|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  |  | | |  |
|  | Personal and Portable Profiles | | |  |
|  | F002032 | | |  |
|  |  | | |  |
|  |  | | |  |
| Document Type | **Feature Implementation Specification (FIS)** | | |  |
| Template Version | **6.1a** | | |  |
| Document ID | **VDOC080627** | | |  |
| Document Location | [**Link to Spec in VSEM**](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=F4W5z4zQx3NrTDAAAAAAAAAAAAA&servername=Production_Server) | | |  |
| Document Owner | **Justin Bauer (jbauer50), Walter Stephens (wstephe1)** | | |  |
| Document Version | **1.2** | | |  |
| Document Status | **Draft** | | |  |
| Date Issued | **2021/03/10** | | |  |
| Date Revised | **2021/06/02** | | |  |
| Document Classification | GIS1 Item Number: | **27.60/35** | |  |
| GIS2 Classification: | **Confidential** | |
|  | | | | |
|  | | | | |
| Document Approval | | | | |
| Person | Role | | Email Confirmation | Date |
|  |  | |  |  |
|  |  | |  |  |

This document contains Ford Motor Company Confidential information. Disclosure of the information contained in any portion of this document is not permitted without the expressed, written consent of a duly authorized representative of Ford Motor Company, Dearborn, Michigan, U.S.A.

Copyright © 2019, Ford Motor Company

Printed Copies Are Uncontrolled

**Important Note**

You need to use the RE specification macros provided by the “RE\_SpecificationMacroTemplate.dotm” (refer to “Utilities” on [page “Specification Templates” in the RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Specification+templates)) to allow seamless VSEM import of the specification content. **Use only these RE specification macros to create requirements** in this specification. Refer to “[How to use the Specification Templates](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates?src=contextnavpagetreemode)” on how to enable and use the macros and the requirements templates in this specification.

# Contents

[Contents 3](#_Toc35000551)

[1 Introduction 3](#_Toc35000552)

[1.1 Document Purpose 3](#_Toc35000553)

[1.2 Document Scope 3](#_Toc35000554)

[1.3 Document Audience 3](#_Toc35000555)

[1.3.1 Stakeholder List 3](#_Toc35000556)

[1.4 Document Organization 3](#_Toc35000557)

[1.4.1 Document Context 3](#_Toc35000558)

[1.4.2 Document Structure 3](#_Toc35000559)

[1.5 Document Conventions 3](#_Toc35000560)

[1.5.1 Requirements Templates 3](#_Toc35000561)

[1.5.1.1 Identification of requirements 3](#_Toc35000562)

[1.5.1.2 Requirements Attributes 3](#_Toc35000563)

[1.6 References 3](#_Toc35000564)

[1.6.1 Ford Documents 3](#_Toc35000565)

[1.6.2 External Documents and Publications 3](#_Toc35000566)

[1.7 Glossary 3](#_Toc35000567)

[1.7.1 Definitions 3](#_Toc35000568)

[1.7.2 Abbreviations 3](#_Toc35000569)

[2 Feature Implementation Overview 3](#_Toc35000570)

[2.1 Description 3](#_Toc35000571)

[2.2 Input Requirements/Documents 3](#_Toc35000572)

[2.3 Lessons Learned 3](#_Toc35000573)

[2.4 Assumptions 3](#_Toc35000574)

[3 Feature Implementation Architecture 3](#_Toc35000575)

[3.1 Functional Architecture 3](#_Toc35000576)

[3.1.1 Description 3](#_Toc35000577)

[3.1.2 Function List 3](#_Toc35000578)

[3.1.3 Signal List 3](#_Toc35000579)

[3.2 Physical Architecture 3](#_Toc35000580)

[3.2.1 E/E Architecture 3](#_Toc35000581)

[3.2.1.1 E/E Architecture Variants 3](#_Toc35000582)

[3.2.1.1.1 E/E Architecture “Architecture Variant 1” 3](#_Toc35000583)

[3.2.1.1.2 E/E Architecture “Architecture Variant 2” 3](#_Toc35000584)

[3.2.1.2 E/E Components 3](#_Toc35000585)

[3.2.1.3 E/E Connections 3](#_Toc35000586)

[3.2.1.4 Signal List 3](#_Toc35000587)

[3.2.2 Software Component Architecture 3](#_Toc35000588)

[3.2.2.1 Description 3](#_Toc35000589)

[3.3 Function Deployment 3](#_Toc35000590)

[3.3.1 Deployment Variants 3](#_Toc35000591)

[3.3.1.1 Deployment “Variant 1” 3](#_Toc35000592)

[3.3.2 Function Allocation 3](#_Toc35000593)

[4 Feature Implementation Modeling 3](#_Toc35000594)

[4.1 Component Interaction Diagrams 3](#_Toc35000595)

[4.1.1 Scenario: “System Startup / Shutdown” 3](#_Toc35000596)

[4.1.2 Scenario: “Normal Operation” 3](#_Toc35000597)

[4.2 Component Interface Behavior Diagrams 3](#_Toc35000598)

[5 Feature Implementation Requirements 3](#_Toc35000599)

[5.1 Functional Safety 3](#_Toc35000600)

[5.1.1 ASIL Decomposition of Technical Safety Requirements 3](#_Toc35000601)

[5.2 Requirements on Components 3](#_Toc35000602)

[5.2.1 Component 1 3](#_Toc35000603)

[5.2.1.1 Technology Function “MyLogicalFunctionA\_Component1” 3](#_Toc35000604)

[5.2.1.1.1 Function Interfaces 3](#_Toc35000605)

[5.2.1.1.1.1 Inputs 3](#_Toc35000606)

[5.2.1.1.1.2 Outputs 3](#_Toc35000607)

[5.2.1.1.1.3 Parameters 3](#_Toc35000608)

[5.2.1.1.1.4 Interface Requirements 3](#_Toc35000609)

[5.2.1.1.2 Function Requirements 3](#_Toc35000610)

[5.2.1.1.2.1 Component Specific Requirements 3](#_Toc35000611)

[5.2.1.2 (Technology) Function “MyLogicalFunctionB\_Component1” 3](#_Toc35000612)

[5.2.2 Component 2 3](#_Toc35000613)

[5.3 Requirements on Connections 3](#_Toc35000614)

[5.3.1 Networks 3](#_Toc35000615)

[5.3.1.1 “CAN Bus xxx” 3](#_Toc35000616)

[5.3.1.1.1 Protocol Requirements 3](#_Toc35000617)

[5.3.1.1.2 Electrical Requirements 3](#_Toc35000618)

[5.3.1.2 “LIN Bus xxx” 3](#_Toc35000619)

[5.3.1.2.1 Protocol Requirements 3](#_Toc35000620)

[5.3.1.2.1.1 Schedule Table 3](#_Toc35000621)

[5.3.1.2.2 Electrical Requirements 3](#_Toc35000622)

[5.3.1.3 “Ethernet xxx” 3](#_Toc35000623)

[5.3.2 HW I/Os 3](#_Toc35000624)

[5.3.2.1 “HW I/O xxx” 3](#_Toc35000625)

[5.4 Requirements on Development Process 3](#_Toc35000626)

[6 Open Concerns 3](#_Toc35000627)

[7 Revision History 3](#_Toc35000628)

[8 Appendix 3](#_Toc35000630)

[8.1 Data Dictionary 3](#_Toc35000631)

[8.1.1 Logical Signals 3](#_Toc35000632)

[8.1.2 Logical Parameters 3](#_Toc35000633)

[8.1.3 Technical Signals 3](#_Toc35000634)

[8.1.3.1 GSDB Signals 3](#_Toc35000635)

[8.1.3.2 HW I/Os 3](#_Toc35000636)

[8.1.3.3 Diagnostic Interfaces 3](#_Toc35000637)

[8.1.3.3.1 DTCs 3](#_Toc35000638)

[8.1.3.3.2 DIDs 3](#_Toc35000639)

[8.1.4 Technical Parameters 3](#_Toc35000640)

[8.1.5 Mappings 3](#_Toc35000641)

[8.1.6 Technical Interfaces 3](#_Toc35000642)

[8.1.6.1 AIS Interfaces 3](#_Toc35000643)

[8.1.6.1.1 Publisher Interfaces 3](#_Toc35000644)

[8.1.6.1.2 Subscriber Interfaces 3](#_Toc35000645)

[8.1.6.2 AUTOSAR Ports 3](#_Toc35000646)

[8.1.7 Messages/APIs 3](#_Toc35000647)

[8.1.7.1 CAN Bus “<Bus Name>” 3](#_Toc35000648)

[8.1.7.2 LIN Bus “<Bus Name>” 3](#_Toc35000649)

[8.1.7.3 AUTOSAR Interfaces 3](#_Toc35000650)

[8.1.7.4 SOA Service Contracts 3](#_Toc35000651)

[8.1.8 Encoding Types 3](#_Toc35000652)

**List of Figures**

[Figure 3‑1: Functional Architecture 3](#_Toc35000653)

[Figure 3‑2 E/E Architecture (SysML Style) 3](#_Toc35000654)

[Figure 3‑3 E/E Architecture (Network Topology Style) 3](#_Toc35000655)

[Figure 3‑4: AUTOSAR compliant SW Component Architecture 3](#_Toc35000656)

**List of Tables**

[Table 1‑1: Electrical Architecture(s) referenced in this document 3](#_Toc35000657)

[Table 1‑2: Ford internal Documents 3](#_Toc35000658)

[Table 1‑3: External documents and publications 3](#_Toc35000659)

[Table 1‑4: Definitions used in this document 3](#_Toc35000660)

[Table 1‑5: Abbreviations used in this document. 3](#_Toc35000661)

[Table 6: Input Requirements/Documents 3](#_Toc35000662)

[Table 3‑1: List of Functions 3](#_Toc35000663)

[Table 3‑2: Electrical Components 3](#_Toc35000664)

[Table 3‑3: E/E Connections 3](#_Toc35000665)

[Table 3‑4: Sample Deployment Diagram 3](#_Toc35000666)

[Table 3‑5: Function Allocation Table (Basic) 3](#_Toc35000667)

[Table 3‑6: Function Allocation Table (Functional Safety Extension) 3](#_Toc35000668)

[Table 5‑1: ASIL Decomposition Table 3](#_Toc35000669)

[Table 5‑2: Input Signal mappings of Function “MyLogicalFunctionA\_Component1” 3](#_Toc35000670)

[Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A 3](#_Toc35000671)

[Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1” 3](#_Toc35000672)

[Table 5‑5: Component Specific Requirements 3](#_Toc35000673)

[Table 5‑6: Inherited Requirements 3](#_Toc35000674)

[Table 6‑1: Open Concerns 3](#_Toc35000675)

# Introduction

## Document Purpose

The Feature Implementation Specification (FIS) specifies the deployment of the logical functions of a feature to an electrical architecture. The FIS specifies all interactions between the ECUs of the electrical architecture required for the feature including the technical signals and the interfaces. It also gives interface and integration requirements, which are specific to the feature for the electrical architecture.

To get more information about the concept of feature, function and component level abstraction refer to the [Ford RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Engineering+for+SW+Enabled+Features).

## Document Scope

***#Hint:*** *The FIS can be used to document multiple deployment variants (refer to chapters “Deployment Variants” and “E/E Architecture Variants”). It is however recommended (except for small features) to have a separate FIS for each variant, because managing multiple variants in the same document easily gets complex and cumbersome.*

***#Functional Safety:*** *For Functional Safety specify only one deployment variant per FIS.*

This FIS describes the deployment of the feature <Feature> to the following electrical architecture(s):

| **Electrical Architecture Name** | **Owner** | **Reference** |
| --- | --- | --- |
| FNV3 |  | <Add VSEM Link> |
|  |  |  |

Table 1‑1: Electrical Architecture(s) referenced in this document

## Document Audience

The FIS is authored by Justin Bauer (jbauer50)/Feature Owner. All Stakeholders, i.e., all people who have a valid interest in the feature implementation should read and, if possible, review the FIS. It needs to be guaranteed, that all stakeholders have access to the currently valid version of the FIS.

***#Hint:*** *The FIS template has the IP Classification “Proprietary” by default. IP Classification “Confidential” might be required in some cases, e.g. by Ford Functional Safety.*

***#Macro:***[*Add Ins -> Edit Document Properties macro*](http://wiki.ford.com/display/RequirementsEngineering/Editing+Specification+Properties)

### Stakeholder List

For the latest list of the function stakeholders and their roles & responsibilities refer to [PPP Stakeholder List](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=S5U9Hrgbx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

***#Hint:*** *Refer to* [*Ford RE Wiki – Stakeholder List*](http://wiki.ford.com/display/RequirementsEngineering/Stakeholder+Analysis) *on how to create a stakeholder list. The stakeholder list should be stored in VSEM in the pseudo folder “General Data Artifacts” of the corresponding feature / function / component.*

## Document Organization

### Document Context

Refer to the [Specification Structure page](http://wiki.ford.com/display/RequirementsEngineering/Specification+templates) in the [Ford RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Engineering+for+SW+Enabled+Features) to understand how the FIS relates to other Ford Requirements Documents and Specifications.

### Document Structure

The structure of this document is explained below:

**Section 1** – Introduction – Giving an explanation how to use this document including responsibilities and the scope of the document. Additionally it contains the revision history and a list of unsettled but known issues that have to be consolidated in future versions. It explains the terminology and gives a clarification of the definitions, concepts and abbreviations used in the document.

**Section 2** – Feature Implementation Description – Giving an overview of the platform and listing assumptions, constraints or dependencies

**Section 3** – Feature Implementation Architecture – Describing 3 Architecture Views:

* Functional Architecture – Showing the logical architecture of functions
* Physical Architecture – Showing the physical architecture (first of all the E/E Architecture), which the Logical Functions get allocated to.
* Software Architecture – Showing the software architecture relevant for the feature (for features with in-house development only)
* Function Deployment – Presenting the allocation of logical functions and signals to the electrical and other components

**Section 4** – Deployment Specific Modeling –Modeling techniques providing additional detail on e.g. interface behavior

**Section 5** – Deployment Specific Requirements – Deployment specific requirements for ECUs, Network Communication, and Process

**Section 6** – List of Open Concerns

**Section 7** – Revision History

**Section 8** – Appendix - Presenting additional data mainly in a tabular form, e.g., a data dictionary

## Document Conventions

### Requirements Templates

Refer to “[How to use the Specification Templates](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates?src=contextnavpagetreemode)” on how to use the specification templates and the VBA macros to create/edit the requirements in the specifications.

The VBA macro enable the import of the specification to VSEM (refer to ["How to import specifications into VSEM as separate requirements"](http://wiki.ford.com/pages/viewpage.action?pageId=104991616&src=contextnavpagetreemode)).

#### Identification of requirements

The unique requirement ID given in the headline of any requirement follows the requirement throughout the development process. The requirement ID format follows a well-defined syntax.

All identifiers in an FIS shall be composed of 4 parts:

* A leading prefix, which indicates the type of requirement (R=Requirement, UC=Use Case, SC=Scenario, …)
* A prefix, which indicates the abstraction level (F=Feature, FNC=Function, CMP = component).
* Followed by a name, indicating the scope, which the requirement belongs to (e.g. feature or function name )
* Ending with the actual requirement number

*Example:*

*R\_CMP\_LockArbitrator\_00004* This is the fourth requirement on component level for the function Lock Arbitrator.

#### Requirements Attributes

Additionally attributes can be added to each requirement. This helps to classify requirements. A [list of available attributes](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes?src=contextnavpagetreemode) is given in the RE Wiki.

## References

### Ford Documents

The list of all Ford internal documents, which are directly related.

| **Reference** | **Title** | **Doc. ID** | **Revision** | **Document Location** |
| --- | --- | --- | --- | --- |
|  | PPP Feature Document | VDOC075158 | 1.2 | [VSEM Link](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=TjpBES7Lx3NrTDAAAAAAAAAAAAA&servername=Production_Server) |
|  | PPP Profile Setting Server Functional Specification | VDOC089824 | 1.1 | [VSEM Link](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=FYV5z4zQx3NrTDAAAAAAAAAAAAA&servername=Production_Server) |
|  | PPP Profile Management Server Functional Specification | VDOC085831 | 1.1 | [VSEM Link](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=FYV5z4zQx3NrTDAAAAAAAAAAAAA&servername=Production_Server) |
|  | PPP Profile Interface Client Functional Specification | VDOC089825 | 1.1 | [VSEM Link](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=FYV5z4zQx3NrTDAAAAAAAAAAAAA&servername=Production_Server) |
|  | Enhanced Memory Feature Specification | VDOC041625 | 3.1 | [VSEM Link](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=S$XhIugux3NrTDAAAAAAAAAAAAA&servername=Production_Server) |
|  | Autosave Decide Functional Spec | VDOC081329 | B | [VSEM Link](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=iBY51bdqx3NrTDAAAAAAAAAAAAA&servername=Production_Server) |
|  | Autosave Compare Functional Spec | VDOC081328 | C | [VSEM Link](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=ioe51bdqx3NrTDAAAAAAAAAAAAA&servername=Production_Server) |
|  | Autosave Feature Specification | VDOC080441 | F | [VSEM Link](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=zNX5hzEYx3NrTDAAAAAAAAAAAAA&servername=Production_Server) |

Table 1‑2: Ford internal Documents

### External Documents and Publications

The list of external documents could include books, reports and online sources.

***#Hint:*** *You may refer to* [*IEEE Citation Reference*](http://www.ieee.org/documents/ieeecitationref.pdf) *on how to format a reference.*

| **Reference** | **Document / Publication** |
| --- | --- |
| [bbb] |  |
|  |  |

Table 1‑3: External documents and publications

## Glossary

### Definitions

| **Definition** | **Description** |
| --- | --- |
| Account ID | Ford Account ID stored in the vehicle used to create a link between vehicle profile ID and user's FordPass Account. |
| Anthropometric Translation | Translation of positional settings from one vehicle to another. Example, translating seat positions from a Navigator to a Mustang. |
| CV Profile Management Server | Cloud based profile server that sends profile settings to the vehicle on a profile import and receives profile settings on a profile export. |
| EM Profile Server | Logical block of the Enhanced Memory system that is responsible for determining the active profile. |
| Entry State | The state the feature is in when there is a newly received profile recall request. |
| Feature Number | A hexadecimal value assigned to a profile setting. This was established by the Feature Based Message Protocol (FBMP) and is used by PPP to create a common understanding of what the profile settings are between vehicles. |
| Feature Value | Determine the status of the active setting. For example, Easy Entry/Exit can be configured ON or OFF or a profile, the feature value can be either ON 0x01 or OFF 0x00. |
| HMI Client | The technology used to display a profile setting's current status to the user. Example, radio stations are displayed to the customer using the SYNC screen as an HMI client, but the stations are stored in the Audio Head Unit. |
| Pers Index | The index number provided to a profile in the vehicle. This helps determine which profile is activated throughout the vehicle. |
| Positional Settings | Any profile setting in the vehicle that deals with a physical position, such as seats, mirrors, pedals, heads up display and steering column. |
| Post-entry State | The state after a profile has been recalled and the identity of the user has not been established. |
| Profile Interface Client | The Profile Interface Client is the subsystem that interfaces with the customer inside of the host vehicle. Any functionality that requires a display or an interaction with the customer isl be provided through this subsystem. The Profile Interface Client is also in charge of alerting the customer if there are any errors within the system. |
| Profile Setting Servers | The Profile Setting Server logical block is a generic term for any vehicle module that owns the execution and storage of profile settings. |
| Profile Settings | All user configurable settings in the vehicle that can be added to the user's profile. Examples include radio stations, ADAS settings, Ambient Light, etc. |
| Secondary Authentication Passcode | A passcode that a user configures for their profile. Entering this passcode provides the user full access to their profile. |

Table 1‑4: Definitions used in this document

### Abbreviations

| **Abbr.** | **Stands for** | **Description** |
| --- | --- | --- |
| SPSS | Subsystem Part Specific Specification | Type of requirement document at Ford. |
| PPP | Personal and Portable Profiles | Abbreviation for the feature define in this specification |
| CV | Connected Vehicle | Identifies systems that are part of the cloud. |

Table 1‑5: Abbreviations used in this document.

# Feature Implementation Overview

## Description

**#Hint:** Give a short overview on what E/E systems / platforms the feature is implemented and what special considerations have to be taken into account for E/E systems / platforms.

Personal and Portable Profiles is implemented on FNV3 architecture. FNV3 has ECG which is used as the Profile Management Server for PPP. There are modules within the PPP boundary that communicate with ECG over Ethernet, CAN-FD, HS CAN and MS CAN. PPP implementation must consider the differences of each communication bus.

## Input Requirements/Documents

**#Hint:** *The table below helps the feature owner to collect relevant input* (requirements, documents, mails, models, …) *while writing the spec. When finalizing the spec, the feature owner should check, if all inputs have been properly considered by derived/outgoing requirements* in chapter “Feature Implementation Requirements” *or the architectural elements.*

*Note:* It is not required to list each input requirement individually in this table, referencing the input document is enough (if relevant document section is indicated).

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference**  (Reference as listed in ch. “References”) | **Section/Requirement** | **Description** | **Derived Requirement**  (optional – reference to requirement in ch. “Feature Implementation Requirements”) |
| **Feature/Function Requirements** | | | |
| PPP Feature Document | Whole Document | Feature level requirements detailing desired use cases and high level flow of the feature. |  |
| PPP Profile Setting Server Functional Specification | Whole Document | Functional requirements detailing the role of the Profile Setting Server (settings ECU’s) within the feature logical boundary. |  |
| PPP Profile Management Server Functional Specification | Whole Document | Functional requirements detailing the role of the Profile Management Server (ECG) within the feature logical boundary |  |
| PPP Profile Interface Client Functional Specification | Whole Document | Functional requirements detailing the role of the Profile Interface Client (SYNC) within the feature logical boundary. |  |
| Enhanced Memory Feature Specification | Whole Document | Feature/Functional requirements detailing the current profile system, which is leveraged and expanded in this document. |  |
| Profiles ARL Draft |  | ARL for vehicle profile systems at FORD. |  |
| **Ford Engineering Standards** | | | |
|  | <Example: some SDS (requirement)> |  |  |
|  |  |  |  |
| **Legal Regulations** | | | |
|  | <Example: some excerpt from ECE or FMVSS> |  |  |
|  |  |  |  |
| **Industry Standards** | | | |
|  | <Example: some ISO/IEEE/SAE or other standard> |  |  |
|  |  |  |  |
| **Other Sources** | | | |
|  | <Example: some stakeholder document> |  |  |
|  |  |  |  |

Table 6: Input Requirements/Documents

## Lessons Learned

**#Classification**: Optional

**#Hint:** Additional information and lessons learned from previous development or related features. A typical source for Lessons Learned is the FMA Quality History.

## Assumptions

**#Classification**: Optional

**#Hint:** A list of assumptions concerning the effects/dependencies of the feature’s deployment as well as (e.g. known limitations). During the course of the feature development most of those assumptions are typically either converted into actual requirements or discarded at some point – such that this chapter ideally remains mostly empty.

1. Common interface design between Profile Management Server (ECG) and all Profile Setting Servers (setting ECU’s)

# Feature Implementation Architecture

## Functional Architecture

**#Hint**: This section depicts the Functional Architecture, i.e., the decomposition into Logical Functions. This architectural step is needed to find the right functional partitioning for the function level.

### Description

**#Hint**: Provide some informal description of the characteristics of the chosen Functional Architecture. Also give some graphical representation of the Functional Architecture. Either SysML activity diagrams or [Data Flow Diagrams](http://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemode) could be used to depict such a Functional Architecture.

**#Link:** [*SysML - Activity Diagrams*](https://pd3.spt.ford.com/sites/SystemsEngineering/SEC/sysml-teamsite/SysML%20Wiki/Activity%20Diagram%20Basics.aspx) or [*RE Wiki - Data Flow Diagrams*](http://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemodehttp://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemode)

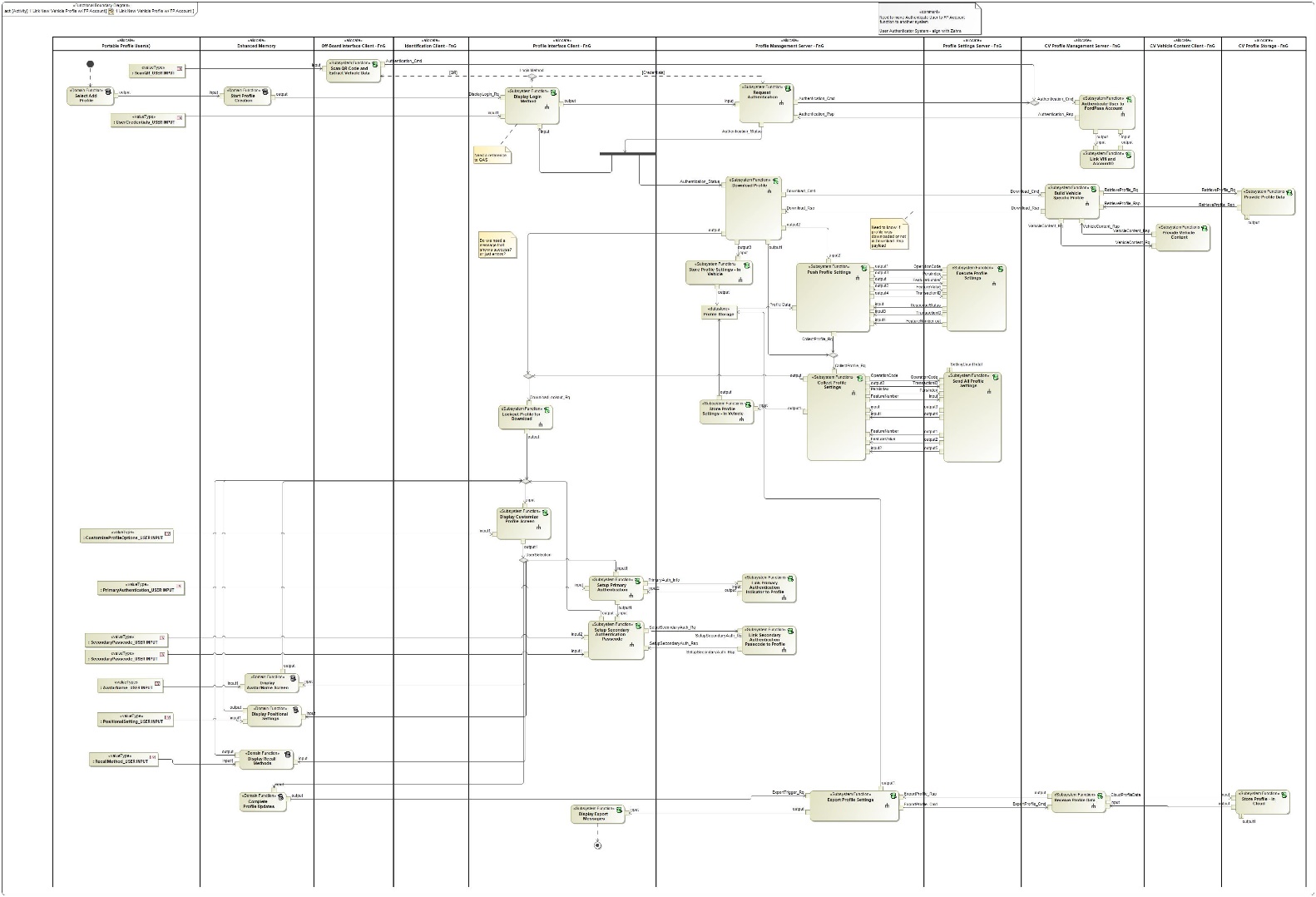


Figure 8: 1 Link New Vehicle Profile w/ FP Account

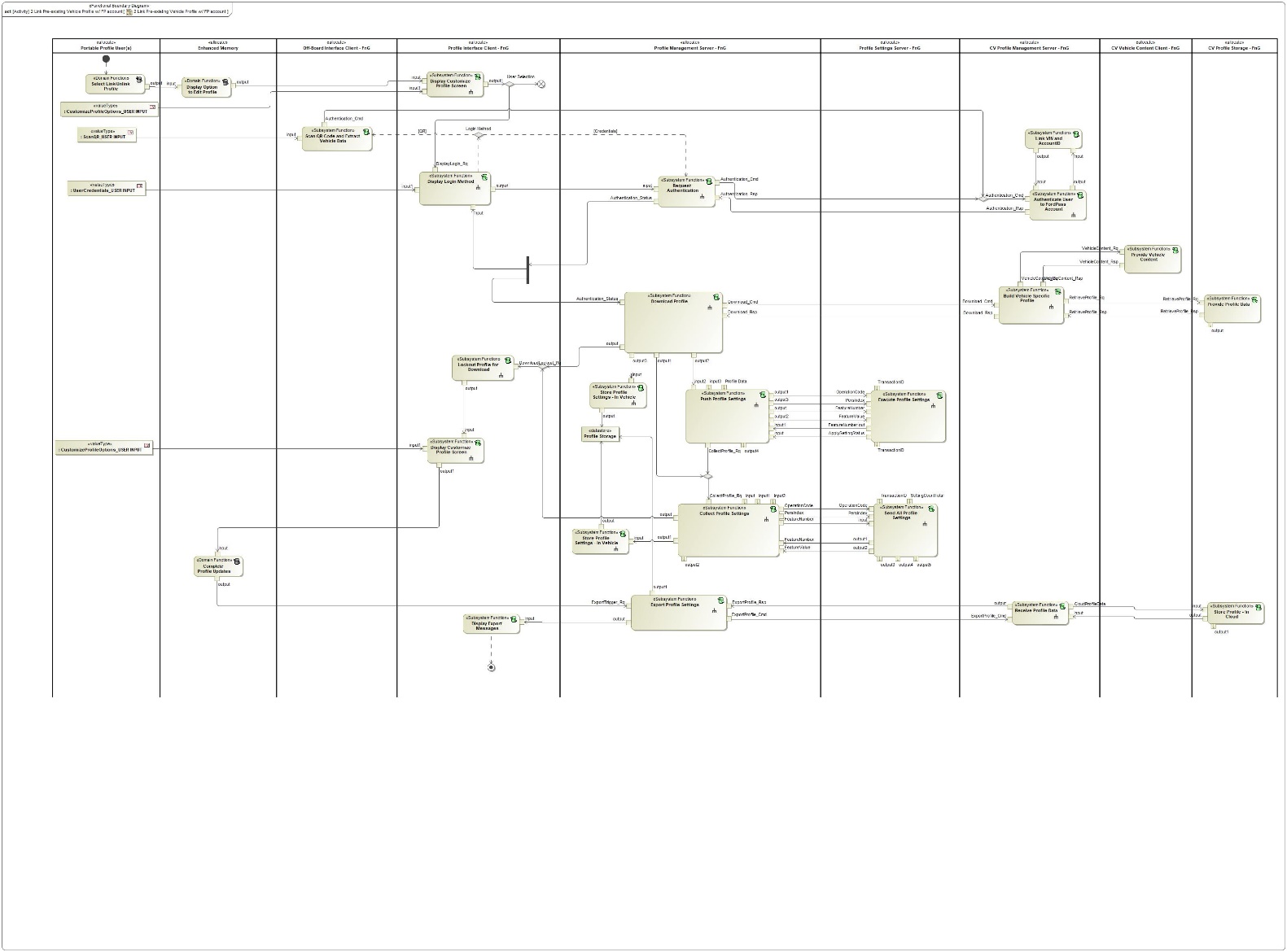


Figure 8: 2 Link Pre-existing Vehicle Profile w/ FP account

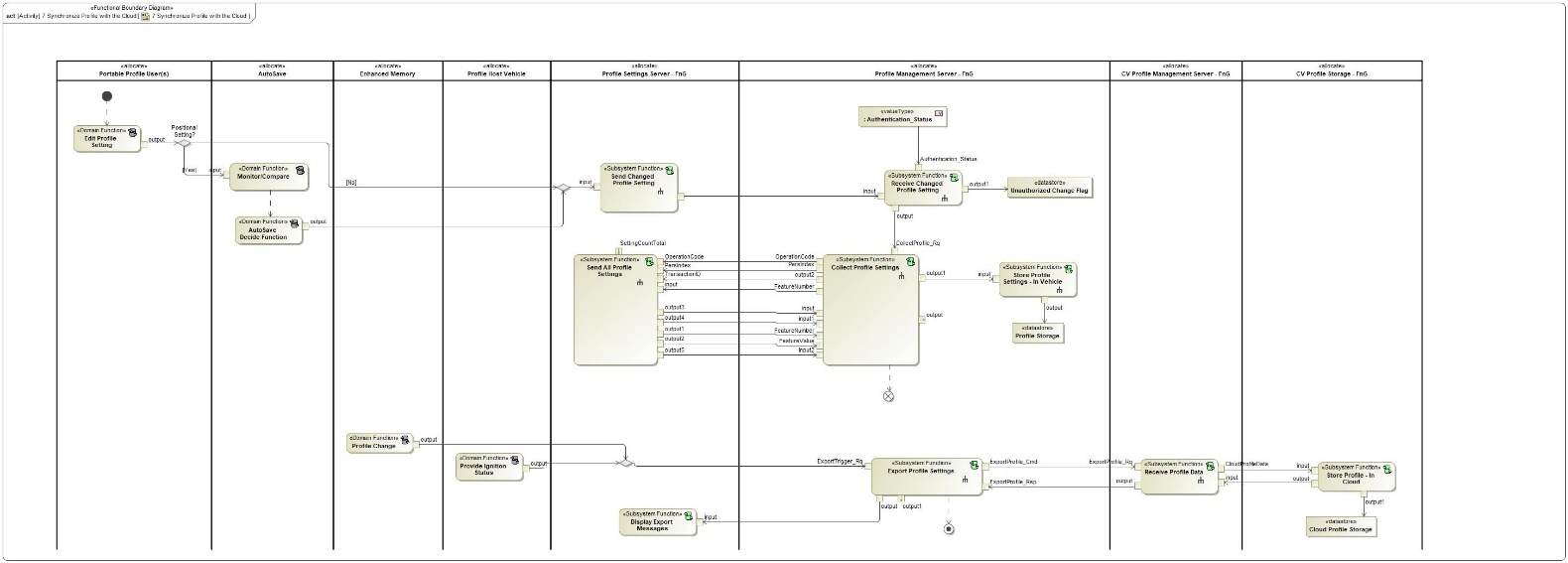


Figure 8: 7 Synchronize Profile with the Cloud

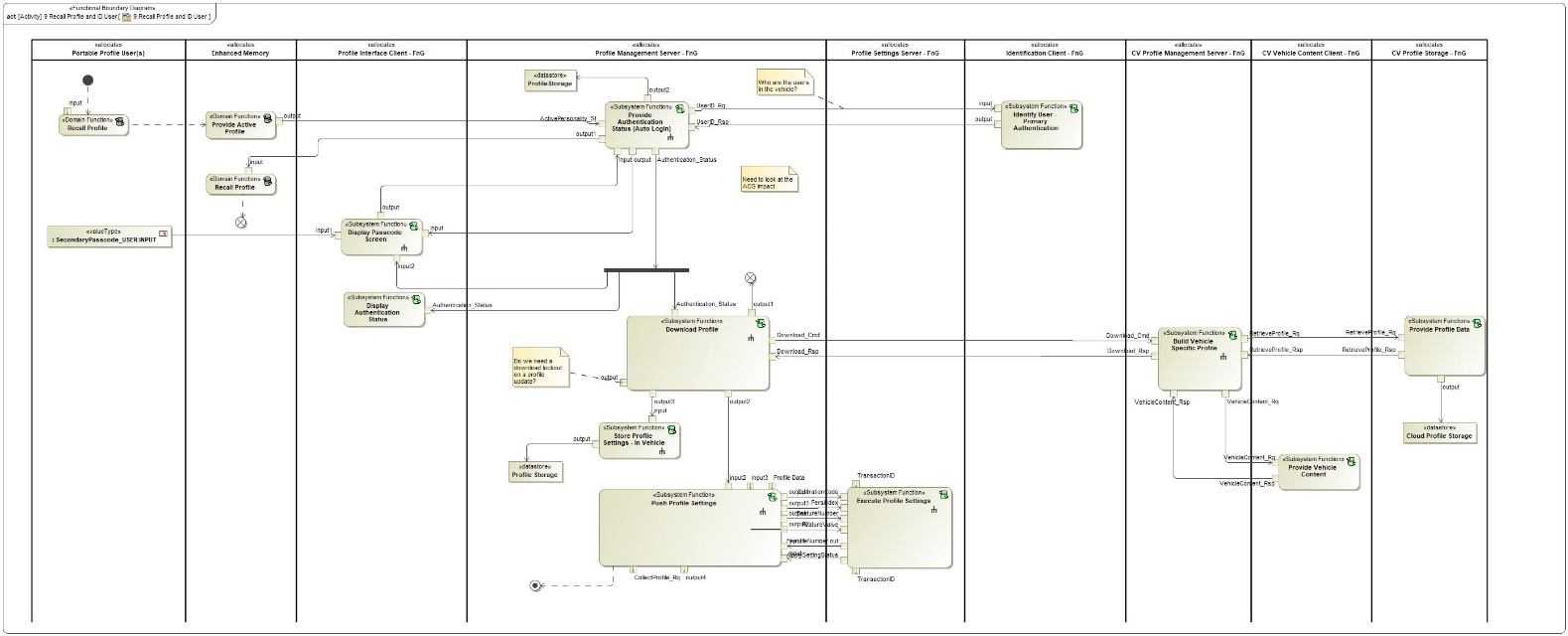


Figure 8: 9 Recall Profile and ID User

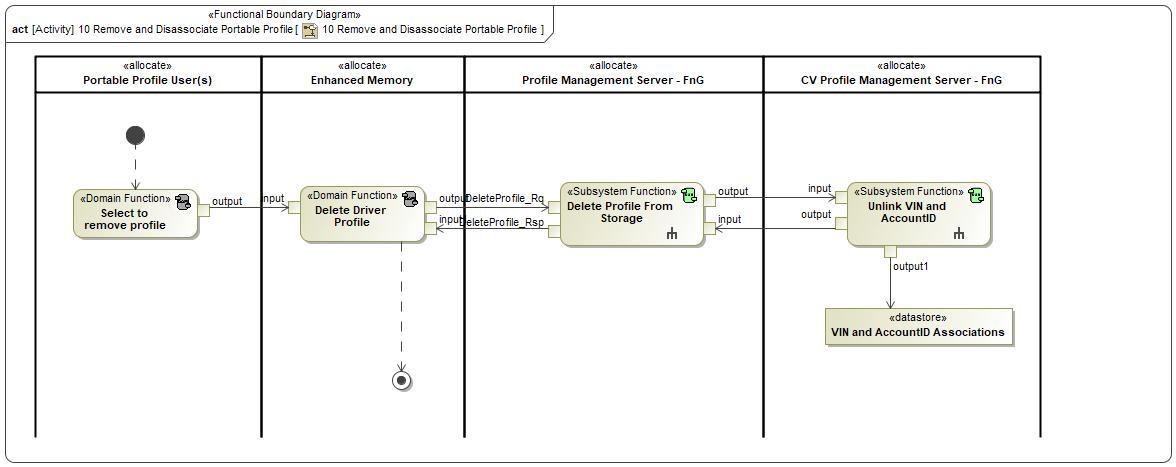


Figure 8: 10 Remove and Disassociate Portable Profile

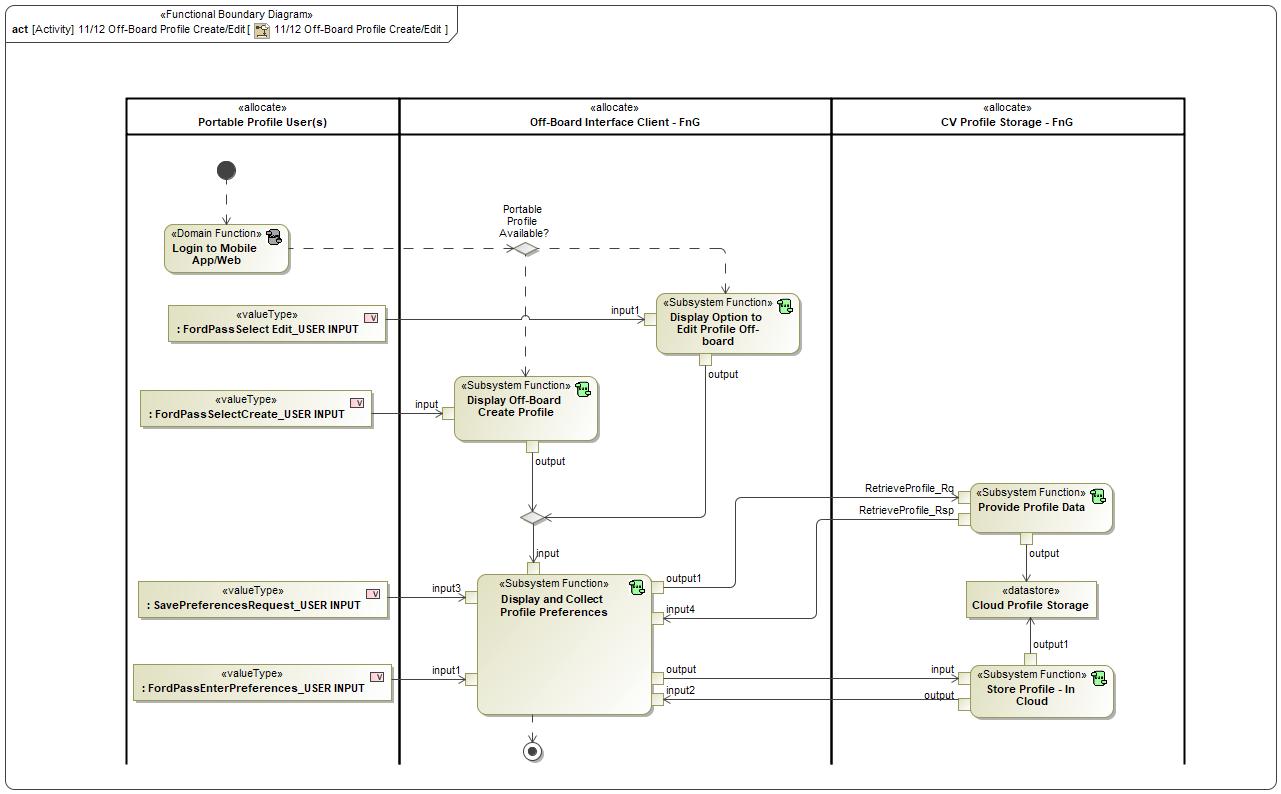


Figure 8: 11/12 Off-Board Profile Create/Edit

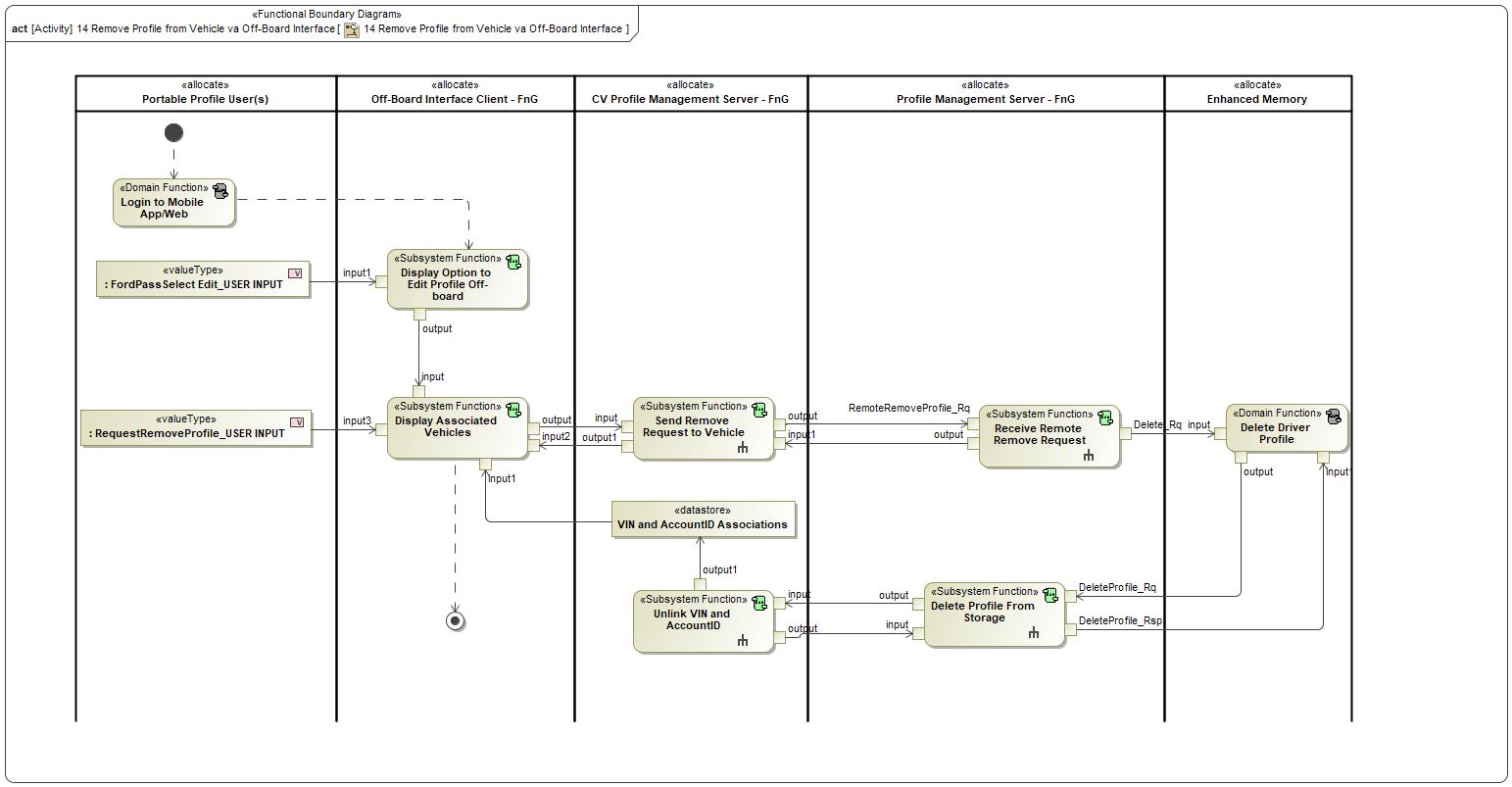


Figure 8: 14 Remove Profile from Vehicle va Off-Board Interface

### Function List

The following functions from the [Global Feature & Function List](https://www.vsemweb.ford.com:443/tc/launchapp?-attach=true&-s=226TCSession&-o=ZmZNi0JHx3NrTDAAAAAAAAAAAAA) are referenced in this Feature Implementation Specification:

|  |  |  |
| --- | --- | --- |
| **Function ID** | **Function Name** | **Function Description** |
| Fn012419/A | Anthropometric Translation - Export | Translates seat profile data from a host vehicle into a normalized seat value. This function uses host vehicle seat data to perform calculation and provides a normalized value as an output. This occurs on each profile export. |
| Fn012439/A | Anthropometric Translation - Import | Translates seat profile data for the applicable vehicle. This function will take a normalized seat position and the host vehicle's seat data and calculate the new seat position as an output. This occurs on an initial profile import. |
| Fn012437/A | Authenticate User to FordPass Account | This function receives the user credentials and authenticates them with the backend system to ensure the credentials are associated to a valid FordPass account. |
| Fn012433/A | Build Vehicle Specific Profile | A cloud function that uses vehicle configuration data and the user's portable profile data to build a vehicle specific profile to be sent to the vehicle. |
| Fn012428/A | Display and Collect Profile Preferences | A function of the Off-Board Interface Client that is used to receive input from the customer to collect profile preferences from outside of the vehicle. This function can be used in the RVS scenario of collecting preferences pre vehicle delivery or when a customer decides to create/edit their profile using the off-board interface. |
| Fn012427/A | Collect Profile Settings | This function collects all in-vehicle profile settings from the Profile Setting Servers. The function interacts with the in-vehicle profile storage to save and store the profile settings. |
| Fn012426/A | Delete Profile From Storage | This function receives a request to delete a profile stored to the vehicle, deletes the profile from storage, and then sends a request to the cloud to disassociate the vehicle from the cloud profile. This function also executes the delete profile when a Master Reset occurs on the vehicle. |
| Fn012461/A | Display Associated Vehicles | This function is allocated to the Off-Board Interface Client and displays the different vehicles that the user's portable profile is currently saved to. The user can select from the list of vehicles and request to remove their profile from that vehicle. |
| Fn012449/A | Display Authentication Status | The function is allocated to the Profile Interface Client which is inside the vehicle. This is an HMI function that displays whether the owner of the active profile has been successfully authenticated by either the primary (PaaK, FR, NFC, etc.) or secondary (PIN) authentication methods. |
| Fn012441/A | Display Login Method | This is an HMI function that is displayed to collect user credentials from the user. The credentials are used to authenticate a user in order to link their profile to their FordPass account and import their profile settings. |
| Fn012464/A | Display Off-Board Create Profile | This function provides the user an interface to create a portable profile from outside a vehicle. |
| Fn012423/A | Display Option to Edit Profile Off-board | Off-Board Interface Client function that provides the user the option to edit a pre-existing portable profile. |
| Fn012454/A | Display Option to Unlink Ford Account | HMI function that provides the capability for the user to unlink their FP account to any pre-existing vehicle profiles. |
| Fn012418/A | Download Profile | This function sends a Download\_Cmd to the cloud to initiate a profile download and receives a profile payload for the authenticated user of the profile via the Download\_Rsp. |
| Fn012467/A | Execute Profile Settings | This is a function of the Profile Setting Servers with the purpose of receiving and applying profile settings that are sent by the Profile Management Server. This function executes profile settings using Feature Number, Feature Value and Pers Index as inputs. The Profile Setting Servers utilize this function when receiving OperationCode = SET from Profile Management Server. |
| Fn012462/A | Export Profile Settings | A function that will export profile settings from the vehicle to the cloud when a new vehicle profile is initially created or when profile settings are changed by the profile user. |
| Fn012445/A | Identify User - Primary Authentication | This function identifies the user that has an active profile in the host vehicle. |
| Fn012459/A | Link Secondary Authentication Passcode to Profile | This function receives a passcode value from the Profile Interface Client and links the passcode to the profile being edited or created. The Secondary Authentication Passcode is intended to be a backup method for a user to be authenticated to their profile. If Primary Authentication fails, Secondary Authentication is prompted. |
| Fn012457/A | Link Primary Authentication Indicator to Profile | This function links a primary authentication indicator (BT device, Face ID, NFC, etc.) to the user's vehicle profile to be used for profile authentication. |
| Fn012420/A | Lockout Profile for Download | This function is allocated to the Profile Interface Client and provides an HMI graphic that locks out the HMI screen so that the customer cannot make any changes while the PPP feature is applying settings. |
| Fn012424/A | Provide Authentication Status (Auto Login) | This function provides the current authentication status of the active profile in the vehicle to other functions of the system. Authentication status is based on the active profile and which user is identified in the vehicle. User identification is provided by the Identification Client. |
| Fn012435/A | Provide Profile Data | This function stores profile data in the cloud and associates the data to a user profile. |
| Fn012429/A | Provide Vehicle Content | This function provides personalized vehicle content for specific vehicles in order to help create a vehicle specific profile. |
| Fn012422/A | Push Profile Settings | This function reads user profile data from the Profile Management Server storage and uses the Personalization Service to send the profile data (Feature Number/Feature Value) to the correct Profile Setting Servers. |
| Fn012447/A | Request Authentication | This function receives the authentication status from the cloud and provides the status to requesting vehicle functions. Function occurs when a customer logs in to a vehicle for the first time. |
| Fn012465/A | Receive Changed Profile Setting | This function receives the values of a changed profile setting for an active profile and arbitrates if the setting should be saved to the profile based on the authentication status of the vehicle user. |
| Fn012450/A | Receive Remote Remove Request | This function receives a remote remove request from the cloud and then sends a request to the Enhanced Memory system to delete the requested profile. |
| Fn012434/A | Setup Secondary Authentication - Passcode | This is an HMI function that receives the secondary authentication passcode from the user. This function is executed when primary authentication of a user is unavailable or fails. |
| Fn012431/A | Receive Profile Data | A cloud function that receives any profile data from the vehicle and passes it along to the cloud profile storage. |
| Fn012436/A | Scan QR Code and Extract Vehicle Data | A function of the Off-Board Interface Client that uses the mobile device's camera to scan a QR code and extract relevant vehicle data to authenticate a user on a specific vehicle. |
| Fn012421/A | Send All Profile Settings | This function sends profile data associated to a user profile that is stored on the Profile Setting Server to the Profile Management Server when the PPPSettings\_Rq with OperationCode = QUERY  is received. The received query operation can be for a specific Feature Number or can be a batch query (FeatureNumber = NULL in PPPSettings\_Rq) of all profile data associated to a profile. |
| Fn012438/A | Send Changed Profile Setting | A function of the Profile Setting Servers that sends a change notification to the Profile Management Server when a change is made to a profile setting of an active profile. The notification does not determine which profile setting was changed, just that a change occurred and the Profile Management Server must query the active profile to refresh the local copy. |
| Fn012443/A | Send Remove Request to Vehicle | A function allocated to the cloud that relays remote remove requests from the Off-Board Interface Client to the Profile Management Server of a specific vehicle. |
| Fn012452/A | Setup Primary Authentication | This is an HMI function that allows the user to setup their primary authentication method for their profile. |
| Fn012442/A | Store Credentials with Profile | This function stores and associates a user's FordPass Credentials with their vehicle profile after the credentials have been authenticated by the cloud. |
| Fn012463/A | Store Profile - In Cloud | Cloud storage of the profile settings that is associated with the user's FordPass account. |
| Fn012455/A | Store Profile Settings - In Vehicle | This function receives profile settings as inputs from multiple function sources and outputs profile settings when the settings are being exported to the cloud or pushed to the Profile Setting Servers. |
| Fn012468/A | Link VIN and AccountID | This is a cloud function that stores all of the VINs that a portable profile is currently available in. |
| Fn012446/A | Unlink FP Account from Vehicle Profile | This function receives an unlink request from the Profile Interface Client, breaks the association between Account ID and vehicle profile, and then sends a request to the cloud to disassociate the vehicle from the cloud profile. |
| Fn012451/A | Unlink VIN and AccountID | A function of the cloud that unlinks the cloud profile from a specific VIN. This occurs when a customer unlinks their FordPass account from a vehicle profile or when the customer deletes their profile from a vehicle. |
|  | Display Customize Profile Screen | This function displays a screen that provides the ability for the user to customize certain parameters/aspects of their profile. |
|  | Display Export Messages | This function communicates any necessary messages to the user while profile settings are being uploaded to the cloud. |
|  | Display Passcode Screen | This function displays a screen that allows the user to enter their passcode. |
|  | Display RVS Option |  |
| Fn012470/A | Display Download Messages | This function communicates any necessary messages to the user while profile settings are being downloaded to the vehicle. |
|  | Restore Profile Settings | This function checks to see if there were any unauthorized changes made to the active profile. If so, the function will request the Profile Management Server to Push the changed profile settings out to the Profile Setting Servers. |

### Signal List

*#Hint: Refer to the Data Dictionary - Logical Signals.*

## Physical Architecture

### E/E Architecture

#### E/E Architecture Variants

**#Classification:** Mandatory – State “No Variants defined”, if not used.

**#Hint:** If different variants of the E/E architecture are specified in this section, list those variants in the table below.

Variants can be expressed based on Variant Options. Typical Variant Options (think of them as Logical Parameters) driven by architecture are e.g. “Network Topology” (e.g. FNV2 or AV) or “Powertrain Type” (e.g. “Electrical Vehicle” vs. “conventional powertrain”. “Conventional Powertrain” might be further split into “ECM+TCM” vs. “PCM”). The optional column “Variant condition” allows to express the dependency of a Variant based on Variant Options/Logical Parameters.

If requirements or certain architectural elements (signals, interfaces, components) are not applicable for all variants, those requirements/elements should state explicitly, which E/E architecture variant they apply to.

**#Link:** [RE Wiki – Variant Management](http://wiki.ford.com/display/RequirementsEngineering/Variant+Management).

|  |  |  |
| --- | --- | --- |
| E/E Architecture Variant Name | Variant Description | Variant Condition (optional) |
| FNV3 with Positionals | FNV3 architecture with positional ECU’s that support PPP (DSM5, MCSM3) | * FNV3 + DSM5 and MCSM3 |
| FNV3 without Positionals | FNV3 architecture with positional ECU’s that do not support PPP (need DSM5 and MCSM3). All positionals disabled if missing DSM5. | * FNV3 – DSM5 and/or MCSM3 |

##### E/E Architecture “Architecture Variant 1”

**#Classification:** Mandatory

**#Hint:** Place a diagram of high level E/E architecture here. Optionally the allocated functions could be shown in the diagram. Either use some SysML like diagram (refer to Figure 3‑2 E/E Architecture (SysML Style)) or enhanced the network topology, which Netcom generates from its master Excel sheet diagrams (refer to Figure 3‑3 E/E Architecture (Network Topology Style)) according to the needs of the feature.

This E/E Architecture variant … supports PPP feature on the FNV3 architecture

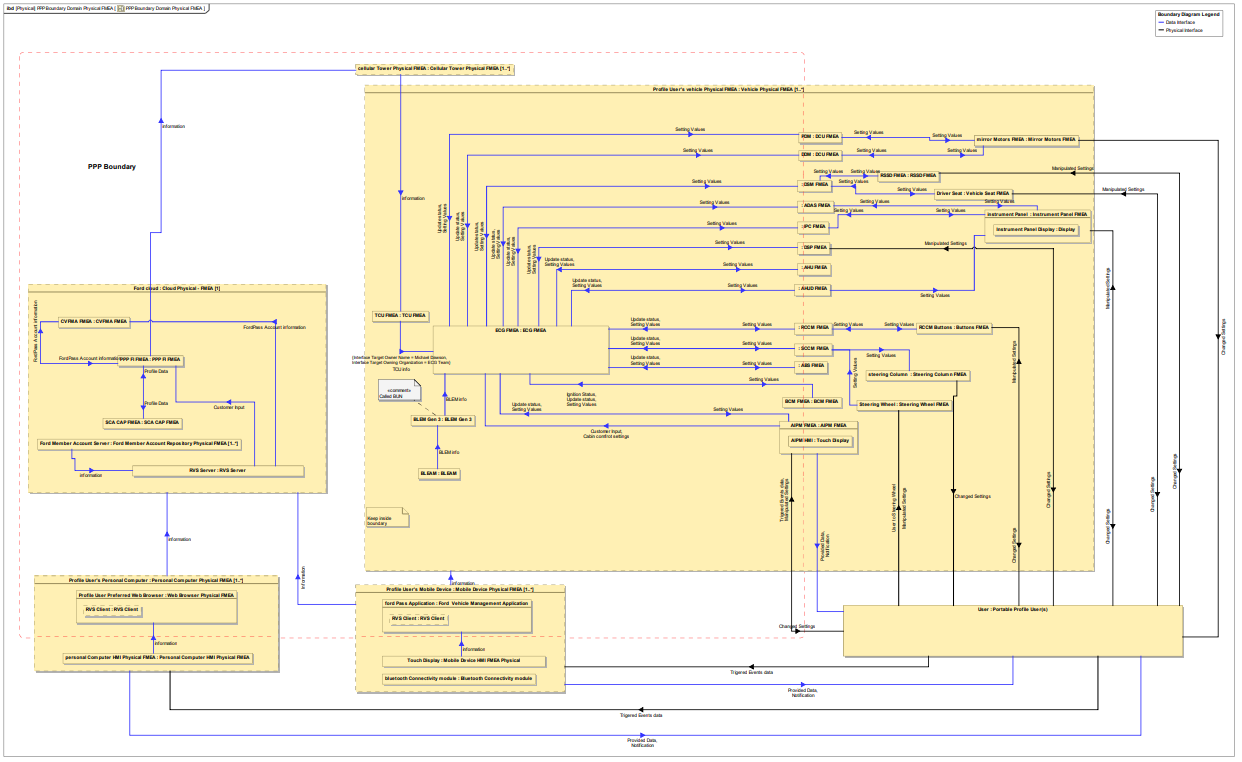


Figure 3‑2 E/E Architecture FNV3 (SysML Style)

#### E/E Components

#Hint: Use component name/acronym as given in the [VSEM Global Core ECU & EE Devices Dictionary](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=BleFgEP3x3NrTDAAAAAAAAAAAAA&servername=Production_Server) If not listed in that database, you may use the use PSF naming convention of the [EDAS signal database in VSEM](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=gPXpSoIbx3NrTDAAAAAAAAAAAAA&servername=Production_Server).  
You may directly link to the corresponding VSEM entry. Refer to the examples below”.

#Links: [*PSF Translate*](https://pd3.spt.ford.com/sites/EESEC3P/PSF_Translate/SitePages/Home.aspx)r (a little utility to search for an EDAS component name in PSF notation)

|  |  |
| --- | --- |
| Component Name | **Description** |
| ABS | Anti-Lock Brake System Control Module |
| IPMA (ADAS) | Image Processing Module “A” |
| ACM (AHU) | Audio Control Module |
| HUD (AHUD) | Heads Up Display |
| APIM | Accessory Protocol Interface Module |
| BCM | Body Control Module |
| DDM (DCU) | Driver Door Module |
| DSM | Driver Seat Module |
| DSP | Digital Signal Processing Module |
| ECG | Enhanced Central Gateway |
| SOBDMC\_HPCM | Sec OBD Ctrl Mod C - HEV P/T Ctrl Mod (Gen 4) |
| IPC | Instrument Panel Cluster Control Module |
| SCMG (MCSM) | Seat Control Module G (massage seats) |
| RCCM | Remote Climate Control Module |
| SCCM | Steering Column Control Module |

Table 3‑2: Electrical Components

#### E/E Connections

#Hint: Lists the E/E connections relevant for the feature and - for network connections - which *Messages* from the *Data Dictionary* are allocated to them. The ‘Connection Type’ is derived from the [*GDT/EDAS Signal Classification*](https://pd3.spt.ford.com/sites/fede/vsem-spls/Shared%20Documents/13-gdt/training/ppt/Signal_Classifications_v6.ppt?web=1). The ‘Protocol’ selection list might not be complete, yet. Add your protocol definition, if needed.

#Links: - [*GDT/EDAS Signal Classification*](https://pd3.spt.ford.com/sites/fede/vsem-spls/Shared%20Documents/13-gdt/training/ppt/Signal_Classifications_v6.ppt?web=1) (as reference for ‘Connection Type’ below)

* [*PSF Translate*](https://pd3.spt.ford.com/sites/EESEC3P/PSF_Translate/SitePages/Home.aspx)r (a little utility to search for an EDAS signal name in PSF notation)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Connection Name | **Connection Type** | **Protocol**  Only if ‘Connection Type’ is “Network”/”RF-Digital” | **Description** | **Allocated Messages**  Only if ‘Connection Type’ is “Network”/”RF-Digital” | **Connected Nodes** |
| FD1-CAN | Network | CAN FD | CAN Flexible Data Enhancement for module to module communication | AbsPppRqst  AbsPppRsp  AbsPppSettingChange  BcmPppRqst  BcmPppRsp  BcmPppSettingChange  HPCMPppRqst  HPCMPppRsp  HPCMPppSettingChange |  |
| FD3-CAN | Network | CAN FD | CAN Flexible Data Enhancement for module to module communication | AdasPppRqst  AdasPppRsp  AdasPppSettingChange  DcuPppRqst  DcuPppRsp  DcuPppSettingChange  DsmPppRqst  DsmPppRsp  DsmPppSettingChange  SccmPppRqst  SccmPppRsp  SccmPppSettingChange |  |
| HS3-CAN | Network | CAN (High Speed) | High Speed Infotainment CAN for module to module communication | AhuPppRqst  AhuPppRsp  AhuPppSettingChange  AhudPppRqst  AhudPppRsp  AhudPppSettingChange  DspPppRqst  DspPppRsp  DspPppSettingChange  IpcPppRqst  IpcPppRsp  IpcPppSettingChange |  |
| MS1-CAN | Network | CAN (Mid Speed) | Medium Speed CAN for module to module communication | McsmPppRqst  McsmPppRsp  McsmPppSettingChange  RccmPppRqst  RccmPppRsp  RccmPppSettingChange |  |
| Ethernet | Network | Ethernet (MQTT) | Module to module communication (mainly ECG to APIM) | ApimPppRqst  ApimPppRsp  ApimPppSettingChange |  |

Table 3‑3: E/E Connections

#### Signal List

***#Hint:*** *Refer to the* [*Data Dictionary*](#_Data_Dictionary) *-* [*Technical Signals*](#_Technical_Signals)*.*

### Software Component Architecture

**#Classification:** Optional – For features with in-house SW development only (remove section otherwise)

***#Hint:*** *For Features with in-house SW development (specifically in an Agile Environment) it is required, that the development team documents and agrees on at least their SW interfaces to the outside world early in the process.*

#### Description

**#Hint**: Provide some informal description of the characteristics of the chosen Software Component Architecture. Also give some graphical representation of the Software Component Architecture. SysML Internal Block Diagrams or [AUTOSAR](http://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemode) Virtual Function Bus models could be used to depict such a Software Component Architecture.

**#Link:** [*SysML – Internal Block Diagrams*](https://pd3.spt.ford.com/sites/SystemsEngineering/SEC/sysml-teamsite/SysML%20Wiki/Internal%20Block%20Diagram%20Basics.aspx) and [*AUTOSAR*](https://www.autosar.org/)

This Software Component Architecture depicts the logical breakdown of the feature components into class descriptions. The architecture diagram details both cloud and vehicle systems required for the feature and their interfaces between one another.

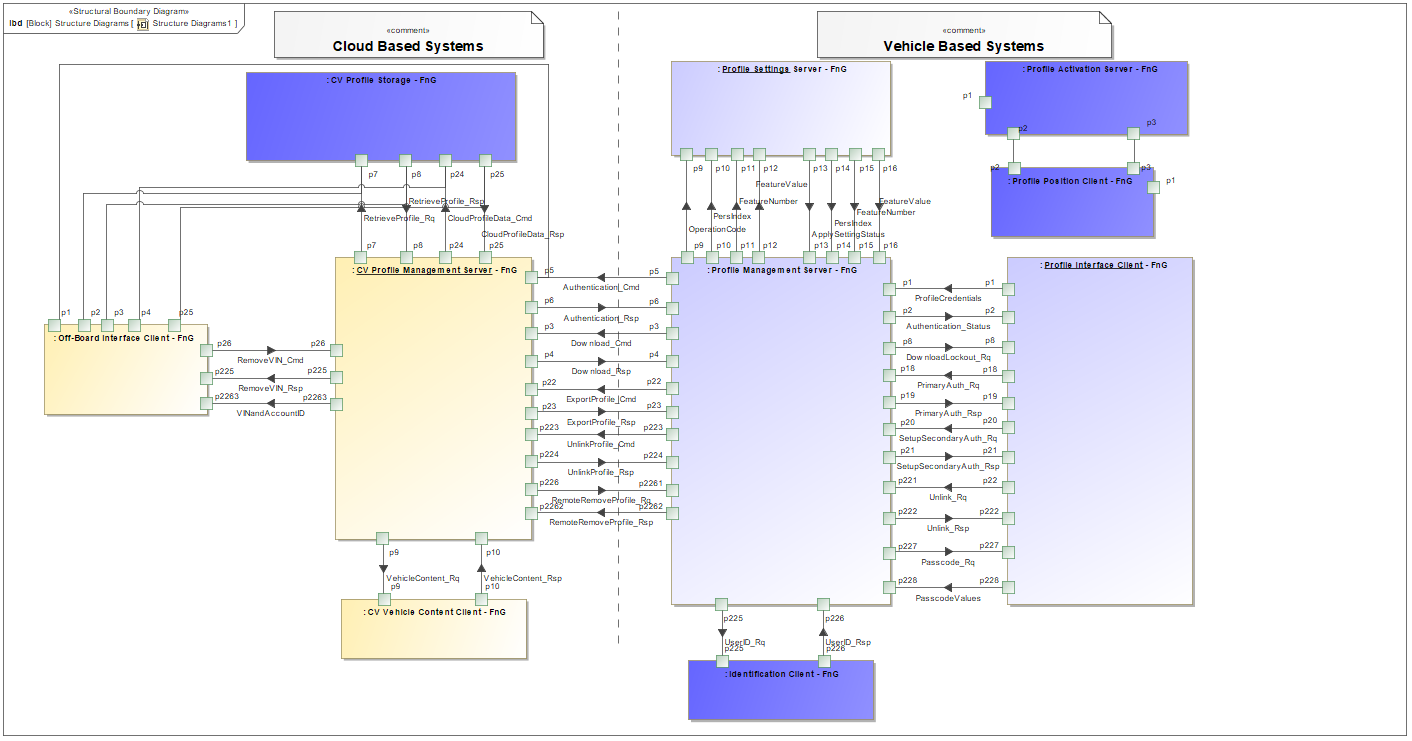


Figure 3‑4: PPP Feature SW Component Architecture

## Function Deployment

***#Hint:*** *This section lists and details the deployment variants of the feature.*

### Deployment Variants

**#Classification:** Mandatory – State “No Variants defined”, if not used.

**#Hint:** If there is more than 1 variant of deployment, the different variant should be listed and described below. Deployment variants are very much driven by E/E architecture variants (refer to section *E/E Architecture Variants*). Nevertheless, Feature/Function variant options might also drive additional deployment variants.

**#For Functional Safety:** Specify each deployment variant in a separate FIS.

No Variants defined.

### Function Allocation

***#Hint:*** *The “Function Allocation Table” shows the mapping of the Logical Functions and the corresponding Technology Functions of a feature to components of the physical architecture as also shown in the deployment diagrams. Typically, there is a 1:1 relationship between (Atomic) Logical and Technology Functions. For details refer to the* *RE Wiki pages* [*“Deriving Implemented Functions from Logical Functions”*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions) *and “*[*Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements%23CascadeRequirements-CascadingVsTraceability)*” When applying MBSE methods please refer to Guideline for Alignment of SW QoS with Ford Starting Model (SysML) for how Logical and Technology Functions in the Ford Starting Model align to Atomic Logical Functions and Technology Functions in RE.*

*For Functional Safety critical features the second table (“Function Allocation Table (Functional Safety Extension)”) has to be additionally filled in*

* *to map Technical Safety Requirements (TSRs) to Technology Functions and hence Components and*
* *to assign an ASIL level to Components and TSRs*

***#Link:***[*RE Wiki - Deriving Implemented Functions from Logical Functions*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

[*Functional Safety Sharepoint*](https://pd3.spt.ford.com/sites/GlobalFunctionalSafety/Pages/default.aspx)

[*Guideline for Alignment of SW QoS with Ford Starting Model (SysML)*](http://wiki.ford.com/display/RequirementsEngineering/Alignment+with+the+Ford+Starting+Model)

| Component | Technology Function Name | Logical Function Name |
| --- | --- | --- |
|
| HPCM | Execute Profile Settings - HPCM | * Execute Profile Settings |
| Send All Profile Settings - HPCM | * Send All Profile Settings |
| Send Changed Profile Settings - HPCM | * Send Changed Profile Setting |
| IPMA\_ADAS | Execute Profile Settings - ADAS | * Execute Profile Settings |
| Send All Profile Settings - ADAS | * Send All Profile Settings |
| Send Changed Profile Settings - ADAS | * Send Changed Profile Setting |
| MCSM | Execute Profile Settings - MCSM | * Execute Profile Settings |
| Send All Profile Settings - MCSM | * Send All Profile Settings |
| Send Changed Profile Settings - MCSM | * Send Changed Profile Setting |
| DCU | Execute Profile Settings - DCU | * Execute Profile Settings |
| Send All Profile Settings - DCU | * Send All Profile Settings |
| Send Changed Profile Settings - DCU | * Send Changed Profile Setting |
| APIM | Execute Profile Settings - APIM | * Execute Profile Settings |
| Send All Profile Settings - APIM | * Send All Profile Settings |
| Send Changed Profile Settings - APIM | * Send Changed Profile Setting |
| Display Authentication Status - APIM | * Display Authentication Status |
| Display Confirmation of Updated Profile - APIM | * Display Confirmation of Updated Profile |
| Display Customize Profile Screen - APIM | * Display Customize Profile Screen |
| Display Error Messages - APIM | * Display Error Messages |
| Display Login Method - APIM | * Display Login Method |
| Display Option to Unlink FP Account - APIM | * Display Option to Unlink FP Account |
| Display Secondary Passcode Screen - APIM | * Display Secondary Passcode Screen |
| Display Welcome Screen with ID Method - APIM | * Display Welcome Screen with ID Method |
| Lockout Profile for Download - APIM | * Lockout Profile for Download |
| Setup Primary Authentication - APIM | * Setup Primary Authentication |
| Setup Secondary Authentication Passcode - APIM | * Setup Secondary Authentication - Passcode |
| SCCM | Execute Profile Settings - SCCM | * Execute Profile Settings |
| Send All Profile Settings - SCCM | * Send All Profile Settings |
| Send Changed Profile Settings- SCCM | * Send Changed Profile Setting |
| AHU | Execute Profile Settings - AHU | * Execute Profile Settings |
| Send All Profile Settings - AHU | * Send All Profile Settings |
| Send Changed Profile Settings - AHU | * Send Changed Profile Setting |
| AHUD | Execute Profile Settings - AHUD | * Execute Profile Settings |
| Send All Profile Settings - AHUD | * Send All Profile Settings |
| Send Changed Profile Settings - AHUD | * Send Changed Profile Setting |
| DSM | Execute Profile Settings - DSM | * Execute Profile Settings |
| Send All Profile Settings - DSM | * Send All Profile Settings |
| Send Changed Profile Settings - DSM | * Send Changed Profile Setting |
| IPC | Execute Profile Settings - IPC | * Execute Profile Settings |
| Send All Profile Settings - IPC | * Send All Profile Settings |
| Send Changed Profile Settings - IPC | * Send Changed Profile Setting |
| RCCM | Execute Profile Settings - RCCM | * Execute Profile Settings |
| Send All Profile Settings - RCCM | * Send All Profile Settings |
| Send Changed Profile Settings - RCCM | * Send Changed Profile Setting |
| BCM | Execute Profile Settings - BCM | * Execute Profile Settings |
| Send All Profile Settings - BCM | * Send All Profile Settings |
| Send Changed Profile Settings - BCM | * Send Changed Profile Setting |
| ECG | Collect Profile Settings - ECG | * Collect Profile Settings |
| Delete Profile From Storage - ECG | * Delete Profile From Storage |
| Download Profile - ECG | * Download Profile |
| Export Profile Settings - ECG | * Export Profile Settings |
| Link Primary Authentication Indicator to Profile - ECG | * Link Primary Authentication Indicator to Profile |
| Link Secondary Authentication Passcode to Profile - ECG | * Link Secondary Authentication Passcode to Profile |
| Provide Authentication Status (Auto Login) - ECG | * Provide Authentication Status (Auto Login) |
| Push Profile Settings - ECG | * Push Profile Settings |
| Receive Remote Remove Request - ECG | * Receive Remote Remove Request |
| Receive Changed Profile Settings - ECG | * Receive Changed Profile Setting |
| Receive RVS Profile Payload - ECG | * Receive RVS Profile Payload |
| Request Authentication - ECG | * Request Authentication |
| Restore Profile Settings - ECG | * Restore Profile Settings |
| Unlink FP Account from Vehicle Profile - ECG | * Unlink FP Account from Vehicle Profile |
| DSP | Execute Profile Settings - DSP | * Execute Profile Settings |
| Send All Profile Settings - DSP | * Send All Profile Settings |
| Send Changed Profile Settings - DSP | * Send Changed Profile Setting |
| ABS | Execute Profile Settings - ABS | * Execute Profile Settings |
| Send All Profile Settings - ABS | * Send All Profile Settings |
| Send Changed Profile Settings - ABS | * Send Changed Profile Setting |

Table 3‑5: Function Allocation Table (Basic)

# Feature Implementation Modeling

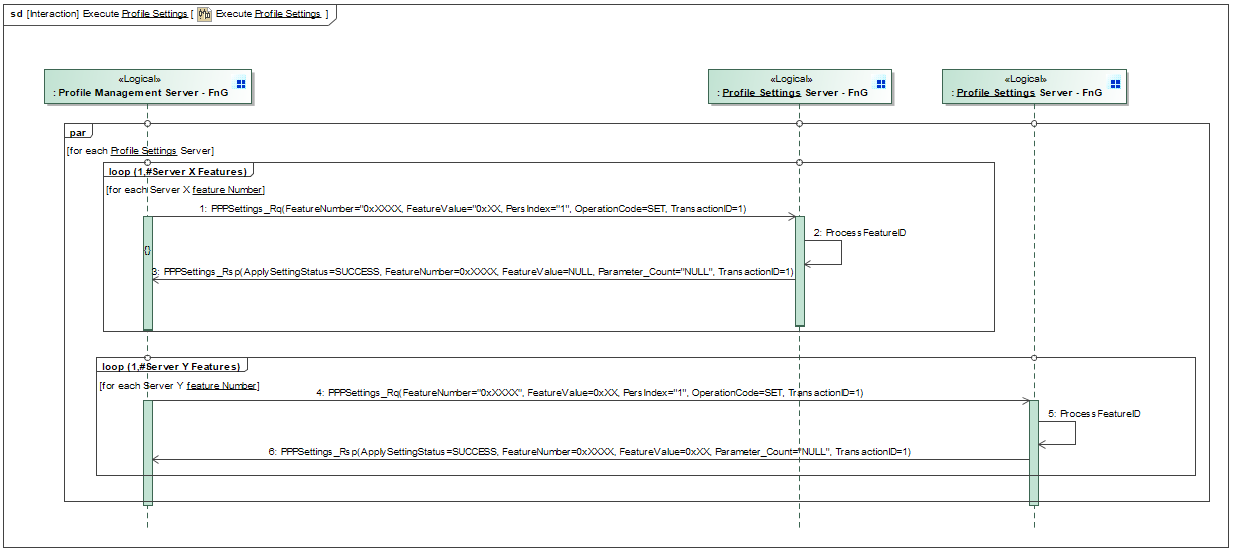
***#Hint:*** *This chapter shall give deployment specific refinements of the customer use cases, which are defined in the Feature Document. Based on activity diagrams and possibly sequence diagrams, this chapter shows, how the overall functionality is decomposed in activities / functions, which can be allocated to single physical components. In addition, this chapter shows how the individual components / activities collaborate.*

## Component Interaction Diagrams

***#Hint:*** *This chapter would typically list sequence charts or activity diagrams with swim lanes for all involved physical components to illustrate the interactions between components for relevant scenarios.*

*End-to-end timing constraints could be placed in the diagrams. End-to-end timing would be relevant if more than 2 nodes (sender / receiver) are involved when information flows from signal generation (e.g. button pressed by user) to signal consumption (e.g. light is switched on).*

### Scenario: “Set Profile”



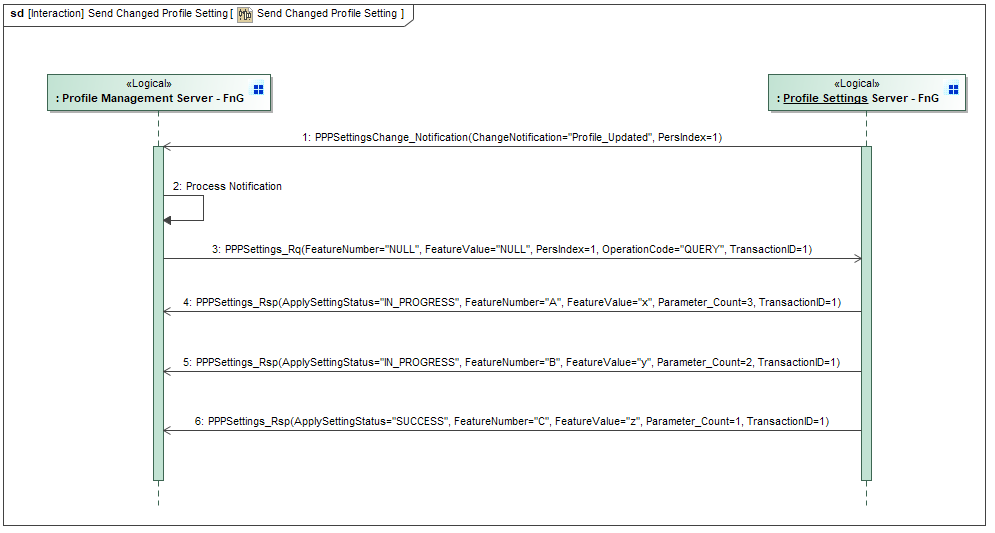
**Figure 12:** Set Profile Sequence Diagram

### Scenario: “Query Profile”



**Figure 13:** Query Profile Sequence Diagram

### Scenario: “Changed Profile Setting”



**Figure 14:** Changed Profile Setting Sequence Diagram

## Component Interface Behavior Diagrams

*#Hint: For complex (application level) interface protocols a protocol state machine would be more appropriate than a bunch of sequence diagrams to illustrate the interactions between components. So, this section would typically show a (protocol) state machine.*

# Feature Implementation Requirements

***#Hint:*** *The Feature Implementation Specification is first of all an architecture document. It shows the Functional and the E/E architecture as well as the deployment of the Functional one to the E/E one.*

## Functional Safety

**#Classification**: Functional Safety only – If not used, remove content and state “Not Applicable”

***#Hint:*** *If feature is not Functional Safety critical, remove subsections from this chapter and state “Feature is not Functional Safety critical”*

### ASIL Decomposition of Technical Safety Requirements

**#Classification**: Functional Safety only – If not used, remove content and state “Not Applicable”

***#Hint:*** *Sometimes an ASIL decomposition of Technical Safety Requirements (TSRs) is required. The (input) TSRs, which are to be decomposed, are derived from FSRs. Those input TSRs are to be specified in this chapter (right above the corresponding ASIL decomposition table). For each input TSR add one “ASIL Decomposition Table”. In the “ASIL Decomposition Table” the derived, decomposed TSRs are referenced by ID and Title. Those TSRs are however not specified in the FIS but in the ECU Functional Specfication.*

**#Macro:** [Add Ins -> Add Requirement macro](http://wiki.ford.com/pages/viewpage.action?pageId=174654231) (select “**Func./Tech. Safety Requirement**” as type)

***#Link:***[*Functional Safety Sharepoint*](https://pd3.spt.ford.com/sites/GlobalFunctionalSafety/Pages/default.aspx) *- Functional Safety Concept*

*Not Applicable*

## Requirements on Components

### ABS

#### Technology Function -87220479.jpg **Execute Profile Settings - ABS**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AbsPpp\_D\_Rq | N/A |  | FD1-CAN::AbsPppRqst |
| PersIndex | AbsPersIdx\_No\_Actl | N/A |  | FD1-CAN::AbsPppRqst |
| FeatureNumber | AbsFeatNum\_No\_Actl | N/A |  | FD1-CAN::AbsPppRqst |
| FeatureValue | AbsFeatValue\_No\_Actl | N/A |  | FD1-CAN::AbsPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - ABS

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | AbsApplySett\_D\_Status | N/A |  | FD1-CAN::AbsPppRsp |
| FeatureNumber | AbsFeatNum\_No\_Res | N/A |  | FD1-CAN::AbsPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - ABS

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
|  |  |  | Choose an item. |  |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings – ABS**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AbsPpp\_D\_Rq | N/A |  | FD1-CAN::AbsPppRqst |
| PersIndex | AbsPersIdx\_No\_Actl | N/A |  | FD1-CAN::AbsPppRqst |
| FeatureNumber | AbsFeatNum\_No\_Actl | N/A |  | FD1-CAN::AbsPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - ABS

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | AbsPersIdx\_No\_Res | N/A |  | FD1-CAN::AbsPppRsp |
| FeatureNumber | AbsFeatNum\_No\_Res | N/A |  | FD1-CAN::AbsPppRsp |
| FeatureValue | AbsFeatValue\_No\_Res | N/A |  | FD1-CAN::AbsPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - ABS

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
|  |  |  | Choose an item. |  |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - ABS**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| N/A | N/A | N/A | N/A | N/A |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - ABS

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | AbsSettChg\_D\_Stat | N/A |  | FD1-CAN::AbsPppSettingChange |
| PersIndex | AbsPersIdx\_No\_Res | N/A |  | FD1-CAN::AbsPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - ABS

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
|  |  |  | Choose an item. |  |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### ADAS

#### Technology Function -87220479.jpg **Execute Profile Settings - ADAS**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AdasPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::AdasPppRqst |
| PersIndex | AdasPersIdx\_No\_Actl | N/A |  | FD3-CAN::AdasPppRqst |
| FeatureNumber | AdasFeatNum\_No\_Actl | N/A |  | FD3-CAN::AdasPppRqst |
| FeatureValue | AdasFeatValue\_No\_Actl | N/A |  | FD3-CAN::AdasPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - ADAS

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | AdasApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::AdasPppRsp |
| FeatureNumber | AdasFeatNum\_No\_Res | N/A |  | FD3-CAN::AdasPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - ADAS

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings – ADAS**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AdasPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::AdasPppRqst |
| PersIndex | AdasPersIdx\_No\_Actl | N/A |  | FD3-CAN::AdasPppRqst |
| FeatureNumber | AdasFeatNum\_No\_Actl | N/A |  | FD3-CAN::AdasPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - ADAS

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | AdasPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::AdasPppRsp |
| FeatureNumber | AdasFeatNum\_No\_Res | N/A |  | FD3-CAN::AdasPppRsp |
| FeatureValue | AdasFeatValue\_No\_Res | N/A |  | FD3-CAN::AdasPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - ADAS

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - ADAS**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - ADAS

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | AdasSettChg\_D\_Stat | N/A |  | FD1-CAN::AdasPppSettingChange |
| PersIndex | AdasPersIdx\_No\_Res | N/A |  | FD1-CAN::AdasPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - ADAS

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### AHU

#### Technology Function -87220479.jpg **Execute Profile Settings - AHU**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AhuPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhuPppRqst |
| PersIndex | AhuPersIdx\_No\_Actl | N/A |  | HS3-CAN::AhuPppRqst |
| FeatureNumber | AhuFeatNum\_No\_Actl | N/A |  | HS3-CAN::AhuPppRqst |
| FeatureValue | AhuFeatValue\_No\_Actl | N/A |  | HS3-CAN::AhuPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - AHU

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | AhuApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhuPppRsp |
| FeatureNumber | AhuFeatNum\_No\_Res | N/A |  | HS3-CAN::AhuPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - AHU

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - AHU**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AhuPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhuPppRqst |
| PersIndex | AhuPersIdx\_No\_Actl | N/A |  | HS3-CAN::AhuPppRqst |
| FeatureNumber | AhuFeatNum\_No\_Actl | N/A |  | HS3-CAN::AhuPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - AHU

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | AhuPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhuPppRsp |
| FeatureNumber | AhuFeatNum\_No\_Res | N/A |  | HS3-CAN::AhuPppRsp |
| FeatureValue | AhuFeatValue\_No\_Res | N/A |  | HS3-CAN::AhuPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - AHU

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - AHU**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - AHU

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | AhuSettChg\_D\_Stat | N/A |  | FD1-CAN::AhuPppSettingChange |
| PersIndex | AhuPersIdx\_No\_Res | N/A |  | FD1-CAN::AhuPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - AHU

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### AHUD

#### Technology Function -87220479.jpg **Execute Profile Settings - AHUD**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AhudPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRqst |
| PersIndex | AhudPersIdx\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |
| FeatureNumber | AhudFeatNum\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |
| FeatureValue | AhudFeatValue\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - AHUD

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | AhudApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRsp |
| FeatureNumber | AhudFeatNum\_No\_Res | N/A |  | HS3-CAN::AhudPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - AHUD

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - AHUD**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AhudPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRqst |
| PersIndex | AhudPersIdx\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |
| FeatureNumber | AhudFeatNum\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - AHUD

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | AhudPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRsp |
| FeatureNumber | AhudFeatNum\_No\_Res | N/A |  | HS3-CAN::AhudPppRsp |
| FeatureValue | AhudFeatValue\_No\_Res | N/A |  | HS3-CAN::AhudPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - AHUD

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - AHUD**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - AHUD

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | AhudSettChg\_D\_Stat | N/A |  | FD1-CAN::AhudPppSettingChange |
| PersIndex | AhudPersIdx\_No\_Res | N/A |  | FD1-CAN::AhudPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - AHUD

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### APIM

#### Technology Function -87220479.jpg **Execute Profile Settings - APIM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | ApimPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::ApimPppRqst |
| PersIndex | ApimPersIdx\_No\_Actl | N/A |  | HS3-CAN::ApimPppRqst |
| FeatureNumber | ApimFeatNum\_No\_Actl | N/A |  | HS3-CAN::ApimPppRqst |
| FeatureValue | ApimFeatValue\_No\_Actl | N/A |  | HS3-CAN::ApimPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - APIM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | ApimApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::ApimPppRsp |
| FeatureNumber | ApimFeatNum\_No\_Res | N/A |  | HS3-CAN::ApimPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - APIM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - APIM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | ApimPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::ApimPppRqst |
| PersIndex | ApimPersIdx\_No\_Actl | N/A |  | HS3-CAN::ApimPppRqst |
| FeatureNumber | ApimFeatNum\_No\_Actl | N/A |  | HS3-CAN::ApimPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - APIM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | ApimPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::ApimPppRsp |
| FeatureNumber | ApimFeatNum\_No\_Res | N/A |  | HS3-CAN::ApimPppRsp |
| FeatureValue | ApimFeatValue\_No\_Res | N/A |  | HS3-CAN::ApimPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - APIM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - APIM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - APIM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | ApimSettChg\_D\_Stat | N/A |  | FD1-CAN::ApimPppSettingChange |
| PersIndex | ApimPersIdx\_No\_Res | N/A |  | FD1-CAN::ApimPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - APIM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### BCM

#### Technology Function -87220479.jpg **Execute Profile Settings - BCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | BcmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD1-CAN::BcmPppRqst |
| PersIndex | BcmPersIdx\_No\_Actl | N/A |  | FD1-CAN::BcmPppRqst |
| FeatureNumber | BcmFeatNum\_No\_Actl | N/A |  | FD1-CAN::BcmPppRqst |
| FeatureValue | BcmFeatValue\_No\_Actl | N/A |  | FD1-CAN::BcmPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - BCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | BcmApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD1-CAN::BcmPppRsp |
| FeatureNumber | BcmFeatNum\_No\_Res | N/A |  | FD1-CAN::BcmPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - BCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - BCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | BcmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD1-CAN::BcmPppRqst |
| PersIndex | BcmPersIdx\_No\_Actl | N/A |  | FD1-CAN::BcmPppRqst |
| FeatureNumber | BcmFeatNum\_No\_Actl | N/A |  | FD1-CAN::BcmPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - BCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | BcmPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD1-CAN::BcmPppRsp |
| FeatureNumber | BcmFeatNum\_No\_Res | N/A |  | FD1-CAN::BcmPppRsp |
| FeatureValue | BcmFeatValue\_No\_Res | N/A |  | FD1-CAN::BcmPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - BCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - BCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - BCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | BcmSettChg\_D\_Stat | N/A |  | FD1-CAN::BcmPppSettingChange |
| PersIndex | BcmPersIdx\_No\_Res | N/A |  | FD1-CAN::BcmPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - BCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### DCU

#### Technology Function -87220479.jpg **Execute Profile Settings - DCU**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | DcuPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::DcuPppRqst |
| PersIndex | DcuPersIdx\_No\_Actl | N/A |  | FD3-CAN::DcuPppRqst |
| FeatureNumber | DcuFeatNum\_No\_Actl | N/A |  | FD3-CAN::DcuPppRqst |
| FeatureValue | DcuFeatValue\_No\_Actl | N/A |  | FD3-CAN::DcuPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - DCU

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | DcuApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::DcuPppRsp |
| FeatureNumber | DcuFeatNum\_No\_Res | N/A |  | FD3-CAN::DcuPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - DCU

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - DCU**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | DcuPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::DcuPppRqst |
| PersIndex | DcuPersIdx\_No\_Actl | N/A |  | FD3-CAN::DcuPppRqst |
| FeatureNumber | DcuFeatNum\_No\_Actl | N/A |  | FD3-CAN::DcuPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - DCU

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | DcuPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::DcuPppRsp |
| FeatureNumber | DcuFeatNum\_No\_Res | N/A |  | FD3-CAN::DcuPppRsp |
| FeatureValue | DcuFeatValue\_No\_Res | N/A |  | FD3-CAN::DcuPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - DCU

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - DCU**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - DCU

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | DcuSettChg\_D\_Stat | N/A |  | FD1-CAN::DcuPppSettingChange |
| PersIndex | DcuPersIdx\_No\_Res | N/A |  | FD1-CAN::DcuPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - DCU

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### DSM

#### Technology Function -87220479.jpg **Execute Profile Settings - DSM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | DsmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::DsmPppRqst |
| PersIndex | DsmPersIdx\_No\_Actl | N/A |  | FD3-CAN::DsmPppRqst |
| FeatureNumber | DsmFeatNum\_No\_Actl | N/A |  | FD3-CAN::DsmPppRqst |
| FeatureValue | DsmFeatValue\_No\_Actl | N/A |  | FD3-CAN::DsmPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - DSM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | DsmApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::DsmPppRsp |
| FeatureNumber | DsmFeatNum\_No\_Res | N/A |  | FD3-CAN::DsmPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - DSM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - DSM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | DsmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::DsmPppRqst |
| PersIndex | DsmPersIdx\_No\_Actl | N/A |  | FD3-CAN::DsmPppRqst |
| FeatureNumber | DsmFeatNum\_No\_Actl | N/A |  | FD3-CAN::DsmPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - DSM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | DsmPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::DsmPppRsp |
| FeatureNumber | DsmFeatNum\_No\_Res | N/A |  | FD3-CAN::DsmPppRsp |
| FeatureValue | DsmFeatValue\_No\_Res | N/A |  | FD3-CAN::DsmPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - DSM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - DSM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - DSM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | DsmSettChg\_D\_Stat | N/A |  | FD1-CAN::DsmPppSettingChange |
| PersIndex | DsmPersIdx\_No\_Res | N/A |  | FD1-CAN::DsmPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - DSM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### DSP

#### Technology Function -87220479.jpg **Execute Profile Settings - DSP**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | DspPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::DspPppRqst |
| PersIndex | DspPersIdx\_No\_Actl | N/A |  | HS3-CAN::DspPppRqst |
| FeatureNumber | DspFeatNum\_No\_Actl | N/A |  | HS3-CAN::DspPppRqst |
| FeatureValue | DspFeatValue\_No\_Actl | N/A |  | HS3-CAN::DspPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - DSP

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | DspApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::DspPppRsp |
| FeatureNumber | DspFeatNum\_No\_Res | N/A |  | HS3-CAN::DspPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - DSP

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - DSP**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | DspPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::DspPppRqst |
| PersIndex | DspPersIdx\_No\_Actl | N/A |  | HS3-CAN::DspPppRqst |
| FeatureNumber | DspFeatNum\_No\_Actl | N/A |  | HS3-CAN::DspPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - DSP

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | DspPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::DspPppRsp |
| FeatureNumber | DspFeatNum\_No\_Res | N/A |  | HS3-CAN::DspPppRsp |
| FeatureValue | DspFeatValue\_No\_Res | N/A |  | HS3-CAN::DspPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - DSP

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - DSP**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - DSP

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | DspSettChg\_D\_Stat | N/A |  | FD1-CAN::DspPppSettingChange |
| PersIndex | DspPersIdx\_No\_Res | N/A |  | FD1-CAN::DspPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - DSP

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### ECG

#### Technology Function -87220479.jpg **Execute Profile Settings - AHUD**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AhudPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRqst |
| PersIndex | AhudPersIdx\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |
| FeatureNumber | AhudFeatNum\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |
| FeatureValue | AhudFeatValue\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - AHUD

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | AhudApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRsp |
| FeatureNumber | AhudFeatNum\_No\_Res | N/A |  | HS3-CAN::AhudPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - AHUD

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-XXXXXX | PPPSettings\_Rsp Message Parameters |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - AHUD**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | AhudPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRqst |
| PersIndex | AhudPersIdx\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |
| FeatureNumber | AhudFeatNum\_No\_Actl | N/A |  | HS3-CAN::AhudPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - AHUD

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | AhudPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRsp |
| FeatureNumber | AhudFeatNum\_No\_Res | N/A |  | HS3-CAN::AhudPppRsp |
| FeatureValue | AhudFeatValue\_No\_Res | N/A |  | HS3-CAN::AhudPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - AHUD

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - AHUD**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - AHUD

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | AhudPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::AhudPppRsp |
| FeatureNumber | AhudFeatNum\_No\_Res | N/A |  | HS3-CAN::AhudPppRsp |
| FeatureValue | AhudFeatValue\_No\_Res | N/A |  | HS3-CAN::AhudPppRsp |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - AHUD

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
|  |  |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### HPCM

#### Technology Function -87220479.jpg **Execute Profile Settings - HPCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | HPCMPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD1-CAN::HPCMPppRqst |
| PersIndex | HPCMPersIdx\_No\_Actl | N/A |  | FD1-CAN::HPCMPppRqst |
| FeatureNumber | HPCMFeatNum\_No\_Actl | N/A |  | FD1-CAN::HPCMPppRqst |
| FeatureValue | HPCMFeatValue\_No\_Actl | N/A |  | FD1-CAN::HPCMPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - HPCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | HPCMApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD1-CAN::HPCMPppRsp |
| FeatureNumber | HPCMFeatNum\_No\_Res | N/A |  | FD1-CAN::HPCMPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - HPCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - HPCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | HPCMPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD1-CAN::HPCMPppRqst |
| PersIndex | HPCMPersIdx\_No\_Actl | N/A |  | FD1-CAN::HPCMPppRqst |
| FeatureNumber | HPCMFeatNum\_No\_Actl | N/A |  | FD1-CAN::HPCMPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - HPCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | HPCMPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD1-CAN::HPCMPppRsp |
| FeatureNumber | HPCMFeatNum\_No\_Res | N/A |  | FD1-CAN::HPCMPppRsp |
| FeatureValue | HPCMFeatValue\_No\_Res | N/A |  | FD1-CAN::HPCMPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - HPCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - HPCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - HPCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | HpcmSettChg\_D\_Stat | N/A |  | FD1-CAN::HpcmPppSettingChange |
| PersIndex | HpcmPersIdx\_No\_Res | N/A |  | FD1-CAN::HpcmPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - HPCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### IPC

#### Technology Function -87220479.jpg **Execute Profile Settings - IPC**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | IpcPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::IpcPppRqst |
| PersIndex | IpcPersIdx\_No\_Actl | N/A |  | HS3-CAN::IpcPppRqst |
| FeatureNumber | IpcFeatNum\_No\_Actl | N/A |  | HS3-CAN::IpcPppRqst |
| FeatureValue | IpcFeatValue\_No\_Actl | N/A |  | HS3-CAN::IpcPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - IPC

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | IpcApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::IpcPppRsp |
| FeatureNumber | IpcFeatNum\_No\_Res | N/A |  | HS3-CAN::IpcPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - IPC

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - IPC**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | IpcPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::IpcPppRqst |
| PersIndex | IpcPersIdx\_No\_Actl | N/A |  | HS3-CAN::IpcPppRqst |
| FeatureNumber | IpcFeatNum\_No\_Actl | N/A |  | HS3-CAN::IpcPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - IPC

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | IpcPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | HS3-CAN::IpcPppRsp |
| FeatureNumber | IpcFeatNum\_No\_Res | N/A |  | HS3-CAN::IpcPppRsp |
| FeatureValue | IpcFeatValue\_No\_Res | N/A |  | HS3-CAN::IpcPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - IPC

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - IPC**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - IPC

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | IpcSettChg\_D\_Stat | N/A |  | FD1-CAN::IpcPppSettingChange |
| PersIndex | IpcPersIdx\_No\_Res | N/A |  | FD1-CAN::IpcPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - IPC

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### MCSM

#### Technology Function -87220479.jpg **Execute Profile Settings - MCSM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | McsmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | MS1-CAN::McsmPppRqst |
| PersIndex | McsmPersIdx\_No\_Actl | N/A |  | MS1-CAN::McsmPppRqst |
| FeatureNumber | McsmFeatNum\_No\_Actl | N/A |  | MS1-CAN::McsmPppRqst |
| FeatureValue | McsmFeatValue\_No\_Actl | N/A |  | MS1-CAN::McsmPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - MCSM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | McsmApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | MS1-CAN::McsmPppRsp |
| FeatureNumber | McsmFeatNum\_No\_Res | N/A |  | MS1-CAN::McsmPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - MCSM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - MCSM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | McsmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | MS1-CAN::McsmPppRqst |
| PersIndex | McsmPersIdx\_No\_Actl | N/A |  | MS1-CAN::McsmPppRqst |
| FeatureNumber | McsmFeatNum\_No\_Actl | N/A |  | MS1-CAN::McsmPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - MCSM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | McsmPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | MS1-CAN::McsmPppRsp |
| FeatureNumber | McsmFeatNum\_No\_Res | N/A |  | MS1-CAN::McsmPppRsp |
| FeatureValue | McsmFeatValue\_No\_Res | N/A |  | MS1-CAN::McsmPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - MCSM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - MCSM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - MCSM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | McsmSettChg\_D\_Stat | N/A |  | FD1-CAN::McsmPppSettingChange |
| PersIndex | McsmPersIdx\_No\_Res | N/A |  | FD1-CAN::McsmPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - MCSM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### RCCM

#### Technology Function -87220479.jpg **Execute Profile Settings - RCCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | RccmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | MS1-CAN::RccmPppRqst |
| PersIndex | RccmPersIdx\_No\_Actl | N/A |  | MS1-CAN::RccmPppRqst |
| FeatureNumber | RccmFeatNum\_No\_Actl | N/A |  | MS1-CAN::RccmPppRqst |
| FeatureValue | RccmFeatValue\_No\_Actl | N/A |  | MS1-CAN::RccmPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - RCCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | RccmApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | MS1-CAN::RccmPppRsp |
| FeatureNumber | RccmFeatNum\_No\_Res | N/A |  | MS1-CAN::RccmPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - RCCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - RCCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | RccmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | MS1-CAN::RccmPppRqst |
| PersIndex | RccmPersIdx\_No\_Actl | N/A |  | MS1-CAN::RccmPppRqst |
| FeatureNumber | RccmFeatNum\_No\_Actl | N/A |  | MS1-CAN::RccmPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - RCCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | RccmPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | MS1-CAN::RccmPppRsp |
| FeatureNumber | RccmFeatNum\_No\_Res | N/A |  | MS1-CAN::RccmPppRsp |
| FeatureValue | RccmFeatValue\_No\_Res | N/A |  | MS1-CAN::RccmPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - RCCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - RCCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - RCCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | RccmSettChg\_D\_Stat | N/A |  | FD1-CAN::RccmPppSettingChange |
| PersIndex | RccmPersIdx\_No\_Res | N/A |  | FD1-CAN::RccmPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - RCCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

### SCCM

#### Technology Function -87220479.jpg **Execute Profile Settings - SCCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | SccmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::SccmPppRqst |
| PersIndex | SccmPersIdx\_No\_Actl | N/A |  | FD3-CAN::SccmPppRqst |
| FeatureNumber | SccmFeatNum\_No\_Actl | N/A |  | FD3-CAN::SccmPppRqst |
| FeatureValue | SccmFeatValue\_No\_Actl | N/A |  | FD3-CAN::SccmPppRqst |

Table 5‑2: Input Signal mappings of Function Execute Profile Settings - SCCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| ApplySettingStatus | SccmApplySett\_D\_Status | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::SccmPppRsp |
| FeatureNumber | SccmFeatNum\_No\_Res | N/A |  | FD3-CAN::SccmPppRsp |

Table 5‑3: Output Signal mappings of Function Execute Profile Settings - SCCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405864 | Sending a Successful ApplySettingStatus |  |
| REQ-405863 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405862 | Validate PPPSettings\_Rq |  |
| REQ-405861 | Execute Profile Settings Performance |  |
| REQ-405859 | Memory Capability |  |
| REQ-405858 | Priority of applying profile settings |  |
| REQ-405857 | Display active profile settings after being applied |  |
| REQ-405855 | PPPSettings\_Rq Message Parameters |  |
| REQ-353147 | Setting Server Unable to Apply Setting |  |
| REQ-352219 | Profile Setting Servers remember valid Feature Number |  |
| REQ-352214 | Receiving and Processing PPPSettings\_Rq |  |
| REQ-421979 | PPPSettings\_Rsp Message Parameters |  |
| REQ-421980 | TransactionID for Executing Profile Settings |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send All Profile Settings - SCCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| OperationCode | SccmPpp\_D\_Rq | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::SccmPppRqst |
| PersIndex | SccmPersIdx\_No\_Actl | N/A |  | FD3-CAN::SccmPppRqst |
| FeatureNumber | SccmFeatNum\_No\_Actl | N/A |  | FD3-CAN::SccmPppRqst |

Table 5‑2: Input Signal mappings of Function Send All Profile Settings - SCCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| PersIndex | SccmPersIdx\_No\_Res | N/A | Name should be Word reference to the “*Technical Interfaces”* name bookmark in the Data Dictionary (e.g. *AIS Interfaces* for CAN signals). | FD3-CAN::SccmPppRsp |
| FeatureNumber | SccmFeatNum\_No\_Res | N/A |  | FD3-CAN::SccmPppRsp |
| FeatureValue | SccmFeatValue\_No\_Res | N/A |  | FD3-CAN::SccmPppRsp |

Table 5‑3: Output Signal mappings of Function Send All Profile Settings - SCCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-405868 | Sending a Successful ApplySettingStatus |  |
| REQ-405867 | PPPSettings\_Rq parameters are out of range |  |
| REQ-405866 | Validate PPPSettings\_Rq |  |
| REQ-352239 | Execute Profile Settings Performance |  |
| REQ-352238 | Memory Capability |  |
| REQ-421982 | SettingCountTotal Parameter |  |
| REQ-421981 | Populating the PPPSetting\_Rsp signal values for query all settings request |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### Technology Function -87220479.jpg **Send Changed Profile Settings - SCCM**

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function Send Changed Profile Settings - SCCM

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Change\_Notification | SccmSettChg\_D\_Stat | N/A |  | FD1-CAN::SccmPppSettingChange |
| PersIndex | SccmPersIdx\_No\_Res | N/A |  | FD1-CAN::SccmPppSettingChange |

Table 5‑3: Output Signal mappings of Function Send Changed Profile Settings - SCCM

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  | Unchanged |  |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
| REQ-421983 | Send Change\_Notification to Profile Management Server |  |
| REQ-421984 | Send Change\_Notification Performance Requirement |  |
| REQ-421985 | Set Change\_Notification to NULL after a change |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

## Requirements on Connections

### Networks

#### “CAN Bus xxx”

***#Hint:*** *For CAN most requirements are defined by Netcom and referenced in* [*VSEM “Multiplexing Specifications” section*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=xcbJ6OwAx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*. Put in this section only those requirements, which deviate from that standard specification.*

*The CAN messages relevant for this feature are listed in the section “Messages” of the “Data Dictionary”.*

##### Protocol Requirements

**#Hint:** *For CAN Ford currently mandates FNOS as SW implementation for the CAN protocol stack. This includes the CAN Network Management and Transport Protocol. If you deviate from this assumption or if you have specific requirements on FNOS, which are not contained in the standard package, put requirements in this section. F*or details the FNOS user guide and application notes could be referenced.

##### Electrical Requirements

**#Hint:** List requirements here, only if they deviate from the SDS CAN.

#### “LIN Bus xxx”

**#Hint:** Place requirements here, which are common to all LIN nodes, but not covered by some SDS LIN.

*The LIN messages relevant for this feature are listed in the section “Messages” of the “Data Dictionary”.*

##### Protocol Requirements

###### Schedule Table

***#Hint:*** *The LIN Schedule Table should be documented in the LDF file. The LDF file could be referenced here*

##### Electrical Requirements

***#Hint:*** *The LIN Schedule Table should be documented in the LDF file. The LDF file could be referenced here*

#### “Ethernet xxx”

***#Hint:*** *On Ethernet (wired or wireless) we will see most likely the DoIP, MQTT or V2x protocols. Those protocols are described in separate specifications and are implemented in the Ford AUTOSAR stack. While DoIP might be not that relevant in this scope, MQTT (together with the Google Protocol Buffer (GPB) serialization of the payload) will become important for all features, which are mapped to a Service Oriented Architecture/Communcation (SoC). Application data (SOA APIs), which is transmitted via MQTT, are listed in the data dictionary section “AUTOSAR Interfaces*

#Hint: Those AUTOSAR Classic (Sender/Receiver and Client/Server) Interfaces, which are used by the feature but not managed in a central repository yet, should be listed here.

SOA Service *”.*

***#Link:***[*http://www.mqtt.org*](http://www.mqtt.org)*, https://developers.google.com/protocol-buffers/docs/proto*

### HW I/Os

**#Hin**t: This section lists all hardwired signals relevant for the feature deployment. Those get typically mapped to VSEM EDAS signals – refer to list of connections in corresponding table in chapter ”E/E Architecture → E/E Connections”. If any specific protocol is used to send/receive signal information or multiplex/demultiplex signals on the HW circuit.

#### “HW I/O xxx”

## Requirements on Development Process

# Open Concerns

***#Hint:*** *The following list presents known issues that have to be discussed or clarified over the course of the on-going requirements engineering.*

| ID | Concern Description | e-Tracker Reference | Status | Solution |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

Table 6‑1: Open Concerns

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Description | Approved by | Responsible |
| 1.0 | 3/10/2021 | Initial version |  | Jbauer50 |
| 1.1 | 6/2/2021 | * Added all technical and logical signals. * Added logical to technical signal mapping for all component functions * Add functional requirement mapping to all component functions * Added E/E Architecture Variants * Added Component Interaction Sequence Diagrams (Set Profile, Query Profile, Changed Profile) * Added allocated messages to E/E Connections table * Added all relevant components to E/E components table * Added feature implementation description to section 2.1 * Added Autosave feature specifications reference to section 1.6.1 * Removed “Receive RVS Payload” function from Function List 3.1.2 * Changed ECM to SOBDMC\_HPCM in E/E components 3.2.1.2 * Added Inherited Requirements for all components for Send Changed Profile Setting function (missed on 1.1 revision) * Added Change\_Notification logical signal * Added more requirements to the 3 functions for each ECU |  | Jbauer50 |

## Template Revisions

*#Important: Do not change this section*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Rev. | Date | Description | Responsible |
| 0 | 2 | 2015-08-05 | * TOC corrected * Document Properties adapted to match needs of VBA macros | Awegman1 |
| 1 | 0 | 2015-11-16 | * Revision History moved to chapter 7 * Table-Styles removed | Awegman1 |
| 1 | 1 | 2016-03-02 | * Rework according to PCL example | Jbaden1 |
| 1 | 2 | 2016-03-22 | * V1.3: Footer formating corrected (Issue 19) * “Constraints” chapter renamed to “Input Requirements” (Issue 20) | Jbaden1 |
| 1 | 3 | 2016-04-20 | * Broken Wiki links repaired | Jbaden1 |
| 2 | 0 | 2016-05-23 | * Prepared for Specification\_Macros.dotm v2.0 * Additional explanations added to ch. 2.2 “Input Requirements” (ARL and SDS requirements often go here) | Jbaden1 |
| 2 | 1 | 2016-07-08 | * Template version added to footer | Jbaden1 |
| 2 | 2 | 2016-07-15 | * Sample SysML diagrams added * Data Dictionary reworked * Alignment with relevant sections in SRD templated | Jbaden1 |
| 3 | 0 | 2016-09-05 | * Lessons learned from IPRB incorporated | Jbaden1 |
| 4 | 0 | 2016-09-27 | * Alignment with QPIP Feature Function Ownership workstream. Platform Spec renamed to Feature Implementation Spec | Jbaden1 |
| 4 | 1 | 2016-11-04 | * Chapters “Purpose” and “Scope” reworked. | Jbaden1 |
| 4 | 1 | 2016-11-10 | * Subsection for “Logical Service Interfaces” added. | Jbaden1 |
| 5 | 0 | 2017-01-13 | * Meta data updated for specification macros, version 3.1 * SW Unit chapter removed for the time being * Green boxes added for user hints | Jbaden1 |
| 5 | 1 | 2017-01-18 | * Minor editorial changes (e.g. hyperlinks highlighted in comments) | Jbaden1 |
| 5 | 1b | 2017-01-20 | * Some editorial corrections * Substructure of old Network Communication (now Connections) moved to Requirements on Connections | Jbaden1 |
| 6 | 0 | 2018-07-24 | * CR53: * Add new cover sheet * Add disclaimer section * Add the following meta-data to the doc properties for the the new cover sheet   + DocGis1ItemNumber   + DocGis2Classification   + DocType   + DocStatus   + DocIssueDate   + DocReleaseDate * CR63: Update FuSa sharepoint references in templates | Jbaden1 |
| 6 | 0 | 2018-08-06 | * CR81: Incorporate lessons learned from System Service Spec pilot (Vehicle Speed) into AFS and FIS | Jbaden1 |
| 6 | 0 | 2018-09-28 | * Broken links to RE Wiki repaired | Jbaden1 |
| 6 | 0 | 2018-10-31 | * Minor corrections on cover sheet and in footer to be more GIS compliant and VSEM aligned * “Overview” and “Description” exchanged in headings (following common sense) | Jbaden1 |
| 6 | 0 | 2018-11-30 | * Update of Functional Safety sections after review by Functional Safety Team * Initial support for variant handling | Jbaden1 |
| 6 | 0 | 2018-12-01 | * Variant condition fields added consistently * Links updated | Jbaden1 |
| 6 | 0 | 2018-12-11 | * Variant condition fields removed from mapping/allocation tables * Mapping tables simplified * Explanatory text for “Variants” sections revised | Jbaden1 |
| 6 | 0a | 2019-01-04 | * Chapter heading “Inherited Function Requirements” removed. Corresponding table renamed to “Requirements not cascaded”. * E/E Connection table got another column for allocated messages * Naming conventions for Implemented Functions corrected (FncName\_CmpName instead of FncName\_on\_CmpName) * Editorial corrections on the cover sheet * Explanatory text added to “Ethernet” section in chapter “Requirements on Connections” * AIS templates updated. Linked to Wiki page | Jbaden1 |
| 6 | 0a | 2019-01-04 | * Minor restructuring in FuSa chapter – after aligning with ECU Functional Spec * Bugfix: table 13 renamed from FTTI table to FHT table, includes a bug fix: each FSR is allocated to only one ECU/component | Jbaden1 |
| 6 | 0b | 2019-02-04 | * Change: Chapter “Interface Requirements” added to “Implemented Function xxx” section (to have a single chapter for to collect subscriber/publisher interface and mapping requirements which to not conform to the corresponding Data Dictionary objects) * Change: “CAN Interface” subsection renamed to “AIS Interfaces” again. Although several Subscriber/Publisher interface attributes are probably CAN bus specific, other attributes seem to be well suited for other networks than CAN. * Change: Chapter “ECU Specific Requirements” renamed to “Component Specific Requirements” in chapter “Implemented Function xxx”. Table “Requirements not cascaded” renamed to “Component Specific Requirements” and refined to describe changes from Logical Function requirements set more formally. This is also to help during VSEM import to identify those requirements of the Logical Function which cannot be simply carried over to the ECU. * Change: Explanatory text in section “Implemented Function xxx” improved. | Jbaden1 |
| 6 | 0c | 2019-02-05 | * Change: Layout of AIS Interfaces in Data Dictionary reworked to enable Excel Import | Jbaden1 |
| 6 | 0c | 2019-02-20 | * Bugfix: In AIS Interfaces none-picklist fields formatted as invisible | Jbaden1 |
| 6 | 1a | 2019-02-05 | Functional Safety related changes:   * Table “Architectural Redundancy Summary” updated * Section “Functional Flows for FTTI ‘xyz’” added to chapter “Component Interaction Diagrams” * Fault Tolerant Time Summary section added to Functional Safety chapter * Chapter “HW Metrics” added | Jbaden1 |
| 6 | 1a | 2019-04-02 | Headings of “Architectural Redundancy Summary” table clarified | Jbaden1 |
| 6 | 1a | 2019-04-10 | * ASIL Decomposition table moved from Function Spec into the Feature Implementation Spec (ASIL Decomposition of Technical Safety Requirements) * 2 alternative versions of the Function Allocation Table (Standard variant vs. Functional Safety variant) placed next to each other. | Jbaden1 |
| 6 | 1a | 2019-05-31 | * Function Allocation Table split into a base (non FuSa) part and a FuSa part to allow a more flexible mapping of MBSE functions (Logical and Technology) to RE functions (Atomic Logical and Implemented). | Jbaden1 |
| 6 | 1a | 2019-05-31 | * “Input Requirement” section reworked (symmetrically to all other templates). * Sections “Functional Flows for FTTI xyz” and “Fault Tolerant Time Summary” removed, because guidance is not available yet. * “Reference” and “Glossary” section moved back to introduction, i.e., to the very beginning of the document (such that also section 2 can already rely on it). * Some mostly editorial changes per request from FuSa team. | Jbaden1 |
| 6 | 1a | 2019-07-02 | * "Important" box added on cover sheet which points to the macros * “Input Requirements” section renamed to Input Information (after discussion with FuSa team) | Jbaden1 |
| 6 | 1a | 2019-07-17 | * Chapter “Message List” removed from CAN and LIN specific chapters of section “Requirements on Connections” | Jbaden1 |
| 6 | 1a | 2019-10-08 | * Chapter “ASIL Decomposition of Technical Safety Requirements”: Input TSRs are specified in the chapter right above the decomposition table. | Jbaden1 |
| 6 | 1a | 2019-10-09 | * Chapter “Service Oriented Communication” moved to section “Messages” in the Data Dictionary. Details from Central SW Wiki about FNV2 SOA added | Jbaden1 |
| 6 | 1a | 2019-10-25 | * Minor updates for HW IOs/Signals * Subsection “Functional Safety” removed from chapter “Feature Implementation Modeling”. Per requrest from FuSa team since no guidance is available how to model e.g. FHT timing diagram. | Jbaden1 |
| 6 | 1a | 2019-05-11 | * Copyright notice shortened and moved to cover sheet and added to footer (to be compliant [with Ford copyright guidelines](http://www.fgti.ford.com/client/NewFGTI/CopyrightNotice.html)) * Term “Disclaimer” no longer used for what is actually only a copyright notice | Jbaden1 |
| 6 | 1a | 2019-22-11 | * Some minor modifications for the SOA APIs/MQTT Messages in the section “Messages” of the Data Dictionary (section references Service Contracts via the API name) * Some minor updates of the Input/Output mapping tables in section “Requirements on Components” for mappings to SOA APIs and EDAS signals. | Jbaden1 |
| 6 | 1a | 2019-12-05 | * Upstream Documents section added to “Input Requirements/Documents” table * Custom style table formatting removed | Jbaden1 |
| 6 | 1a | 2020-01-07 | * Some fine tuning for naming conventions of E/E components and connections. * List of HW I/O signal types reduced to RF-A, RF-D, D, A, Networked and PWM. * Protocol column added to the E/E connection table | Jbaden1 |
| 6 | 1a | 2020-01-07 | * “HW Metric” and “Architecture Redundancy Summary” sections removed per request from the Functional Architecture Team (based on Governance Board decision [FSTGB-97](mailto:TrackLite%20%23%20FSTGB-97:%20https://www.tracklite.ford.com/prweb/PRAuth/TrackLiteSSO?pyActivity=@baseclass.RedirectAndRunWraper&ThreadName=WorkLinkThread&bPurgeTargetThread=true&AccessGroupName=FSTGB:ProjectAdministrators&Location=pyActivity%3DWork-.Open%26Action%3DReview%26HarnessPurpose%3DReview%26InsHandle%3DFORD-FSTGB-WORK+FSTGB-97)) * “Functional Safety” chapter moved to “Feature Implementation Requirements” section. “Function Allocation” chapter seemed no longer appropriate. | Jbaden1 |
| 6 | 1a | 2020-01-07 | * Ordering of fields in AIS interfaces tables modified to conform with the Macro Template and the Importer Sheet * Page Header: no longer in bold letters | Jbaden1 |
| 6 | 1a | 2020-03-09 | * Missing doc property “LatestSigMappingID” and “LatestAisInterfaceID” added * doc property “CopyrightDate” re-formatted to text and copyright date field in footer corrected * Version numbering re-initialized as 0.1 * Init value of version/revision date set to “yyyy/mm/dd” instead of “yyyy-mm-dd” to be in line with the “Edit Document Property” dialog * Type of “Latest….ID” doc properties changed from Text to Number | Jbaden1 |
| 6 | 1a | 2020-03-11 | * “Mapping” table removed from template. Has been migrated to macro. | Jbaden1 |
| 6 | 1a | 2020-03-13 | * Separate chapter “Technical Safety Requirements” removed. Content already covered by Allocation Table in chapter Function Allocation. * “Implemented Function” replaced by term “Technology Function” | Jbaden1 |

# Appendix

## Data Dictionary

### Logical Signals

**#Hint:** Logical Signals are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link**: [*RE Wiki – Adding a Logical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Logical+Signal+or+Parameter)

**#Macro**: Add Ins -> Add Requirement macro (select “Logical Signal” as type)

ApplySettingStatus

Determines if a profile setting had successfully been set by the Profile Setting Server.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | 0x0 | Final Result – Success |
| 0x1 | Final Result – Fail |
| 0x2 | Final Result – Information |
| 0x3 | Intermediate Result - Wait |
| **Unit** | |  |

Change\_Notification

Notifies the Profile Management Server that a change has been made to the active profile.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | 0x0 | NULL |
| 0x1 | PROFILE\_UPDATED |
| 0x2 | DATA\_LOST |
| **Unit** | |  |

FeatureNumber

Signal that carries the hexadecimal values assigned to each setting that is stored in the profile.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Continuous |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value | 0x0000 |
| Max Value | 0xFFFF |
| Resolution | 0x1 |
| Offset | None |
| **Unit** | |  |

FeatureValue

Carries the setting configuration for a Feature Number. Feature Numbers and Feature Values are always paired together.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Continuous |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value | 0x0000 |
| Max Value | 0xFFFF |
| Resolution | 0x1 |
| Offset | None |
| **Unit** | |  |

OperationCode

Determines which operation the Profile Setting Server is requested to execute. Depending on the operation code, the following signals could contain more or less data.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | 0x00 | Reserved |
| 0x01 | Query |
| 0x02 | Set |
| 0x03 | Import |
| 0x04 | Export |
| 0x05 | Copy |
| 0x06 – 0xFF | Reserved |
| **Unit** | |  |

PersIndex

Determines the profile number that the Profile Setting Server needs to execute an operation on.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | 0x00 | Pers1 |
| 0x01 | Pers2 |
| 0x02 | Pers3 |
| 0x03 | Pers4 |
| 0x04 | Vehicle |
| 0x05 – 0xFF | Reserved |
| **Unit** | |  |

Authentication\_Status

Internal vehicle status that broadcasts if a profile is authenticated or not. Profile Management Server is the transmitter of the signal and the receiver is the Profile Interface Client.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | WAITING |
| **0x1** | SUCCESS |
| **0x2** | FAIL |
| **Unit** | |  |

DownloadLockout\_Rq

A command sent to the Profile Interface Client when the Profile Management Server is actively downloading and applying profile data. Signal prompts Profile Interface Client to display a lockout screen and communicates status.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | NULL |
| **0x1** | ON |
| **0x2** | OFF |
| **0x3** | ERROR |
| **Unit** | |  |

ExportMessage\_Rq

Signal to communicate the status of a profile being exported to the cloud. Signal used to determine errors/failures that occurred during profile export.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | NULL |
| **0x0** | ERROR |
| **Unit** | |  |

PasscodeValues

Passcode values received by the user and passed to Profile Management Server. Signal also communicates if the user selected to "skip" the secondary authentication.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | NULL |
| **0x1** | SKIP |
|  | PASSCODE VALUES |
| **Unit** | |  |

Passcode\_Rq

Signal from Profile Management Server to Profile Interface Client to display the passcode screen.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | NULL |
| **0x1** | NEW |
| **0x2** | ERROR |
| **Unit** | |  |

PrimaryAuth\_Rq

Signal from Profile Interface Client to Profile Management Server to store Primary Authentication for the active user.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

PrimaryAuth\_Rsp

Signal from Profile Management Server to Profile Interface Client providing the status of the Primary Authentication linking to the active profile.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

ProfileCredentials

Credential values that are collected by Profile Interface Client and sent to the Profile Management Server.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) |  | Username  Password |
| **Unit** | | String |

SetupSecondaryAuth\_Rq

Passcode values sent from Profile Interface Client to Profile Management Server during initial passcode creation.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) |  | Passcode Values |
| **Unit** | |  |

SetupSecondaryAuth\_Rsp

Signal from Profile Management Server to Profile Interface Client communicating status of secondary authentication passcode creation for the profile being created/edited.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | NULL |
| **0x1** | SUCCESS |
| **0x2** | ERROR |
| **Unit** | |  |

Unlink\_Rq

Request from the Profile Interface Client to the Profile Management Server to unlink a profile from its linked Ford account.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) |  | PersIndex  UserID |
| **Unit** | |  |

Unlink\_Rsp

Response to the Unlink\_Rq indicating a Success or Failure when unlinking a profile from a Ford Account.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | NULL |
|  | **0x1** | SUCCESS |
|  | **0x2** | FAIL |
| **Unit** | |  |

ActivePersonality\_St

Indicates the active vehicle profile.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| Value  (Discrete  Encoding) | 0x0 | Pers1 |
| 0x1 | Pers2 |
| 0x2 | Pers3 |
| 0x3 | Pers4 |
| 0x4 | Vehicle |
| 0x5 | NotDetermined |
| 0x6 | NotUsed |
| 0x7 | NotUsed |
| **Unit** | |  |

ApplySettingStatus

Determines if a profile setting had successfully been set by the Profile Setting Server.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | 0x0 | Final Result – Success |
| 0x1 | Final Result – Fail |
| 0x2 | Final Result – Information |
| 0x3 | Intermediate Result - Wait |
| **Unit** | |  |

Authentication\_Status

Internal vehicle status that broadcasts if a profile is authenticated or not. Profile Management Server is the transmitter of the signal and the receiver is the Profile Interface Client.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

DeleteProfile\_Rq

A command to the Profile Management Server to delete a specific profile. The profile identifier shall be a parameter of the DeleteProfile\_Rq

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) |  | Profile Name |
| **Unit** | | String |

DeleteProfile\_Rsp

Profile Management Server response to the DeleteProfile\_Rq. Indicates Success or Failure.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | SUCCESS |
| **0x1** | FAIL |
| **Unit** | | String |

Delete\_Rq

Request from Profile Management Server to Enhanced Memory system to delete a profile.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

DownloadLockout\_Rq

A command sent to the Profile Interface Client when the Profile Management Server is actively downloading and applying profile data. Signal prompts Profile Interface Client to display a lockout screen and communicate status.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | NULL |
| **0x1** | ON |
| **0x2** | OFF |
| **0x3** | ERROR |
| **Unit** | |  |

ExportMessage\_Rq

Signal to communicate the status of a profile being exported to the cloud. Signal used to determine errors/failures that occurred during profile export.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

ExportTrigger\_Rq

Signal received by the Profile Management Server that triggers an export of profile settings to the CV Profile Management Server.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

FeatureNumber

Signal that carries the hexadecimal values assigned to each setting that is stored in the profile.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

FeatureValue

Carries the setting configuration for a Feature Number. Feature Numbers and Feature Values are always paired together.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

OperationCode

Determines which operation the Profile Setting Server is requested to execute. Depending on the operation code, the following signals could contain more or less data.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

PasscodeValues

Passcode values received by the user and passed to Profile Management Server. Signal also communicates if the user selected to "skip" the secondary authentication.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Passcode\_Rq

Signal from Profile Management Server to Profile Interface Client to display the passcode screen.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

PersIndex

Determines the profile number that the Profile Setting Server needs to execute an operation on.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | 0x00 | Pers1 |
| 0x01 | Pers2 |
| 0x02 | Pers3 |
| 0x03 | Pers4 |
| 0x04 | Vehicle |
| 0x05 – 0xFF | Reserved |
| **Unit** | |  |

PrimaryAuth\_Rq

Signal from Profile Interface Client to Profile Management Server to store Primary Authentication for the active user.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

PrimaryAuth\_Rsp

Signal from Profile Management Server to Profile Interface Client providing the status of the Primary Authentication linking to the active profile.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

ProfileComplete\_St

Status sent by the Profile Management Server to the Enhanced Memory Profile Interface Client to indicate that a profile has been successfully created and exported to the cloud.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

ProfileCredentials

Credential values that are collected by Profile Interface Client and sent to the Profile Management Server.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

SetupSecondaryAuth\_Rq

Passcode values sent from Profile Interface Client to Profile Management Server during initial passcode creation.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

SetupSecondaryAuth\_Rsp

Signal from Profile Management Server to Profile Interface Client communicating status of secondary authentication passcode creation for the profile being created/edited.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Unlink\_Rq

Request from the Profile Interface Client to the Profile Management Server to unlink a profile from its linked Ford account.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Unlink\_Rsp

Response to the Unlink\_Rq indicating a Success or Failure when unlinking a profile from a Ford Account.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

UserIDRecall\_Rq

Recall request based on a user ID.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

UserID\_Rq

Request to the authentication manager to ID the user of the vehicle.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

UserID\_Rsp

User ID information sent by the Identification Client to the Profile Management Server that indicates who is identified in the vehicle.

|  |  |  |
| --- | --- | --- |
| **ASIL** | |  |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

### Logical Parameters

**#Hint:** Logical Parameters are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link**: [*RE Wiki – Adding a Logical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Logical+Signal+or+Parameter)

**#Macro:** Add Ins -> Add Requirement macro (select “Logical Parameter” as type)

### Technical Signals

**#Hint:** This section lists all GSDB + GDT + SW signals relevant for the feature deployment.

**#Link**: [*RE Wiki – Adding a Technical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Signal+or+Parameter)

**#Macro:** Add Ins -> Add Requirement macro (select “Technical Signal” as type)

#### GSDB Signals

**#Hint:** This part of the Data Dictionary lists signals, which should go to the GSDB in VSEM, but do not exist in the GSDB in VSEM yet, but are or will be requested for the GSDB. Those would go temporarily to this section in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server) in VSEM.

###TSG\_PPP\_00001### ECUPpp\_D\_Rq

Action codes for the modules that determine how the rest of the signals in the message should be interpreted.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0x0 - NULL |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Discrete  Encoding) | 0x0 | NULL |
| 0x1 | Query |
| 0x2 | SET |
| 0x3 | COPY |
| 0x4 | NotUsed\_1 |
| 0x5 | NotUsed\_2 |
| 0x6 | NotUsed\_3 |
| 0x7 | NotUsed\_4 |
| **Unit** | | SED |

###TSG\_PPP\_00002### ECUPppId\_No\_Actl

Unique ID for each request, this has range 1-7. Value 0 is reserved for representing RESET scenarios.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0x0 |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Continuous Encoding) | Min Value | 0 |
| Max Value | 7 |
| Resolution | 1 |
| Offset | 0 |
| **Unit** | | unitless |

###TSG\_PPP\_00003### ECUPersIdx\_No\_Actl

SED signal for profile identification.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0x0 |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Discrete  Encoding) | 0x0 | Profile 1 |
| 0x1 | Profile 2 |
| 0x2 | Profile 3 |
| 0x3 | Profile 4 |
| 0x4 | Vehicle/Guest |
| **Unit** | |  |

###TSG\_PPP\_00004### ECUFeatNum\_No\_Actl

This is an unitless signal representing Feature code of the parameter. Range is 0x0-0xFFFF where 0xFFFF is reserved for “All Settings”.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0xFFFF |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Continuous Encoding) | Min Value | 0 (0x0000) |
| Max Value | 65535 (0xFFFF) |
| Resolution | 1 |
| Offset | 0 |
| **Unit** | | unitless |

###TSG\_PPP\_00005### ECUFeatValue\_No\_Actl

This is an unitless signal representing parameter values stored in NVM of profile setting server. Range is 0x0 – 0xFFFF. This parameter shall be ignored by the profile setting server when action codes are “Query” or “Copy”

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0xFFFF |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Continuous Encoding) | Min Value | 0 (0x0000) |
| Max Value | 65535 (0xFFFF) |
| Resolution | 1 |
| Offset | 0 |
| **Unit** | |  |

###TSG\_PPP\_00006### ECUApplySett\_D\_Status

Response code for the received request.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0x0 - NULL |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Discrete  Encoding) | 0x0 | NULL |
| 0x1 | SUCCESS |
| 0x2 | FAILED |
| 0x3 | IN\_PROGRESS |
| 0x4 | NotUsed\_1 |
| 0x5 | NotUsed\_2 |
| 0x6 | NotUsed\_3 |
| 0x7 | NotUsed\_4 |
| **Unit** | | SED |

###TSG\_PPP\_00007### ECUPppID\_No\_Res

This is an unitless signal needed to transmit the same ID received in the request, profile setting server will echo back the same value for ECG to corelate the response to request.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0 |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Continuous Encoding) | Min Value | 0 |
| Max Value | 7 |
| Resolution | 1 |
| Offset | 0 |
| **Unit** | | unitless |

###TSG\_PPP\_00008### ECUPersIndex\_No\_Res

SED signal for profile identification.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0x0 |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Discrete  Encoding) | 0x0 | Profile 1 |
| 0x1 | Profile 2 |
| 0x2 | Profile 3 |
| 0x3 | Profile 4 |
| 0x4 | Vehicle/Guest |
| **Unit** | |  |

###TSG\_PPP\_00009### ECURespIndex\_No\_Res

This is an unitless signal significant for Query All use case. Profile setting server shall send the total number of parameters with response and keep decrementing it till it reaches 1.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0 |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Continuous Encoding) | Min Value | 0 (0x0) |
| Max Value | 32 (0x20) |
| Resolution | 1 |
| Offset | 0 |
| **Unit** | | unitless |

###TSG\_PPP\_00010### ECUFeatNum\_No\_Res

This is an unitless signal representing Feature code of the parameter. Range is 0x0-0xFFFF where 0xFFFF is reserved for “All Settings”.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0xFFFF |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Continuous Encoding) | Min Value | 0 (0x0000) |
| Max Value | 65535 (0xFFFF) |
| Resolution | 1 |
| Offset | 0 |
| **Unit** | | unitless |

###TSG\_PPP\_00011### ECUFeatValue\_No\_Res

This is an unitless signal representing parameter values stored in NVM of profile setting server. Range is 0x0 – 0xFFFF. This parameter shall be ignored by the profile setting server when action codes are “Query” or “Copy”

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0xFFFF |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Continuous Encoding) | Min Value | 0 (0x0000) |
| Max Value | 65535 (0xFFFF) |
| Resolution | 1 |
| Offset | 0 |
| **Unit** | |  |

###TSG\_PPP\_00012### ECUSettChg\_D\_Stat

This is state encoded signal to send notification to ECG service.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Init Default Value** | | 0x0 |
| **Encoding Type Name** | |  |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed, | | |
| **Value**  (Discrete  Encoding) | 0x0 | NULL |
| 0x1 | PROFILE\_UPDATED |
| 0x2 | DATA\_LOST |
| **Unit** | | unitless |

#### HW I/Os

**#Hint:** This chapter lists signals, which will be mapped to hardwired I/Os. Those get typically refer to VSEM EDAS signals (or input/output signals of device transmittals in VSEM GDT).

#### Diagnostic Interfaces

**#Hint:** This chapter lists Diagnostic Interfaces (DTCs and DIDs), which get mapped to Logical Parameters in context of the Technology Functions in chapter “Parameters” of the Function Interfaces. Those DTC/DID names should match the names in the diagnostics specification (Part 2).

**#ToDo:** Currently the template below is just a proposal. A macro still needs to be created

##### DTCs

###<DTC\_<ID>>### <DTC Name>

<Some Description of the DTC.

Refer to VSEM document “[Diagnostic Fault Coverage and DTC Numbers](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=yAUtrNhnx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

[Design Consideration](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=yAUtrNhnx3NrTDAAAAAAAAAAAAA&servername=Production_Server)”, what to fill into the attributes below>

|  |  |
| --- | --- |
| **Test Period Time** |  |
| **Test Run Criteria,** |  |
| **Enable Criteria (EC)** |  |
| **Applicable** |  |
| **FailureTypeBytes** |  |
| **Test Period Time** |  |
| **Test Run Criteria,** |  |

##### DIDs

**#Hint**: This section lists diagnostic DID which Technical Parameters get mapped to.

**#Todo**: A proper template derived from the Part 2 spec still needs to be created.

### Technical Parameters

**#Hint:** This section lists all Method 2, Method 3 and calibration parameters relevant for the feature deployment.

**#Link**: [*RE Wiki – Adding a Technical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Signal+or+Parameter)

**#Macro:** [Add Ins -> Add Requirement macro](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#HowtousetheSpecificationTemplates-AddNewRequirement) (select “Technical Parameter” as type)

### Mappings

**#Hint**: This section lists mapping objects for Logical Signals / Parameters to their GSDB + GDT + SW counterparts (1:N mapping is supported). Mapping objects are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link:** [RE Wiki – Adding a Signal or Parameter Mapping](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

**#Macro:** Add Ins -> Add Requirement macro (select “Mapping” as type)

### Technical Interfaces

**#Hint:** This section lists port/interface details, which define how network/SW/HW signals are published / subscribed.

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

#### AIS Interfaces

**#Hint:** This chapter lists the AIS subscriber and publisher interface objects (managed in VSEM), which are needed to deploy the feature to the E/E architecture. If AIS interfaces do not yet exist in VSEM, those may temporarily be managed as a workaround in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link:** [System Engineering Portal – AIS Release 3.2](https://pd3.spt.ford.com/sites/fede/vsem-spls/Shared%20Documents/02-ais/methods/AIS%20Methods%20Document.pptx?web=1)  
[RE Wiki - AIS Interfaces](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface#AddingaTechnicalInterface-AisInterfaces)

[*Publisher Interface AIS in VSEM*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=zjYtY3Jcx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

[*Subscriber Interface AIS in VSEM*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=LSYtewY7x3NrTDAAAAAAAAAAAAA&servername=Production_Server)

**#Macro:** Add Ins -> Add Requirement macro (select “AIS Subscriber If” or “AIS Publisher If” as type)

##### Publisher Interfaces

##### Subscriber Interfaces

#### AUTOSAR Ports

**#Hint:** Those AUTOSAR Classic (provided and required) ports, which are used by the feature but are not managed in a central repository yet, could be listed here.

### Messages/APIs

#### CAN Bus “<Bus Name>”

**#Hint:** This section gives the relevant extract from the [Central Message Database (CMDB) in VSEM](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server) .

###<MSG\_MessageID### AbsPppRqst

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CAN ID** | **Transmission Mode** | **Period** | **Signal Names** | **Transmitter(s)** | **Receiver(s)** |
|  |  |  | AbsPpp\_D\_Rq | ECG | Abs |
| AbsPppId\_No\_Actl |
| AbsPersIds\_No\_Actl |
| AbsProfileMode\_D\_Stat |
| AbsFeatNum\_No\_Actl |
| AbsFeatValue\_No\_Actl |

#### LIN Bus “<Bus Name>”

#### AUTOSAR Interfaces

**#Hint:** Those AUTOSAR Classic (Sender/Receiver and Client/Server) Interfaces, which are used by the feature but not managed in a central repository yet, should be listed here.

#### SOA Service Contracts

**#Hint:** This part of the Data Dictionary lists Service APIs/MQTT messages and embedded data elements, which are used for the Service Oriented Architecture (SOA). If those APIs/MQTT messages already exist e.g. in the [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog), simply add a reference to those yet.

Information on FNV2 SOA can be found in the ECG wiki page

* MQTT Topic Naming: [*FNV2-SOA: MQTT Topic and Message Structure*](https://www.eesewiki.ford.com/display/ecg/FNV2-SOA%3A+MQTT+Topic+and+Message+Structure?src=sidebar)
* message syntax and proper naming can be found [*SOA API Messaging Guidelines*](https://www.eesewiki.ford.com/x/Q7rKAg)

For examples what to fill into the table fields below refer to [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog)

###<ServiceContractID>### Service Contract Name

<Service contract purpose/behavior>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Messaging Pattern | Frequency  (For Data Broadcast Only) | Message Data Element(s)  (Must Match GPB) or applicable CAN signal | Description of Data Element(s) | Topic Name |
| Choose an item. |  | GBP Data element / CAN Signal name 1 | Detailed encoding of data element 1 |  |
| … |  |  |
| GBP Data element / CAN Signal name 1 | Detailed encoding of data element 3 |  |

### Encoding Types

**#Link:** [*RE Wiki – Adding Encoding Types*](http://wiki.ford.com/display/RequirementsEngineering/Adding+an+Encoding+Type)

**#Macro:** Add Ins -> Add Requirement macro (select “Encoding Type” as type)

Document ends here.