT_OTHER

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x04U |

#### 4.5.2.2.68. PROG_E_COHCHK_INCORRECT_SW_HW

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.69. PROG_E_COHCHK_INCORRECT_SW_SW

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.70. PROG_E_COHPRECHK_CORRECT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.5.2.2.71. PROG_E_COHPRECHK_INCORRECT_HW_SW

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.72. PROG_E_COHPRECHK_INCORRECT_SW_SW

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.73. PROG_E_COHPRECHK_INTERNAL_ERROR

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 0x01U |

#### 4.5.2.2.74. PROG_E_ERASED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.75. PROG_E_NOT_ERASED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.5.2.2.76. PROG_E_NOT_OK

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.77. PROG_E_OK

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.5.2.2.78. PROG_E_RFS_DRIVER_FAIL

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x05U |

#### 4.5.2.2.79. PROG_E_RFS_VERSION_FAIL

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x06U |

#### 4.5.2.2.80. PROG_FALSE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0U |

#### 4.5.2.2.81. PROG_FLASH_ROUTINES_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x06U |

#### 4.5.2.2.82. PROG_HSM_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x08U |

#### 4.5.2.2.83. PROG_MAX_LENGTH_CHECKMEMORY_ANSWER

| | |
|---|---|
| **Purpose** | |
| **Value** | 6U |

#### 4.5.2.2.84. PROG_MAX_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"num:dectoint(GM/MAX_PARTITION)"!]U |

#### 4.5.2.2.85. PROG_MAX_RD_PER_BLOCK

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"num:dectoint(DownloadVerification/MaxNumberOfRDPerBlock)"!]U |

#### 4.5.2.2.86. PROG_MAX_REGION_ALLOWED

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | [!"num:dectoint(GM/MAX_REGION_ALLOWED)"!]U |

#### 4.5.2.2.87. PROG_MEMORY_ASYNCHRONOUS

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.88. PROG_MEMORY_NB

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"$MEMORY_MAX"!]U |

#### 4.5.2.2.89. PROG_MEMORY_NOTUSED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.90. PROG_MEMORY_SYNCHRONOUS

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.91. PROG_MEM_ACCESS_TYPE_NONE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.5.2.2.92. PROG_MEM_ACCESS_TYPE_READ

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.93. PROG_MEM_ACCESS_TYPE_READ_WRITE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.94. PROG_MEM_ACCESS_TYPE_WRITE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.95. PROG_MEM_OPERATION_TYPE_ERASE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.96. PROG_MEM_OPERATION_TYPE_READ

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.97. PROG_MEM_OPERATION_TYPE_WRITE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.98. PROG_MEM_TYPE_CUSTOM

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x05U |

#### 4.5.2.2.99. PROG_MEM_TYPE_EEPROM

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 0x01U |

#### 4.5.2.2.100. PROG_MEM_TYPE_FLASH

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.5.2.2.101. PROG_MEM_TYPE_FLASH_EXT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x04U |

#### 4.5.2.2.102. PROG_MEM_TYPE_INIT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0xFFU |

#### 4.5.2.2.103. PROG_MEM_TYPE_RAM

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.104. PROG_MEM_TYPE_SCRATCHPAD

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.105. PROG_MIN_VAL_TO_WRITE

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"$MIN_VAL_TO_WRITE_FOR_ALL_MEMORIES"!]U |

#### 4.5.2.2.106. PROG_PEC_NO_ERROR

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0000U |

#### 4.5.2.2.107. PROG_PROT_CALIBRATION_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.5.2.2.108. PROG_PSI_INVALID

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.5.2.2.109. PROG_PSI_PROGRAMMED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.5.2.2.110. PROG_PSI_REVOKED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.111. PROG_RESET_CAUSE_DSC01

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.5.2.2.112. PROG_RESET_CAUSE_DSC02

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 0x02U |

### 4.5.2.2.113. PROG_RESET_CAUSE_ER

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

### 4.5.2.2.114. PROG_RESET_CAUSE_S3_TIMEOUT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

### 4.5.2.2.115. PROG_SBA_OK

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x80U |

### 4.5.2.2.116. PROG_SBA_PARTITION

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x04U |

### 4.5.2.2.117. PROG_SEGMENT_NB

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"$SEGMENT_MAX"!]U |

### 4.5.2.2.118. PROG_SUBJECT_NAME_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | 16U |

#### 4.5.2.2.119. PROG_TRUE

| **Purpose** | |
|---|---|
| **Value** | 1U |

#### 4.5.2.2.120. PROG_VERIFY_CRC

| **Purpose** | |
|---|---|
| **Value** | 0x00U |

#### 4.5.2.2.121. PROG_VERIFY_HASH

| **Purpose** | |
|---|---|
| **Value** | 0x02U |

#### 4.5.2.2.122. PROG_VERIFY_SIGNATURE

| **Purpose** | |
|---|---|
| **Value** | 0x01U |

### 4.5.2.3. Objects

#### 4.5.2.3.1. m_aubKeyData

| **Purpose** | |
|---|---|
| Type | `const u8` |

#### 4.5.2.3.2. m_aubPROGPublicModulus

| **Purpose** | |
|---|---|
| Type | `const u8` |

### 4.5.2.3.3. m_ubCheckMemoryStatus

| **Purpose** | |
|---|---|
| Type | `u8` |

### 4.5.2.3.4. m_ubFailedCheckMemoryCount

| **Purpose** | |
|---|---|
| Type | `u8` |

### 4.5.2.3.5. m_ubSimulateProgSessionWithResponse

| **Purpose** | |
|---|---|
| Type | `u8` |

### 4.5.2.3.6. m_ulMacKeyKength

| **Purpose** | |
|---|---|
| Type | `const u32` |

### 4.5.2.3.7. m_ulPROGPublicExponent

| **Purpose** | |
|---|---|
| Type | `const u32` |

### 4.5.2.3.8. tFirstCheckMemoryAnswerInfo

| **Purpose** | |
|---|---|
| Type | |

### 4.5.2.3.9. ubDiagStatus

| **Purpose** | |
|---|---|

| | |
|---|---|
| Type | u8 |

#### 4.5.2.3.10. uwLength

| **Purpose** | |
|---|---|
| Type | u16 |

### 4.5.2.4. Functions

#### 4.5.2.4.1. PROG_ActiveSBL

| **Purpose** | UDS callback for ActiveSBL. | |
|---|---|---|
| **Synopsis** | tUdsStatus **PROG_ActiveSBL** ( u16 * puwLen , u8 * aubUdsData ); | |
| **Parameters (in,out)** | puwLen | pointer on data length |
| | aubUdsData | pointer on data |
| **Return Value** | UDS status | |
| | UDS_ACK | positive response |
| | UDS_NRC_xxx | negative response |
| | UDS_NRC_31 | Input pointer parameters NULL value |
| **Description** | This function handles the ActiveSBL UDS request. It shall be configured in UDS Tresos Studio plugin for the request RC (0x31 0x01 0x03 0x01) with this exact name and with Callback_Origin set to OTHER. | |

#### 4.5.2.4.2. PROG_ActiveSBL_Check

| **Purpose** | Check if SBL is valid and compatible. | |
|---|---|---|
| **Synopsis** | tProgStatus **PROG_ActiveSBL_Check** ( void ); | |
| **Return Value** | Result of check | |
| | PROG_E_OK | SBL is valid and compatible with PBL |
| | PROG_E_NOT_OK | SBL is not valid and/or incompatible with PBL |

### 4.5.2.4.3. PROG_AnswerSuccesiveCheckMemoryRequests

| | |
|---|---|
| **Purpose** | Called in CHECK_MEMORY_FINISH state. |
| **Synopsis** | `void PROG_AnswerSuccesiveCheckMemoryRequests ( void );` |

### 4.5.2.4.4. PROG_AutoControl

| | | |
|---|---|---|
| **Purpose** | UDS callback for AutoControl. | |
| **Synopsis** | `tUdsStatus PROG_AutoControl ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the AutoControl UDS request. It shall be configured in UDS Tresos Studio plugin for the request RC of autocontrol of the application with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.5. PROG_AutoControl_Process

| | |
|---|---|
| **Purpose** | manage asynchronous autocontrol |
| **Synopsis** | `void PROG_AutoControl_Process ( void );` |

### 4.5.2.4.6. PROG_BckdManage

| | |
|---|---|
| **Purpose** | Manage function to be called as fast as possible to perform background actions. |
| **Synopsis** | `void PROG_BckdManage ( void );` |

### 4.5.2.4.7. PROG_CRC

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `tProgStatus PROG_CRC ( void );` |

| | | |
|---|---|---|
| **Return Value** | | |

### 4.5.2.4.8. PROG_CRC_Compare

| | | |
|---|---|---|
| **Purpose** | Compare the expected and the calculated CRCs. | |
| **Synopsis** | `tProgStatus` **`PROG_CRC_Compare`** `( void );` | |
| **Return Value** | Result of treatment | |
| | `PROG_E_OK` | Compare OK |
| | `PROG_E_BUSY` | Compare in progress |
| | `PROG_E_NOT_OK` | Compare finished on error |

### 4.5.2.4.9. PROG_CalcCrc16

| | | |
|---|---|---|
| **Purpose** | Called to calculate CRC. | |
| **Synopsis** | `void` **`PROG_CalcCrc16`** `( const u8 * aubCrcData , u32 ulReadLength , u16 * uwCrcValue );` | |
| **Parameters (in)** | `aubCrcData` | Data to add in the CRC calculation |
| | `ulReadLength` | Data length to add in the CRC calculation |
| **Parameters (out)** | `uwCrcValue` | Pointer to variable where to set the CRC result |

### 4.5.2.4.10. PROG_CalcCrc32

| | | |
|---|---|---|
| **Purpose** | Called to calculate CRC. | |
| **Synopsis** | `void` **`PROG_CalcCrc32`** `( const u8 * aubCrcData , u32 ulReadLength , u32 * ulCrcValue );` | |
| **Parameters (in)** | `aubCrcData` | Data to add in the CRC calculation |
| | `ulReadLength` | Data length to add in the CRC calculation |
| **Parameters (out)** | `ulCrcValue` | Pointer to variable where to set the CRC result |

### 4.5.2.4.11. PROG_CheckDecompHeaderStatus

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Synopsis** | `void PROG_CheckDecompHeaderStatus ( void );` |

### 4.5.2.4.12. PROG_CheckMemory

| | | |
|---|---|---|
| **Purpose** | UDS callback for CheckMemory. | |
| **Synopsis** | `tUdsStatus PROG_CheckMemory ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the CheckMemory UDS request. It shall be configured in UDS Tresos Studio plugin for the request XXXXXX with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.13. PROG_CheckPartialSegmentListCrc

| | |
|---|---|
| **Purpose** | This API allow to perform the partial software CRC computation over all programmed segment. |
| **Synopsis** | `void PROG_CheckPartialSegmentListCrc ( void );` |

### 4.5.2.4.14. PROG_CheckPartialSwCvnStatus

| | | |
|---|---|---|
| **Purpose** | This API allow to perform the partial software CVN check. | |
| **Synopsis** | `tProgStatus PROG_CheckPartialSwCvnStatus ( tProgStatus eProgStatus );` | |
| **Return Value** | | |

### 4.5.2.4.15. PROG_CheckProgRequest

| | |
|---|---|
| **Purpose** | API that check if a programming request has been received by the application. |

| | | |
|---|---|---|
| **Synopsis** | `u8 PROG_CheckProgRequest ( void );` | |
| **Return Value** | Result of check | |
| | `PROG_BOOT_REPROG` | Reprogramming request has been received |
| | `PROG_BOOT_NO_REPROG` | No reprogramming request received |
| **Description** | Callback is called: at Bootloader startup to know if a programming request has been received in Application<br>Callback shall implement: get information from application if a programming request has been received (e.g: read a flag from noinit RAM shared between Bootloader and Application) | |

### 4.5.2.4.16. PROG_CheckProgrammingCounter

| | | |
|---|---|---|
| **Purpose** | API for checking the programming counter. | |
| **Synopsis** | `tProgStatus PROG_CheckProgrammingCounter ( u8 ubBlockId );` | |
| **Parameters (in)** | `ubBlockId` | BlockID |
| **Return Value** | | |

### 4.5.2.4.17. PROG_CheckProgrammingDependencies

| | | |
|---|---|---|
| **Purpose** | UDS callback for CheckProgrammingDependencies. | |
| **Synopsis** | `tUdsStatus PROG_CheckProgrammingDependencies ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the CheckProgrammingDependencies UDS request. It shall be configured in UDS Tresos Studio plugin for the request RC CheckProgrammingDependencies (0x31 0x01 0xFF 0x01) with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.18. PROG_CheckProgrammingPreCondition

| | |
|---|---|
| **Purpose** | Function providing the programming pre-conditions check status. |
| **Synopsis** | `tProgStatus` **`PROG_CheckProgrammingPreCondition`** `( void );` |
| **Return Value** | Programming Pre-conditions check Status |

### 4.5.2.4.19. PROG_CheckProgrammingRequest

| | |
|---|---|
| **Purpose** | Check programming request. |
| **Synopsis** | `tProgBoolean` **`PROG_CheckProgrammingRequest`** `( void );` |
| **Return Value** | Programming Request Check Status |

### 4.5.2.4.20. PROG_CheckValidAppl

| | | |
|---|---|---|
| **Purpose** | | |
| **Synopsis** | `tProgBoolean` **`PROG_CheckValidAppl`** `( void );` | |
| **Return Value** | | |

### 4.5.2.4.21. PROG_Check_Prg_Dep_Check

| | | |
|---|---|---|
| **Purpose** | Called to verify if CheckProgDependencies routine can be executed. | |
| **Synopsis** | `tProgStatus` **`PROG_Check_Prg_Dep_Check`** `( void );` | |
| **Return Value** | Result of check | |
| | `PROG_E_OK` | check is allowed |
| | `PROG_E_NOT_OK` | check is not allowed |

### 4.5.2.4.22. PROG_CloseProgrammingSession

| | | |
|---|---|---|
| **Purpose** | request to close the programming session | |
| **Synopsis** | `void` **`PROG_CloseProgrammingSession`**<br>`( tUdsChangeReason eUdsChangeReason );` | |
| **Parameters (in)** | `eUdsChangeReason` | reason why the close is requested |
| **Description** | This function is called to request the close of the programming session | |

### 4.5.2.4.23. PROG_CommunicationControl

| | | |
|---|---|---|
| **Purpose** | UDS callback for CommunicationControl. | |
| **Synopsis** | `tUdsStatus` **`PROG_CommunicationControl`**<br>`( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the CommunicationControl UDS request. It shall be configured in UDS Tresos Studio plugin for the request CC (0x23) with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.24. PROG_ComputeMessageDigest

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void` **`PROG_ComputeMessageDigest`** `( void );` |

### 4.5.2.4.25. PROG_ControlDTCSetting

| | | |
|---|---|---|
| **Purpose** | UDS callback for ControlDTCSetting. | |
| **Synopsis** | `tUdsStatus` **`PROG_ControlDTCSetting`**<br>`( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the ControlDTCSetting UDS request. It shall be configured in UDS Tresos Studio plugin for the request CDTS (0x14) with this exact name and with Callback_Origin set to OTHER. | |

#### 4.5.2.4.26. PROG_CopySBATicket

| | | |
|---|---|---|
| **Purpose** | Copy the SBA ticket to the provided RAM buffer. | |
| **Synopsis** | `tProgStatus` **`PROG_CopySBATicket`** `( u8 * pubRamBuffer );` | |
| **Parameters (out)** | `pubRamBuffer` | pointer to a RAM buffer where to copy the SBA ticket |
| **Return Value** | Result | |
| | `PROG_E_OK` | Copy ok |
| | `PROG_E_NOT_OK` | Copy failed |
| **Description** | Callback is called: On Bootloader startup during SBA check<br>Callback shall implement: the reading from non volatile memory of the SBA ticket (822 bytes long, starting with the data type) | |

#### 4.5.2.4.27. PROG_CustCheckProgPrecond

| | | |
|---|---|---|
| **Purpose** | Check if all the programming pre-conditions are met. | |
| **Synopsis** | `tProgStatus` **`PROG_CustCheckProgPrecond`** `( void );` | |
| **Return Value** | state | |
| | `PROG_E_OK` | All the programming pre-conditions are met |
| | `PROG_E_NOT_OK` | At least one programming pre-condition is NOT met |
| **Description** | Callback is called: On Programming precondition check<br>Callback shall implement: programming precondition check. Inform Bootloader if the ECU is in a state where programming can be performed. If condition are not correct programming will be rejected by Bootloader and negative response will be sent on the network. | |

#### 4.5.2.4.28. PROG_CustCheckProgPrecondList

| | |
|---|---|
| **Purpose** | Check if all the programming pre-conditions are met. |

| | | |
|---|---|---|
| **Synopsis** | `void PROG_CustCheckProgPrecondList ( u8 * pubProgrammingConditionNumber , u8 * paubConditionList );` | |
| **Parameters (out)** | `pubProgrammingConditionNumber` | Number of failed Programming Conditions that shall be returned in the response |
| | `paubConditionList` | List of failed conditions |
| **Description** | Callback is called: On Programming precondition check<br>Callback shall implement: programming precondition check. Inform Bootloader if the ECU is in a state where programming can be performed. If condition are not correct programming will be rejected by Bootloader and negative response will be sent on the network. | |

### 4.5.2.4.29. PROG_CustomCalcInactiveBankReadAddr

| | | |
|---|---|---|
| **Purpose** | Calculate the read address on inactive memory bank. | |
| **Synopsis** | `u32 PROG_CustomCalcInactiveBankReadAddr ( u32 ulAddr );` | |
| **Parameters (in)** | `ulAddr` | Address on active memory bank |
| **Return Value** | Calculated address on inactive bank | |

### 4.5.2.4.30. PROG_CustomCalcInactiveBankWriteAddr

| | | |
|---|---|---|
| **Purpose** | Calculate the write address on inactive memory bank. | |
| **Synopsis** | `u32 PROG_CustomCalcInactiveBankWriteAddr ( u32 ulAddr );` | |
| **Parameters (in)** | `ulAddr` | Write address on active memory bank |
| **Return Value** | Calculated address on inactive memory bank | |

### 4.5.2.4.31. PROG_CustomCheckCertificateVerification

| | | |
|---|---|---|
| **Purpose** | Get the result of the certificate verification. | |
| **Synopsis** | `tProgStatus PROG_CustomCheckCertificateVerification ( void );` | |
| **Return Value** | eProgStatus success of the certificate verififcation | |
| | `PROG_E_OK` | Certificate verification passed |
| | `PROG_E_NOT_OK` | Certificate verification failed |
| **Description** | Callback is called: On checkMemory request | |

| | |
|---|---|
| | Callback shall implement: the certificate verification or its result |

### 4.5.2.4.32. PROG_CustomCheckCompatibilityId

| | | |
|---|---|---|
| **Purpose** | Callback used to check the CompatibilityId. | |
| **Synopsis** | `tProgStatus` **`PROG_CustomCheckCompatibilityId`** `( u8 * pubCompatibilityIdAddress , u8 ubLen );` | |
| **Parameters (in)** | `pubCompatibilityIdAddress` | CompatibilityId address |
| | `ubLen` | CompatibilityId length |
| **Return Value** | return status | |
| | `PROG_E_OK` | |
| | `PROG_E_NOT_OK` | |

### 4.5.2.4.33. PROG_CustomCheckRollbackId

| | | |
|---|---|---|
| **Purpose** | Callback used to check the RollbackId. | |
| **Synopsis** | `tProgStatus` **`PROG_CustomCheckRollbackId`** `( u8 * pubRollbackIdAddress , u8 ubLen );` | |
| **Parameters (in)** | `pubRollbackIdAddress` | RollbackId address |
| | `ubLen` | RollbackId length |
| **Return Value** | return status | |
| | `PROG_E_OK` | |
| | `PROG_E_NOT_OK` | |

### 4.5.2.4.34. PROG_CustomCheckSigningInfo

| | | |
|---|---|---|
| **Purpose** | Callback used to check the SigningInfo (SigningName and SigningKeyIdentifier). | |
| **Synopsis** | `tProgStatus` **`PROG_CustomCheckSigningInfo`** `( u8 * pubSigningInfoAddress , u8 ubLen );` | |
| **Parameters (in)** | `pubSigningInfoAddress` | SigningInfo address |
| | `ubLen` | SigningInfo length |
| **Return Value** | return status | |

| | | |
|---|---|---|
| | PROG_E_OK | |
| | PROG_E_NOT_OK | |

### 4.5.2.4.35. PROG_CustomCheckTargetName

| | | |
|---|---|---|
| **Purpose** | Callback used to check the TargetName. | |
| **Synopsis** | tProgStatus **PROG_CustomCheckTargetName** ( u8 * pubTargetNameAddress , u8 ubLen ); | |
| **Parameters (in)** | pubTargetNameAddress | TargetName address |
| | ubLen | TargetName length |
| **Return Value** | return status | |
| | PROG_E_OK | |
| | PROG_E_NOT_OK | |

### 4.5.2.4.36. PROG_CustomCheckUuid

| | | |
|---|---|---|
| **Purpose** | Callback used to check the Uuid. | |
| **Synopsis** | tProgStatus **PROG_CustomCheckUuid** ( u8 * pubUuidAddress , u8 ubLen ); | |
| **Parameters (in)** | pubUuidAddress | Uuid address |
| | ubLen | Uuid length |
| **Return Value** | return status | |
| | PROG_E_OK | |
| | PROG_E_NOT_OK | |

### 4.5.2.4.37. PROG_CustomChecksumCalc

| | | |
|---|---|---|
| **Purpose** | Get result of checksum calculation. | |
| **Synopsis** | tProgStatus **PROG_CustomChecksumCalc** ( u16 * puwCalculatedCks ); | |
| **Parameters (out)** | puwCalculatedCks | pointer to calculated checksum |
| **Return Value** | state | |
| | PROG_E_OK | Calculation finished successfully |
| | PROG_E_BUSY | Calculation in progress |

| | | |
|---|---|---|
| | PROG_E_NOT_OK | Calculation finished on error |
| **Description** | Callback is called: To get result of a checksum calculation, after PROG_CustomStartChecksumCalc/PROG_CustomUpdateChecksumCalc calls<br>Callback shall implement: Provide result of checksum calculation. Checksum calculation is customer specific (Checksum, CRC16,...) | |

### 4.5.2.4.38. PROG_CustomCoherencyCheck

| | | |
|---|---|---|
| **Purpose** | This API is called to do the coherency check treatment. | |
| **Synopsis** | `tProgStatus` **`PROG_CustomCoherencyCheck`** `( tProgCohChkResult * eCohChkResult );` | |
| **Parameters (out)** | eCohChkResult | The result of the coherency check. Can be: 0-correct, 1-incorrect, 2-incorrect error SW-HW, 3-incorrect error SW-SW, 4-incorrect other error |
| **Return Value** | Coherency Check result | |
| | PROG_E_OK | when the check has finished |
| | PROG_E_BUSY | if the check is on going |
| **Description** | Callback is called: on the coherency check request reception<br>Callback shall implement: the algorithm performing the coherency check of the previously programmed blocks (E.g.: checking of blocks versions compatibility). | |

### 4.5.2.4.39. PROG_CustomCvnVerification

| | | |
|---|---|---|
| **Purpose** | Callback for CVN check. | |
| **Synopsis** | `tProgStatus` **`PROG_CustomCvnVerification`** `( u8 ubLogicalBlockId , const u8 * paubExpectedCvn );` | |
| **Parameters (in)** | ubLogicalBlockId | Block identifier value |
| | paubExpectedCvn | CVN value |
| **Return Value** | state | |
| | PROG_E_OK | Treatment finish successfully |
| | PROG_E_BUSY | Treatment is in progress |
| | PROG_E_NOT_OK | Error happened during treatment |

| | |
|---|---|
| **Description** | Callback is called: on reception of verify partial sw<br>Callback shall implement: the check of CVN value |

### 4.5.2.4.40. PROG_CustomCvnVerificationStatus

| | | |
|---|---|---|
| **Purpose** | Callback for CVN check. | |
| **Synopsis** | `tProgStatus` **`PROG_CustomCvnVerificationStatus`** `( void );` | |
| **Return Value** | state | |
| | `PROG_E_OK` | Treatment finish successfully |
| | `PROG_E_BUSY` | Treatment is in progress |
| | `PROG_E_NOT_OK` | Error happened during treatment |
| **Description** | Callback is called: on reception of verify partial sw<br>Callback shall implement: the return of CVN status check updated in the verification callback | |

### 4.5.2.4.41. PROG_CustomDecryptData

| | | |
|---|---|---|
| **Purpose** | Callback that shall request data decryption before writing them to memory Callback is called: receiving a TransferData before the decompression (if activated). | |
| **Synopsis** | `tProgStatus` **`PROG_CustomDecryptData`** `( u8 ubEncryptionMethod , u8 * pubData , u16 DataLength );` | |
| **Parameters (in)** | `ubEncryptionMethod` | Encrypting method indicator (from RequestDownload dataFormatIdentifier field) |
| | `DataLength` | received data length |
| **Parameters (in,out)** | `pubData` | received data pointer (points to the encrypted data and callback implementation shall copy decrypted data at the same location than the encrypted one.) |
| **Return Value** | Result of the decryption | |
| | `PROG_E_OK` | Decryption finish successfully |
| | `PROG_E_NOT_OK` | Error happened during decryption |
| **Description** | Callback shall implement: If needed, it shall decrypt the received data according to the EncryptingMethod. | |

### 4.5.2.4.42. PROG_CustomDownloadNotification

| | | |
|---|---|---|
| **Purpose** | Notification of a download event. | |
| **Synopsis** | `tProgStatus` **`PROG_CustomDownloadNotification`** `( u32 ulStartAddress , u32 ulMemorySize );` | |
| **Parameters (in)** | `ulStartAddress` | received start address value |
| | `ulMemorySize` | received memory size value |
| **Return Value** | eProgStatus success of the operation | |
| | `PROG_E_OK` | |
| | `PROG_E_NOT_OK` | |
| **Description** | Callback is called: On reception of RequestDownload routine<br>Callback shall implement: provide information that the Flash Memory was programmed and should be erased before a new writing and customer specific implementation on download event | |

### 4.5.2.4.43. PROG_CustomGetAsymPublicKey

| | | |
|---|---|---|
| **Purpose** | Get the public key modulus and exponent when using asymmetric cryptography. Used in SA_InitCrypto. | |
| **Synopsis** | `void` **`PROG_CustomGetAsymPublicKey`** `( const u8 ** paubPublicModulus , u32 * pulPublicExponent );` | |
| **Parameters (out)** | `paubPublicModulus` | Pointer to asymmetric cryptography public key modulus array |
| | `pulPublicExponent` | Pointer to asymmetric cryptography public key exponent |

### 4.5.2.4.44. PROG_CustomGetComputedBootloaderChecksum

| | |
|---|---|
| **Purpose** | This API is called to get the computed Bootloader checksum stored in non-volatile memory. |
| **Synopsis** | `void` **`PROG_CustomGetComputedBootloaderChecksum`** `( u8 * pubComputedChecksum );` |

| | | |
|---|---|---|
| **Parameters (out)** | pubComputedChecksum | Pointer where to copy the checksum. |
| **Description** | Callback is called: Before starting Bootloader to verify the computed checksum<br>Callback shall implement: get from non-volatile memory the Bootloader checksum and copy it to the provided pointer | |

### 4.5.2.4.45. PROG_CustomGetEcuId

| | | |
|---|---|---|
| **Purpose** | API to be called in order to get the ECU Id from a custom location. | |
| **Synopsis** | void **PROG_CustomGetEcuId** ( u8 * paubEcuId ); | |
| **Parameters (out)** | paubEcuId | Pointer to ECU Id |
| **Description** | This API is called to get the ECU ID during the TransferData. The implementation of this API shall permit to give the EcuId to the bootloader from a custom location (Flash, RAM, custom configuration, etc.). | |

### 4.5.2.4.46. PROG_CustomGetEraseStatus

| | | |
|---|---|---|
| **Purpose** | Get the erase status of the memory block. | |
| **Synopsis** | tProgEraseStatus **PROG_CustomGetEraseStatus** ( u8 ubBlockId ); | |
| **Parameters (in)** | ubBlockId | ID of the Memory block to be erased |
| **Return Value** | state | |
| | PROG_E_NOT_ERASED | |
| | PROG_E_ERASED | |
| **Description** | Callback is called: On reception of Erase routine<br>Callback shall implement: provide information if logical block is already erase and that erase shall be skipped | |

### 4.5.2.4.47. PROG_CustomGetExpectedCrc

| | |
|---|---|
| **Purpose** | This API is called in order to get the expected CRC corresponding to the required logical block, if this one is not found in the request. |
| **Synopsis** | void **PROG_CustomGetExpectedCrc** ( u8 ubLogicalBlockId , u32 * pulExpectedCrc ); |

| | | |
|---|---|---|
| **Parameters (in)** | `ubLogicalBlockId` | The logical block on which corresponding CRC is required |
| **Parameters (out)** | `pulExpectedCrc` | The expected CRC |
| **Description** | Callback is called: During CRC verification<br>Callback shall implement: extract from downloaded software the expected CRC value | |

#### 4.5.2.4.48. PROG_CustomGetMacKey

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void` **`PROG_CustomGetMacKey`** `( const u8 ** paubKeyData , u32 * pulKeyLength );` |

#### 4.5.2.4.49. PROG_CustomGetNextSectorAddr

| | | |
|---|---|---|
| **Purpose** | Get next sector start address. | |
| **Synopsis** | `tProgAddressType` **`PROG_CustomGetNextSectorAddr`** `( tProgAddressType uMemAddress );` | |
| **Parameters (in)** | `uMemAddress` | Memory address of reference sector |
| **Return Value** | Start address of the next sector. | |
| **Description** | Callback is called: After an erase operation to set the beginning of the next sector address<br>Callback shall implement: Operation to get the next sector address | |

#### 4.5.2.4.50. PROG_CustomGetProgCounter

| | | |
|---|---|---|
| **Purpose** | API to get the stored value of the programming counter. | |
| **Synopsis** | `u16` **`PROG_CustomGetProgCounter`** `( u8 ubBlockId );` | |
| **Parameters (in)** | `ubBlockId` | BlockID |
| **Return Value** | programming counter on 16 bits | |
| **Description** | Callback is called: Before erasing the block<br>Callback shall implement: return the current value of the programming counter | |

### 4.5.2.4.51. PROG_CustomGetResetCause

| | | |
|---|---|---|
| **Purpose** | Restore the reset cause and the need of response. | |
| **Synopsis** | `void` **`PROG_CustomGetResetCause`** `( tProgResCause * pubResetCause , tProgBoolean * pubSendResp );` | |
| **Parameters (out)** | `pubResetCause` | pointer to the reset cause |
| | `pubSendResp` | Provide information if positive response shall be sent depending of the value of the suppressPositiveResponse bit from the request TRUE: response shall be sent / FALSE: no response shall be sent |
| **Description** | Callback is called: At Bootloader startup to get the UDS request that has caused the reset.<br>Callback shall implement: provide the cause of the reset (UDS request) that has been set by application or Bootloader (by call to PROG_CustomStoreResetCause) | |

### 4.5.2.4.52. PROG_CustomGetResumeAddress

| | | |
|---|---|---|
| **Purpose** | Get resume address. | |
| **Synopsis** | `u32` **`PROG_CustomGetResumeAddress`** `( u8 ubBlockId );` | |
| **Parameters (in)** | `ubBlockId` | Index of the logical block |
| **Return Value** | | |
| **Description** | Callback is called: On reception of ReadDataByIdenfifier for DID "Reprogramming Resume Information" | |

### 4.5.2.4.53. PROG_CustomGetSegmentList

| | | |
|---|---|---|
| **Purpose** | Retrieve segment list stored in memory. | |
| **Synopsis** | `void` **`PROG_CustomGetSegmentList`** `( tSegmentListType * pstSegList );` | |
| **Parameters (in,out)** | `pstSegList` | pointer on structure where to copy the data |
| **Description** | Callback is called: This function is called in case reprogramming shall be resume in order to get all data that have been previously written for the logical block | |

| | |
|---|---|
| | Callback shall implement: Copy from memory of data that have been previously stored with PROG_CustomStoreSegmentList |

### 4.5.2.4.54. PROG_CustomIncrementProgCounter

| | | |
|---|---|---|
| **Purpose** | API to increment the programming counter for the erased logical block. | |
| **Synopsis** | `tProgStatus PROG_CustomIncrementProgCounter ( u8 ubBlockId );` | |
| **Parameters (in)** | `ubBlockId` | BlockID |
| **Return Value** | Result of incrementation operation | |
| | `PROG_E_OK` | incrementation operation finished successfully |
| | `PROG_E_NOT_OK` | incrementation operation error happened |
| **Description** | This API is called to increment the programming counter of the block. The maximum value shall be limited at 0xFFFF and it shall not overflow.<br>Callback is called: Before erasing the block<br>Callback shall implement: increment the current value of the programming counter | |

### 4.5.2.4.55. PROG_CustomInvalidateBootloaderChecksum

| | |
|---|---|
| **Purpose** | This API is called to set the Bootloader checksum as invalid. |
| **Synopsis** | `void PROG_CustomInvalidateBootloaderChecksum ( void );` |
| **Description** | Callback is called: After Bootloader checksum computation, before updating the checksum.<br>Callback shall implement: Set in non-volatile memory the Bootloader checksum validity flag with invalid value |

### 4.5.2.4.56. PROG_CustomIsFirstProgramming

| | |
|---|---|
| **Purpose** | Get the status of the Flash memory if it's full erased or not (i.e first download on this ECU). |
| **Synopsis** | `tProgBoolean PROG_CustomIsFirstProgramming ( void );` |
| **Return Value** | eIsFirstProgramming status return by the function (PROG_TRUE / PROG_FALSE) |

| | |
|---|---|
| **Description** | Callback is called: On reception of Erase routine to skip erasing if memory has never been written<br>Callback shall implement: provide information if this is the first Flash programming |

### 4.5.2.4.57. PROG_CustomIsValidBootloaderChecksum

| | | |
|---|---|---|
| **Purpose** | This API is called to know if the stored Bootloader checksum is valid. | |
| **Synopsis** | `tProgBoolean` **`PROG_CustomIsValidBootloaderChecksum`** `( void );` | |
| **Return Value** | Validity status | |
| | `PROG_E_OK` | Checksum is valid |
| | `PROG_E_NOT_OK` | Checksum is invalid |
| **Description** | Callback is called: Before reading the bootloader checksum<br>Callback shall implement: get from non-volatile memory the Bootloader checksum status | |

### 4.5.2.4.58. PROG_CustomMemGetJobStatus

| | | |
|---|---|---|
| **Purpose** | Get the status of memory job. | |
| **Synopsis** | `tProgStatus` **`PROG_CustomMemGetJobStatus`** `( void );` | |
| **Return Value** | eProgStatus success of the operation(s) | |
| | `PROG_E_OK` | |
| | `PROG_E_NOT_OK` | |
| | `PROG_E_BUSY` | |
| **Description** | Callback is called: After each memory access operation<br>Callback shall implement: After PROG_CustomMemoryErase/PROG_CustomMemoryWrite/PROG_CustomMemoryRead returns PROG_E_BUSY, this callback is called periodically until getting a status different from PROG_E_BUSY. | |

### 4.5.2.4.59. PROG_CustomMemoryAccessNotification

| | |
|---|---|
| **Purpose** | Notification of memory access to allow customers to place their routines. |

| | | |
|---|---|---|
| **Synopsis** | `tProgStatus PROG_CustomMemoryAccessNotification ( tProgMemType eMemType , tOperationType eOperationType , tProgAddressType uMemAddress , tDataLength uLength , tDataBufferType PROG_FAR_POINTER paubDataBuffer );` | |
| **Parameters (in)** | `eMemType` | Memory type (RAM, Flash and Flash Ext) |
| | `eOperationType` | Operation type (Read, write and erase) |
| | `uMemAddress` | Start address |
| | `uLength` | Data length |
| | `paubDataBuffer` | Data buffer |
| **Return Value** | eProgStatus success of the operation(s) | |
| | `PROG_E_OK` | |
| | `PROG_E_NOT_OK` | |
| **Description** | Callback is called: After successful memory data access<br>Callback shall implement: Operation that need to be performed after a memory data access | |

#### 4.5.2.4.60. PROG_CustomSetAppValidity

| | | |
|---|---|---|
| **Purpose** | Set the Application Validity. | |
| **Synopsis** | `void PROG_CustomSetAppValidity ( u32 ulAddress , u32 ulEndAddress );` | |
| **Parameters (in)** | `ulAddress` | Start address of sector on which the CRC has succeeded |
| | `ulEndAddress` | End address of sector on which the CRC has succeeded |
| **Description** | Callback is called: After Checksum computation has succeeded<br>Callback shall implement: Update the application validity flag as valid (additional customer coherency check can be required to conclude on application validity) | |

#### 4.5.2.4.61. PROG_CustomSetApplicationChecksum

| | |
|---|---|
| **Purpose** | This API is called to store the computed Application checksum in non-volatile memory. |

| | | |
|---|---|---|
| **Synopsis** | void **PROG_CustomSetApplicationChecksum** ( u8 * pubComputedChecksum , u16 uwBlockIdentifier ); | |
| **Parameters (in)** | pubComputedChecksum | Pointer to Application checksum to store. |
| | uwBlockIdentifier | Block Identifier. |
| **Description** | Callback is called: Before sending response to CheckMemory request for authenticated block<br>Callback shall implement: store in non-volatile memory the Application checksum (will be compare at next startup with one calculated by application). It shall also update the checksum computed by application. | |

### 4.5.2.4.62. PROG_CustomSetBootloaderChecksum

| | | |
|---|---|---|
| **Purpose** | This API is called to store the computed Bootloader checksum in non-volatile memory. | |
| **Synopsis** | void **PROG_CustomSetBootloaderChecksum** ( u8 * pubComputedChecksum ); | |
| **Parameters (in)** | pubComputedChecksum | Pointer to Bootloader checksum to store. |
| **Description** | Callback is called: After Bootloader checksum computation at Bootloader start<br>Callback shall implement: store in non-volatile memory the Bootloader checksum | |

### 4.5.2.4.63. PROG_CustomSetDownloadVerificationSuccess

| | | |
|---|---|---|
| **Purpose** | Callback called after comparing the expected checksum or signature and the calculated one. | |
| **Synopsis** | tProgStatus **PROG_CustomSetDownloadVerificationSuccess** ( u8 ubLogicalBlockId , u8 ubLogicalSegmentId , tProgBoolean ubCompareSuccess ); | |
| **Parameters (in)** | ubLogicalBlockId | The id of the logical block for which the compare was done |
| | ubLogicalSegmentId | The id of the logical segment for which the compare was done (ignore if the full logical block is verified) |
| | ubCompareSuccess | TRUE if the 2 elements of the comparison are identical |
| **Return Value** | state | |

| | | |
|---|---|---|
| | PROG_E_OK | Treatment finish successfully |
| | PROG_E_NOT_OK | Error happened during treatment |
| **Description** | Callback is called: After a successful or unsuccessful download verification<br>Callback shall implement: update of the logical block validity status | |

### 4.5.2.4.64. PROG_CustomSetEraseStatus

| | | |
|---|---|---|
| **Purpose** | Set the erase status of the memory block. | |
| **Synopsis** | `tProgStatus` **`PROG_CustomSetEraseStatus`** `( u8 ubBlockId , tProgEraseStatus eEraseStatus );` | |
| **Parameters (in)** | ubBlockId | ID of the Memory block to be erased |
| | eEraseStatus | New erase status |
| **Return Value** | eProgStatus success of the erase status update | |
| | PROG_E_OK | |
| | PROG_E_NOT_OK | |
| **Description** | Callback is called: After successful logical block erasing and RequestDownload request reception<br>Callback shall implement: storage of the logical block erase status | |

### 4.5.2.4.65. PROG_CustomStartChecksumCalc

| | |
|---|---|
| **Purpose** | Initialization of the Custom Checksum calculation. |
| **Synopsis** | `void` **`PROG_CustomStartChecksumCalc`** `( void );` |
| **Description** | Callback is called: On start of a checksum calculation<br>Callback shall implement: Initialization of checksum calculation. Checksum calculation is customer specific (Checksum, CRC16,...) |

### 4.5.2.4.66. PROG_CustomStoreResetCause

| | |
|---|---|
| **Purpose** | Store the reset cause and the need of response. |
| **Synopsis** | `void` **`PROG_CustomStoreResetCause`** `( tProgResCause ubResetCause , tProgBoolean ubSendResp );` |

| | | |
|---|---|---|
| **Parameters (in)** | `ubResetCause` | the reset cause |
| | `ubSendResp` | the need of response according to suppressPositiveResponse bit from the request |
| **Description** | Callback is called: Before Bootloader perform a reset to set reset cause (UDS request that has caused the reset).<br>Callback shall implement: storage of the reset cause (UDS request) | |

### 4.5.2.4.67. PROG_CustomStoreResumeAddress

| | | |
|---|---|---|
| **Purpose** | Store resume address. | |
| **Synopsis** | `void` **`PROG_CustomStoreResumeAddress`**<br>`( u8 ubBlockId , u32 ulAddress );` | |
| **Parameters (in)** | `ubBlockId` | Index of the logical block |
| | `ulAddress` | Address to store |
| **Description** | Callback is called: During reprogramming to store resume address | |

### 4.5.2.4.68. PROG_CustomStoreSegmentList

| | | |
|---|---|---|
| **Purpose** | Store segment list in memory. | |
| **Synopsis** | `void` **`PROG_CustomStoreSegmentList`**<br>`( tSegmentListType * pstSegList );` | |
| **Parameters (in)** | `pstSegList` | address of structure to copy |
| **Description** | Callback is called: This function is called on RequestTransferExit to store information about the downloaded segment<br>Callback shall implement: Copy to memory of the segment list | |

### 4.5.2.4.69. PROG_CustomUpdateChecksumCalc

| | |
|---|---|
| **Purpose** | Update (transfer data blocks) for Custom Checksum calculation. |
| **Synopsis** | `void` **`PROG_CustomUpdateChecksumCalc`**<br>`( u8 * pubData , u32 ulDataSize );` |

| | | |
|---|---|---|
| **Parameters (in)** | `pubData` | pointer to the data to compute |
| | `ulDataSize` | Length of data to compute |
| **Description** | Callback is called: After a PROG_CustomStartChecksumCalc call and when data to be used for checksum calculation have been read from Flash memory<br>Callback shall implement: Checksum calculation. Checksum calculation is customer specific (Checksum, CRC16,...) | |

#### 4.5.2.4.70. PROG_CustomValidateBootloaderChecksum

| | |
|---|---|
| **Purpose** | This API is called to set the Bootloader checksum as valid. |
| **Synopsis** | `void` **`PROG_CustomValidateBootloaderChecksum`** `( void );` |
| **Description** | Callback is called: After Bootloader checksum computation, after updating the checksum.<br>Callback shall implement: Set in non-volatile memory the Bootloader checksum validity flag with valid value |

#### 4.5.2.4.71. PROG_CustomWriteCRC

| | | |
|---|---|---|
| **Purpose** | Callback for CRC storage. | |
| **Synopsis** | `void` **`PROG_CustomWriteCRC`** `( u32 ulCrcVal );` | |
| **Parameters (in)** | `ulCrcVal` | CRC value |
| **Description** | Callback is called: After CRC calculation<br>Callback shall implement: storage of the CRC value for further use | |

#### 4.5.2.4.72. PROG_CustomWriteProgStatus

| | | |
|---|---|---|
| **Purpose** | Callback storing the programming status structure. | |
| **Synopsis** | `void` **`PROG_CustomWriteProgStatus`** `( u32 ulProgrammingStatus );` | |
| **Parameters (in)** | `ulProgrammingStatus` | Programming Status (4 Bytes) |
| **Description** | Callback is called: After Programming status update<br>Callback shall implement: Storage of Programming status in RAM. The storage in non-volatile memory shall be done before the ECU is | |

### 4.5.2.4.73. PROG_DisableECCCheck

| | |
|---|---|
| **Purpose** | Callback that shall disable ECC if needed Callback is called: When Bootloader perform read access on Flash that can be unprogrammed and can cause ECC error. |
| **Synopsis** | void **PROG_DisableECCCheck** ( void ); |
| **Description** | Callback shall implement: If needed, disabling of ECC check Hardware specific) |

### 4.5.2.4.74. PROG_Do_CheckHash

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | void **PROG_Do_CheckHash** ( void ); |

### 4.5.2.4.75. PROG_Do_CheckPrgDependencies

| | |
|---|---|
| **Purpose** | Called to calculate the CRC for CheckProgDependencies routine. |
| **Synopsis** | void **PROG_Do_CheckPrgDependencies** ( void ); |

### 4.5.2.4.76. PROG_Do_CheckSignature

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | void **PROG_Do_CheckSignature** ( void ); |

### 4.5.2.4.77. PROG_Do_CoherencyCheck

| | |
|---|---|
| **Purpose** | Called to do the Coherency Check treatment. |
| **Synopsis** | void **PROG_Do_CoherencyCheck** ( void ); |

### 4.5.2.4.78. PROG_Do_CompareKey

| | |
|---|---|
| **Purpose** | API called to get seed result. |
| **Synopsis** | void **PROG_Do_CompareKey** ( void ); |

### 4.5.2.4.79. PROG_Do_GetSeed

| | |
|---|---|
| **Purpose** | API called to get seed. |
| **Synopsis** | `void PROG_Do_GetSeed ( void );` |

### 4.5.2.4.80. PROG_DrvDown_IsFlashRoutinesPresent

| | | |
|---|---|---|
| **Purpose** | Returns the value of m_ubFlashRoutinesPresent, that represents the presence of the flash routines in RAM. | |
| **Synopsis** | `tProgBoolean PROG_DrvDown_IsFlashRoutinesPresent ( void );` | |
| **Return Value** | Result of treatment | |
| | `PROG_TRUE` | Flash routines are present in RAM |
| | `PROG_FALSE` | Flash routines are not present in RAM |
| **Description** | This function is called to verify the presence of the flash routines | |

### 4.5.2.4.81. PROG_Dsc01Cbk

| | | |
|---|---|---|
| **Purpose** | UDS callback for DiagnosticSessionControlDefault. | |
| **Synopsis** | `tUdsStatus PROG_Dsc01Cbk ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the DiagnosticSessionControlDefault UDS request. It shall be configured in UDS Tresos Studio plugin for the request DSC (0x10 0x01) with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.82. PROG_Dsc03Cbk

| | |
|---|---|
| **Purpose** | UDS callback for DiagnosticSessionControlExtended. |
| **Synopsis** | `tUdsStatus PROG_Dsc03Cbk ( u16 * puwLen , u8 * aubUdsData );` |

| | | |
|---|---|---|
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| **Description** | This function handles the DiagnosticSessionControlExtended UDS request. It shall be configured in UDS Tresos Studio plugin for the request DSC (0x10 0x03) with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.83. PROG_EcuReset

| | | |
|---|---|---|
| **Purpose** | UDS callback for EcuReset. | |
| **Synopsis** | `tUdsStatus PROG_EcuReset ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the EcuReset UDS request. It shall be configured in UDS Tresos Studio plugin for the request ER (0x11 0x01) with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.84. PROG_EnableECCCheck

| | |
|---|---|
| **Purpose** | Callback that shall enable ECC if needed Callback is called: After Bootloader has performed a read access on Flash that can be unprogrammed and can cause ECC error. |
| **Synopsis** | `void PROG_EnableECCCheck ( void );` |
| **Description** | Callback shall implement: If needed, enabling of ECC check (Hardware specific) |

### 4.5.2.4.85. PROG_Entry_ActiveSBL

| | |
|---|---|
| **Purpose** | Called on entry to ActiveSBL state. |
| **Synopsis** | `void PROG_Entry_ActiveSBL ( void );` |

### 4.5.2.4.86. PROG_Entry_Alive

| | |
|---|---|
| **Purpose** | Called on entry to Alive state. |
| **Synopsis** | `void PROG_Entry_Alive ( void );` |

### 4.5.2.4.87. PROG_Entry_AutoControl

| | |
|---|---|
| **Purpose** | Called on entry to AutoControl state. |
| **Synopsis** | `void PROG_Entry_AutoControl ( void );` |

### 4.5.2.4.88. PROG_Entry_CheckDependenciesFinish

| | |
|---|---|
| **Purpose** | Called to send CheckProgDependencies routine response. |
| **Synopsis** | `void PROG_Entry_CheckDependenciesFinish ( void );` |

### 4.5.2.4.89. PROG_Entry_CheckHash

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_CheckHash ( void );` |

### 4.5.2.4.90. PROG_Entry_CheckMemory

| | |
|---|---|
| **Purpose** | Check memory programming. |
| **Synopsis** | `void PROG_Entry_CheckMemory ( void );` |

### 4.5.2.4.91. PROG_Entry_CheckMemoryCompute

| | |
|---|---|
| **Purpose** | Check memory programming. |
| **Synopsis** | `void PROG_Entry_CheckMemoryCompute ( void );` |

### 4.5.2.4.92. PROG_Entry_CheckMemoryFinish

| | |
|---|---|
| **Purpose** | Called on entry to CHECK_MEMORY_FINISH state. |

| | |
|---|---|
| **Synopsis** | `void PROG_Entry_CheckMemoryFinish ( void );` |

### 4.5.2.4.93. PROG_Entry_ChecksumByRange

| | |
|---|---|
| **Purpose** | Init of the Checksum Calculation. |
| **Synopsis** | `void PROG_Entry_ChecksumByRange ( void );` |

### 4.5.2.4.94. PROG_Entry_CoherencyPreCheck

| | |
|---|---|
| **Purpose** | Called to do the Coherency Pre Check treatment. |
| **Synopsis** | `void PROG_Entry_CoherencyPreCheck ( void );` |

### 4.5.2.4.95. PROG_Entry_CompareKey

| | |
|---|---|
| **Purpose** | Called on entry to CompareKey state. |
| **Synopsis** | `void PROG_Entry_CompareKey ( void );` |

### 4.5.2.4.96. PROG_Entry_CompareKeyCheck

| | |
|---|---|
| **Purpose** | Called on entry to CompareKey state. |
| **Synopsis** | `void PROG_Entry_CompareKeyCheck ( void );` |

### 4.5.2.4.97. PROG_Entry_DecompHeader

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_DecompHeader ( void );` |

### 4.5.2.4.98. PROG_Entry_DefaultSession

| | |
|---|---|
| **Purpose** | Called on entry to DefaultSession state. |
| **Synopsis** | `void PROG_Entry_DefaultSession ( void );` |

### 4.5.2.4.99. PROG_Entry_EcuReset

| | |
|---|---|
| **Purpose** | Called on entry to EcuReset state. |
| **Synopsis** | `void PROG_Entry_EcuReset ( void );` |

### 4.5.2.4.100. PROG_Entry_Erase

| | |
|---|---|
| **Purpose** | Called on entry to Erase state. |
| **Synopsis** | `void PROG_Entry_Erase ( void );` |

### 4.5.2.4.101. PROG_Entry_EraseCheck

| | |
|---|---|
| **Purpose** | Called on entry to EraseCheck state. |
| **Synopsis** | `void PROG_Entry_EraseCheck ( void );` |

### 4.5.2.4.102. PROG_Entry_EraseFinish

| | |
|---|---|
| **Purpose** | Called on entry to EraseFinish state. |
| **Synopsis** | `void PROG_Entry_EraseFinish ( void );` |

### 4.5.2.4.103. PROG_Entry_EraseNRC78

| | |
|---|---|
| **Purpose** | Called on entry to EraseNRC78 state. |
| **Synopsis** | `void PROG_Entry_EraseNRC78 ( void );` |

### 4.5.2.4.104. PROG_Entry_EraseTransmitNRC78

| | |
|---|---|
| **Purpose** | Called on entry to EraseTransmitNRC78 state. |
| **Synopsis** | `void PROG_Entry_EraseTransmitNRC78 ( void );` |

### 4.5.2.4.105. PROG_Entry_ExtendedSession

| | |
|---|---|
| **Purpose** | Called on entry to ExtendedSession state. |

| | |
|---|---|
| **Synopsis** | `void PROG_Entry_ExtendedSession ( void );` |

### 4.5.2.4.106. PROG_Entry_GetSeed

| | |
|---|---|
| **Purpose** | Called on entry to GetSeed state. |
| **Synopsis** | `void PROG_Entry_GetSeed ( void );` |

### 4.5.2.4.107. PROG_Entry_GetSeedCheck

| | |
|---|---|
| **Purpose** | Called on entry to GetSeed state. |
| **Synopsis** | `void PROG_Entry_GetSeedCheck ( void );` |

### 4.5.2.4.108. PROG_Entry_HSMUpdate_TDFinish

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_HSMUpdate_TDFinish ( void );` |

### 4.5.2.4.109. PROG_Entry_INIT

| | |
|---|---|
| **Purpose** | Called on entry to INIT state. |
| **Synopsis** | `void PROG_Entry_INIT ( void );` |

### 4.5.2.4.110. PROG_Entry_PartialVerificationCrc

| | |
|---|---|
| **Purpose** | Called on VerifyPartialSoftwareChecksum routine reception. |
| **Synopsis** | `void PROG_Entry_PartialVerificationCrc ( void );` |

### 4.5.2.4.111. PROG_Entry_PreInit

| | |
|---|---|
| **Purpose** | Called on entry to PreInit state. |
| **Synopsis** | `void PROG_Entry_PreInit ( void );` |

### 4.5.2.4.112. PROG_Entry_ProgrammingSession

| | |
|---|---|
| **Purpose** | Called on entry to ProgrammingSession state. |
| **Synopsis** | `void PROG_Entry_ProgrammingSession ( void );` |

### 4.5.2.4.113. PROG_Entry_RD

| | |
|---|---|
| **Purpose** | Called on entry to RD state. |
| **Synopsis** | `void PROG_Entry_RD ( void );` |

### 4.5.2.4.114. PROG_Entry_RD_Finish

| | |
|---|---|
| **Purpose** | Called on entry to RD Finish state. |
| **Synopsis** | `void PROG_Entry_RD_Finish ( void );` |

### 4.5.2.4.115. PROG_Entry_RD_Signature

| | |
|---|---|
| **Purpose** | Called on entry to RD SIGNATURE state. |
| **Synopsis** | `void PROG_Entry_RD_Signature ( void );` |

### 4.5.2.4.116. PROG_Entry_RTE

| | |
|---|---|
| **Purpose** | Called on entry to RTE state. |
| **Synopsis** | `void PROG_Entry_RTE ( void );` |

### 4.5.2.4.117. PROG_Entry_RTEFailed

| | |
|---|---|
| **Purpose** | Called on entry to RTEFailed state. |
| **Synopsis** | `void PROG_Entry_RTEFailed ( void );` |

### 4.5.2.4.118. PROG_Entry_RTEFinish

| | |
|---|---|
| **Purpose** | Called on entry to RTEFinish state. |

| | |
|---|---|
| **Synopsis** | `void PROG_Entry_RTEFinish ( void );` |

#### 4.5.2.4.119. PROG_Entry_Reset

| | |
|---|---|
| **Purpose** | Called on entry to Reset state. |
| **Synopsis** | `void PROG_Entry_Reset ( void );` |

#### 4.5.2.4.120. PROG_Entry_ResumeVerification

| | |
|---|---|
| **Purpose** | Entry function for Resume verification state. |
| **Synopsis** | `void PROG_Entry_ResumeVerification ( void );` |

#### 4.5.2.4.121. PROG_Entry_Resume_Finish

| | |
|---|---|
| **Purpose** | Entry function for Resume verification state. |
| **Synopsis** | `void PROG_Entry_Resume_Finish ( void );` |

#### 4.5.2.4.122. PROG_Entry_SblSynch

| | |
|---|---|
| **Purpose** | Called on entry to SblSynch state. |
| **Synopsis** | `void PROG_Entry_SblSynch ( void );` |

#### 4.5.2.4.123. PROG_Entry_SecureChecksumFailed

| | |
|---|---|
| **Purpose** | Send response in the case of a failure on secure checksum calculation. |
| **Synopsis** | `void PROG_Entry_SecureChecksumFailed ( void );` |
| **Description** | This function is called at the end of the Check Memory in the case of a failure on secure checksum calculation |

#### 4.5.2.4.124. PROG_Entry_SignatureCheck

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_SignatureCheck ( void );` |

### 4.5.2.4.125. PROG_Entry_Sleep

| | |
|---|---|
| **Purpose** | Called on entry to Sleep state. |
| **Synopsis** | `void PROG_Entry_Sleep ( void );` |

### 4.5.2.4.126. PROG_Entry_Streaming

| | |
|---|---|
| **Purpose** | Called on entry to Streaming state. |
| **Synopsis** | `void PROG_Entry_Streaming ( void );` |

### 4.5.2.4.127. PROG_Entry_TD

| | |
|---|---|
| **Purpose** | Called on entry to TD state. |
| **Synopsis** | `void PROG_Entry_TD ( void );` |

### 4.5.2.4.128. PROG_Entry_TD_Failed

| | |
|---|---|
| **Purpose** | Called on entry to TD_Failed state. |
| **Synopsis** | `void PROG_Entry_TD_Failed ( void );` |

### 4.5.2.4.129. PROG_Entry_TD_Header

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_TD_Header ( void );` |

### 4.5.2.4.130. PROG_Entry_UpdatePSI

| | |
|---|---|
| **Purpose** | Called on entry to UpdatePSI state. |
| **Synopsis** | `void PROG_Entry_UpdatePSI ( void );` |

### 4.5.2.4.131. PROG_Entry_ValidateSBASignature

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_ValidateSBASignature ( void );` |

### 4.5.2.4.132. PROG_Entry_ValidateSBASignerInfo

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_ValidateSBASignerInfo ( void );` |

### 4.5.2.4.133. PROG_Entry_ValidateSignature

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_ValidateSignature ( void );` |

### 4.5.2.4.134. PROG_Entry_ValidateSignerInfo

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_Entry_ValidateSignerInfo ( void );` |

### 4.5.2.4.135. PROG_Entry_Write

| | |
|---|---|
| **Purpose** | Called on entry to Write state. |
| **Synopsis** | `void PROG_Entry_Write ( void );` |

### 4.5.2.4.136. PROG_Entry_WriteFingerprint

| | |
|---|---|
| **Purpose** | Called on entry to WriteFingerprint state. |
| **Synopsis** | `void PROG_Entry_WriteFingerprint ( void );` |

### 4.5.2.4.137. PROG_Erase

| | |
|---|---|
| **Purpose** | Called in Erase state. |
| **Synopsis** | `void PROG_Erase ( void );` |

### 4.5.2.4.138. PROG_EraseMemoryRequest

| | |
|---|---|
| **Purpose** | UDS callback for EraseMemoryRequest. |

| | | |
|---|---|---|
| **Synopsis** | `tUdsStatus` **`PROG_EraseMemoryRequest`** `( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the EraseMemoryRequest UDS request. It shall be configured in UDS Tresos Studio plugin for the request RC Erase Memory (0x31 0x01 0xFF 0x00) with this exact name and with Callback_Origin set to OTHER. | |

#### 4.5.2.4.139. PROG_Erase_Guard

| | | |
|---|---|---|
| **Purpose** | Called before going to PROG_Erase state. | |
| **Synopsis** | `tProgStatus` **`PROG_Erase_Guard`** `( void );` | |
| **Return Value** | Result of check | |
| | `PROG_E_OK` | Erase request accepted |
| | `PROG_E_NOT_OK` | Erase request not accepted |

#### 4.5.2.4.140. PROG_Exit_CheckMemory

| | |
|---|---|
| **Purpose** | Check memory programming. |
| **Synopsis** | `void` **`PROG_Exit_CheckMemory`** `( void );` |

#### 4.5.2.4.141. PROG_Exit_CheckMemoryFinish

| | |
|---|---|
| **Purpose** | Called on exit to CHECK_MEMORY_FINISH state. |
| **Synopsis** | `void` **`PROG_Exit_CheckMemoryFinish`** `( void );` |

#### 4.5.2.4.142. PROG_Exit_INIT

| | |
|---|---|
| **Purpose** | Called on entry to INIT state. |

| | |
|---|---|
| **Synopsis** | `void PROG_Exit_INIT ( void );` |

### 4.5.2.4.143. PROG_Exit_PartialVerificationCrc

| | |
|---|---|
| **Purpose** | Called on VerifyPartialSoftwareChecksum routine reception. |
| **Synopsis** | `void PROG_Exit_PartialVerificationCrc ( void );` |

### 4.5.2.4.144. PROG_Exit_TD_Write

| | |
|---|---|
| **Purpose** | Called on exit from TD_Write state. |
| **Synopsis** | `void PROG_Exit_TD_Write ( void );` |

### 4.5.2.4.145. PROG_FlashPage

| | | |
|---|---|---|
| **Purpose** | Called to write one or more flash pages. | |
| **Synopsis** | `u8 PROG_FlashPage ( u32 ulAddress , u8 ubPages , u32 * pulDataBuffer , u16 uwDataBufferLenght );` | |
| **Parameters (in)** | `ulAddress` | Address of the first page to write |
| | `ubPages` | number of pages to write |
| | `pulDataBuffer` | pointer to the buffer of data to write |
| | `uwDataBufferLenght` | Exact length of data to write |
| **Return Value** | Result of the write operation | |
| | `PROG_FLASH_PAGE_E_OK` | write is successfull |
| | `PROG_FLASH_PAGE_E_NOT_OK` | write is not successfull |
| **Description** | This API can be called by the Framework to write one or more specific flash pages. | |

### 4.5.2.4.146. PROG_GetActiveCurrentSession

| | | |
|---|---|---|
| **Purpose** | UDS callback for GetActiveCurrentSession. | |
| **Synopsis** | `tUdsStatus PROG_GetActiveCurrentSession ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |

| | | |
|---|---|---|
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the GetActiveCurrentSession UDS request. It shall be configured in UDS Tresos Studio plugin for the request RDBI with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.147. PROG_GetComputedBootloaderChecksum

| | | |
|---|---|---|
| **Purpose** | API for getting the previously computed Bootloader checksum. | |
| **Synopsis** | `tProgStatus` **`PROG_GetComputedBootloaderChecksum`** `( u8 * pubComputedChecksum );` | |
| **Parameters (out)** | `pubComputedChecksum` | Computed Checksum (if return is PROG_E_OK) |
| **Return Value** | Operation status | |
| | `PROG_E_OK` | Checksum has been provide |
| | `PROG_E_NOT_OK` | Checksum is invalid and cannot be provided |

### 4.5.2.4.148. PROG_GetCurrentDiagApp

| | | |
|---|---|---|
| **Purpose** | UDS callback for GetCurrentDiagApp. | |
| **Synopsis** | `tUdsStatus` **`PROG_GetCurrentDiagApp`** `( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the GetCurrentDiagApp UDS request. It shall be configured in UDS Tresos Studio plugin for the request RDBI with this exact name and with Callback_Origin set to OTHER. | |

#### 4.5.2.4.149. PROG_GetDidF0F3

| | | |
|---|---|---|
| **Purpose** | Called by external module to get data of DID F0F3. | |
| **Synopsis** | `void PROG_GetDidF0F3 ( u8 * aubData );` | |
| **Parameters (out)** | `aubData` | DID data |
| **Description** | This API is called by an external value to get the data of the DID F0F3 (EcuId) | |

#### 4.5.2.4.150. PROG_GetDidF0F6

| | | |
|---|---|---|
| **Purpose** | Called by external module to get data of DID F0F6. | |
| **Synopsis** | `void PROG_GetDidF0F6 ( u8 * aubData );` | |
| **Parameters (out)** | `aubData` | DID data |
| **Description** | This API is called by an external value to get the data of the DID F0F6 (EcuId) | |

#### 4.5.2.4.151. PROG_GetKeyNBIDValue

| | |
|---|---|
| **Purpose** | Callback retrieving the Key NBID value stored in memory. |
| **Synopsis** | `u16 PROG_GetKeyNBIDValue ( void );` |
| **Return Value** | Key NBID value on 16 bits |
| **Description** | Callback is called: during reprogramming process to get the key NBID of the current application stored in NVM.<br>It will be used to know if the new download application can be accepted or not<br>Callback shall implement: reading from non volatile memory of the Key NBID value |

#### 4.5.2.4.152. PROG_GetMacKey

| | | |
|---|---|---|
| **Purpose** | Get the key used to compute MAC checksum. | |
| **Synopsis** | `void PROG_GetMacKey ( Csm_SymKeyType * pstMacKey );` | |
| **Parameters (out)** | `pstMacKey` | Pointer to the Key |
| **Description** | This function is called to get the key used for MAC computation | |

### 4.5.2.4.153. PROG_GetNBIDValue

| | |
|---|---|
| **Purpose** | Callback retrieving the NBID value store in memory. |
| **Synopsis** | `u16 PROG_GetNBIDValue ( void );` |
| **Return Value** | NBID value on 16 bits |
| **Description** | Callback is called: during reprogramming process to get the NBID of the current application stored in NVM.<br>It will be used to know if the new download application can be accepted or not<br>Callback shall implement: reading from non volatile memory of the NBID value |

### 4.5.2.4.154. PROG_GetNetworkStatus

| | |
|---|---|
| **Purpose** | Function providing the network status. |
| **Synopsis** | `u16 PROG_GetNetworkStatus ( void );` |
| **Return Value** | Network Status |

### 4.5.2.4.155. PROG_GetProgCntrLockVal

| | | |
|---|---|---|
| **Purpose** | API for getting the programming counter lock value (maximum programming counter) for a specific block. | |
| **Synopsis** | `tProgStatus PROG_GetProgCntrLockVal ( u8 ubBlockId , u16 * puwProgCntMax );` | |
| **Parameters (in)** | `ubBlockId` | BlockID |
| **Parameters (out)** | `puwProgCntMax` | Programming Counter lock value |
| **Return Value** | Operation status | |
| | `PROG_E_OK` | Block index has been found |
| | `PROG_E_NOT_OK` | Block index has not been found |

### 4.5.2.4.156. PROG_GetSBIFlagValue

| | |
|---|---|
| **Purpose** | Retrieve the SBI flag value stored in memory. |

| | |
|---|---|
| **Synopsis** | `u8 PROG_GetSBIFlagValue ( void );` |
| **Return Value** | SBI flag value on 8 bits |
| **Description** | Callback is called: during reprogrammimg process to update the SBI flag for the SBA ticket stored in Secure RAM.<br>Callback shall implement: Reading in secure volatile memory of the SBI flag value |

### 4.5.2.4.157. PROG_GetSecurityLevel

| | | |
|---|---|---|
| **Purpose** | Get the current security level. | |
| **Synopsis** | `u8 PROG_GetSecurityLevel ( void );` | |
| **Return Value** | the current security level | |
| | `0x00U` | ECU is locked |
| | `0x0nU` | ECU is in security level n |
| **Description** | This function handles the Security Level feature managed by UDS. It shall be configured in UDS Tresos Studio plugin (if security check is needed). | |

### 4.5.2.4.158. PROG_GetSeed_Unlocked

| | |
|---|---|
| **Purpose** | Called in GetSeed_Unlocked state. |
| **Synopsis** | `void PROG_GetSeed_Unlocked ( void );` |

### 4.5.2.4.159. PROG_GetSuppressBitFromAppli

| | | |
|---|---|---|
| **Purpose** | Get the status of the suppress positive response bit from the last reprogramming request in application. | |
| **Synopsis** | `u8 PROG_GetSuppressBitFromAppli ( void );` | |
| **Return Value** | Suppression bit value | |
| | `TRUE` | The suppress positive response bit was set (response will not be sent) |
| | `FALSE` | The suppress positive response bit was not set (response will be sent) |
| **Description** | Callback is called: At startup when Bootloader shall send a response to a request that has been received in application | |

| | |
|---|---|
| | Callback shall implement: get from application information if the suppressPositiveResponse bit was set in the received request (e.g: read a flag from noinit RAM shared between Bootloader and Application) |

#### 4.5.2.4.160. PROG_Guard_RD_Check_RTEFinish

| | | |
|---|---|---|
| **Purpose** | Called receiving a RD request after a RTE. | |
| **Synopsis** | `tProgStatus` **`PROG_Guard_RD_Check_RTEFinish`** `( void );` | |
| **Return Value** | Result of check | |
| | `PROG_E_OK` | RD request accepted |
| | `PROG_E_NOT_OK` | RD request not accepted |

#### 4.5.2.4.161. PROG_HSMStatusManage

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void` **`PROG_HSMStatusManage`** `( void );` |

#### 4.5.2.4.162. PROG_HSMUpdate_TD

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void` **`PROG_HSMUpdate_TD`** `( void );` |

#### 4.5.2.4.163. PROG_Impl10_CheckDataBlocksResult

| | |
|---|---|
| **Purpose** | Called on entry to HASH_CHECK. |
| **Synopsis** | `void` **`PROG_Impl10_CheckDataBlocksResult`** `( void );` |

#### 4.5.2.4.164. PROG_Impl10_CheckMemoryAllowed

| | |
|---|---|
| **Purpose** | Called on guard of check memory to check whether it is allowed or not. |
| **Synopsis** | `tProgStatus` **`PROG_Impl10_CheckMemoryAllowed`** `( void );` |

| | | |
|---|---|---|
| **Return Value** | | |

### 4.5.2.4.165. PROG_Impl10_CompareDataBlockHash

| | |
|---|---|
| **Purpose** | Called on entry to RTE_COMPARE_HASH state. |
| **Synopsis** | `void PROG_Impl10_CompareDataBlockHash ( void );` |

### 4.5.2.4.166. PROG_Impl10_Do_CheckHashOfKey

| | |
|---|---|
| **Purpose** | Called during the write production key state to check if hash is finished. |
| **Synopsis** | `void PROG_Impl10_Do_CheckHashOfKey ( void );` |

### 4.5.2.4.167. PROG_Impl10_Do_HashMoreUnwrittenData

| | |
|---|---|
| **Purpose** | Called while RTE_COMPARE_HASH is active, it updates the hash calculation according to the length mentioned in VBT. |
| **Synopsis** | `void PROG_Impl10_Do_HashMoreUnwrittenData ( void );` |

### 4.5.2.4.168. PROG_Impl10_Entry_CheckMemoryFailed

| | |
|---|---|
| **Purpose** | Called on entry to CHECK_MEMORY_FAILED state. |
| **Synopsis** | `void PROG_Impl10_Entry_CheckMemoryFailed ( void );` |

### 4.5.2.4.169. PROG_Impl10_Entry_CheckReceivedKey

| | |
|---|---|
| **Purpose** | Called upon the reception of write key request. |
| **Synopsis** | `void PROG_Impl10_Entry_CheckReceivedKey ( void );` |

### 4.5.2.4.170. PROG_Impl10_Entry_SignatureCheck

| | |
|---|---|
| **Purpose** | Called on entry to SIGNATURE_CHECK. |

| | |
|---|---|
| **Synopsis** | `void PROG_Impl10_Entry_SignatureCheck ( void );` |

### 4.5.2.4.171. PROG_Impl10_Entry_WriteKeyFinished

| | |
|---|---|
| **Purpose** | Called on when the hash calculation is finished. |
| **Synopsis** | `void PROG_Impl10_Entry_WriteKeyFinished ( void );` |

### 4.5.2.4.172. PROG_Impl10_FinalizeHash

| | |
|---|---|
| **Purpose** | Called on entry to RTE_COMPARE_HASH. |
| **Synopsis** | `void PROG_Impl10_FinalizeHash ( void );` |

### 4.5.2.4.173. PROG_Impl10_GenerateMac

| | |
|---|---|
| **Purpose** | Called on entry to MAC_GENERATATION, it generates the mac for each software part. |
| **Synopsis** | `void PROG_Impl10_GenerateMac ( void );` |

### 4.5.2.4.174. PROG_Init

| | |
|---|---|
| **Purpose** | Initialize PROG module. |
| **Synopsis** | `void PROG_Init ( void );` |

### 4.5.2.4.175. PROG_InvalidateBlock

| | | |
|---|---|---|
| **Purpose** | API that invalidate the logical block. | |
| **Synopsis** | `tProgStatus PROG_InvalidateBlock ( u8 ubBlockId );` | |
| **Parameters (in)** | `ubBlockId` | The ID of the block that will be erased |
| **Return Value** | Result application invalidation | |
| | `PROG_E_OK` | Treatment finish successfully |
| | `PROG_E_NOT_OK` | Error happened during treatment |
| **Description** | Callback is called: On Erase routine reception | |

| | |
|---|---|
| | Callback shall implement: 1- customer code that shall be executed before performing an erasing of a logical block<br>2- It shall invalidate the logical block that will be erased to make sure no jump to the application will be done if an error occurred and the application is not fully erased or reprogrammed |

#### 4.5.2.4.176. PROG_InvalidateSection

| | | |
|---|---|---|
| **Purpose** | Callback invalidating the application markers. | |
| **Synopsis** | `tProgStatus` **`PROG_InvalidateSection`** `( tProgAddressType ulStartAddress , u32 ulEraseLength , tUdsStatus * ErrorCode );` | |
| **Parameters (in)** | `ulStartAddress` | Erased Start address of the segment that will be erased |
| | `ulEraseLength` | requested erase length |
| **Parameters (out)** | `ErrorCode` | UDS error code that shall be return in case of error during API treatment |
| **Return Value** | Result application invalidation | |
| | `PROG_E_OK` | Treatment finish successfully |
| | `PROG_E_NOT_OK` | Error happened during treatment (ErrorCode shall be filled in this case) |
| **Description** | Callback is called: On Erase routine reception<br>Callback shall implement: 1- customer code that shall be executed before performing an erasing<br>2- It shall invalidate the application or the current erased section to make sure no jump to the application will be done if an error occurred and the application is not fully erased or reprogrammed | |

#### 4.5.2.4.177. PROG_InvalidateSection_BlockID

| | | |
|---|---|---|
| **Purpose** | API that invalidate the application marker. | |
| **Synopsis** | `tProgStatus` **`PROG_InvalidateSection_BlockID`** `( u8 ubBlockId );` | |
| **Parameters (in)** | `ubBlockId` | The ID of the block that will be erased |
| **Return Value** | Result application invalidation | |

| | | |
|---|---|---|
| | PROG_E_OK | Treatment finish successfully |
| | PROG_E_NOT_OK | Error happened during treatment |
| **Description** | Callback is called: On Erase routine reception<br>Callback shall implement: 1- customer code that shall be executed before performing an erasing of a logical block<br>2- It shall invalidate the logical block that will be erased to make sure no jump to the application will be done if an error occurred and the application is not fully erased or reprogrammed | |

#### 4.5.2.4.178. PROG_IsValidApplication

| | | |
|---|---|---|
| **Purpose** | Callback checking if the application is valid or not. | |
| **Synopsis** | `tProgBoolean PROG_IsValidApplication ( void );` | |
| **Return Value** | Result of check | |
| | TRUE | Application is valid |
| | FALSE | Application is not valid or not present |
| **Description** | Callback is called: at startup and on some routine.<br>Callback shall implement: It shall verify that the application with all its dependencies are correctly flashed and return the result of the check | |

#### 4.5.2.4.179. PROG_JumpToApplication

| | |
|---|---|
| **Purpose** | Callback performing jump to application software. |
| **Synopsis** | `void PROG_JumpToApplication ( void );` |
| **Description** | Callback is called: at Bootloader startup if application is valid and shall be executed<br>Callback shall implement: jump to application start address |

#### 4.5.2.4.180. PROG_JumpToSBL

| | |
|---|---|
| **Purpose** | Called to perform the jump to SBL. |
| **Synopsis** | `void PROG_JumpToSBL ( void );` |

#### 4.5.2.4.181. PROG_Manage

| | |
|---|---|
| **Purpose** | Manage function to be called periodically. |
| **Synopsis** | `void PROG_Manage ( void );` |

#### 4.5.2.4.182. PROG_MessageDigestCheck

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_MessageDigestCheck ( void );` |

#### 4.5.2.4.183. PROG_OpenProgrammingSession

| | |
|---|---|
| **Purpose** | Reception of a programming session request. |
| **Synopsis** | `void PROG_OpenProgrammingSession ( void );` |

#### 4.5.2.4.184. PROG_PreInit

| | |
|---|---|
| **Purpose** | Called in PreInit state. |
| **Synopsis** | `void PROG_PreInit ( void );` |

#### 4.5.2.4.185. PROG_RD_Check

| | | |
|---|---|---|
| **Purpose** | Called receiving a RD request. | |
| **Synopsis** | `tProgStatus PROG_RD_Check ( void );` | |
| **Return Value** | Result of check | |
| | `PROG_E_OK` | RD request accepted |
| | `PROG_E_NOT_OK` | RD request not accepted |

#### 4.5.2.4.186. PROG_RTE

| | |
|---|---|
| **Purpose** | Called on cyclically in RTE state. |

| | |
|---|---|
| **Synopsis** | `void PROG_RTE ( void );` |

### 4.5.2.4.187. PROG_RangeChecksumFinish

| | | |
|---|---|---|
| **Purpose** | Finishing the checksum calculation. | |
| **Synopsis** | `tProgStatus PROG_RangeChecksumFinish ( void );` | |
| **Return Value** | state | |
| | `PROG_E_OK` | Calculation finished successfully |
| | `PROG_E_BUSY` | Calculation in progress |
| | `PROG_E_NOT_OK` | Calculation finished on error |

### 4.5.2.4.188. PROG_RequestDownload

| | | |
|---|---|---|
| **Purpose** | UDS callback for RequestSeed. | |
| **Synopsis** | `tUdsStatus PROG_RequestDownload ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the RequestSeed UDS request. It shall be configured in UDS Tresos Studio plugin for the request RD (0x34) with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.189. PROG_RequestSeed

| | | |
|---|---|---|
| **Purpose** | UDS callback for RequestSeed. | |
| **Synopsis** | `tUdsStatus PROG_RequestSeed ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |

| | | |
|---|---|---|
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the RequestSeed UDS request. It shall be configured in UDS Tresos Studio plugin for the request SA1 (0x27 0x01) with this exact name and with Callback_Origin set to OTHER. | |

#### 4.5.2.4.190. PROG_RequestTransferExit

| | | |
|---|---|---|
| **Purpose** | UDS callback for RequestTransferExit. | |
| **Synopsis** | `tUdsStatus PROG_RequestTransferExit ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the RequestTransferExit UDS request. It shall be configured in UDS Tresos Studio plugin for the request RTE (0x37) with this exact name and with Callback_Origin set to OTHER. | |

#### 4.5.2.4.191. PROG_ResReprog_CheckSegmentListVerif

| | |
|---|---|
| **Purpose** | Provide next segment to verify. |
| **Synopsis** | `void PROG_ResReprog_CheckSegmentListVerif ( void );` |
| **Description** | This function is called after verification of a segment to get information of the next segment |

#### 4.5.2.4.192. PROG_SBASignatureCheck

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Synopsis** | `void PROG_SBASignatureCheck ( void );` |

### 4.5.2.4.193. PROG_SBASignerInfoCheck

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_SBASignerInfoCheck ( void );` |

### 4.5.2.4.194. PROG_SendKey

| | | |
|---|---|---|
| **Purpose** | UDS callback for SendKey. | |
| **Synopsis** | `tUdsStatus PROG_SendKey ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | UDS status | |
| | `UDS_ACK` | positive response |
| | `UDS_NRC_xxx` | negative response |
| | `UDS_NRC_31` | Input pointer parameters NULL value |
| **Description** | This function handles the SendKey UDS request. It shall be configured in UDS Tresos Studio plugin for the request SA2 (0x27 0x02) with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.195. PROG_SendNRC78

| | |
|---|---|
| **Purpose** | Send the NRC_78 instantaneously to gain time. |
| **Synopsis** | `void PROG_SendNRC78 ( void );` |

### 4.5.2.4.196. PROG_Send_NRC

| | | |
|---|---|---|
| **Purpose** | Called to send negative response. | |
| **Synopsis** | `void PROG_Send_NRC ( tUdsStatus eUdsStatus );` | |
| **Parameters (in)** | `eUdsStatus` | Error code to use in negative response |

#### 4.5.2.4.197. PROG_SetKeyNBIDValue

| | | |
|---|---|---|
| **Purpose** | Retrieve the Key NBID value store in memory. | |
| **Synopsis** | `void PROG_SetKeyNBIDValue ( u16 uwKeyNBID );` | |
| **Parameters (in)** | `uwKeyNBID` | New value of the Key NBID to be store in memory |
| **Description** | Callback is called: during reprogramming process to update the key NBID of the downloaded application stored in NVM.<br>Callback shall implement: Writting in non volatile memory of the Key NBID value | |

#### 4.5.2.4.198. PROG_SetNBIDValue

| | | |
|---|---|---|
| **Purpose** | Retrieve the NBID value store in memory. | |
| **Synopsis** | `void PROG_SetNBIDValue ( u16 uwNBID );` | |
| **Parameters (in)** | `uwNBID` | New value of the NBID to be store in memory |
| **Description** | Callback is called: during reprogramming process to update the NBID of the downloaded application stored in NVM.<br>Callback shall implement: Writting in non volatile memory of the NBID value | |

#### 4.5.2.4.199. PROG_SetNetworkStatus

| | | |
|---|---|---|
| **Purpose** | Inform of network status change. | |
| **Synopsis** | `void PROG_SetNetworkStatus ( u16 uwNetworkStatus );` | |
| **Parameters (in)** | `uwNetworkStatus` | new network status |
| **Description** | This function is called on change of the network status | |

#### 4.5.2.4.200. PROG_SetProgrammingStatus

| | |
|---|---|
| **Purpose** | Update the programming status. |
| **Synopsis** | `void PROG_SetProgrammingStatus ( u32 ulProgrammingStatusMask , tProgBoolean ubStatus );` |

| | | |
|---|---|---|
| **Parameters (in)** | `ulProgrammingStatusMask` | the mask for a specific failure |
| | `ubStatus` | failure status (0 - present, 1 - not present) |

### 4.5.2.4.201. PROG_SetSBIFlagValue

| | | |
|---|---|---|
| **Purpose** | Set the SBI flag value in secure memory. | |
| **Synopsis** | `void PROG_SetSBIFlagValue ( u8 ubSBI );` | |
| **Parameters (in)** | `ubSBI` | New value of the SBI flag to be stored in memory |
| **Description** | Callback is called: during startup to update the SBI flag for the SBA ticket stored in Secure RAM.<br>Callback shall implement: Writing in secure volatile memory of the SBI flag value | |

### 4.5.2.4.202. PROG_SignatureCheck

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_SignatureCheck ( void );` |

### 4.5.2.4.203. PROG_SignerInfoCheck

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | `void PROG_SignerInfoCheck ( void );` |

### 4.5.2.4.204. PROG_SimulateOpenProgSession

| | |
|---|---|
| **Purpose** | Request to simulation a programming session opening. |
| **Synopsis** | `void PROG_SimulateOpenProgSession ( void );` |

### 4.5.2.4.205. PROG_SkipPage

| | |
|---|---|
| **Purpose** | Called by the Flash driver to know if the page can be written. |

| | | |
|---|---|---|
| **Synopsis** | u8 **PROG_SkipPage** ( u32 * uAddr ); | |
| **Parameters (in)** | uAddr | pointer on Address of the page to write |
| **Return Value** | Result of the check | |
| | PROG_NO_SKIP_PAGE | page can be written |
| | others | write is not allowed |
| **Description** | This API is called by the Flash driver to know if it is authorized to write a Flash pages.<br>The API will forward the request to the framework | |

#### 4.5.2.4.206. PROG_Streaming

| | |
|---|---|
| **Purpose** | Called in Streaming state. |
| **Synopsis** | void **PROG_Streaming** ( void ); |

#### 4.5.2.4.207. PROG_StreamingFrameReceived

| | | |
|---|---|---|
| **Purpose** | Reception of a streaming frame. | |
| **Synopsis** | void **PROG_StreamingFrameReceived** ( u16 ulReceivedDataLength , u8 * aubData ); | |
| **Parameters (in)** | ulReceivedDataLength | Data Length in the received frame |
| | aubData | Pointer to buffer where received data are located |
| **Description** | This function is called on reception of a streaming frame | |

#### 4.5.2.4.208. PROG_SwitchApplicationMode

| | |
|---|---|
| **Purpose** | |
| **Synopsis** | void **PROG_SwitchApplicationMode** ( void ); |

#### 4.5.2.4.209. PROG_SwitchApplicationModeInd

| | |
|---|---|
| **Purpose** | Called before Bootloader perform a jump to application. |

| | |
|---|---|
| **Synopsis** | `void PROG_SwitchApplicationModeInd ( void );` |
| **Description** | Callback is called: Before jumping to Application<br>Callback shall implement: customer code that shall be executed before jumping to application |

### 4.5.2.4.210. PROG_TD

| | | |
|---|---|---|
| **Purpose** | Called in TD state. | |
| **Synopsis** | `tProgStatus PROG_TD ( void );` | |
| **Return Value** | | |

### 4.5.2.4.211. PROG_TpRxInd

| | | |
|---|---|---|
| **Purpose** | called on frame reception | |
| **Synopsis** | `void PROG_TpRxInd ( tTpMsgIdx uMsgIdx , u8 ebStatus );` | |
| **Parameters (in)** | `uMsgIdx` | Identifier of the transmission frame |
| | `ebStatus` | status of the transmission |
| **Description** | This function is called on a diagnostic frame reception | |

### 4.5.2.4.212. PROG_TpStartOfReceptionInd

| | | |
|---|---|---|
| **Purpose** | called on frame reception | |
| **Synopsis** | `void PROG_TpStartOfReceptionInd ( u8 ubStatus );` | |
| **Parameters (in)** | `ubStatus` | status of the transmission |
| **Description** | This function is called on a diagnostic frame reception | |

### 4.5.2.4.213. PROG_TpTxConf

| | |
|---|---|
| **Purpose** | called on confirmation of frame transmission |
| **Synopsis** | `void PROG_TpTxConf ( tTpMsgIdx uMsgIdx , u8 ebStatus );` |

| | | |
|---|---|---|
| **Parameters (in)** | uMsgIdx | Identifier of the transmission frame |
| | ebStatus | status of the transmission |
| **Description** | This function is called on confirmation of a diagnostic response transmission | |

### 4.5.2.4.214. PROG_TransferData

| | | |
|---|---|---|
| **Purpose** | UDS callback for TransferData. | |
| **Synopsis** | `tUdsStatus PROG_TransferData ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | puwLen | pointer on data length |
| | aubUdsData | pointer on data |
| **Return Value** | UDS status | |
| | UDS_ACK | positive response |
| | UDS_NRC_xxx | negative response |
| | UDS_NRC_31 | Input pointer parameters NULL value |
| **Description** | This function handles the TransferData UDS request. It shall be configured in UDS Tresos Studio plugin for the request TD (0x36) with this exact name and with Callback_Origin set to OTHER. | |

### 4.5.2.4.215. PROG_UpdatePSI

| | | |
|---|---|---|
| **Purpose** | UDS callback for UpdatePSI. | |
| **Synopsis** | `tUdsStatus PROG_UpdatePSI ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | puwLen | pointer on data length |
| | aubUdsData | pointer on data |
| **Return Value** | UDS status | |
| | UDS_ACK | positive response |
| | UDS_NRC_xxx | negative response |
| | UDS_NRC_31 | Input pointer parameters NULL value |
| **Description** | This function handles the UpdatePSI UDS request. It shall be configured in UDS Tresos Studio plugin for the request XXXXXX with this exact name and with Callback_Origin set to OTHER. | |

#### 4.5.2.4.216. PROG_VerificationOnTheFly

| | | |
|---|---|---|
| **Purpose** | Called in CRC state. | |
| **Synopsis** | `tProgStatus PROG_VerificationOnTheFly ( void );` | |
| **Return Value** | | |

#### 4.5.2.4.217. PROG_VerifySectionCrc

| | | |
|---|---|---|
| **Purpose** | Callback verifying the application is correctly downloaded and setting the application validity marker. | |
| **Synopsis** | `tProgCompleteStatus PROG_VerifySectionCrc ( void );` | |
| **Return Value** | Result of check | |
| | `TRUE` | Application is valid |
| | `FALSE` | Application is not valid or not present |
| **Description** | Callback is called: on CheckApplicationValidation routine at the end of the programming sequence.<br>Callback shall implement: verification that each segment is fully programmed and setting of the validity marker if necessary. It then returns the status of the check. | |

#### 4.5.2.4.218. PROG_Write

| | |
|---|---|
| **Purpose** | Called in Write state. |
| **Synopsis** | `void PROG_Write ( void );` |

#### 4.5.2.4.219. PROG_WriteCheck

| | |
|---|---|
| **Purpose** | Called in WriteCheck state. |
| **Synopsis** | `void PROG_WriteCheck ( void );` |

#### 4.5.2.4.220. PROG_WriteFingerprintCheck

| | |
|---|---|
| **Purpose** | Called on loop in WriteFingerprint state. |

| | | |
|---|---|---|
| **Synopsis** | void **PROG_WriteFingerprintCheck** ( void ); | |

### 4.5.2.4.221. Prog_CustomGetAdditionalProgrammingConditionalFlags

| | | |
|---|---|---|
| **Purpose** | get additional programming conditional flags | |
| **Synopsis** | void **Prog_CustomGetAdditionalProgrammingConditionalFlags** ( u8 * pubFlag ); | |
| **Parameters (out)** | pubFlag | pointer to a variable to get the additional programming conditional flag |
| **Description** | This API is called during the processing of the check programming dependencies routine | |

### 4.5.2.4.222. Prog_CustomGetECUInternalProgrammingFlag

| | | |
|---|---|---|
| **Purpose** | get the programming conditions flag | |
| **Synopsis** | void **Prog_CustomGetECUInternalProgrammingFlag** ( u8 * pubFlag ); | |
| **Parameters (out)** | pubFlag | pointer to a variable to get the programming conditions flag |
| **Description** | This API is called during the processing of the check programming dependencies routine | |

### 4.5.2.4.223. Prog_CustomGetProgrammingConditionsFlag

| | | |
|---|---|---|
| **Purpose** | get the ecu internal programming flags | |
| **Synopsis** | void **Prog_CustomGetProgrammingConditionsFlag** ( u8 * pubFlag ); | |
| **Parameters (out)** | pubFlag | pointer to a variable to get the ecu internal programming flags |
| **Description** | This API is called during the processing of the check programming dependencies routine | |

### 4.5.2.4.224. Prog_CustomGetProgrammingTolerantConditionsFlag

| | | |
|---|---|---|
| **Purpose** | get the programming tolerant conditions flag | |

| | | |
|---|---|---|
| **Synopsis** | void **Prog_CustomGetProgrammingTolerantConditionsFlag** ( u8 * pubFlag ); | |
| **Parameters (out)** | pubFlag | pointer to a variable to get the programming tolerant conditions flag |
| **Description** | This API is called during the processing of the check programming dependencies routine | |

### 4.5.2.4.225. Prog_CustomIsProdKeyPresent

| | |
|---|---|
| **Purpose** | Check if the production key is written or not. |
| **Synopsis** | boolean **Prog_CustomIsProdKeyPresent** ( void ); |
| **Return Value** | boolean TRUE : Key exists, FALSE : Key does not exist |
| **Description** | Callback is called: before writing or reading the production key<br>Callback shall implement: check if the key exists or not |

### 4.5.2.4.226. Prog_CustomReadKeyChecksum

| | | |
|---|---|---|
| **Purpose** | Get the checksum of the key. | |
| **Synopsis** | void **Prog_CustomReadKeyChecksum** ( u8 * aubKeyChecksum ); | |
| **Parameters (out)** | aubKeyChecksum | key checksum |
| **Description** | Callback is called: On receiving RDBI of the production key<br>Callback shall implement: get the key checksum from the non volatile memory | |

### 4.5.2.4.227. Prog_CustomWriteKey

| | | |
|---|---|---|
| **Purpose** | write the production key used in signature verification | |
| **Synopsis** | void **Prog_CustomWriteKey** ( u32 ulExponent , u8 * aubModulus , u8 * aubKeyChecksum ); | |
| **Parameters (in)** | ulExponent | key exponent |
| | aubModulus | key modulus |
| | aubKeyChecksum | key checksum |

| | |
|---|---|
| **Description** | Callback is called: On receving WDBI for the production key<br>Callback shall implement: write the key and its checksum in non volatile memory |

### 4.5.2.4.228. Prog_GetEssApplicationStartAddress

| | | |
|---|---|---|
| **Purpose** | Get the address of the jump to the application. | |
| **Synopsis** | `void` **`Prog_GetEssApplicationStartAddress`** `( u32 * ulApplicationStartAddress );` | |
| **Parameters (out)** | `ulApplicationStartAddress` | Application start address |
| **Description** | Callback is called: On switching to application mode<br>Callback shall implement: get the application start address from the ESS | |

### 4.5.2.4.229. Prog_GetEssLength

| | | |
|---|---|---|
| **Purpose** | Get the length of the ESS. | |
| **Synopsis** | `void` **`Prog_GetEssLength`** `( u32 * ulEssLength );` | |
| **Parameters (out)** | `ulEssLength` | ESS length |
| **Description** | Callback is called: On calculating MAC of ESS<br>Callback shall implement: get the length of the ESS | |

### 4.5.2.4.230. Prog_GetEssLogicalBlockId

| | | |
|---|---|---|
| **Purpose** | Get ESS logical block Identifier. | |
| **Synopsis** | `tProgStatus` **`Prog_GetEssLogicalBlockId`** `( u8 ubBlockIndex , u16 * pulBlockIdent );` | |
| **Parameters (in)** | `ubBlockIndex` | Block identifier |
| **Parameters (out)** | `pulBlockIdent` | identifier of the request logical block in ESS |
| **Return Value** | eProgStatus success of the operation | |
| | `PROG_E_OK` | Information is available |
| | `PROG_E_NOT_OK` | Information is not available |

| | |
|---|---|
| **Description** | Callback is called: On RD or Erase routine to identify the downloaded block<br>Callback shall implement: return identifier of the request logical block in ESS |

### 4.5.2.4.231. Prog_GetEssLogicalBlockLength

| | | |
|---|---|---|
| **Purpose** | Get ESS logical block Length. | |
| **Synopsis** | `tProgStatus` **`Prog_GetEssLogicalBlockLength`** `( u8 ubBlockIndex , u32 * pulBlockLength );` | |
| **Parameters (in)** | `ubBlockIndex` | Block identifier |
| **Parameters (out)** | `pulBlockLength` | length of the request logical block in ESS |
| **Return Value** | eProgStatus success of the operation | |
| | `PROG_E_OK` | Information is available |
| | `PROG_E_NOT_OK` | Information is not available |
| **Description** | Callback is called: On RD or Erase routine to identify the downloaded block<br>Callback shall implement: return length of the request logical block in ESS | |

### 4.5.2.4.232. Prog_GetEssLogicalBlockNbr

| | | |
|---|---|---|
| **Purpose** | Get ESS number of logical block. | |
| **Synopsis** | `tProgStatus` **`Prog_GetEssLogicalBlockNbr`** `( u8 * pubBlockNbr );` | |
| **Parameters (out)** | `pubBlockNbr` | number of logical block in ESS |
| **Return Value** | eProgStatus success of the operation | |
| | `PROG_E_OK` | Information is available |
| | `PROG_E_NOT_OK` | Information is not available |
| **Description** | Callback is called: On RD or Erase routine to identify the downloaded block<br>Callback shall implement: return number of logical block in ESS | |

### 4.5.2.4.233. Prog_GetEssLogicalBlockStartAddr

| | |
|---|---|
| **Purpose** | Get ESS logical block address. |

| | | |
|---|---|---|
| **Synopsis** | tProgStatus **Prog_GetEssLogicalBlockStartAddr** ( u8 ubBlockIndex , u32 * pulBlockAddr ); | |
| **Parameters (in)** | `ubBlockIndex` | Block identifier |
| **Parameters (out)** | `pulBlockAddr` | address of the request logical block in ESS |
| **Return Value** | eProgStatus success of the operation | |
| | `PROG_E_OK` | Information is available |
| | `PROG_E_NOT_OK` | Information is not available |
| **Description** | Callback is called: On RD or Erase routine to identify the downloaded block<br>Callback shall implement: return address of the request logical block in ESS | |

### 4.5.2.4.234. Prog_GetEssLogicalBlockVerifTable

| | | |
|---|---|---|
| **Purpose** | Get ESS logical block VBT address. | |
| **Synopsis** | tProgStatus **Prog_GetEssLogicalBlockVerifTable** ( u8 ubBlockIndex , u32 * pulVBTAddress ); | |
| **Parameters (in)** | `ubBlockIndex` | Block identifier |
| **Parameters (out)** | `pulVBTAddress` | address of the verification block table of the request logical block in ESS |
| **Return Value** | eProgStatus success of the operation | |
| | `PROG_E_OK` | Information is available |
| | `PROG_E_NOT_OK` | Information is not available |
| **Description** | Callback is called: On RD to get Verification Block Table location<br>Callback shall implement: return address of the verification block table of the request logical block in ESS | |

### 4.5.2.4.235. Prog_GetEssStartAddr

| | | |
|---|---|---|
| **Purpose** | Get the start address of the ESS. | |
| **Synopsis** | void **Prog_GetEssStartAddr** ( u32 * ulEssStartAddress ); | |
| **Parameters (out)** | `ulEssStartAddress` | ESS start address |
| **Description** | Callback is called: On calculating MAC | |

| | |
|---|---|
| | Callback shall implement: get the start address of the ESS |

#### 4.5.2.4.236. Prog_GetEssValidityStatus

| | | |
|---|---|---|
| **Purpose** | Get ESS validity status. | |
| **Synopsis** | `tProgStatus` **`Prog_GetEssValidityStatus`** `( void );` | |
| **Return Value** | eProgStatus success of the operation | |
| | `PROG_E_OK` | ESS is valid |
| | `PROG_E_NOT_OK` | is not valid |
| **Description** | Callback is called: Before using information from ESS<br>Callback shall implement: ESS validity status | |

#### 4.5.2.4.237. Prog_GetEssVerifTable

| | | |
|---|---|---|
| **Purpose** | Get ESS VBT address. | |
| **Synopsis** | `tProgStatus` **`Prog_GetEssVerifTable`** `( u32 * pulVBTAddress );` | |
| **Parameters (out)** | `pulVBTAddress` | address of the verification block table of the request logical block in ESS |
| **Return Value** | eProgStatus success of the operation | |
| | `PROG_E_OK` | Information is available |
| | `PROG_E_NOT_OK` | Information is not available |
| **Description** | Callback is called: On RD to get Verification Block Table location<br>Callback shall implement: return address of the verification block table of the request logical block in ESS | |

## 4.5.3. Integration notes

### 4.5.3.1. Exclusive areas

Exclusive areas information is not available for this module.

#### 4.5.3.2. Production errors

Production errors information is not available for this module.

#### 4.5.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

Memory mapping information is not available for this module.

#### 4.5.3.4. Integration requirements

**WARNING** ⚠

**Integration requirements list is not exhaustive**
The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the Prog module.

## 4.6. SA

### 4.6.1. Configuration parameters

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| CommonPublishedInformation | 1..1 | **Label:** Common Published Information<br>Common container, aggregated by all modules. It contains published information about vendor and versions. |
| PublishedInformation | 1..1 | **Label:** EB Published Information<br>Additional published parameters not covered by CommonPublishedInformation container. |

| Containers included | | |
|---|---|---|
| General | 1..1 | This container describes the general properties of the node. |

### 4.6.1.1. CommonPublishedInformation

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ArMajorVersion | 1..1 |
| ArMinorVersion | 1..1 |
| ArPatchVersion | 1..1 |
| SwMajorVersion | 1..1 |
| SwMinorVersion | 1..1 |
| SwPatchVersion | 1..1 |
| ModuleId | 1..1 |
| VendorId | 1..1 |
| Release | 1..1 |

| Parameter Name | ArMajorVersion |
|---|---|
| **Label** | AUTOSAR Major Version |
| **Description** | Major version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| Parameter Name | ArMinorVersion |
|---|---|
| **Label** | AUTOSAR Minor Version |
| **Description** | Minor version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |

| | |
|---|---|
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ArPatchVersion** |
|---|---|
| **Label** | AUTOSAR Patch Version |
| **Description** | Patch level version number of AUTOSAR specification on which the appropriate implementation is based on. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMajorVersion** |
|---|---|
| **Label** | Software Major Version |
| **Description** | Major version number of the vendor specific implementation of the module. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 1 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMinorVersion** |
|---|---|
| **Label** | Software Minor Version |
| **Description** | Minor version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 6 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwPatchVersion** |
|---|---|

| | |
|---|---|
| **Label** | Software Patch Version |
| **Description** | Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 3 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ModuleId** |
|---|---|
| **Label** | Numeric Module ID |
| **Description** | Module ID of this module from Module List |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **VendorId** |
|---|---|
| **Label** | Vendor ID |
| **Description** | Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 1 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **Release** |
|---|---|
| **Label** | Release Information |
| **Multiplicity** | 1..1 |
| **Type** | STRING_LABEL |
| **Default value** | |
| **Configuration class** | **PublishedInformation:** |

| | |
|---|---|
| **Origin** | Elektrobit Automotive GmbH |

### 4.6.1.2. PublishedInformation

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| PbcfgMSupport | 1..1 |

| **Parameter Name** | **PbcfgMSupport** |
|---|---|
| **Label** | PbcfgM support |
| **Description** | Specifies whether or not the SA can use the PbcfgM module for post-build support. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

### 4.6.1.3. General

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| MANAGE_PERIOD | 1..1 |
| Seed_Type | 1..1 |
| Security_Algorithm_Type | 1..1 |
| CsmRandomGenerateConfigId | 1..1 |
| CsmRandomSeedConfigId | 1..1 |
| Security_Access_Seed_Length | 1..1 |
| Enable_Static_Seed | 1..1 |
| Compare_Key_Type | 1..1 |

| Parameters included | |
|---|---|
| CsmSignatureVerifyConfigId | 1..1 |
| Security_Access_Key_Length | 1..1 |
| Enable_Antiscanning | 1..1 |
| Security_Access_As_Timer | 1..1 |
| Security_Access_As_Retry_Counter | 1..1 |
| Decomp_Out_Buffer_size | 1..1 |
| Static_Key_0 | 1..1 |
| Static_Key_1 | 1..1 |
| Static_Key_2 | 1..1 |
| Static_Key_3 | 1..1 |
| Static_Key_4 | 1..1 |

| Parameter Name | MANAGE_PERIOD |
|---|---|
| Label | SA Manage Period |
| Description | Period of the periodical SA task.<br>This period must be multiple of EB periodical value in EB module configuration. |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Range | >=1 |
| Origin | EB |

| Parameter Name | Seed_Type |
|---|---|
| Label | Seed Type |
| Description | Specify the type of seed that should be used.<br>▶ Case Standard: Seed is free timer based.<br>▶ Case Cryptographic_Random: Seed is generated using cryptographic random.<br>▶ Case Cryptographic_PUN: Seed is PUN based. |
| Multiplicity | 1..1 |

| | |
|---|---|
| **Type** | ENUMERATION |
| **Default value** | Standard |
| **Range** | Standard<br>Cryptographic_Random<br>Cryptographic_PUN |
| **Origin** | EB |

| **Parameter Name** | **Security_Algorithm_Type** |
|---|---|
| **Label** | Security Algorithm Type |
| **Description** | Specify the security algorithm that should be used.<br>▶ Case Standard: Standard security algorithm of the OEM will be used.<br>▶ Case Custom: custom algorithm implemented by the user in a callback will be sed. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | Standard |
| **Range** | Standard<br>Custom |
| **Origin** | EB |

| **Parameter Name** | **CsmRandomGenerateConfigId** |
|---|---|
| **Label** | CsmRandomGenerateConfigId |
| **Description** | Reference a *CsmRandomGenerate*<br>Dependencies:<br>▶ Reference shall be valid |
| **Multiplicity** | 1..1 |
| **Type** | REFERENCE |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **CsmRandomSeedConfigId** |
|---|---|
| **Label** | CsmRandomSeedConfigId |
| **Description** | Reference a *CsmRandomSeed*<br>Dependencies: |

| | |
|---|---|
| | ▶ Reference shall be valid |
| **Multiplicity** | 1..1 |
| **Type** | REFERENCE |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **Security_Access_Seed_Length** |
|---|---|
| **Label** | Security Access Seed Length |
| **Description** | Specify the size of the seed for SecurityAccess service. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 3 |
| **Origin** | EB |

| **Parameter Name** | **Enable_Static_Seed** |
|---|---|
| **Label** | Enable Static Seed |
| **Description** | Specify if the Static Seed should be used.<br>▶ Case Disabled: A new Seed is generated on each time a GetSeed request is received.<br>▶ Case Enabled: A new Seed is generated only if the precedent one was used by the tester to generate a key and send an request. Else the last generated seed is returned as response to the GetSeed request. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Compare_Key_Type** |
|---|---|
| **Label** | Compare Key Type |
| **Description** | Specify the type of key comparison that should be used.<br>▶ Case Standard: Byte to byte key compare.<br>▶ Case Verify_Signature: Key signature verification.<br>▶ Case PUN: PUN extracted from received key compared with stored PUN. |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |

| | |
|---|---|
| **Default value** | Standard |
| **Range** | Standard<br>Verify_Signature<br>PUN |
| **Origin** | EB |

| **Parameter Name** | **CsmSignatureVerifyConfigId** |
|---|---|
| **Label** | CsmSignatureVerifyConfigId |
| **Description** | Reference a *CsmSignatureVerify*<br>Dependencies:<br>▶ Reference shall be valid |
| **Multiplicity** | 1..1 |
| **Type** | REFERENCE |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **Security_Access_Key_Length** |
|---|---|
| **Label** | Security Access Key Length |
| **Description** | Specify the size of the key for SecurityAccess service. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 3 |
| **Origin** | EB |

| **Parameter Name** | **Enable_Antiscanning** |
|---|---|
| **Label** | Enable Antiscanning |
| **Description** | Specify if the Anti-scanning is enabled. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Security_Access_As_Timer** |
|---|---|
| **Label** | Security Access As Timer |
| **Description** | Specify the value of the Anti-scanning Lock Timer SecurityAccess service. |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 10000 |
| **Origin** | EB |

| **Parameter Name** | **Security_Access_As_Retry_Counter** |
|---|---|
| **Label** | Security Access As Retry Counter |
| **Description** | Specify the value of the Anti-scanning Retry Counter for SecurityAccess service. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 3 |
| **Origin** | EB |

| **Parameter Name** | **Decomp_Out_Buffer_size** |
|---|---|
| **Label** | Decompression Output Buffer Size |
| **Description** | Size of the Output decompression buffer.<br>This buffer is used to store the data decompressed and will be used for write operation. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x400 |
| **Range** | >=0 |
| **Origin** | EB |

| **Parameter Name** | **Static_Key_0** |
|---|---|
| **Description** | First byte of the constant security key. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x04 |
| **Range** | >=0<br><=255 |
| **Origin** | EB |

| **Parameter Name** | **Static_Key_1** |
|---|---|

| | |
|---|---|
| **Description** | Second byte of the constant security key. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x03 |
| **Range** | >=0<br><=255 |
| **Origin** | EB |

| **Parameter Name** | **Static_Key_2** |
|---|---|
| **Description** | Third byte of the constant security key. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x02 |
| **Range** | >=0<br><=255 |
| **Origin** | EB |

| **Parameter Name** | **Static_Key_3** |
|---|---|
| **Description** | Fouth byte of the constant security key. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x01 |
| **Range** | >=0<br><=255 |
| **Origin** | EB |

| **Parameter Name** | **Static_Key_4** |
|---|---|
| **Description** | Fifth byte of the constant security key. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0x00 |
| **Range** | >=0<br><=255 |

| | |
|---|---|
| **Origin** | EB |

# 4.6.2. Application programming interface (API)

## 4.6.2.1. Type definitions

### 4.6.2.1.1. tAntiscanInfo

| | | |
|---|---|---|
| **Purpose** | | |
| **Type** | struct | |
| **Members** | `u8 ubRetryCnt` | |
| | `u32 ulLockTimer` | |
| | `tSaBoolean ubAsLocked` | |

### 4.6.2.1.2. tDecompressStateType

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

### 4.6.2.1.3. tSaBoolean

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

### 4.6.2.1.4. tSaCsmState

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

#### 4.6.2.1.5. tSaStatus

| | |
|---|---|
| **Purpose** | |
| **Type** | `u8` |

### 4.6.2.2. Macro constants

#### 4.6.2.2.1. LZSS_BREAK_EVEN

| | |
|---|---|
| **Purpose** | |
| **Value** | (u8)( ( 1U + LZSS_INDEX_BIT_COUNT + LZSS_LENGTH_BIT_COUNT ) / 9U ) |

#### 4.6.2.2.2. LZSS_END_OF_STREAM

| | |
|---|---|
| **Purpose** | |
| **Value** | (u8)0U |

#### 4.6.2.2.3. LZSS_INDEX_BIT_COUNT

| | |
|---|---|
| **Purpose** | |
| **Value** | (u8)10U |

#### 4.6.2.2.4. LZSS_LENGTH_BIT_COUNT

| | |
|---|---|
| **Purpose** | |
| **Value** | (u8)4U |

#### 4.6.2.2.5. LZSS_MOD_WINDOW

| | |
|---|---|
| **Purpose** | |
| **Value** | (u16)( ( a ) & ( LZSS_WINDOW_SIZE - 1U ) ) |

#### 4.6.2.2.6. LZSS_WINDOW_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | (u16)((u16)(1U) << LZSS_INDEX_BIT_COUNT ) |

#### 4.6.2.2.7. SA_ANTISCANNING_ENABLED

| | |
|---|---|
| **Purpose** | |
| **Value** | STD_ON |

#### 4.6.2.2.8. SA_AS_LOCK_TIMER

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"num:i(General/Security_Access_As_Timer div General/MANAGE_PERIOD)"!]U |

#### 4.6.2.2.9. SA_AS_MAX_NB_RETRY

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"num:i(General/Security_Access_As_Retry_Counter)"!]U |

#### 4.6.2.2.10. SA_CHALLENGE_BIT

| | |
|---|---|
| **Purpose** | |
| **Value** | 64U |

#### 4.6.2.2.11. SA_COMPARE_KEY_STANDARD

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.6.2.2.12. SA_COMPARE_KEY_TYPE

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | [!IF "General/Compare_Key_Type = 'Verify_Signature'"!]SA_COMPARE_KEY_VERIFY_SIGNATURE[!ELSE!]SA_COMPARE_KEY_STANDARD[!ENDIF!] |

### 4.6.2.2.13. SA_COMPARE_KEY_VERIFY_SIGNATURE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

### 4.6.2.2.14. SA_COMPRESSION_DISABLED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

### 4.6.2.2.15. SA_COMPRESSION_ENABLED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

### 4.6.2.2.16. SA_COMPRESSION_STATE

| | |
|---|---|
| **Purpose** | |
| **Value** | [!WS "5"!][!IF "node:exists(as:modconf('Prog')) and (as:modconf('Prog')/General/Enable_Compression = 'true')"!]SA_COMPRESSION_ENABLED[!ELSE!]SA_COMPRESSION_DISABLED[!ENDIF!][!CR!] |

### 4.6.2.2.17. SA_CRY_EXPONENT_ENABLED

| | |
|---|---|
| **Purpose** | |
| **Value** | STD_OFF |

### 4.6.2.2.18. SA_CSM_RANDOM_GENERATE_ID

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"name(as:ref(as:modconf('SA')/General/CsmRandomGenerateConfigId))"!] |

#### 4.6.2.2.19. SA_CSM_RANDOM_SEED_ID

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"name(as:ref(as:modconf('SA')/General/CsmRandomSeedConfigId))"!] |

#### 4.6.2.2.20. SA_CSM_SIG_VERIFY_ID

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"name(as:ref(as:modconf('SA')/General/CsmSignatureVerifyConfigId))"!] |

#### 4.6.2.2.21. SA_CSM_STATE_INIT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.6.2.2.22. SA_CSM_WRAPPER_43_USED

| | |
|---|---|
| **Purpose** | |
| **Value** | STD_ON |

#### 4.6.2.2.23. SA_DECOMP_COMPLETE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x10U |

#### 4.6.2.2.24. SA_DECOMP_COMPRESSLEN

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x08U |

#### 4.6.2.2.25. SA_DECOMP_COMPRESSPOS

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x04U |

#### 4.6.2.2.26. SA_DECOMP_FINISH

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x40U |

#### 4.6.2.2.27. SA_DECOMP_INIT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.6.2.2.28. SA_DECOMP_IN_PROGRESS

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x20U |

#### 4.6.2.2.29. SA_DECOMP_OUT_BUF_SIZE

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"num:i(General/Decomp_Out_Buffer_size)"!]U |

#### 4.6.2.2.30. SA_DECOMP_UNCOMPRESSED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.6.2.2.31. SA_ERR_NULL_PTR

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x80U |

#### 4.6.2.2.32. SA_E_BUSY

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

#### 4.6.2.2.33. SA_E_NOK_AS_LOCKED

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x05U |

#### 4.6.2.2.34. SA_E_NOK_INVALID_KEY

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x04U |

#### 4.6.2.2.35. SA_E_NOT_OK

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

#### 4.6.2.2.36. SA_E_OK

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.6.2.2.37. SA_FALSE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0U |

#### 4.6.2.2.38. SA_IDLE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

#### 4.6.2.2.39. SA_KEY_LEN

| | |
|---|---|
| **Purpose** | |
| **Value** | ([!"num:dectoint(General/Security_Access_Key_Length)"!]U) |

#### 4.6.2.2.40. SA_MANAGE_PERIOD

| | |
|---|---|
| **Purpose** | |
| **Value** | [!"num:i(General/MANAGE_PERIOD)"!]U |

#### 4.6.2.2.41. SA_PUBLIC_KEY_LENGTH

| | |
|---|---|
| **Purpose** | |
| **Value** | 32U |

#### 4.6.2.2.42. SA_RANDOM_GEN_STATE_GENERATE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x06U |

#### 4.6.2.2.43. SA_RANDOM_GEN_STATE_STANDBY

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x07U |

#### 4.6.2.2.44. SA_RANDOM_NUMBER_LENGTH

| | |
|---|---|
| **Purpose** | |
| **Value** | SA_SEED_LEN |

#### 4.6.2.2.45. SA_SECURITY_ALOGORITHM_CUSTOM

| | |
|---|---|
| **Purpose** | |
| **Value** | 0X01U |

#### 4.6.2.2.46. SA_SECURITY_ALOGORITHM_STANDARD

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | 0X00U |

### 4.6.2.2.47. SA_SECURITY_ALOGORITHM_TYPE

| | |
|---|---|
| **Purpose** | |
| **Value** | [!IF "as:modconf('SA')/General/Security_Algorithm_Type = 'Standard'"!]SA_SECURITY_ALOGORITHM_STANDARD[!ELSE!]SA_SECURITY_ALOGORITHM_CUSTOM[!ENDIF!] |

### 4.6.2.2.48. SA_SEED_CSM_RANDOM

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x01U |

### 4.6.2.2.49. SA_SEED_GEN_STATE_INIT

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x02U |

### 4.6.2.2.50. SA_SEED_GEN_STATE_START

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x03U |

### 4.6.2.2.51. SA_SEED_GEN_STATE_UPDATE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x05U |

### 4.6.2.2.52. SA_SEED_LEN

| | |
|---|---|
| **Purpose** | |

| | |
|---|---|
| **Value** | ([!"num:dectoint(General/Security_Access_Seed_Length)"!]U) |

#### 4.6.2.2.53. SA_SEED_STANDARD

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x00U |

#### 4.6.2.2.54. SA_SEED_TYPE

| | |
|---|---|
| **Purpose** | |
| **Value** | [!IF "as:modconf('SA')/General/Seed_Type = 'Cryptographic_Random'"!]SA_SEED_CSM_RANDOM[!ELSE!]SA_SEED_STANDARD[!ENDIF!] |

#### 4.6.2.2.55. SA_SIGNATURE_CHECK_FINISH

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0AU |

#### 4.6.2.2.56. SA_SIGNATURE_CHECK_STANDBY

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x0CU |

#### 4.6.2.2.57. SA_SIGNATURE_CHECK_START

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x08U |

#### 4.6.2.2.58. SA_SIGNATURE_CHECK_UPDATE

| | |
|---|---|
| **Purpose** | |
| **Value** | 0x09U |

#### 4.6.2.2.59. SA_SIGNATURE_LENGTH

| | |
|---|---|
| **Purpose** | |
| **Value** | SA_KEY_LEN |

#### 4.6.2.2.60. SA_STATIC_KEY_LEN

| | |
|---|---|
| **Purpose** | |
| **Value** | 5U |

#### 4.6.2.2.61. SA_STATIC_SEED_ENABLED

| | |
|---|---|
| **Purpose** | |
| **Value** | STD_ON |

#### 4.6.2.2.62. SA_TRUE

| | |
|---|---|
| **Purpose** | |
| **Value** | 1U |

#### 4.6.2.2.63. SA_USE_CRYPTO

| | |
|---|---|
| **Purpose** | |
| **Value** | STD_ON |

### 4.6.2.3. Objects

#### 4.6.2.3.1. m_aubSAPublicModulus

| | |
|---|---|
| **Purpose** | |
| Type | `const u8` |

### 4.6.2.3.2. m_aubStaticKey

| **Purpose** | |
|---|---|
| Type | `const u8` |

### 4.6.2.3.3. m_ulSAPublicExponent

| **Purpose** | |
|---|---|
| Type | `const u32` |

## 4.6.2.4. Functions

### 4.6.2.4.1. SA_CompareKey

| **Purpose** | Compare the received and calcluated key. | |
|---|---|---|
| **Synopsis** | `tProgStatus` **`SA_CompareKey`** `( const u8 * aubReceivedKey );` | |
| **Parameters (in)** | `aubReceivedKey` | input buffer with the received key from the network |
| **Return Value** | Result of comparison | |
| | `PROG_STATUS_OK` | Both key are the same |
| | `PROG_STATUS_NOT_OK` | Both key are different |
| **Description** | This function is called upon correct SA2 request It compare the received key from the diagnostic request to the one calculated into SA_ComputeSeed It then return the result | |

### 4.6.2.4.2. SA_CsmNotification

| **Purpose** | API is a unique callback called by CSM module to treat random generation, encryption, decryption, hash and signature verification. | |
|---|---|---|
| **Synopsis** | `Std_ReturnType` **`SA_CsmNotification`** `( Csm_ReturnType eCsmResult );` | |
| **Parameters (in)** | `eCsmResult` | Csm treatment result |

| | | |
|---|---|---|
| **Return Value** | | |

### 4.6.2.4.3. SA_CustomGetAsymPublicKey

| | | |
|---|---|---|
| **Purpose** | Get the public key when using asymmetric cryptography. Used in SA_InitCrypto. | |
| **Synopsis** | `void SA_CustomGetAsymPublicKey ( const u8 ** paubPublicModulus , u32 * pulPublicExponent );` | |
| **Parameters (out)** | `paubPublicModulus` | Pointer to public key modulus array |
| | `pulPublicExponent` | Pointer to public key exponent |

### 4.6.2.4.4. SA_CustomGetLastRandomNumber

| | | |
|---|---|---|
| **Purpose** | Get the last random number stored. | |
| **Synopsis** | `void SA_CustomGetLastRandomNumber ( u8 * aubDataRandomNumber );` | |
| **Parameters (out)** | `aubDataRandomNumber` | Generated random number array |
| **Description** | Called in SA_GenerateRandomCallback when Csm is in SA_SEED_GEN_STATE_-START state | |

### 4.6.2.4.5. SA_CustomRestoreAsRetryCnt

| | |
|---|---|
| **Purpose** | API that restores the security access anti-scanning retry counter. |
| **Synopsis** | `u8 SA_CustomRestoreAsRetryCnt ( void );` |
| **Return Value** | Value of security access anti-scanning retry counter |
| **Description** | This callback is called at Bootloader startup (if anti-scanning feature is activated) to get the retry counter value from non-volatile memory. |

### 4.6.2.4.6. SA_CustomStoreAsRetryCnt

| | | |
|---|---|---|
| **Purpose** | API that stores the security access anti-scanning retry counter. | |
| **Synopsis** | `void SA_CustomStoreAsRetryCnt ( u8 RetryCntValue );` | |
| **Parameters (in)** | `RetryCntValue` | Value of security access anti-scanning retry counter |

| | | |
|---|---|---|
| **Description** | This callback is called on reception of SecurityAccess service in case anti-scanning feature is activated. The counter value provided as parameter shall be stored in non-volatile memory. | |

#### 4.6.2.4.7. SA_CustomStoreRandomNumber

| | | |
|---|---|---|
| **Purpose** | Store the generated random number. | |
| **Synopsis** | `void SA_CustomStoreRandomNumber ( u8 * pubDataRandomNumber );` | |
| **Parameters (in)** | `aubDataRandomNumber` | Generated random number array |
| **Description** | Called in SA_GenerateRandomCallback when Csm is in SA_RANDOM_GEN_-STATE_GENERATE state | |

#### 4.6.2.4.8. SA_DecompInputParamInit

| | |
|---|---|
| **Purpose** | reinit decompression input param |
| **Synopsis** | `void SA_DecompInputParamInit ( void );` |
| **Description** | This function Reinitializes the input decompression parameter after each TD |

#### 4.6.2.4.9. SA_DecompWriteDataConfirmation

| | | |
|---|---|---|
| **Purpose** | freed the written data from output buffer | |
| **Synopsis** | `void SA_DecompWriteDataConfirmation ( u16 uwLength );` | |
| **Parameters (in)** | `uwLength` | length written with sucess that shall be freed from output buffer |
| **Description** | This function allows to freed the written data from output buffer | |

#### 4.6.2.4.10. SA_DecompressData

| | | |
|---|---|---|
| **Purpose** | store the data to be decompress | |
| **Synopsis** | `void SA_DecompressData ( u8 * pubData , u16 uwDataLength );` | |
| **Parameters (in)** | `pubData` | Input buffer where the data to decompress are stored |

| | | |
|---|---|---|
| | uwDataLength | Length of data to decompress from the buffer |
| **Description** | This function is used to store the data received from the network to an input buffer. The decompression will be done asynchronously afterward. | |

### 4.6.2.4.11. SA_DecompressInit

| | |
|---|---|
| **Purpose** | Decompression variable initialization. |
| **Synopsis** | `void SA_DecompressInit ( void );` |
| **Description** | This function is called at init and for each request Download to intialize all decompression variable. |

### 4.6.2.4.12. SA_DecompressManage

| | |
|---|---|
| **Purpose** | decompress a byte of the input buffer |
| **Synopsis** | `void SA_DecompressManage ( void );` |
| **Description** | This function Decompress a byte of the input buffer data with LZSS algorithm. Each decompressed byte is then store into an Output buffer |

### 4.6.2.4.13. SA_GetDecompressedData

| | | |
|---|---|---|
| **Purpose** | Accessor to get the decompressed data. | |
| **Synopsis** | `tDecompressStateType SA_GetDecompressedData ( u8 ** pubDecompData , u16 * puwDecompressLength );` | |
| **Parameters (out)** | pubDecompData | Output buffer where the data decompressed are copied |
| | puwDecompressLength | Total length of data decompressed |
| **Return Value** | | |
| **Description** | This function allows to get the decompressed data from the buffer | |

### 4.6.2.4.14. SA_GetSeed

| | |
|---|---|
| **Purpose** | Compute the key from an static key and a random Seed. |

| | | |
|---|---|---|
| **Synopsis** | tSaStatus **SA_GetSeed** ( u8 * aubSeed ); | |
| **Parameters (out)** | aubSeed | Output buffer to send back the random Seed |
| **Return Value** | Result of GetSeed action | |
| **Description** | This function is called upon correct SA1 request It first call SA_GetRandomValue to get a random value. Then it computes the key following an algorithm from this random seed and a static key predefined. The random generated Seed is then send back to lower layer. | |

#### 4.6.2.4.15. SA_GetStatus

| | | |
|---|---|---|
| **Purpose** | API called by PROG module to get the SecurityAccess status. | |
| **Synopsis** | tSaStatus **SA_GetStatus** ( void ); | |
| **Return Value** | | |

#### 4.6.2.4.16. SA_Init

| | |
|---|---|
| **Purpose** | Initialize layer. |
| **Synopsis** | void **SA_Init** ( void ); |
| **Description** | This function initializes the SA layer, shall be called only once at ECU startup |

#### 4.6.2.4.17. SA_Manage

| | |
|---|---|
| **Purpose** | Manage the SA layer periodic task. |
| **Synopsis** | void **SA_Manage** ( void ); |
| **Description** | This function is the periodic function managing the SA layer |

## 4.6.3. Integration notes

### 4.6.3.1. Exclusive areas

Exclusive areas information is not available for this module.

### 4.6.3.2. Production errors

Production errors information is not available for this module.

### 4.6.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

Memory mapping information is not available for this module.

### 4.6.3.4. Integration requirements

**WARNING** ⚠

**Integration requirements list is not exhaustive**

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the SA module.

# 4.7. Uds

## 4.7.1. Configuration parameters

| **Containers included** | | |
|---|---|---|
| **Container name** | **Multiplicity** | **Description** |
| CommonPublishedInformation | 1..1 | **Label:** Common Published Information<br>Common container, aggregated by all modules. It contains published information about vendor and versions. |
| PublishedInformation | 1..1 | **Label:** EB Published Information<br>Additional published parameters not covered by CommonPublishedInformation container. |

| Containers included | | |
|---|---|---|
| General | 1..1 | |
| Session | 8..8 | Allows the configuration of 8 sessions. The "Name" column is a fixed value which has to be compliant with naming below:<br>▶ DEFAULT<br>▶ PROGRAMMING<br>▶ EXTENDED<br>▶ SUPPLIER<br>▶ OTHER_01<br>▶ OTHER_02<br>▶ OTHER_03<br>▶ OTHER_04<br>A total of 8 sessions can be configured. It is mandatory that these sessions are defined in the list. |
| Service | 0..n | This container contains the standard service configuration |
| Supplier_Services | 1..1 | |
| Service_DID | 0..n | This container contains the RDBI, WDBI, IOCBI and RSDBI service configuration. |
| Routine_Control | 0..n | This container contains the Routine Control services configuration.<br>This container contains the Routine Control services configuration. |
| Service_OBD | 0..n | This container contains the OBD services configuration. |

### 4.7.1.1. CommonPublishedInformation

| Parameters included | |
|---|---|
| **Parameter name** | **Multiplicity** |
| ArMajorVersion | 1..1 |
| ArMinorVersion | 1..1 |
| ArPatchVersion | 1..1 |
| SwMajorVersion | 1..1 |
| SwMinorVersion | 1..1 |

| Parameters included | |
|---|---|
| SwPatchVersion | 1..1 |
| ModuleId | 1..1 |
| VendorId | 1..1 |
| Release | 1..1 |

| Parameter Name | ArMajorVersion |
|---|---|
| Label | AUTOSAR Major Version |
| Description | Major version number of AUTOSAR specification on which the appropriate implementation is based on. |
| Multiplicity | 1..1 |
| Type | INTEGER_LABEL |
| Default value | 0 |
| Configuration class | **PublishedInformation:** |
| Origin | Elektrobit Automotive GmbH |

| Parameter Name | ArMinorVersion |
|---|---|
| Label | AUTOSAR Minor Version |
| Description | Minor version number of AUTOSAR specification on which the appropriate implementation is based on. |
| Multiplicity | 1..1 |
| Type | INTEGER_LABEL |
| Default value | 0 |
| Configuration class | **PublishedInformation:** |
| Origin | Elektrobit Automotive GmbH |

| Parameter Name | ArPatchVersion |
|---|---|
| Label | AUTOSAR Patch Version |
| Description | Patch level version number of AUTOSAR specification on which the appropriate implementation is based on. |
| Multiplicity | 1..1 |
| Type | INTEGER_LABEL |
| Default value | 0 |
| Configuration class | **PublishedInformation:** |

| | |
|---|---|
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMajorVersion** |
|---|---|
| **Label** | Software Major Version |
| **Description** | Major version number of the vendor specific implementation of the module. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 3 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwMinorVersion** |
|---|---|
| **Label** | Software Minor Version |
| **Description** | Minor version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 9 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **SwPatchVersion** |
|---|---|
| **Label** | Software Patch Version |
| **Description** | Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 2 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **ModuleId** |
|---|---|
| **Label** | Numeric Module ID |
| **Description** | Module ID of this module from Module List |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 0 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **VendorId** |
|---|---|
| **Label** | Vendor ID |
| **Description** | Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER_LABEL |
| **Default value** | 1 |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

| **Parameter Name** | **Release** |
|---|---|
| **Label** | Release Information |
| **Multiplicity** | 1..1 |
| **Type** | STRING_LABEL |
| **Default value** | |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

### 4.7.1.2. PublishedInformation

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| PbcfgMSupport | 1..1 |

| **Parameter Name** | **PbcfgMSupport** |
|---|---|
| **Label** | PbcfgM support |
| **Description** | Specifies whether or not the Uds can use the PbcfgM module for post-build support. |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Configuration class** | **PublishedInformation:** |
| **Origin** | Elektrobit Automotive GmbH |

### 4.7.1.3. General

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Standard | 1..1 |
| UDS_MANAGE_PERIOD | 1..1 |
| SecurityCheck | 1..1 |
| SecurityFunction | 1..1 |
| RC_NRC_IMPLEMENTATION | 1..1 |
| DID_NRC_IMPLEMENTATION | 1..1 |
| UDS_MAX_DID_MULTI_RDBI | 1..1 |
| SPREC_IN_RESPONSE | 1..1 |
| RESPONSE_PENDING | 1..1 |
| TIMER_RESPONSE_PENDING_CHECK | 1..1 |
| RELOAD_TSTOPDIAG | 1..1 |
| Ext_ResponsePending_Manage_Call | 1..1 |
| UDS_P2_ADJUST | 0..1 |
| P2STAR_ADJUST | 0..1 |

| **Parameter Name** | **Standard** |
|---|---|
| **Description** | Notify the variants OEM in which the plugin is used. |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | ISO |
| **Range** | ISO |
| **Origin** | EB |

| **Parameter Name** | **UDS_MANAGE_PERIOD** |
|---|---|
| **Description** | Specifies the period of the manage task in ms.<br>This period must be multiple of EB periodical value in EB module configuration. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 2 |
| **Range** | >=1 |
| **Origin** | EB |

| **Parameter Name** | **SecurityCheck** |
|---|---|
| **Description** | **Security check feature:**<br>▶ `Activated`: Enable API and internal code to manage NRC_33 (Security Access Denied)<br>▶ `Deactivated`: Disable API and internal code to manage NRC_33 (Security Access Denied) |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **SecurityFunction** |
|---|---|
| **Description** | The name of the function which will be in charge to get the current security level.<br>Note:<br>▶ Its prototype shall be: u8 SecurityFunction_Name(void);<br>Example:<br>▶ PROG_GetSecurityLevel |

| | |
|---|---|
| | ▶ TELE_GetSecurityLevel<br>▶ ... |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Origin** | EB |

| **Parameter Name** | **RC_NRC_IMPLEMENTATION** |
|---|---|
| **Description** | Only for Routine Control service ($31)<br>Define NRC used in case of Service and Sub-service are implemented but not in current session<br>Range: [0x00 ; 0xFF]<br>Default: NRC_31 |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 49 |
| **Range** | <=255<br>>=0 |
| **Origin** | EB |

| **Parameter Name** | **DID_NRC_IMPLEMENTATION** |
|---|---|
| **Description** | Only for service with DID<br>Define NRC used in case of Service and DID are implemented but not in current session<br>Range: [0x00 ; 0xFF]<br>Default: NRC_31 |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 49 |
| **Range** | <=255<br>>=0 |
| **Origin** | EB |

| **Parameter Name** | **UDS_MAX_DID_MULTI_RDBI** |
|---|---|

| | |
|---|---|
| **Description** | Define the maximum number of DIDs allowed into a request.<br>Example:<br>▶ If a maximum of 4 DID is accepted into a request, then UDS_MAX_DID_MULTI_RDBI shall be set to 4. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 1 |
| **Range** | <=1000<br>>=1 |
| **Origin** | EB |

| **Parameter Name** | **SPREC_IN_RESPONSE** |
|---|---|
| **Description** | **SPREC (Session Parameter RECord) functionality**<br>SPREC (Session Parameter RECord) refers to UDS_P2 and UDS_P2STAR values<br>▶ `Activated`: SPREC parameter is sent within the DSC ($10) positive response<br>▶ `Deactivated`: SPREC parameter not present in DSC ($10) positive response |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |
| **Origin** | EB |

| **Parameter Name** | **RESPONSE_PENDING** |
|---|---|
| **Description** | **Response pending functionality**<br>▶ `Activated`: Enable API and internal code to manage NRC_78 (response pending)<br>▶ `Deactivated`: Disable API and internal code to manage NRC_78 (response pending) |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | true |

| | |
|---|---|
| **Origin** | EB |

| **Parameter Name** | **TIMER_RESPONSE_PENDING_CHECK** |
|---|---|
| **Description** | **External NRC_78 (response pending) timeout increment, allowing increment under interrupt using hardware timer**<br>▶ `Activated`: NRC_78 timeout increment is managed externally by calling UDS_ResponsePending_TimCntManage() function<br>▶ `Deactivated`: NRC_78 timeout increment is managed internally in UDS_-Manage call |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **RELOAD_TSTOPDIAG** |
|---|---|
| **Description** | **Reload session timeout functionality**<br>API: `void UDS_ReloadTStopDiag (void)`<br>This function can be called by the customer application, in order to maintain a non standard session opened<br>It allows the ECU to maintain the current session for a duration equal to T_Stop_-Diag (5 s).<br>This functionality can be used for the NRC78 periodic answers, if the request duration is very long.<br>▶ `Activated`: API UDS_ReloadTStopDiag available<br>▶ `Deactivated`: API UDS_ReloadTStopDiag unavailable |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **Ext_ResponsePending_Manage_Call** |
|---|---|
| **Label** | External Response Pending Manage Call |
| **Description** | This parameter allow the user to call the API UDS_ResponsePending_Manage by himself |

| | |
|---|---|
| | This allow to manage the response pending timer separatly than the basic scheduling (e.g specific os task with higher priority) |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **UDS_P2_ADJUST** |
|---|---|
| **Label** | P2 Adjust |
| **Description** | This parameter is used to guarantee that the diagnostic response is available on the bus before reaching P2 by adjusting the current P2ServerMax<br>P2Timing = P2ServerMax - P2Adjust<br>Bootloader will send a response within P2Timing ms after receiving the request.<br>The parameter value is defined in ms and must be a multiple of UDS_MANAGE_PERIOD.<br>This timing is common to all sessions. |
| **Multiplicity** | 0..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Origin** | EB |

| **Parameter Name** | **P2STAR_ADJUST** |
|---|---|
| **Label** | P2* Adjust |
| **Description** | This parameter is used to guarantee that the diagnostic response is available on the bus before reaching P2Star by adjusting the current P2StarServerMax<br>P2StarTiming = P2StarServerMax - P2StarAdjust<br>Bootloader will send a response within P2StarTiming ms after the previous NRC78.<br>The parameter value is defined in ms and must be a multiple of UDS_MANAGE_PERIOD.<br>This timing is common to all sessions. |
| **Multiplicity** | 0..1 |
| **Type** | INTEGER |

| | |
|---|---|
| **Default value** | 0 |
| **Origin** | EB |

### 4.7.1.4. Session

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Identifier | 1..1 |
| UDS_P2 | 1..1 |
| UDS_P2STAR | 1..1 |

| **Parameter Name** | **Identifier** |
|---|---|
| **Description** | Specifies the hexadecimal value of the session<br>Range:[0x00 ; 0xFF] |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Origin** | EB |

| **Parameter Name** | **UDS_P2** |
|---|---|
| **Description** | First transmission timing of the NRC78 after the diagnostic request. (ms)<br>This time must be a multiple of UDS_MANAGE_PERIOD.<br>This timing is common to all sessions. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 50 |
| **Origin** | EB |

| **Parameter Name** | **UDS_P2STAR** |
|---|---|
| **Description** | Periodic transmission of the NRC78. (ms)<br>This time must be a multiple of UDS_MANAGE_PERIOD.<br>This timing is common to all sessions. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |

| | |
|---|---|
| **Default value** | 5000 |
| **Origin** | EB |

### 4.7.1.5. Service

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Service | 1..1 |
| SubService | 1..1 |
| Mode | 1..1 |
| Default | 1..1 |
| Programming | 1..1 |
| Extended | 1..1 |
| Supplier | 1..1 |
| Other_1 | 1..1 |
| Other_2 | 1..1 |
| Other_3 | 1..1 |
| Other_4 | 1..1 |
| Length | 1..1 |
| SecurityLevel | 1..1 |
| Callback | 1..1 |
| Callback_Origin | 1..1 |

| **Parameter Name** | **Service** |
|---|---|
| **Description** | Specifies the service.<br>▶ DSC:Diagnostic Session Control ($10)<br>▶ ER:ECU Reset ($11)<br>▶ SA:Security Access ($27)<br>▶ CC:Communication Control ($28)<br>▶ TP:Tester Present ($3E)<br>▶ RTE:Request Transfert Exit ($37)<br>▶ TD:Transfert Data ($36)<br>▶ RU:Request Upload ($35) |

| | |
|---|---|
| | RD:Request Download ($34)<br>RMBA:Read Memory By Address ($23)<br>WMBA:Write Memory By Address ($3D)<br>RDTCI:Read DTC Information ($19)<br>CDTCI:Clear Diagnostic Information ($14)<br>CDTCS:Control DTC Setting ($85)<br>LC:Link Control ($87) |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | DSC |
| **Range** | DSC<br>ER<br>SA<br>CC<br>TP<br>RTE<br>TD<br>RU<br>RD<br>RMBA<br>WMBA<br>RDTCI<br>CDTCI<br>CDTCS<br>LC |
| **Origin** | EB |

| **Parameter Name** | **SubService** |
|---|---|
| **Description** | Specifies the sub-service.<br>Range:[0x00 ; 0xFF] |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |

| | |
|---|---|
| **Range** | >=0<br><=0xFF |
| **Origin** | EB |

| **Parameter Name** | **Mode** |
|---|---|
| **Description** | **Addressing mode**<br>▶ `Physical`: only physical addressing available for this request<br>▶ `Functional`: only functional addressing available for this request (broadcast)<br>▶ `Both`: physical and functional addressing available for this request |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | Physical |
| **Range** | Physical<br>Functional<br>Both |
| **Origin** | EB |

| **Parameter Name** | **Default** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in default session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Programming** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Programming session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Extended** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Extended session. |

| | |
|---|---|
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Supplier** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Supplier session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_1** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_1 session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_2** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_2 session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_3** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_3 session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_4** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_4 session. |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Length** |
|---|---|
| **Description** | Specifies the length of the request.<br>A null length disable the length filtering. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0 |
| **Origin** | EB |

| **Parameter Name** | **SecurityLevel** |
|---|---|
| **Description** | Specifies the security level required for the request.<br>▶ 0: no check required<br>▶ 1: SA1 / SA2<br>▶ 2: SA3 / SA4<br>▶ ... |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0<br><=63 |
| **Origin** | EB |

| **Parameter Name** | **Callback** |
|---|---|
| **Description** | Specifies the callback name.<br>this callback will be called by UDS if the request is valid (session allowed + length correct + parameters correct) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | |

| | |
|---|---|
| **Origin** | EB |

| **Parameter Name** | **Callback_Origin** |
|---|---|
| **Description** | Select the layer where the callback is defined: EB_cbk, APP_cbk or other one.<br>▶ EB: callback is defined in EB_Prg.c<br>▶ APP: callback is defined in APP_Prg.c<br>▶ OTHER: callback is defined in other file (not APP or EB) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | APP |
| **Range** | EB<br>APP<br>OTHER |
| **Origin** | EB |

### 4.7.1.6. Supplier_Services

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| BA | 1..1 |
| BB | 1..1 |
| BC | 1..1 |
| BD | 1..1 |
| BE | 1..1 |

| **Parameter Name** | **BA** |
|---|---|
| **Description** | The service with SID (service identifier) 0xBA is supplier defined.<br>If you need to use this service, activate the current configuration variable. Then the UDS_CustomSupplier_BA callback will be provided to manage it. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **BB** |
|---|---|
| **Description** | The service with SID (service identifier) 0xBB is supplier defined.<br>If you need to use this service, activate the current configuration variable. Then the UDS_CustomSupplier_BB callback will be provided to manage it. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **BC** |
|---|---|
| **Description** | The service with SID (service identifier) 0xBC is supplier defined.<br>If you need to use this service, activate the current configuration variable. Then the UDS_CustomSupplier_BC callback will be provided to manage it. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **BD** |
|---|---|
| **Description** | The service with SID (service identifier) 0xBD is supplier defined.<br>If you need to use this service, activate the current configuration variable. Then the UDS_CustomSupplier_BD callback will be provided to manage it. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | EB |

| **Parameter Name** | **BE** |
|---|---|
| **Description** | The service with SID (service identifier) 0xBE is supplier defined.<br>If you need to use this service, activate the current configuration variable. Then the UDS_CustomSupplier_BE callback will be provided to manage it. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |

| | |
|---|---|
| **Origin** | EB |

### 4.7.1.7. Service_DID

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Service | 1..1 |
| DID | 1..1 |
| Mode | 1..1 |
| Default | 1..1 |
| Programming | 1..1 |
| Extended | 1..1 |
| Supplier | 1..1 |
| Other_1 | 1..1 |
| Other_2 | 1..1 |
| Other_3 | 1..1 |
| Other_4 | 1..1 |
| Length | 1..1 |
| SecurityLevel | 1..1 |
| Callback | 1..1 |
| Callback_Origin | 1..1 |

| **Parameter Name** | **Service** |
|---|---|
| **Description** | Specifies the service.<br>▶ RDBI:ReadDataByIdentifier ($22)<br>▶ WDBI:WriteDataByIdentifier ($2E)<br>▶ IOCBI:InputOutputControlByIdentifier ($2F)<br>▶ RSDBI:ReadScalingDataByIdentifier ($24) |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | RDBI |
| **Range** | RDBI<br>WDBI |

| | |
|---|---|
| | IOCBI<br>RSDBI |
| **Origin** | EB |

| **Parameter Name** | **DID** |
|---|---|
| **Description** | Specifies the DID.<br>Range:[0x0000 ; 0xFFFF] |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0<br><=0xFFFF |
| **Origin** | EB |

| **Parameter Name** | **Mode** |
|---|---|
| **Description** | **Addressing mode**<br>▶ `Physical`: only physical addressing available for this request<br>▶ `Functional`: only functional addressing available for this request (broadcast)<br>▶ `Both`: physical and functional addressing available for this request |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | Physical |
| **Range** | Physical<br>Functional<br>Both |
| **Origin** | EB |

| **Parameter Name** | **Default** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in default session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |

| | |
|---|---|
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Programming** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Programming session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Extended** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Extended session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Supplier** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Supplier session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_1** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_1 session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_2** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_2 session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| Parameter Name | Other_3 |
|---|---|
| Description | Switches ON if the Service/sub-service is implemented in Other_3 session. |
| Multiplicity | 1..1 |
| Type | BOOLEAN |
| Default value | false |
| Origin | AUTOSAR_ECUC V1.0.0 |

| Parameter Name | Other_4 |
|---|---|
| Description | Switches ON if the Service/sub-service is implemented in Other_4 session. |
| Multiplicity | 1..1 |
| Type | BOOLEAN |
| Default value | false |
| Origin | AUTOSAR_ECUC V1.0.0 |

| Parameter Name | Length |
|---|---|
| Description | Specifies the expected length to accept the request (SID + DID = 3).<br>A null length disable the length filtering. |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 3 |
| Range | >=0 |
| Origin | EB |

| Parameter Name | SecurityLevel |
|---|---|
| Description | Specifies the security level required for the request.<br>▶ 0: no check required<br>▶ 1: SA1 / SA2<br>▶ 2: SA3 / SA4<br>▶ ... |
| Multiplicity | 1..1 |
| Type | INTEGER |
| Default value | 0 |
| Range | >=0 |

| | |
|---|---|
| | <=63 |
| **Origin** | EB |

| **Parameter Name** | **Callback** |
|---|---|
| **Description** | Specifies the callback name.<br>this callback will be called by UDS if the request is valid (session allowed + length correct + parameters correct) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | |
| **Origin** | EB |

| **Parameter Name** | **Callback_Origin** |
|---|---|
| **Description** | Select the layer where the callback is defined: EB_cbk, APP_cbk or other one.<br>▶ EB: callback is defined in EB_Prg.c<br>▶ APP: callback is defined in APP_Prg.c<br>▶ OTHER: callback is defined in other file (not APP or EB) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | APP |
| **Range** | EB<br>APP<br>OTHER |
| **Origin** | EB |

### 4.7.1.8. Routine_Control

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| SubService | 1..1 |
| Mode | 1..1 |
| Routine_Identifier | 1..1 |
| Default | 1..1 |

| Parameters included | |
|---|---|
| Programming | 1..1 |
| Extended | 1..1 |
| Supplier | 1..1 |
| Other_1 | 1..1 |
| Other_2 | 1..1 |
| Other_3 | 1..1 |
| Other_4 | 1..1 |
| Length | 1..1 |
| SecurityLevel | 1..1 |
| Callback | 1..1 |
| Callback_Origin | 1..1 |

| **Parameter Name** | **SubService** |
|---|---|
| **Description** | Only for PSA<br>RoutineControlType<br>▶ 0x01: Start Routine<br>▶ 0x02: Stop Routine<br>▶ 0x03: Status Request |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0 |
| **Origin** | EB |

| **Parameter Name** | **Mode** |
|---|---|
| **Description** | **Addressing mode**<br>▶ Physical: only physical addressing available for this request<br>▶ Functional: only functional addressing available for this request (broadcast)<br>▶ Both: physical and functional addressing available for this request |
| **Multiplicity** | 1..1 |
| **Type** | STRING |

| | |
|---|---|
| **Default value** | Physical |
| **Range** | Physical<br>Functional<br>Both |
| **Origin** | EB |

| **Parameter Name** | **Routine_Identifier** |
|---|---|
| **Description** | Specifies the routine identifier.<br>Range:[0x0000 ; 0xFFFF]<br>Specifies the routine identifier. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0<br><=0xFFFF |
| **Origin** | EB |

| **Parameter Name** | **Default** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in default session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Programming** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Programming session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Extended** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Extended session. |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Supplier** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Supplier session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_1** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_1 session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_2** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_2 session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_3** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_3 session. |
| **Multiplicity** | 1..1 |
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Other_4** |
|---|---|
| **Description** | Switches ON if the Service/sub-service is implemented in Other_4 session. |
| **Multiplicity** | 1..1 |

| | |
|---|---|
| **Type** | BOOLEAN |
| **Default value** | false |
| **Origin** | AUTOSAR_ECUC V1.0.0 |

| **Parameter Name** | **Length** |
|---|---|
| **Description** | Specifies the routine control length to accept the request (SID + subfonction + RI = 4).<br>A null length disable the length filtering. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 4 |
| **Range** | >=0 |
| **Origin** | EB |

| **Parameter Name** | **SecurityLevel** |
|---|---|
| **Description** | Specifies the security level required for the request.<br>▶ 0: no check required<br>▶ 1: SA1 / SA2<br>▶ 2: SA3 / SA4<br>▶ ... |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0<br><=63 |
| **Origin** | EB |

| **Parameter Name** | **Callback** |
|---|---|
| **Description** | Specifies the callback name.<br>this callback will be called by UDS if the request is valid (session allowed + length correct + parameters correct) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |

| | |
|---|---|
| **Default value** | |
| **Origin** | EB |

| **Parameter Name** | **Callback_Origin** |
|---|---|
| **Description** | Select the layer where the callback is defined: EB_cbk, APP_cbk or other one.<br>▶ EB: callback is defined in EB_Prg.c<br>▶ APP: callback is defined in APP_Prg.c<br>▶ OTHER: callback is defined in other file (not APP or EB) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | APP |
| **Range** | EB<br>APP<br>OTHER |
| **Origin** | EB |

### 4.7.1.9. Service_OBD

| **Parameters included** | |
|---|---|
| **Parameter name** | **Multiplicity** |
| Service | 1..1 |
| Mode | 1..1 |
| Length | 1..1 |
| Callback | 1..1 |
| Callback_Origin | 1..1 |

| **Parameter Name** | **Service** |
|---|---|
| **Description** | Specifies the OBD service.<br>▶ OBD_SID_00 ($00)<br>▶ OBD_SID_01 ($01)<br>▶ OBD_SID_02 ($02)<br>▶ OBD_SID_03 ($03)<br>▶ OBD_SID_04 ($04) |

| | |
|---|---|
| | OBD_SID_05 ($05)<br>OBD_SID_06 ($06)<br>OBD_SID_07 ($07)<br>OBD_SID_08 ($08)<br>OBD_SID_09 ($09)<br>OBD_SID_0A ($0A)<br>OBD_SID_0B ($0B)<br>OBD_SID_0C ($0C)<br>OBD_SID_0D ($0D)<br>OBD_SID_0E ($0E)<br>OBD_SID_0F ($0F) |
| **Multiplicity** | 1..1 |
| **Type** | ENUMERATION |
| **Default value** | OBD_SID_00 |
| **Range** | OBD_SID_00<br>OBD_SID_01<br>OBD_SID_02<br>OBD_SID_03<br>OBD_SID_04<br>OBD_SID_05<br>OBD_SID_06<br>OBD_SID_07<br>OBD_SID_08<br>OBD_SID_09<br>OBD_SID_10<br>OBD_SID_11<br>OBD_SID_12<br>OBD_SID_13<br>OBD_SID_14<br>OBD_SID_15 |
| **Origin** | EB |

| **Parameter Name** | **Mode** |
|---|---|

| | |
|---|---|
| **Description** | **Addressing mode**<br>▶ `Physical`: only physical addressing available for this request<br>▶ `Functional`: only functional addressing available for this request (broadcast)<br>▶ `Both`: physical and functional addressing available for this request |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | Physical |
| **Range** | Physical<br>Functional<br>Both |
| **Origin** | EB |

| **Parameter Name** | **Length** |
|---|---|
| **Description** | Specifies the exact length to accept the request.<br>A null length disable the length filtering. |
| **Multiplicity** | 1..1 |
| **Type** | INTEGER |
| **Default value** | 0 |
| **Range** | >=0 |
| **Origin** | EB |

| **Parameter Name** | **Callback** |
|---|---|
| **Description** | Specifies the callback name.<br>this callback will be called by UDS if the request is valid (session allowed + length correct + parameters correct) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | |
| **Origin** | EB |

| **Parameter Name** | **Callback_Origin** |
|---|---|
| **Description** | Select the layer where the callback is defined: EB_cbk, APP_cbk or other one. |

| | |
|---|---|
| | ▶ EB: callback is defined in EB_Prg.c<br>▶ APP: callback is defined in APP_Prg.c<br>▶ OTHER: callback is defined in other file (not APP or EB) |
| **Multiplicity** | 1..1 |
| **Type** | STRING |
| **Default value** | APP |
| **Range** | EB<br>APP<br>OTHER |
| **Origin** | EB |

## 4.7.2. Application programming interface (API)

### 4.7.2.1. Objects

#### 4.7.2.1.1. m_astDiagSrvCfg1

| **Purpose** | structure for Diagnostic services configuration 1, stored in ROM |
|---|---|
| Type | `const tUdsSrvCfg1` |

#### 4.7.2.1.2. m_astDiagSrvCfg2

| **Purpose** | structure for Diagnostic services configuration 2, stored in ROM |
|---|---|
| Type | `const tUdsSrvCfg2` |

#### 4.7.2.1.3. m_astDiagSrvCfg3

| **Purpose** | structure for Diagnostic services configuration 3, stored in ROM |
|---|---|

| | |
|---|---|
| Type | `const tUdsSrvCfg3` |

#### 4.7.2.1.4. m_astDiagSrvCfg5

| | |
|---|---|
| **Purpose** | structure for Diagnostic services configuration 5, stored in ROM |
| Type | `const tUdsSrvCfg5` |

### 4.7.2.2. Functions

#### 4.7.2.2.1. UDS_CbkOnRxRequestInd

| | | |
|---|---|---|
| **Purpose** | Callback for the diagnostic request (configuration). | |
| **Synopsis** | `tUdsStatus` **`UDS_CbkOnRxRequestInd`**<br>`( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | Diagnostic response status | |
| | `UDS_ACK` | Positive response |
| | `UDS_NRC_xxx` | Negative response code. |
| **Description** | The diagnostic request is valid (filtering completed). The callback configured for this service is used. | |

#### 4.7.2.2.2. UDS_CustomPositiveAnswerInd

| | | |
|---|---|---|
| **Purpose** | Notification of positive answer. | |
| **Synopsis** | `void` **`UDS_CustomPositiveAnswerInd`** `( con-`<br>`st u16 * puwLen , const u8 * aubUdsData );` | |
| **Parameters (in)** | `puwLen` | received data length |
| | `aubUdsData` | received data pointer |
| **Description** | Notification in order to give possibility to the user to execute an action before the positive answer transmission. | |

#### 4.7.2.2.3. UDS_CustomSupplier_BA

| | | |
|---|---|---|
| **Purpose** | API that check if a supplier request has been received. | |
| **Synopsis** | `tUdsStatus` **`UDS_CustomSupplier_BA`** `( u16 * puwLen , u8 * aubUdsData , tUdsAddrMode eUdsAddrMode , tUdsAddrType eUdsAddrType );` | |
| **Parameters (in)** | `eUdsAddrMode` | Addressing mode information (PHYSICAL 0x01U , FUNCTIONAL 0x02U , PHYSICAL_FUNCTIONAL 0x03U) |
| | `eUdsAddrType` | Addressing type information (DIAG 0x00U , OBD 0x01U , DIAG_OBD 0x02U) |
| **Parameters (in,out)** | `puwLen` | received data length, response length (no response if 0) |
| | `aubUdsData` | received data pointer, data to transmit |
| **Return Value** | Result of check | |
| | `UDS_ACK` | BA supplier request has been received successfully |
| | `UDS_NRC_xxx` | Negative response code. |
| | `UDS_ERR_COHE` | SDF error detected (no response) |
| **Description** | Callback is called: receiving a BA supplier request.<br>Callback shall implement: BA supplier service | |

#### 4.7.2.2.4. UDS_CustomSupplier_BB

| | | |
|---|---|---|
| **Purpose** | API that check if a supplier request has been received. | |
| **Synopsis** | `tUdsStatus` **`UDS_CustomSupplier_BB`** `( u16 * puwLen , u8 * aubUdsData , tUdsAddrMode eUdsAddrMode , tUdsAddrType eUdsAddrType );` | |
| **Parameters (in)** | `eUdsAddrMode` | Addressing mode information (PHYSICAL 0x01U / FUNCTIONAL 0x02U / PHYSICAL_FUNCTIONAL 0x03U) |
| | `eUdsAddrType` | Addressing type information (DIAG 0x00U / OBD 0x01U / DIAG_OBD 0x02U) |
| **Parameters (in,out)** | `puwLen` | received data length, response length (no response if 0) |

| | | |
|---|---|---|
| | aubUdsData | received data pointer, data to transmit |
| **Return Value** | Result of check | |
| | UDS_ACK | BB supplier request has been received successfully |
| | UDS_NRC_xxx | Negative response code. |
| | UDS_ERR_COHE | SDF error detected (no response) |
| **Description** | Callback is called: receiving a BB supplier request.<br>Callback shall implement: BB supplier service | |

#### 4.7.2.2.5. UDS_CustomSupplier_BC

| | | |
|---|---|---|
| **Purpose** | API that check if a supplier request has been received. | |
| **Synopsis** | tUdsStatus **UDS_CustomSupplier_BC** ( u16 * puwLen , u8 * aubUdsData , tUdsAddrMode eUdsAddrMode , tUdsAddrType eUdsAddrType ); | |
| **Parameters (in)** | eUdsAddrMode | Addressing mode information (PHYSICAL 0x01U / FUNCTIONAL 0x02U / PHYSICAL_FUNCTIONAL 0x03U) |
| | eUdsAddrType | Addressing type information (DIAG 0x00U / OBD 0x01U / DIAG_OBD 0x02U) |
| **Parameters (in,out)** | puwLen | received data length, response length (no response if 0) |
| | aubUdsData | received data pointer, data to transmit |
| **Return Value** | Result of check | |
| | UDS_ACK | BC supplier request has been received successfully |
| | UDS_NRC_xxx | Negative response code. |
| | UDS_ERR_COHE | SDF error detected (no response) |
| **Description** | Callback is called: receiving a BC supplier request.<br>Callback shall implement: BC supplier service | |

#### 4.7.2.2.6. UDS_CustomSupplier_BD

| | |
|---|---|
| **Purpose** | API that check if a supplier request has been received. |

| | | |
|---|---|---|
| **Synopsis** | tUdsStatus **UDS_CustomSupplier_BD** ( u16 * puwLen , u8 * aubUdsData , tUdsAddrMode eUdsAddrMode , tUdsAddrType eUdsAddrType ); | |
| **Parameters (in)** | eUdsAddrMode | Addressing mode information (PHYSICAL 0x01U / FUNCTIONAL 0x02U / PHYSICAL_FUNCTIONAL 0x03U) |
| | eUdsAddrType | Addressing type information (DIAG 0x00U / OBD 0x01U / DIAG_OBD 0x02U) |
| **Parameters (in,out)** | puwLen | received data length, response length (no response if 0) |
| | aubUdsData | received data pointer, data to transmit |
| **Return Value** | Result of check | |
| | UDS_ACK | BD supplier request has been received successfully |
| | UDS_NRC_xxx | Negative response code. |
| | UDS_ERR_COHE | SDF error detected (no response) |
| **Description** | Callback is called: receiving a BD supplier request.<br>Callback shall implement: BD supplier service | |

### 4.7.2.2.7. UDS_CustomSupplier_BE

| | | |
|---|---|---|
| **Purpose** | API that check if a supplier request has been received. | |
| **Synopsis** | tUdsStatus **UDS_CustomSupplier_BE** ( u16 * puwLen , u8 * aubUdsData , tUdsAddrMode eUdsAddrMode , tUdsAddrType eUdsAddrType ); | |
| **Parameters (in)** | eUdsAddrMode | Addressing mode information (PHYSICAL 0x01U / FUNCTIONAL 0x02U / PHYSICAL_FUNCTIONAL 0x03U) |
| | eUdsAddrType | Addressing type information (DIAG 0x00U / OBD 0x01U / DIAG_OBD 0x02U) |
| **Parameters (in,out)** | puwLen | received data length, response length (no response if 0) |
| | aubUdsData | received data pointer, data to transmit |
| **Return Value** | Result of check | |
| | UDS_ACK | BE supplier request has been received successfully |
| | UDS_NRC_xxx | Negative response code. |

| | | |
|---|---|---|
| | `UDS_ERR_COHE` | SDF error detected (no response) |
| **Description** | Callback is called: receiving a BE supplier request.<br>Callback shall implement: BE supplier service | |

#### 4.7.2.2.8. UDS_GetCurrentSession

| | | |
|---|---|---|
| **Purpose** | Accessor for the current session. | |
| **Synopsis** | `tUdsSessionType UDS_GetCurrentSession ( void );` | |
| **Return Value** | Session type | |
| | `UDS_SESSION_DEFAULT` | Default session |
| | `UDS_SESSION_PROGRAMMING` | Programming session |
| | `UDS_SESSION_EXTENDED_DIAG` | Extended session |
| | `UDS_SESSION_SUPPLIER` | Supplier session |
| | `UDS_SESSION_OTHER_0x` | Other session (01/02/03/04) |
| **Description** | Accessor for the current session. | |

#### 4.7.2.2.9. UDS_Init

| | |
|---|---|
| **Purpose** | Initialize layer. |
| **Synopsis** | `void UDS_Init ( void );` |
| **Description** | This function initializes UDS layer, shall be called only once at ECU startup. Current sesssion is set to DEFAULT session. Sesssion timeout is deactivated. |

#### 4.7.2.2.10. UDS_IsOBDService

| | | |
|---|---|---|
| **Purpose** | Test if the current service is OBD type. | |
| **Synopsis** | `tUdsBoolean UDS_IsOBDService ( void );` | |
| **Return Value** | Result of treatment | |
| | `UDS_FALSE` | Current service is not OBD type |
| | `UDS_TRUE` | Current service is OBD type |

### 4.7.2.2.11. UDS_LongRequestEnd

| | | |
|---|---|---|
| **Purpose** | Response pending completion: final response received from the application. | |
| **Synopsis** | `void UDS_LongRequestEnd ( u16 uwLen , u8 * aubUdsData , tUdsStatus eStatus );` | |
| **Parameters (in)** | `uwLen` | data length |
| | `aubUdsData` | pointer on response data |
| | `eStatus` | Diagnostic response status (UDS_ACK for positive response, UDS_NRC_xxx for negative response) |
| **Description** | This API shall be called in order to provide an answer after an long processing request. This suppose that a callback (configured on a service) has generated a NRC 78 answer.<br>The complete data buffer must have been updated by the application starting from index0. The 3 first data bytes have been written with "7F <service> 78" on first NACK_-78 response. | |

### 4.7.2.2.12. UDS_LongRequestRespTxConf

| | |
|---|---|
| **Purpose** | Response pending treatment: transmission acknowledgement for intermediate NRC_-78 or final response. |
| **Synopsis** | `void UDS_LongRequestRespTxConf ( void );` |
| **Description** | This API is useful for response pending management. It shall be called on transmission acknowledgement for intermediate NRC_78 message. The aim is to ensure that the transport protocol layer is not already treating NRC_78 message, when the final response is given by the application. If a NRC_78 message is under transmission, the final response transmission is delayed until TxConf notification. |

### 4.7.2.2.13. UDS_LongRequestResponseInd

| | | |
|---|---|---|
| **Purpose** | Response pending management: request for intermediate NRC_78 or final response transmission. | |
| **Synopsis** | `tUdsStatus UDS_LongRequestResponseInd ( u16 uwLen , u8 * aubUdsData );` | |
| **Parameters (in)** | `uwLen` | final length for response |

| | | |
|---|---|---|
| **Parameters (in,out)** | `aubUdsData` | pointer on data |
| **Return Value** | Diagnostic response status | |
| | `UDS_ACK` | Positive response: the transmission is pending |
| | `UDS_NRC_xxx` | Negative response code (just need to be different from UDS_ACK) : the transmission request is rejected for undetailed reason. |
| **Description** | Response pending is in progress. The UDS layer ensure cyclic NRC_78 transmissions and final response transmission. This callback is used to request transmission. | |

#### 4.7.2.2.14. UDS_Manage

| | |
|---|---|
| **Purpose** | Regular tick of the layer. |
| **Synopsis** | `void UDS_Manage ( void );` |
| **Description** | Ensure cyclic tasks of the layer: it manages the session counter and throws timeout notification. If session timeout occurs, the layer automatically switches to the default diagnostic session. |

#### 4.7.2.2.15. UDS_P2AboutToExpireInd

| | |
|---|---|
| **Purpose** | Notification just before the P2/P2_STAR timeout. |
| **Synopsis** | `void UDS_P2AboutToExpireInd ( void );` |
| **Description** | Notification in order to give possibility to the application to execute an action before P2/P2_STAR timeout. |

#### 4.7.2.2.16. UDS_ReloadTStopDiag

| | |
|---|---|
| **Purpose** | Reload session timer. |
| **Synopsis** | `void UDS_ReloadTStopDiag ( void );` |
| **Description** | This function checks the state of the current session (standard or other). If the current session is non standard (diagnostic or programming session for example), the timer T_Stop_Diag is reloaded with its maximum value: m_uUdsSessTimeout. It allows the |

| | |
|---|---|
| | ECU to maintain the current session for a duration equal to T_Stop_Diag (5 s). This functionality can be used for the NRC78 periodic answers, if the request treatment needs more than T_Stop_Diag to be completed. |

### 4.7.2.2.17. UDS_ResponsePending_Manage

| | |
|---|---|
| **Purpose** | Regular tick of the layer. |
| **Synopsis** | `void UDS_ResponsePending_Manage ( void );` |
| **Description** | Ensure cyclic tasks of the layer: response pending, it ensures cyclic NRC_78 message transmissions and final response transmission. |

### 4.7.2.2.18. UDS_ResponsePending_TimCntManage

| | |
|---|---|
| **Purpose** | Response Pending Manage from ISR of STM Timer. |
| **Synopsis** | `void UDS_ResponsePending_TimCntManage ( void );` |
| **Description** | External NRC_78 (response pending) timeout increment, allowing increment under interrupt using hardware timer. This API is called from ISR of STM timer if TIMER_-RESPONSE_PENDING_CHECK is enabled in UDS configuration |

### 4.7.2.2.19. UDS_RxRequest

| | | |
|---|---|---|
| **Purpose** | Treatment of diagnostic request. | |
| **Synopsis** | `tUdsBoolean UDS_RxRequest ( u16 * puwLen , u8 * aubUdsData );` | |
| **Parameters (in,out)** | `puwLen` | pointer on data length (request) |
| | `aubUdsData` | pointer on data (request) |
| **Return Value** | Result of treatment | |
| | `UDS_FALSE` | puwLen and/or aubUdsData are NULL pointers |
| | `UDS_TRUE` | when puwLen is different that zero |
| **Description** | This function performs the processing of a received diagnostic request. It automatically handles the TesterPresent service as well as the status request of routine control request (RC). In the case of a configured service, the function calls the corresponding callback m_astDiagServiceCfg1[].pfuRxRequestInd, m_astDiagServiceCfg2[].pfuRxRequestInd, m_astDiagServiceCfg3[].pfuRxRequestInd or m_astDi- | |

| | |
|---|---|
| | agServiceCfg5[].pfuRxRequestInd In the case of a service not configured, the function automatically sends a negative appropriated response. |

### 4.7.2.2.20. UDS_RxRequestWithAddrMode

| | | |
|---|---|---|
| **Purpose** | Treatment of diagnostic request with addressing mode. | |
| **Synopsis** | `tUdsBoolean` **`UDS_RxRequestWithAddrMode`** `( u16 * puwLen , u8 * aubUdsData , tUdsAddrMode eUdsAddrMode , tUdsAddrType eUdsAddrType );` | |
| **Parameters (in)** | `eUdsAddrMode` | addressing mode (request) (UDS_ADDR_PHYSICAL, UDS_ADDR_FUNCTIONAL) |
| | `eUdsAddrType` | diagnostic type (request) (UDS_TYPE_DIAG, UDS_TYPE_OBD, UDS_TYPE_DIAG_OBD) |
| **Parameters (in,out)** | `puwLen` | pointer on data length |
| | `aubUdsData` | pointer on data |
| **Return Value** | Result of treatment | |
| | `UDS_FALSE` | puwLen and/or aubUdsData are NULL pointers |
| | `UDS_TRUE` | when puwLen is different that zero |
| **Description** | This function performs the processing of a received diagnostic request with addressing mode. It automatically handles the TesterPresent service as well as the status request of routine control request (RC). In the case of a configured service, the function calls the corresponding callback m_astDiagServiceCfg1[].pfuRxRequestInd, m_astDiagServiceCfg2[].pfuRxRequestInd, m_astDiagServiceCfg3[].pfuRxRequestInd or m_astDiagServiceCfg5[].pfuRxRequestInd In the case of a service not configured with physical addressing mode, the function automatically sends a negative appropriated response. In the case of a service not configured with functional addressing mode, a negative answer is transmitted in specific cases only. Addressing mode comply to 14229-1:2005(E) specification. The diagnostic type is also evaluated to reject some services not defined in OBD diagnostic. | |

### 4.7.2.2.21. UDS_SessionStatusInd

| | |
|---|---|
| **Purpose** | Notification for diagnostic session transition. |

| | | |
|---|---|---|
| **Synopsis** | `void` **`UDS_SessionStatusInd`** `( tUdsSessionType eUdsNewSessType , tUdsSessionType eUdsOldSessType , tUdsChangeReason eUdsChangingCause );` | |
| **Parameters (in)** | `eUdsNewSessType` | new session |
| | `eUdsOldSessType` | old session |
| | `eUdsChangingCause` | explicit request (UDS_SESSION_CHANGE_REQUESTED) or session timeout (UDS_SESSION_TIMEOUT) |
| **Description** | It provides old and new sessions, with the reason for the transition. | |

#### 4.7.2.2.22. UDS_StopNRC78Timer

| | |
|---|---|
| **Purpose** | Response pending management: Stop NRC78 timer. |
| **Synopsis** | `void` **`UDS_StopNRC78Timer`** `( void );` |
| **Description** | API used internally in bootloader to stop NRC78 timer while FLASH erasing treatment (code execution from RAM). |

#### 4.7.2.2.23. UDS_StopSessionTimer

| | |
|---|---|
| **Purpose** | Stop session timer. |
| **Synopsis** | `void` **`UDS_StopSessionTimer`** `( void );` |
| **Description** | This function allows to deactivate the T_Stop_Diag timer. |

## 4.7.3. Integration notes

### 4.7.3.1. Exclusive areas

Exclusive areas information is not available for this module.

### 4.7.3.2. Production errors

Production errors information is not available for this module.

### 4.7.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

Memory mapping information is not available for this module.

### 4.7.3.4. Integration requirements

**WARNING** ⚠

**Integration requirements list is not exhaustive**

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the Uds module.

# 5. Bibliography