Personal and Portable Profiles

<<Feature>>

(F002032/B;1)

|  |  |  |  |
| --- | --- | --- | --- |
| Document Type | **Feature Implementation Specification (FIS)** | |  |
| Template Version | **6.1a** | |  |
| SysML Report Template Version | **O Beta (11/24/2019)** | |  |
| Document ID | **ffst01.10\_featuredocument\_sysmlreporttemplate** | |  |
| Document Location |  | |  |
| Document Owner | **Justin Bauer, Walter Stephens** | |  |
| Document Revision | **FD0** | |  |
| Document Status | **Draft** | |  |
| Date Issued | **2021/03/10** | |  |
| Date Revised | **2021/03/10** | |  |
| Document Classification | GIS1 Item Number: | **27.60/35** |  |
| GIS2 Classification: | **Confidential** |

|  |  |  |  |
| --- | --- | --- | --- |
| Document Approval | | | |
| Person | Role | Email | 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

**Disclaimer**

**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.**

This document contains information developed and accumulated by and for FORD MOTOR COMPANY. As such, it is a proprietary document, which, if disseminated to unauthorized persons, would provide others with restricted information, data, or procedures not otherwise available, exposing the FORD MOTOR COMPANY to potential harm.

Employees and suppliers having custody of this specification or authorized to use it must be cognizant of its proprietary nature and ensure that the information herein is not made available to unauthorized persons.

FORD MOTOR COMPANY reserves the right to protect this work as an unpublished copyrighted work in the event of an inadvertent or deliberate unauthorized publication. FORD MOTOR COMPANY also reserves its rights under copyright laws to protect this work as a published work.

This document or portions thereof shall not be distributed outside FORD MOTOR COMPANY without prior written consent. Refer all questions concerning disclosure to the author(s) or any duly authorized representative of Ford Motor Company.

**Copyright** © **2019 Ford Motor Company**

**Contents**

1 Introduction 6

1.1 Document Purpose 6

1.2 Document Scope 6

1.3 Document Audience 6

1.3.1 Stakeholder List 6

1.4 Document Organization 6

1.4.1 Document Context 6

1.4.2 Document Structure 6

1.5 Document Conventions 7

1.5.1 Requirements Templates 7

1.5.1.1 Identification of requirements 7

1.5.1.2 Requirement Attributes 7

2 Feature Implementation Overview 8

2.1 Description 8

2.2 Input Requirements 8

2.3 Assumptions 9

2.4 References 9

2.4.1 Ford Documents 9

2.4.2 External Documents and Publications 10

2.5 Glossary 10

2.5.1 Definitions 10

2.5.2 Abbreviations 11

3 Feature Implementation Architecture 12

3.1 Functional Architecture 12

3.1.1 Description 12

3.1.2 Function List 12

3.1.2.1 Functions of 13

3.1.2.2 Functions of 14

3.1.3 Signal List 14

3.2 Physical Architecture 18

3.2.1 E/E Architecture 18

3.2.1.1 E/E Architecture Variants 19

3.2.1.2 E/E Components 20

3.2.1.3 E/E Connections 20

3.2.1.4 Signal List 22

3.2.2 Software Component Architecture 22

3.2.2.1 Description 22

3.3 Function Deployment 22

3.3.1 Deployment Variants 23

3.3.1.1 Main (Only) variant 23

3.3.1.2 Deployment “Variant 1” 23

3.3.2 Function Allocation 24

3.3.2.1 Functional Safety 25

4 Feature Implementation Modeling 27

4.1 Component Interaction Diagrams 27

4.1.1 Scenario: “System Startup / Shutdown” 27

4.1.2 Scenario: “Normal Operation” 27

4.1.3 Functional Safety 28

4.1.3.1 Fault Handling Time Analysis 28

4.1.3.2 Requirements Derivation Diagram 29

4.2 Component Interface Behavior Diagrams 29

5 Feature Implementation Requirements 30

6 Open Concerns 31

7 Revision History 32

8 Appendix 33

8.1 Data Dictionary 33

8.1.1 Logical Signals 33

8.1.2 Logical Parameters 33

8.1.3 Technical Signals 33

8.1.3.1 GSDB Signals 34

8.1.3.2 Service Oriented Communication 34

8.1.3.3 Hardwired Signals 34

8.1.3.4 Diagnostic Interfaces 34

8.1.4 Technical Parameters 36

8.1.5 Mappings 36

8.1.6 Technical Interfaces 37

8.1.6.1 AIS Interfaces 37

8.1.6.2 Service Oriented Communcation (SoC) Interfaces 38

8.1.6.3 AUTOSAR Ports (SW Interfaces) 38

8.1.7 Messages 38

8.1.7.1 CAN Bus “<Bus Name>” 38

8.1.7.2 LIN Bus “<Bus Name>” 39

8.1.8 Encoding Types 39

**List of Figures**

Figure 1: 21

Figure 2:  24

Figure 3: 30

**List of Tables**

Table 1‑1: Electrical Architecture(s) referenced in this document 6

Table 1‑2: Functions referenced in this document 6

Table 2‑1: Ford Internal Documents 10

Table 2‑2: Ford Internal Documents *(not specified in SysML model)* 11

Table 2‑3: External Documents and Publications 11

Table 2‑4: External Documents and Publications *(not specified in SysML model)* 11

Table 2‑5: Definitions Used In This Document 12

Table 2‑6: Abbreviations used in this document 12

Table 3‑1: List of Functions 15

Table 3‑2: List of ‑‑ Functions 15

Table 3‑3 List of signals sent by 18

Table 3‑4 List of signals received by 19

Table 3‑5: Electrical Components 21

Table 3‑6: E/E Connections for 23

Table 3‑7: Function Allocation Table 26

Table 3‑8: Function Allocation Table 27

Table 3‑9: Architectural Redundancy Summary 27

Table 4‑1: Fault Handling Time Table 30

Table 5‑1: Input Signal mappings of ‑ 34

Table 5‑2: Output Signal mappings of ‑ 35

Table 5‑3: Parameter mappings of ‑ 36

Table 5‑4: Component Specific Requirements 37

Table 5‑5: Inherited Requirements 37

Table 5‑6: Input Signal mappings of ‑ 39

Table 5‑7: Output Signal mappings of ‑ 40

Table 5‑8: Parameter mappings of ‑ 40

Table 5‑9: Component Specific Requirements 40

Table 5‑10: Inherited Requirements 40

Table 5‑11: Input Signal mappings of Component: ‑ 41

Table 5‑12: Output Signal mappings of Component: ‑ 42

# 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 - . 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **CDSID** | **Contact Info** | **Role** | **Stakeholder Group** |
| Naomee Rahman |  |  |  |  |
| Joshep Isho |  |  |  |  |
| Michael Dawson |  |  |  |  |
| John Ricks |  |  |  |  |
| Srikanth Hari |  |  |  |  |
| Hina Noor |  |  |  |  |
| John Doe | Jdoe |  | Model Architect | CVPP Basic Design |
| Scott Watkins |  |  |  |  |
| Michaek Alievsky |  |  |  |  |
| Dennis Lenkowski |  |  |  |  |
| Anthony King |  |  |  |  |
|  |  |  |  |  |
| John Murphy |  |  |  |  |
| Patrick Brown |  |  |  |  |
| Gregory Reed |  |  |  |  |
| Farhan Sethi |  |  |  |  |
| John Rentis |  |  |  |  |
| John Correia |  |  |  |  |
| Christopher Van Auken |  |  |  |  |
| Jonathan laquinto |  |  |  |  |
| Dave Walus |  |  |  |  |
| Mike Gulyas |  |  |  |  |
| Evangelos Foutis |  |  |  |  |
| Timothy Thieves |  |  |  |  |
| Walter Stephens |  |  |  |  |
| William Falconer |  |  |  |  |
| Dennis Trombley |  |  |  |  |
| Stephan Coe |  |  |  |  |
| Francis Luk |  |  |  |  |
| Vijay Srikakolapu |  |  |  |  |
| Ibaa Al-Hayek |  |  |  |  |

## 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.

### 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.

## 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. |
| Action Validation Criteria | Validation criteria for these actions |
| 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. |
| Driver Actions or Others | Description of driver actions or other people |
| 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. |
| FTTI | Fault Tolerance Time Interval |
| Functional Redundancies | Functional redundancy - fault tolerance |
| High speed | Approximately more than 52 mph (83 kph) |
| 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. |
| Low speed | Approximately 12 to 36 mph (19 to 58 kph ) |
| Medium speed | Approximately 36 mph to 52 mph (58 to 83 kph) |
| 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. |
| PPP | Personal and Portable Profiles |
| 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. |
| RFI | Reduced Functionality Interval |
| Secondary Authentication Passcode | A passcode that a user configures for their profile. Entering this passcode provides the user full access to their profile. |
| SPSS | Subsystem Part Specific Specification. Type of requirement document at Ford. |
| term glossary | A term glossary is a table of agreed upon definitions for terms used in project development that may provide clarity or avoid confusion to stakeholders. |
| TLA | Three Letter Acronym |
| Very Low Speed | Approximately 0 to 12 mph (0 to 19 kph) |

Table 1‑4: Definitions used in this document

### Abbreviations

*No acronyms specified.*

# 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.

F002032/B;1 Personal and Portable Profiles

Develop a cloud-based platform / ecosystem to deliver a personalized in-vehicle experience that is seamlessly transferrable between Ford/Lincoln vehicles. This would reduce the time and effort a customer needs to setup / customize a vehicle (known pain point for subscription based ownership). Creating a personalized experience will increase customer loyalty and provide a data platform for vehicle and cloud based predictions, Digital Assistant, in-vehicle purchases and other monetization strategies. This in-vehicle profile will link with the centralized member profile for a common profile across all channels of the ecosystem.

## 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** | | | |
|  | <Example:  id + title of relevant Function Spec> | <Example: “Function requirements of Logical Function …”> | <Note: If you reference a requirement in this column, then that requirement should have a trace link in its [“Source”/”Source Req.” attribute](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) field pointing back to the input requirement (or to a requirement inside the input document) given in this table row> |
|  |  |  |  |
| **Ford Engineering Standards** | | | |
|  | <Example: some SDS (requirement)> |  |  |
|  |  |  |  |
| **Legal Regulations** | | | |
|  | Compliance with FMVSS101 | The Feature shall comply with FMVSS101. |  |
|  |  |  |  |
| **Industry Standards** | | | |
|  | ISO 26262 | The system should be developed according to Ford's implementation of Functional Safety. |  |
|  |  |  |  |
| **Other Sources** | | | |
|  |  |  |  |
|  | Recall profile | Every time that the user walks up to a ford/lincoln vehicle that he has seen before and saved settings to, the vehicle shall recall the user prefered settings. |  |
|  | Absolute Position Seat 6 | When a seat is equipped with power memory recline, an absolute position point shall be defined as full forward recline |  |
|  | Absolute Position Seat 2 | When a seat is equipped with power memory height adjust movement, an absolute position point shall be defined as full down track travel position |  |
|  | Absolute Position Seat 3 | When a seat is equipped with power memory cushion tilt movement, an absolute position point shall be defined as full tilt down position |  |
|  | Profile Export | When a user Ford Account is attached to a portable profile, the vehicle using portable profiles shall transfer any new vehicle/ portable profile settings added with any Ford/Lincoln vehicle in the Ecosystem. |  |
|  | Absolute Position Pedal 1 | When a pedal system is equipped with power memory pedal adjustment, an absolute position point shall be defined as the pedal in the farthest away position from the occupant |  |
|  | Absolute Position Seat 1 | When a seat is equipped with power memory fore/aft movement, an absolute position point shall be defined as full rear track travel position |  |
|  | Absolute Position Steering Column 2 | When a steering column is equipped with power memory tele-in/tele-out movement, an absolute position point shall be defined as full tele-in |  |
|  | Absolute Position Seat 5 | When a seat is equipped with power memory right cushion extension movement, an absolute position point shall be defined as no extension |  |
|  | Temporary Profile | The portable profiles feature shall support a type of portable profile called Temporary portable profile. |  |
|  | User Authentication | The portable profiles feature shall automaticaly authenticate any vehicle ocuppant on approach or entry for any vehicle the occupant has previously paired with. |  |
|  | Absolute Position Seat 4 | When a seat is equipped with power memory left cushion extension movement, an absolute position point shall be defined as no extension |  |
|  | Absolute Position Seat 8 | When a seat is equipped with power memory lumbar, an absolute position point shall be defined as full off or flush with seat (not engaged) \*Covers if there is ever a non-bladder lumbar memory feature added |  |
|  | Profile Enhancing data enable | the portable profiles feature shall enable customer portable profile data to be collected form user profiles. |  |
|  | Profile Routine support | When the vehicle implements any routines prompted/excecuted by habits/patterns/geolocation, the portable profiles feature shall support the exectution of this routines by providing any data collected from the user, vehicle and preferences. |  |
|  | Absolute Position Seat 10 | When a seat is equipped with power memory headrest up/down movement, an absolute position point shall be defined as full down |  |
|  | Restore Profile | The Portable Profile Feature shall store old profile versions so that a user can restore older profile versions |  |
|  | Absolute Position Seat 9 | When a seat is equipped with power memory headrest fore/aft movement, an absolute position point shall be defined as full aft (full back) |  |
|  | Preferencess accesibility | The portable profiles feature shall be capable to access user vehicle preferences at anytime (Mobile Device, Vehicle). |  |
|  | Absolute Position Steering Column 1 | When a steering column is equipped with power memory tilt up/down movement, an absolute position point shall be defined as full tilt up in the travel window |  |
|  | Sub-portable profile | The portable profiles feature shall allow portable profiles to contain sub profiles. |  |
|  | Temporary profile remove | When a Ford/Lincoln Customer temporaly uses a Ford/Lincoln vehicle, the portable profiles feature shall remove the user prefered settings after the temporay usage of the vehicle is finished. |  |
|  | Feature enable/disable | When Portable Profile feature is available in a Ford/Lincoln vehicle, the vehicle shall allow the portable profiles feature to be enabled/disabled. |  |
|  | to Who Profile data is shared | The portable profiles feature shall allow Ford Motor Company to determine to who the data collected from user profiles is shared with. |  |
|  | Absolute Position Seat 7 | When a seat is equipped with power memory upper back pivot, an absolute position point shall be defined as full rearward recline |  |
|  |  |  |  |

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.

1. There needs to be a clear way to wipe customer data from a vehicle.
2. Selecting a driver profile should set priority on Bluetooth connection. The owner of the active profile should have their phone connected via Bluetooth.
3. Positional settings (seats, mirrors, steering wheel, AHUD) should be part of the Portable Profile. Not having these settings is seen as a big miss with the competition.
4. Customers want the ability to access their Driver Profile from a mobile application.
5. Profile displays should be simple and easy to use so there are no added distractions or complications for the driver.
6. Key Fobs are not a completely reliable way to apply a driver profile to a vehicle. Families with 3 or more users of a vehicle cannot all have a key fob to associate to their profile.
7. After servicing a vehicle, the driver's profile should still be available or easily accessible. A customer does not want to have to reset or restore settings after servicing their vehicle.

## 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.

No Assumptions specified.

# 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)

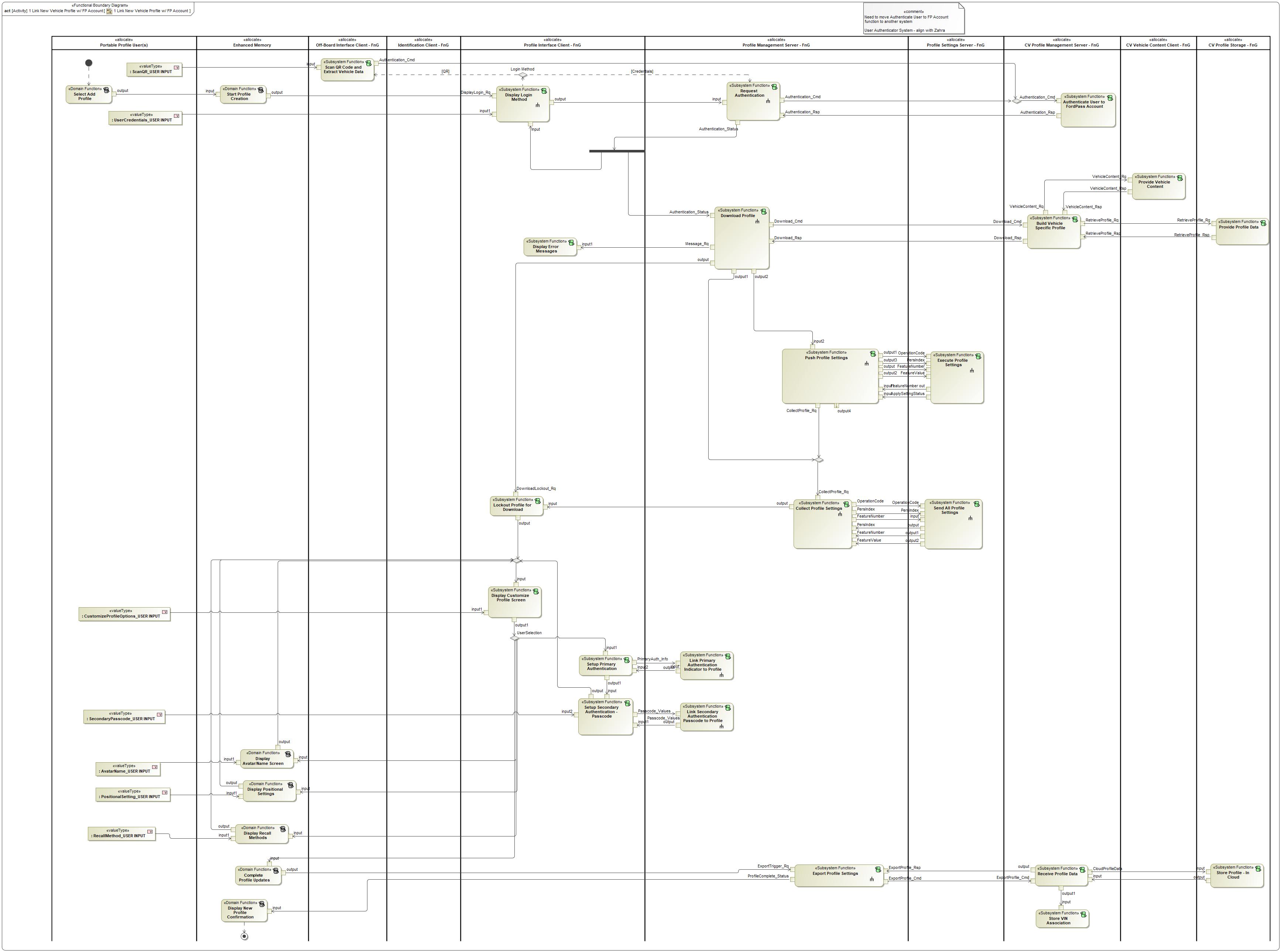


Figure 8: 1 Link New Vehicle Profile w/ FP Account

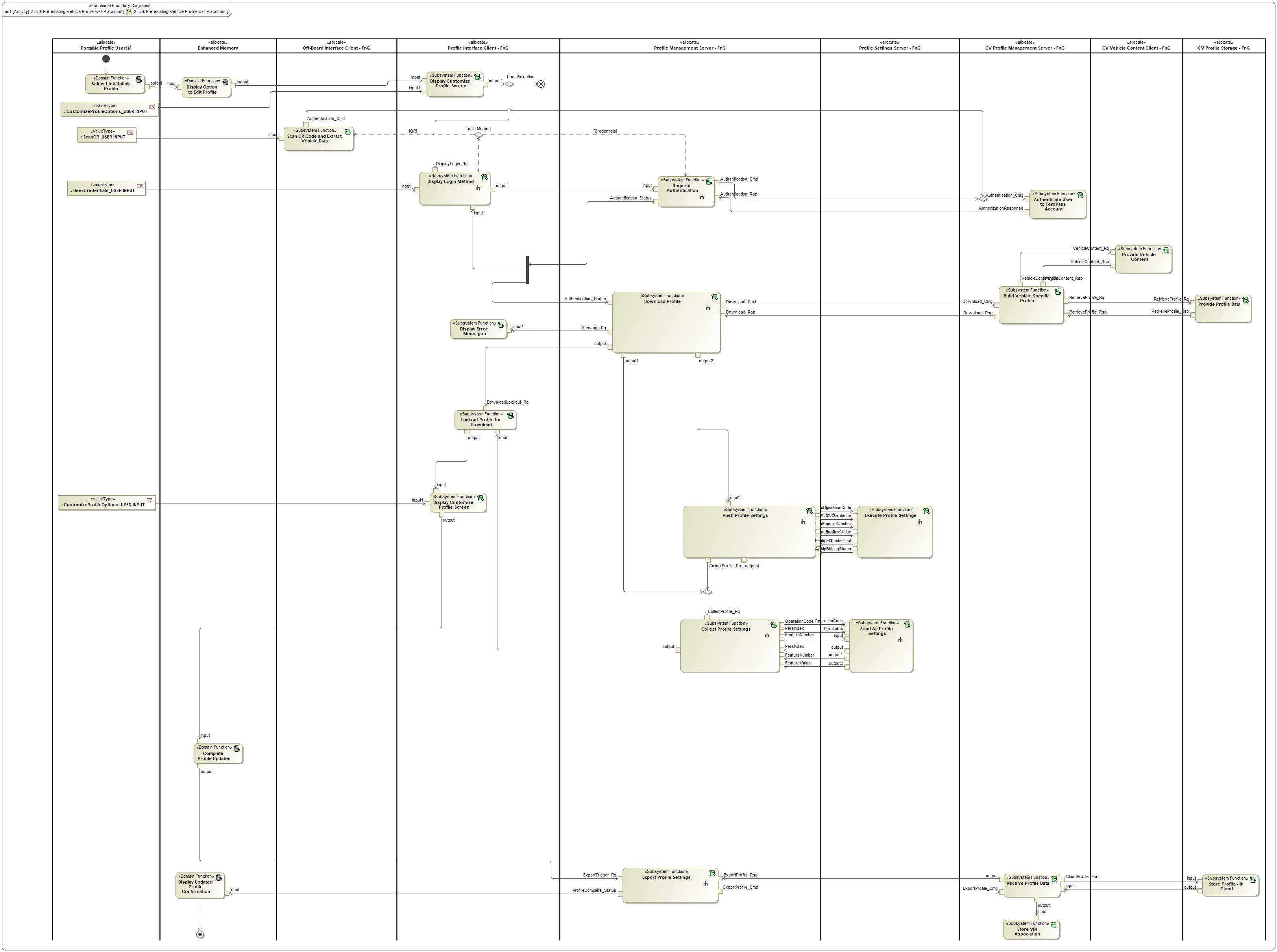


Figure 8: 2 Link Pre-existing Vehicle Profile w/ FP account

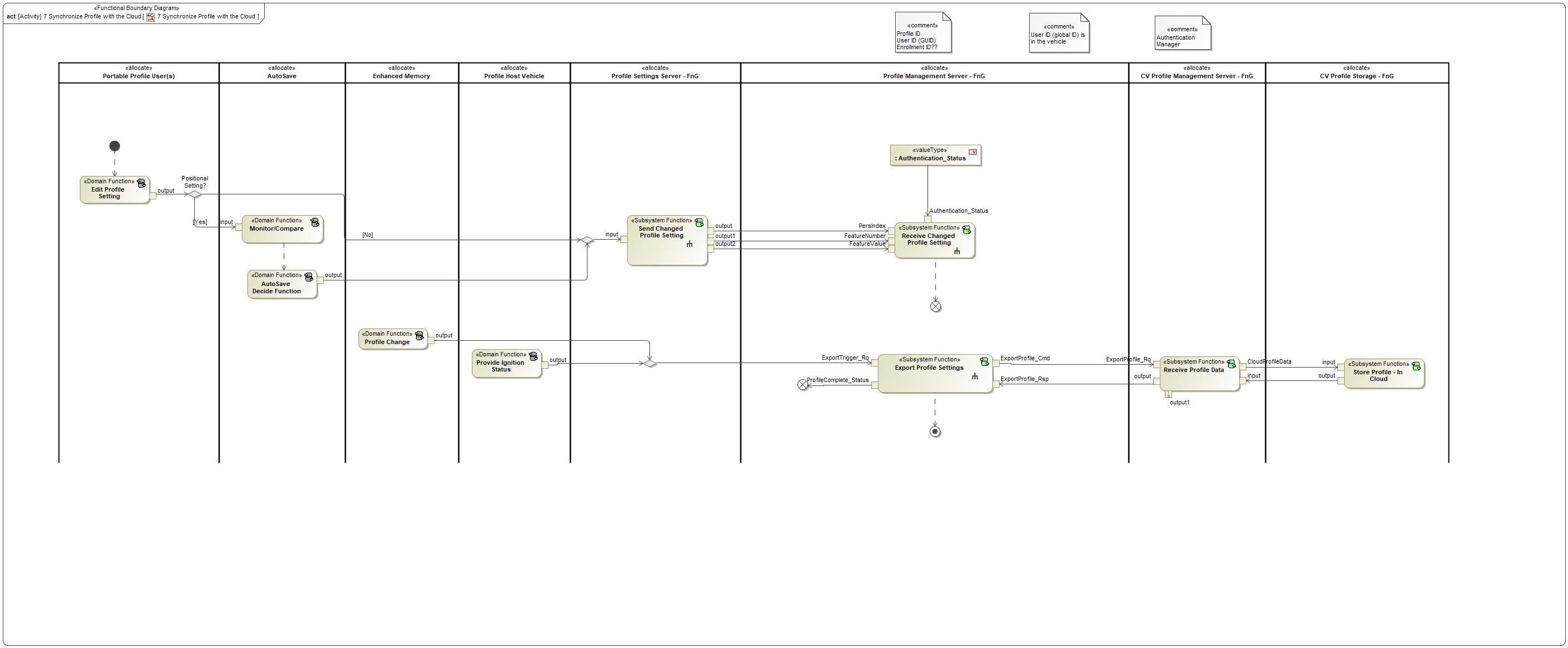


Figure 8: 7 Synchronize Profile with the Cloud

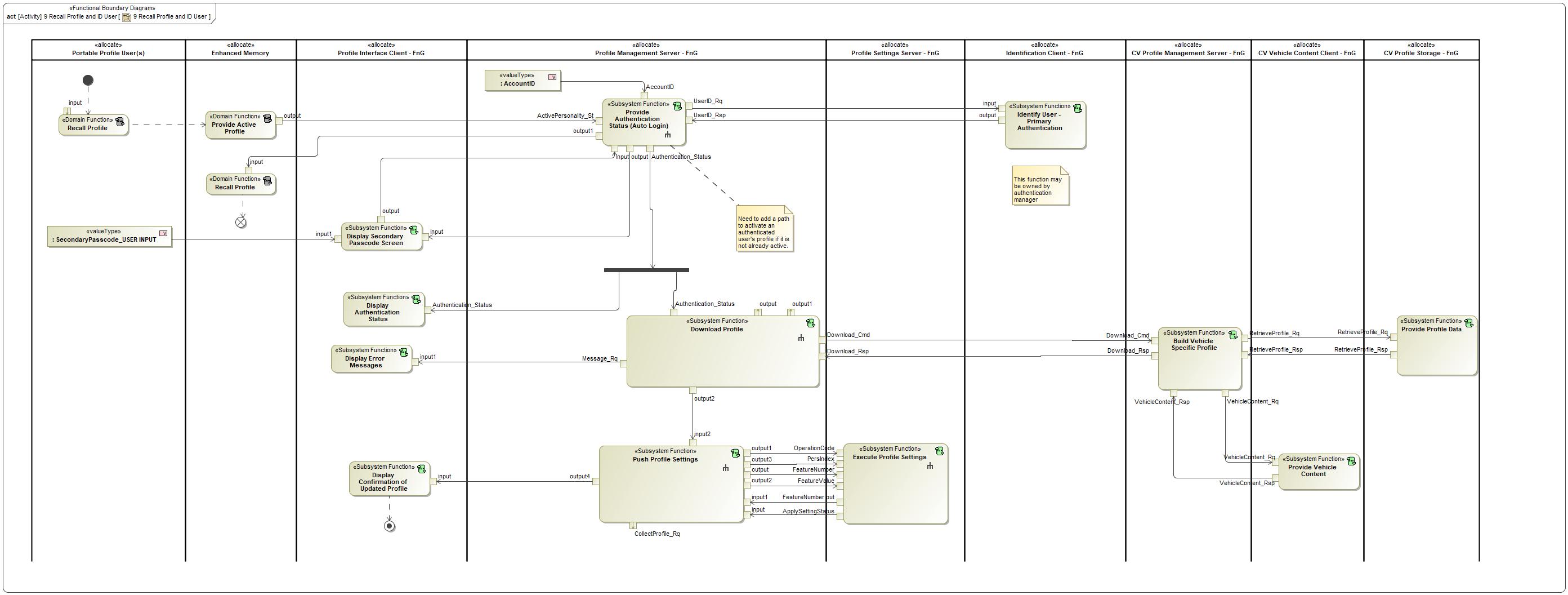


Figure 8: 9 Recall Profile and ID User

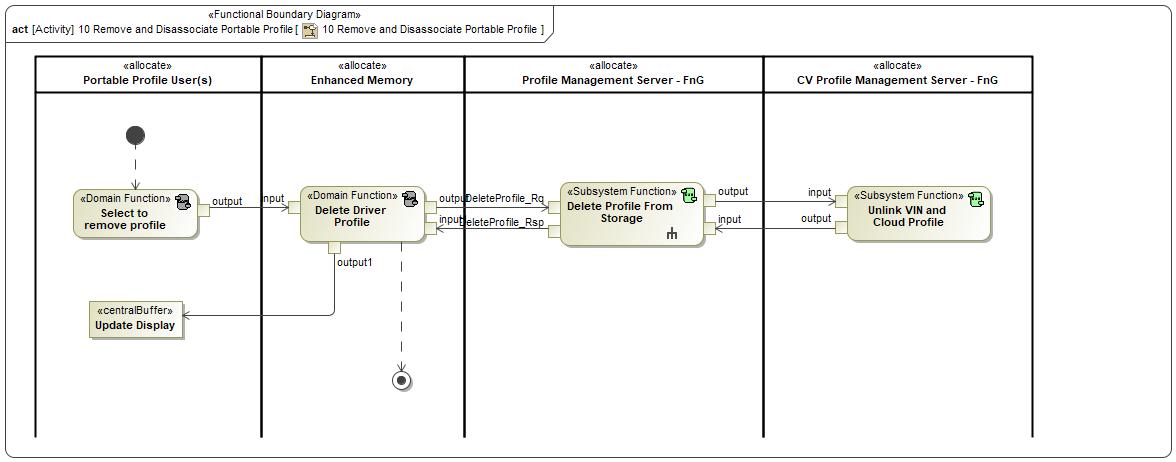


Figure 8: 10 Remove and Disassociate Portable Profile

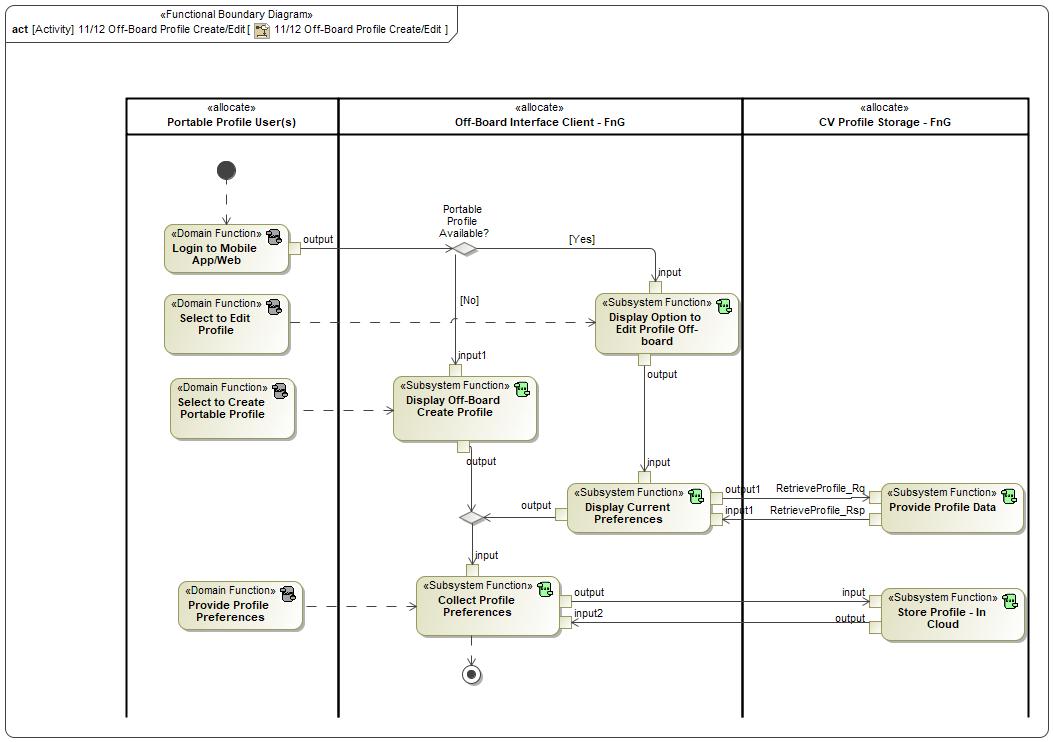


Figure 8: 11/12 Off-Board Profile Create/Edit

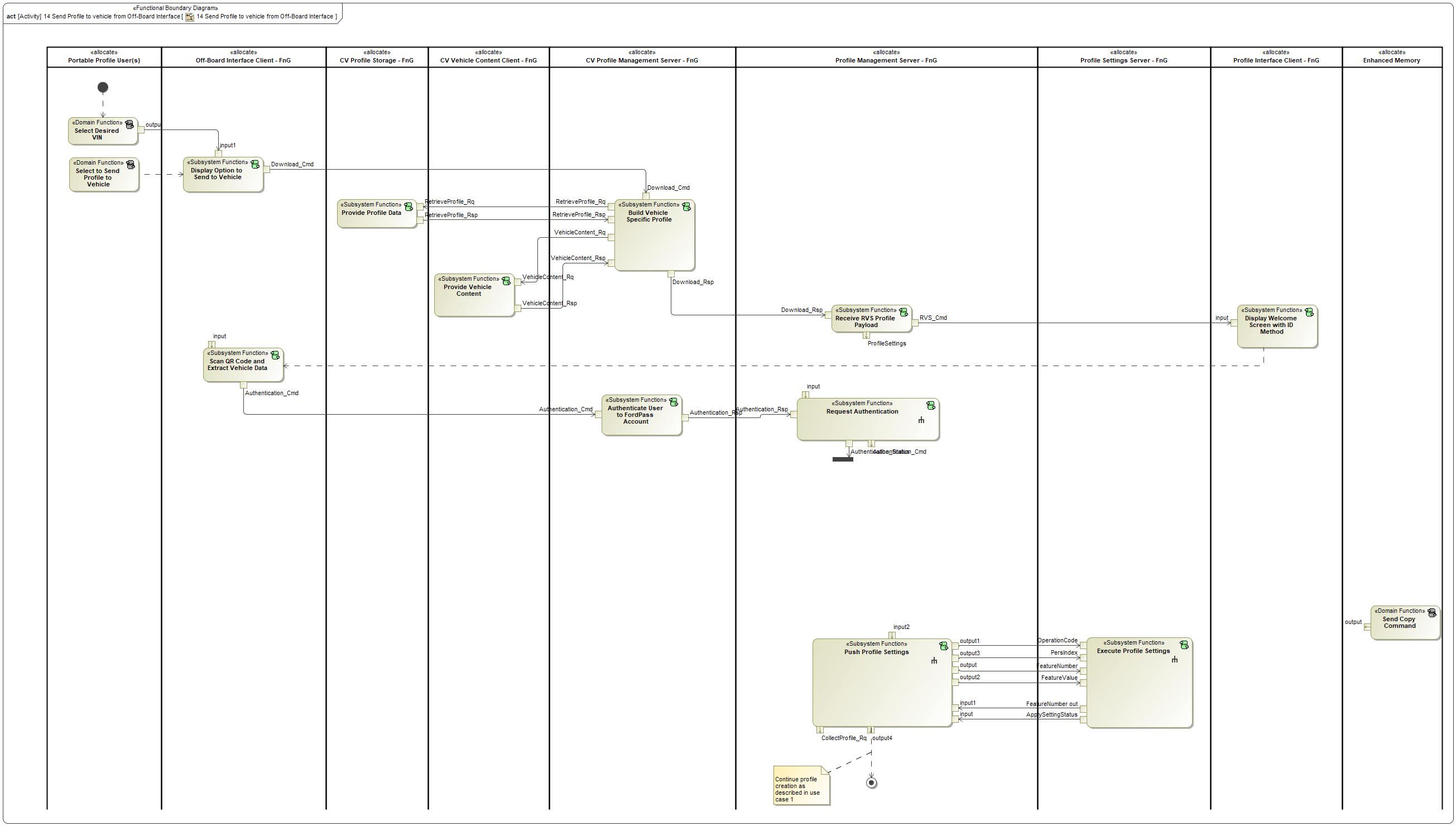


Figure 8: 14 Send Profile to vehicle from Off-Board Interface

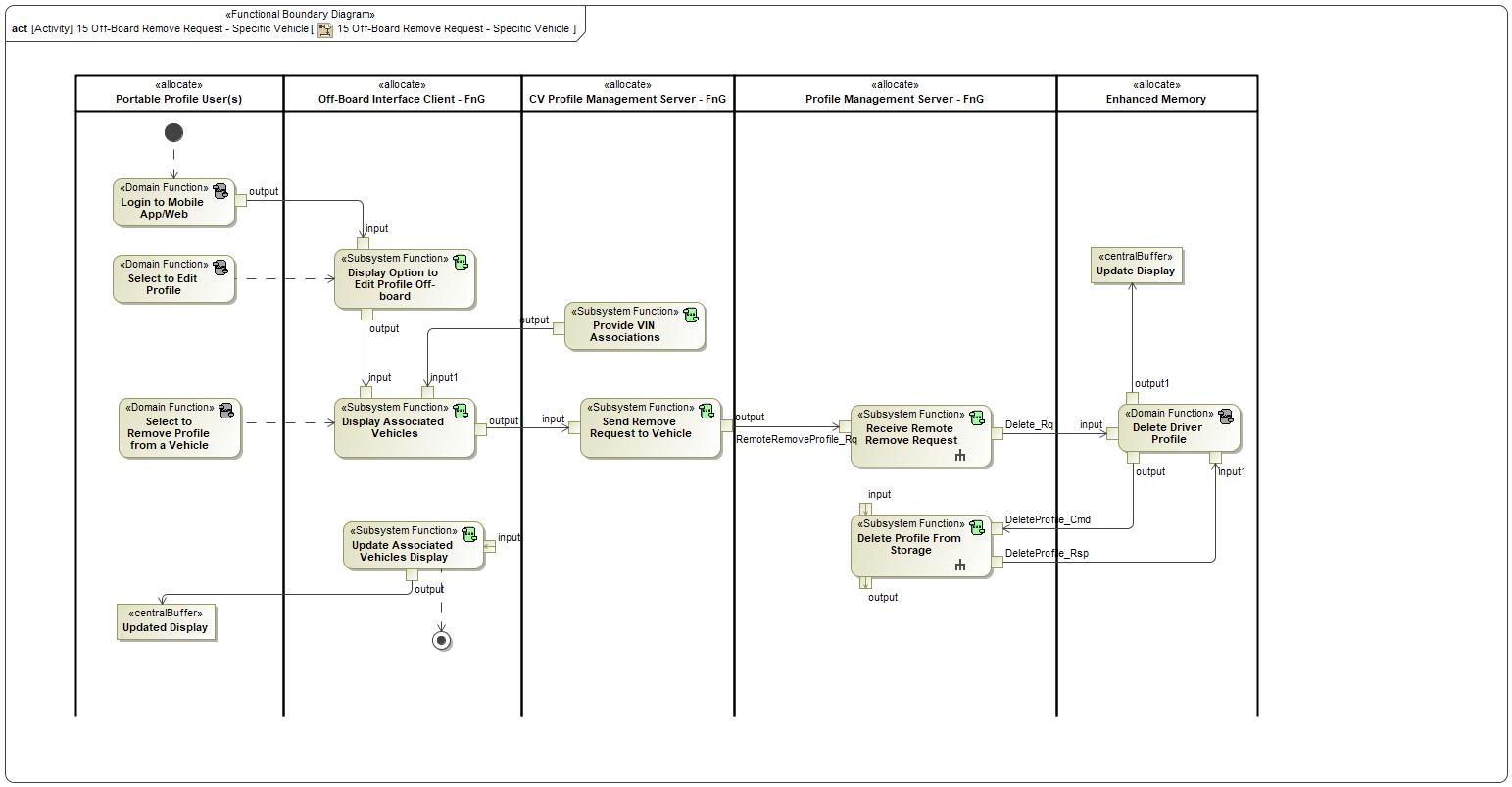


Figure 8: 15 Off-Board Remove Request - Specific Vehicle

### 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 |
| --- | --- | --- |
|  | *(activity)* Receive Profile Data | *(activity)* A cloud function that receives any profile data from the vehicle and passes it along to the cloud profile storage. |
|  | *(activity)* Execute Profile Settings | *(activity)* This is a function of the Profile Setting Servers with the purpose of receiving, storing and applying (for synchronization) 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. |
|  | *(activity)* Display Positional Settings |  |
|  | *(activity)* Lockout Profile for Download | *(activity)* 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. |
|  | *(activity)* Select Add Profile | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Collect Profile Settings | *(activity)* 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. |
|  | *(activity)* Request Authentication | *(activity)* 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. |
|  | *(activity)* Display Avatar/Name Screen |  |
|  | *(activity)* Setup Primary Authentication | *(activity)* This is an HMI function that allows the user to setup their primary authentication method for their profile. |
|  | *(activity)* Complete Profile Updates |  |
|  | *(activity)* Display Recall Methods |  |
|  | *(activity)* Start Profile Creation | *(activity)* DOMAIN Function - Enhanced Memory |
|  | *(activity)* Provide Profile Data | *(activity)* This function stores profile data in the cloud and associates the data to a user profile. |
|  | *(activity)* Link Primary Authentication Indicator to Profile | *(activity)* 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. |
|  | *(activity)* Display Customize Profile Screen | *(activity)* This function displays a screen that provides the ability for the user to customize certain parameters/aspects of their profile. |
|  | *(activity)* Authenticate User to FordPass Account | *(activity)* This function receives the user credentials and authenticates them with the backend system to ensure the credentials are associated to a valid FordPass account. |
|  | *(activity)* Setup Secondary Authentication - Passcode | *(activity)* 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. |
|  | *(activity)* Scan QR Code and Extract Vehicle Data | *(activity)* 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. |
|  | *(activity)* Download Profile | *(activity)* 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. |
|  | *(activity)* Display Error Messages | *(activity)* This function communicates error messages from other functions to the user. |
|  | *(activity)* Display Login Method | *(activity)* 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. |
|  | *(activity)* Push Profile Settings | *(activity)* This function receives the ActiveProfile\_St and StoredVehicleProfileSettings to populate the ApplySettings\_Rq with the correct profile settings and active Pers Index in order to push the profile settings from the Profile Management Server to the Profile Setting Servers. |
|  | *(activity)* Display New Profile Confirmation | *(activity)* DOMAIN Function - Enhanced Memory |
|  | *(activity)* Export Profile Settings | *(activity)* 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. |
|  | *(activity)* Store VIN Association | *(activity)* This is a cloud function that stores all of the VINs that a portable profile is currently available in. |
|  | *(activity)* Link Secondary Authentication Passcode to Profile | *(activity)* 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. |
|  | *(activity)* Store Profile - In Cloud | *(activity)* Cloud storage of the profile settings that is associated with the user's FordPass account. |
|  | *(activity)* Send All Profile Settings | *(activity)* This function sends all profile data stored on the Profile Setting Server to the Profile Management Server when the PPPSettings\_Rq with OperationCode = Query is received. This function is mainly used during initial profile creation in the vehicle. |
|  | *(activity)* Build Vehicle Specific Profile | *(activity)* 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. |
|  | *(activity)* Provide Vehicle Content | *(activity)* This function provides personalized vehicle content for specific vehicles in order to help create a vehicle specific profile. |

Table 17: List of Functions on 1 Link New Vehicle Profile w/ FP Account

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* Download Profile | *(activity)* 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. |
|  | *(activity)* Send All Profile Settings | *(activity)* This function sends all profile data stored on the Profile Setting Server to the Profile Management Server when the PPPSettings\_Rq with OperationCode = Query is received. This function is mainly used during initial profile creation in the vehicle. |
|  | *(activity)* Display Customize Profile Screen | *(activity)* This function displays a screen that provides the ability for the user to customize certain parameters/aspects of their profile. |
|  | *(activity)* Provide Profile Data | *(activity)* This function stores profile data in the cloud and associates the data to a user profile. |
|  | *(activity)* Scan QR Code and Extract Vehicle Data | *(activity)* 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. |
|  | *(activity)* Store VIN Association | *(activity)* This is a cloud function that stores all of the VINs that a portable profile is currently available in. |
|  | *(activity)* Display Error Messages | *(activity)* This function communicates error messages from other functions to the user. |
|  | *(activity)* Display Login Method | *(activity)* 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. |
|  | *(activity)* Display Option to Edit Profile | *(activity)* DOMAIN Function |
|  | *(activity)* Display Customize Profile Screen | *(activity)* This function displays a screen that provides the ability for the user to customize certain parameters/aspects of their profile. |
|  | *(activity)* Provide Vehicle Content | *(activity)* This function provides personalized vehicle content for specific vehicles in order to help create a vehicle specific profile. |
|  | *(activity)* Authenticate User to FordPass Account | *(activity)* This function receives the user credentials and authenticates them with the backend system to ensure the credentials are associated to a valid FordPass account. |
|  | *(activity)* Push Profile Settings | *(activity)* This function receives the ActiveProfile\_St and StoredVehicleProfileSettings to populate the ApplySettings\_Rq with the correct profile settings and active Pers Index in order to push the profile settings from the Profile Management Server to the Profile Setting Servers. |
|  | *(activity)* Export Profile Settings | *(activity)* 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. |
|  | *(activity)* Receive Profile Data | *(activity)* A cloud function that receives any profile data from the vehicle and passes it along to the cloud profile storage. |
|  | *(activity)* Lockout Profile for Download | *(activity)* 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. |
|  | *(activity)* Display Updated Profile Confirmation | *(activity)* DOMAIN Function |
|  | *(activity)* Complete Profile Updates |  |
|  | *(activity)* Select Link/Unlink Profile | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Collect Profile Settings | *(activity)* 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. |
|  | *(activity)* Execute Profile Settings | *(activity)* This is a function of the Profile Setting Servers with the purpose of receiving, storing and applying (for synchronization) 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. |
|  | *(activity)* Build Vehicle Specific Profile | *(activity)* 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. |
|  | *(activity)* Request Authentication | *(activity)* 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. |
|  | *(activity)* Store Profile - In Cloud | *(activity)* Cloud storage of the profile settings that is associated with the user's FordPass account. |

Table 17: List of Functions on 2 Link Pre-existing Vehicle Profile w/ FP account

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* AutoSave Decide Function | *(activity)* DOMAIN Function |
|  | *(activity)* Edit Profile Setting | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Store Profile - In Cloud | *(activity)* Cloud storage of the profile settings that is associated with the user's FordPass account. |
|  | *(activity)* Export Profile Settings | *(activity)* 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. |
|  | *(activity)* Receive Profile Data | *(activity)* A cloud function that receives any profile data from the vehicle and passes it along to the cloud profile storage. |
|  | *(activity)* Send Changed Profile Setting | *(activity)* A function of the Profile Setting Servers that sends changed profile setting's Feature Number and new Feature Value to the Profile Management Server when the customer makes a change to an active profile setting. |
|  | *(activity)* Receive Changed Profile Setting | *(activity)* 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. |
|  | *(activity)* Profile Change | *(activity)* DOMAIN Function - Enhanced Memory |
|  | *(activity)* Monitor/Compare | *(activity)* DOMAIN Function |
|  | *(activity)* Provide Ignition Status | *(activity)* DOMAIN Function - Host Vehicle |

Table 17: List of Functions on 7 Synchronize Profile with the Cloud

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* Display Confirmation of Updated Profile | *(activity)* An HMI function that is used to communicate the status of an updated profile to the customer. |
|  | *(activity)* Push Profile Settings | *(activity)* This function receives the ActiveProfile\_St and StoredVehicleProfileSettings to populate the ApplySettings\_Rq with the correct profile settings and active Pers Index in order to push the profile settings from the Profile Management Server to the Profile Setting Servers. |
|  | *(activity)* Identify User - Primary Authentication | *(activity)* This function identifies the user that has an active profile in the host vehicle. |
|  | *(activity)* Display Authentication Status | *(activity)* The function is allocated to the Profile Interface Client which is inside the vehicle. This is an HMI function that will display the authentication status to a customer that is attempting to login to the vehicle. |
|  | *(activity)* Provide Vehicle Content | *(activity)* This function provides personalized vehicle content for specific vehicles in order to help create a vehicle specific profile. |
|  | *(activity)* Recall Profile | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Provide Authentication Status (Auto Login) | *(activity)* 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. |
|  | *(activity)* Provide Active Profile | *(activity)* DOMAIN Function - Enhanced Memory |
|  | *(activity)* Display Error Messages | *(activity)* This function communicates error messages from other functions to the user. |
|  | *(activity)* Download Profile | *(activity)* 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. |
|  | *(activity)* Provide Profile Data | *(activity)* This function stores profile data in the cloud and associates the data to a user profile. |
|  | *(activity)* Display Secondary Passcode Screen | *(activity)* This function displays a screen that allows the user to enter their passcode. |
|  | *(activity)* Build Vehicle Specific Profile | *(activity)* 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. |
|  | *(activity)* Recall Profile | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Execute Profile Settings | *(activity)* This is a function of the Profile Setting Servers with the purpose of receiving, storing and applying (for synchronization) 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. |

Table 17: List of Functions on 9 Recall Profile and ID User

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* Unlink VIN and Cloud Profile | *(activity)* 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. |
|  | *(activity)* Delete Driver Profile | *(activity)* DOMAIN Function - Enhanced Memory |
|  | *(activity)* Delete Profile From Storage | *(activity)* 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. |
|  | *(activity)* Select to remove profile | *(activity)* DOMAIN Function - Profile User |

Table 17: List of Functions on 10 Remove and Disassociate Portable Profile

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* Login to Mobile App/Web | *(activity)* DOMAIN Function |
|  | *(activity)* Display Current Preferences | *(activity)* An Off-Board Interface Client function that shows the customer their current preferences for their portable profile settings. The function collects these settings from the CV Profile Storage. |
|  | *(activity)* Display Off-Board Create Profile | *(activity)* This function provides the user an interface to create or edit a portable profile from outside a vehicle. |
|  | *(activity)* Store Profile - In Cloud | *(activity)* Cloud storage of the profile settings that is associated with the user's FordPass account. |
|  | *(activity)* Select to Create Portable Profile | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Display Option to Edit Profile Off-board | *(activity)* Off-Board Interface Client function that provides the user the option to edit a pre-existing portable profile. |
|  | *(activity)* Collect Profile Preferences | *(activity)* 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. |
|  | *(activity)* Provide Profile Data | *(activity)* This function stores profile data in the cloud and associates the data to a user profile. |
|  | *(activity)* Provide Profile Preferences | *(activity)* DOMAIN Functions |
|  | *(activity)* Select to Edit Profile | *(activity)* DOMAIN Function |

Table 17: List of Functions on 11/12 Off-Board Profile Create/Edit

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* Push Profile Settings | *(activity)* This function receives the ActiveProfile\_St and StoredVehicleProfileSettings to populate the ApplySettings\_Rq with the correct profile settings and active Pers Index in order to push the profile settings from the Profile Management Server to the Profile Setting Servers. |
|  | *(activity)* Select Desired VIN | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Select to Send Profile to Vehicle | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Authenticate User to FordPass Account | *(activity)* This function receives the user credentials and authenticates them with the backend system to ensure the credentials are associated to a valid FordPass account. |
|  | *(activity)* Execute Profile Settings | *(activity)* This is a function of the Profile Setting Servers with the purpose of receiving, storing and applying (for synchronization) 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. |
|  | *(activity)* Provide Vehicle Content | *(activity)* This function provides personalized vehicle content for specific vehicles in order to help create a vehicle specific profile. |
|  | *(activity)* Request Authentication | *(activity)* 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. |
|  | *(activity)* Display Welcome Screen with ID Method | *(activity)* This function is used during the Remote Vehicle Setup (RVS) scenario and is allocated to the Profile Interface Client. This function displays a welcome screen to the customer and provides a method for the customer to provide positive ID. |
|  | *(activity)* Send Copy Command | *(activity)* DOMAIN Function |
|  | *(activity)* Display Option to Send to Vehicle | *(activity)* This function provides the user interface that allows the user to push their portable profile to a specific vehicle. |
|  | *(activity)* Build Vehicle Specific Profile | *(activity)* 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. |
|  | *(activity)* Provide Profile Data | *(activity)* This function stores profile data in the cloud and associates the data to a user profile. |
|  | *(activity)* Scan QR Code and Extract Vehicle Data | *(activity)* 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. |
|  | *(activity)* Receive RVS Profile Payload | *(activity)* A function that is similar to the "Download Profile" function that receives the Remote Vehicle Setup (RVS) payload and temporarily caches the profile until there is positive ID of the user. |

Table 17: List of Functions on 14 Send Profile to vehicle from Off-Board Interface

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* Display Associated Vehicles | *(activity)* 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. |
|  | *(activity)* Select to Remove Profile from a Vehicle | *(activity)* DOMAIN Function - Profile User |
|  | *(activity)* Update Associated Vehicles Display |  |
|  | *(activity)* Delete Driver Profile | *(activity)* DOMAIN Function - Enhanced Memory |
|  | *(activity)* Delete Profile From Storage | *(activity)* 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. |
|  | *(activity)* Select to Edit Profile | *(activity)* DOMAIN Function |
|  | *(activity)* Receive Remote Remove Request | *(activity)* 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. |
|  | *(activity)* Send Remove Request to Vehicle | *(activity)* 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. |
|  | *(activity)* Display Option to Edit Profile Off-board | *(activity)* Off-Board Interface Client function that provides the user the option to edit a pre-existing portable profile. |
|  | *(activity)* Provide VIN Associations | *(activity)* This is a cloud function that provides the information of all of the VIN's that are associated to the user's portable profile. |
|  | *(activity)* Login to Mobile App/Web | *(activity)* DOMAIN Function |

Table 17: List of Functions on 15 Off-Board Remove Request - Specific Vehicle

### Signal List

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

## Physical Architecture

### E/E Architecture

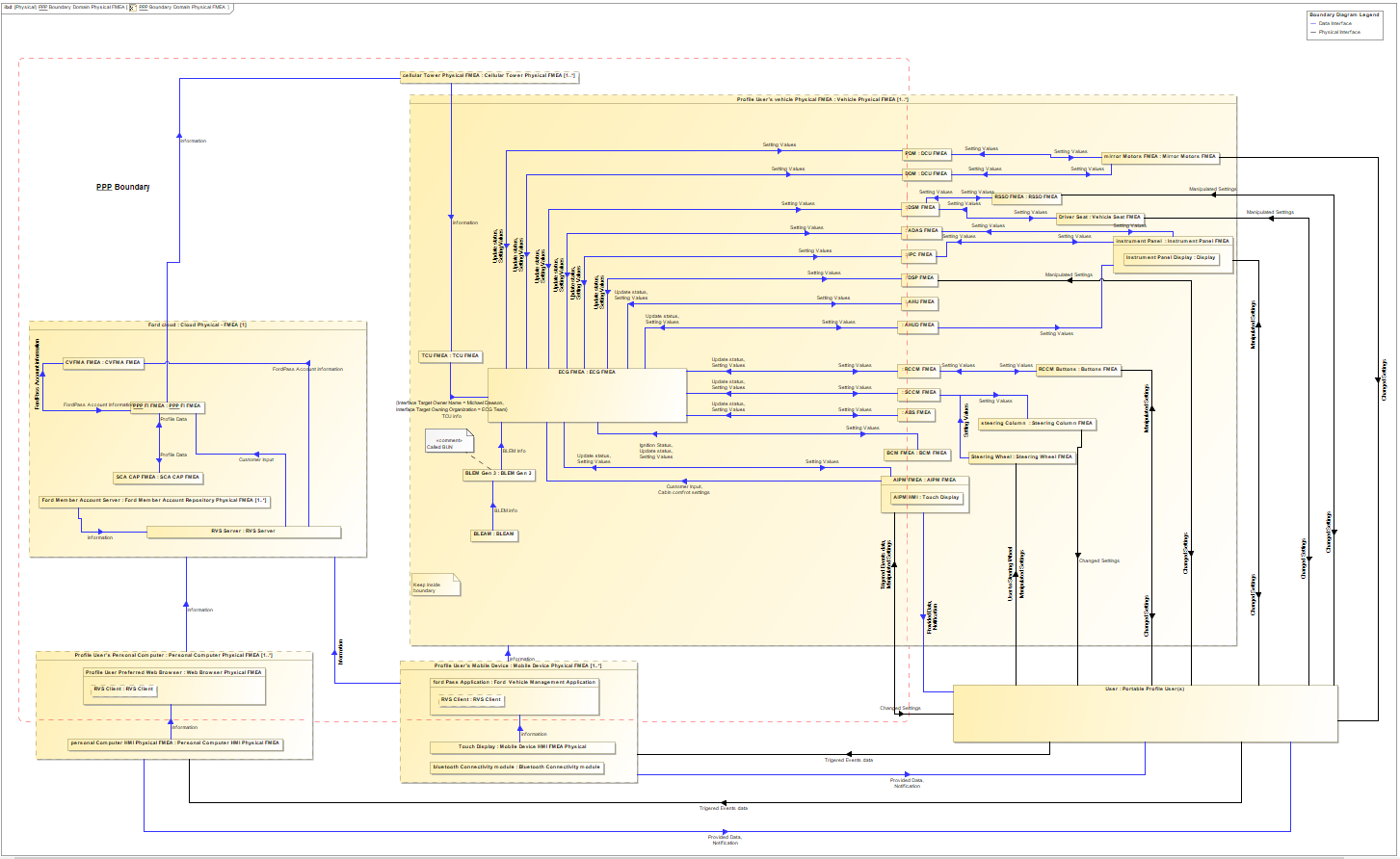


Figure 3‑4: PPP Physical Architecture

#### 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).

No Variants Defined

#### 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** |

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** |
| <Give a Connection Name>  *#Hint:*   * *For ‘Connection Type’ “Network” check with Netcom for naming conventions for busses/networks* * *For other ‘Connection Types’  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 VSEM entry. Refer to the “Event Notification Signal” example below”. | Choose an item. | Choose an item. | <Provide a brief description> | <Give a list of relevant messages >  *#Hint:*  *The message name should be linked.*  *E.g.*   * *for CAN signals to the VSEM CMDB (refer e.g. to* [*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)*).* * *for SOA Service API data elements to the SOA Service API or MQTT/FTCP message in the corresponding central repository (e.g.* [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog)*)*   *If a message is not yet managed in VSEM or any other central repository, add a link to the section “Messages” in the Data Dictionary. In the subsections of that data dictionary chapter you may add a definition of your message.* | <Give a list of relevant nodes> |
| e..g. HS-CAN4 | Network | CAN (High Speed) | Infotainment High Speed CAN bus | … | … |
| e..g. [CELLULAR TCUB WIFI](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=WZe1wsPXx3NrTDAAAAAAAAAAAAA&servername=Production_Server%5e) | RF-Digital | WiFi (FTCP) |  | … | … |
| e.g. [CR167·CTRL MOD. - RCM # EVENT NOTIFICATION SIGNAL 1](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=LjXtx$M9x3NrTDAAAAAAAAAAAAA&servername=Production_Server) | PMW | n/a | Event Notification Signal | n/a |  |
|  |  |  |  |  |  |

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/)

## 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.

|  |  |  |
| --- | --- | --- |
| **Deployment Variant Name** | Variant Description | Variant Condition (optional) |
| CAN | Normal CAN messages between ECU’s |  |
| CAN Transport Protocol | Dedicated TP Channel between ECU’s |  |
| Ethernet | Ethernet Communication between supporting ECU’s |  |
|  |  |  |

#### Deployment “Variant 1”

***#Classification:*** *Optional*

***#Hint:*** *Add a deployment diagram (e.g. a SysML Activity Diagram where the actions represent the Technology Functions and the swimlanes represent the components) and some explanatory text about the variant to this section. The naming of the Technology Functions should make clear, what Logical Function it had been derived from (e.g. VehicleSpeedCalculation(Wheelbased)\_ABS)*

This deployment variant … <add some explanatory text here>



Table 3‑4: Sample Deployment Diagram

### 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 |
| --- | --- | --- |
|
| ECM | Execute Profile Settings - ECM | * Execute Profile Settings |
| Send All Profile Settings - ECM | * Send All Profile Settings |
| Send Changed Profile Settings - ECM | * Send Changed Profile Setting |
| ADAS FMEA | 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 FMEA | 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 |
| AIPM FMEA | 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 FMEA | 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 FMEA | 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 FMEA | 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 FMEA | 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 FMEA | 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 FMEA | 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 FMEA | 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 FMEA | 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 FMEA | 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 FMEA | 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: “System Startup / Shutdown”

***#Hint:*** *This chapter shall define specific scenarios / sequences (e.g. power-up initialization, ignition after-run, NM sleep / wake up, MicroHybrid start / stop inhibition.*

### Scenario: “Normal Operation”

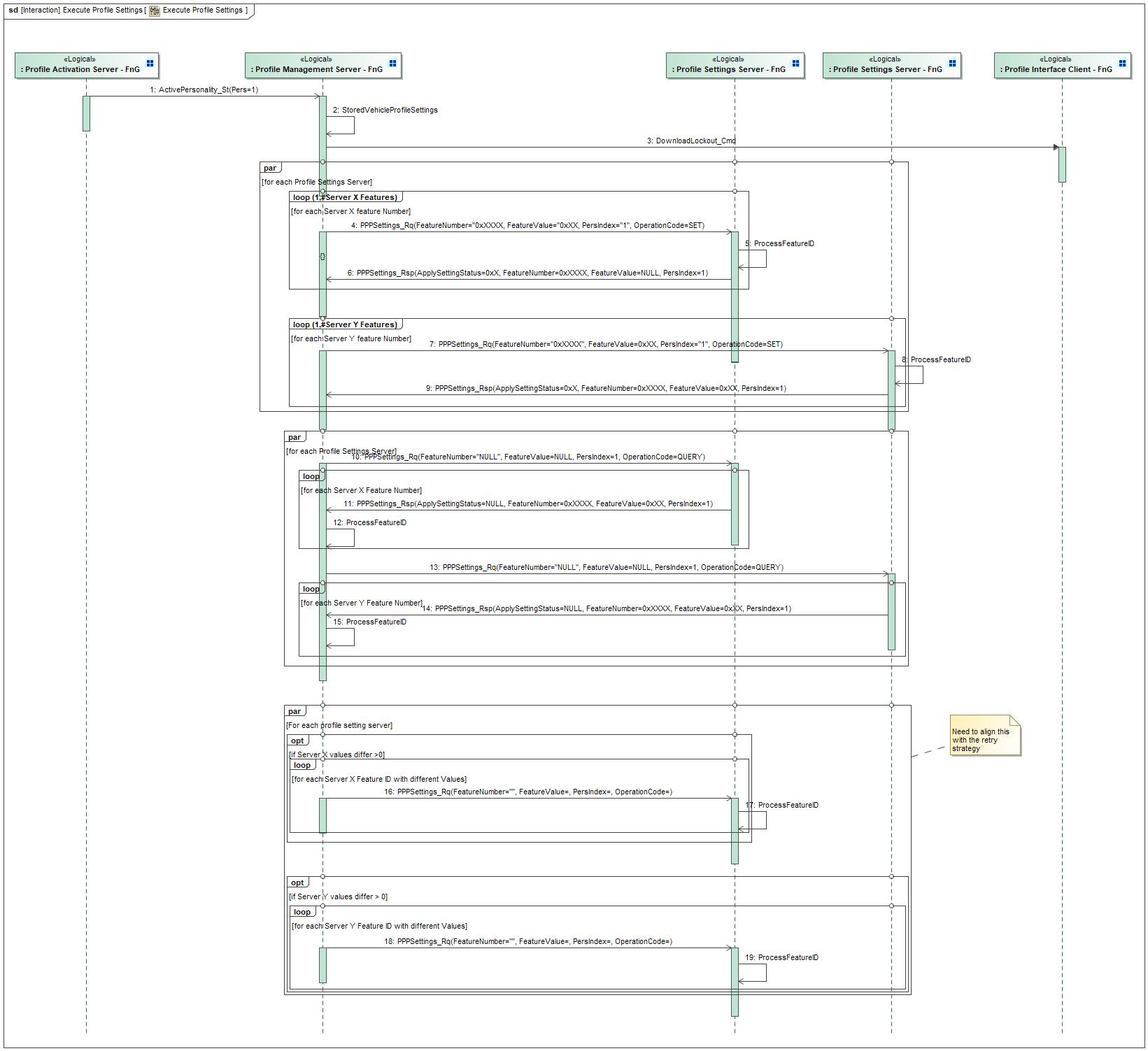


Figure 8: Execute Profile Settings

See Section 4.2 for State Machine Diagram illustrating operation.

## 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*

## Requirements on Components

### AHUD FMEA

AHUD FMEA

#### 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -602524511.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -602524511.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### DCU FMEA

DCU FMEA

#### Technology Function -602524511.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -602524511.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -539531057.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### ADAS FMEA

ADAS FMEA

#### Technology Function -539531057.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -539531057.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -539531057.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### IPC FMEA

IPC FMEA

#### Technology Function -254316882.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -254316882.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -254316882.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### ABS FMEA

ABS FMEA

#### Technology Function -254316882.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -2086633416.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -2086633416.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### BCM FMEA

BCM FMEA

#### Technology Function -2086633416.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -2086633416.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1710163615.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### AHU FMEA

AHU FMEA

#### Technology Function -1710163615.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1710163615.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 2074241822.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### ECG FMEA

ECG FMEA

#### Technology Function 2074241822.jpg **Collect Profile Settings - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Collect Profile Settings - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Collect Profile Settings - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 2074241822.jpg **Delete Profile From Storage - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Delete Profile From Storage - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Delete Profile From Storage - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 2074241822.jpg **Download Profile - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Download Profile - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Download Profile - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1964147481.jpg **Export Profile Settings - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Export Profile Settings - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Export Profile Settings - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1964147481.jpg **Link Primary Authentication Indicator to Profile - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Link Primary Authentication Indicator to Profile - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Link Primary Authentication Indicator to Profile - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1964147481.jpg **Link Secondary Authentication Passcode to Profile - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Link Secondary Authentication Passcode to Profile - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Link Secondary Authentication Passcode to Profile - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1964147481.jpg **Provide Authentication Status (Auto Login) - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Provide Authentication Status (Auto Login) - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Provide Authentication Status (Auto Login) - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 2120196228.jpg **Push Profile Settings - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Push Profile Settings - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Push Profile Settings - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 2120196228.jpg **Receive Remote Remove Request - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Receive Remote Remove Request - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Receive Remote Remove Request - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 2120196228.jpg **Receive Changed Profile Settings - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Receive Changed Profile Settings - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Receive Changed Profile Settings - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 2120196228.jpg **Receive RVS Profile Payload - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Receive RVS Profile Payload - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Receive RVS Profile Payload - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1916201523.jpg **Request Authentication - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Request Authentication - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Request Authentication - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1916201523.jpg **Restore Profile Settings - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Restore Profile Settings - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Restore Profile Settings - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1916201523.jpg **Unlink FP Account from Vehicle Profile - ECG**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Unlink FP Account from Vehicle Profile - ECG

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Unlink FP Account from Vehicle Profile - ECG

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### AIPM FMEA

AIPM FMEA

#### Technology Function 1916201523.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1057661821.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1057661821.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -361410294.jpg **Display Authentication Status - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Display Authentication Status - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Display Authentication Status - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -361410294.jpg **Display Confirmation of Updated Profile - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Display Confirmation of Updated Profile - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Display Confirmation of Updated Profile - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -361410294.jpg **Display Customize Profile Screen - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Display Customize Profile Screen - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Display Customize Profile Screen - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1494991138.jpg **Display Error Messages - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Display Error Messages - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Display Error Messages - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1494991138.jpg **Display Login Method - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Display Login Method - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Display Login Method - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1494991138.jpg **Display Option to Unlink FP Account - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Display Option to Unlink FP Account - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Display Option to Unlink FP Account - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1494991138.jpg **Display Secondary Passcode Screen - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Display Secondary Passcode Screen - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Display Secondary Passcode Screen - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1494991138.jpg **Display Welcome Screen with ID Method - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Display Welcome Screen with ID Method - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Display Welcome Screen with ID Method - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1494991138.jpg **Lockout Profile for Download - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Lockout Profile for Download - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Lockout Profile for Download - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1494991138.jpg **Setup Primary Authentication - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Setup Primary Authentication - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Setup Primary Authentication - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -1494991138.jpg **Setup Secondary Authentication Passcode - 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Setup Secondary Authentication Passcode - APIM

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Setup Secondary Authentication Passcode - 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### DSP FMEA

DSP FMEA

#### Technology Function -315134170.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -315134170.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -315134170.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### DSM FMEA

DSM FMEA

#### Technology Function -519359658.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -519359658.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -519359658.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### MCSM

MCSM

#### Technology Function -384924636.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -384924636.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -384924636.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### ECM

ECM

#### Technology Function -384924636.jpg **Execute Profile Settings - ECM**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -934034915.jpg **Send All Profile Settings - ECM**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -934034915.jpg **Send Changed Profile Settings - ECM**

***#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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### 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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### RCCM FMEA

RCCM FMEA

#### Technology Function -934034915.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function -934034915.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1266185694.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

### SCCM FMEA

SCCM FMEA

#### Technology Function 1266185694.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1266185694.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

#### Technology Function 1266185694.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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

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** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

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.*

## 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*](https://developers.google.com/protocol-buffers/docs/proto)

### HW I/Os

**#Hint**: 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 | Current version of MagicDraw Implementation Spec Export Template is a WIP and does not properly export all Content |  | Open |  |
| 2 | No specific implementation requirements for each ECU. Functional Spec contains the level of detail needed. |  | Open |  |
| 3 | CAN signals not yet available and will be added for the next rev of the Implementation Spec. |  |  |  |
| 4 |  |  |  |  |

Table 6‑1: Open Concerns

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Description | Approved by | Responsible |
| A |  | Initial version |  | Jbaden1 |
|  |  |  |  |  |

## 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)

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Description** | **Details** |
| **Create\_PP = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Phone Selection for Protection** | User deciding which phone to user for profile protection. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPPSettings\_Rsp** | Success or Failure response that the Profile Settings Server provides to the Profile Management Server after profile settings are applied. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Unlink Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Customer Input** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Connection Trigger** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **TranslatedProfileData** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle ON/OFF input** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ProfilePayload\_Rq** | This signal will request the CV Preference Management System to collect user profile in order to send it to the vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall | Import | Create** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Driver Door Status** | Vehicle tells PPP when the driver door goes from closed to ajar. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **SeatData** | All of the information needed to perform anthropometric translation. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Change\_Setting\_Rq** | User's input to changes to their Portable Profile settings. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Mobile Device Authentication** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Preference Status** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **TCU info** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **LinkProfile\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Profile\_Creation\_Confirmation = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Personal Profile Status** | Profile active/inactive status. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **DisplayWelcome\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Search Response** | Mobile devices in vicinity of host vehicle responding to a search request. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Available Profiles** | Available profiles as a result of an identification search | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Cloud Authentication** | PPP requesting the cloud to authenticate a user's credentials. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RetrieveData\_Rq** | This signal is a request sent to the CV Profile Storage to retrieve a user's profile data from storage. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **User Detected** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **User Credentials** | User credentials being sent from PPP to cloud to be verified. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Setting Categorization** | Profile Attribute deciding how to categorize profile data. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RemoveProfile\_Cmd** | This signal is a sent from the cloud to a specific vehicle that requests the removal of a user's portable profile from the specified vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPP Creation Confirmation** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PositiveID\_enabled** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Imported Profile** | Notify Enhanced Memory that there is a new imported profile in the vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Profile Data** | Target 5000 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Identification Search** | Portable Profiles searching for available profiles. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ID Request** | A request to collect credentials from the user to establish positive identification. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **AuthStatusIndicator** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Auth\_Status = Profile\_Exists** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPP\_Set\_Settings2** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Notification** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ActivePersonality\_St** | Indicates the active vehicle profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import Other Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Feature Enable Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **DownloadSuccess** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Synch\_Flag** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Disable Feature Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **SynchEnable\_St** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Remote Remove Request** | Remove profile request coming from outside of the vehicle will be sent from the Profile Management Server to Enhanced Memory so that Enhanced Memory can perform a delete profile operation. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Setting Selections** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **FordPass Account information** | Target 5000 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **VINTargetedRemove\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Link Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPP\_Set\_Settings** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key On and Profile PP Enabled** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Display\_Cred\_Error\_Timeout = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Data** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **BeginPrimaryAuth** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ButtonPairing\_St** | Status to indicate the memory seat button that is paired to a profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Passcode\_Rq** | Signal is sent to the Profile Interface client when the primary authentication method failed to identify the user. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **BLEM info** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Changes to Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PushSettings\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **CredentialValidation\_St** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Ignition Status** | Vehicle tells PPP what the ignition status of the vehicle is. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Changed Settings** | Target 2 Newtons | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPP\_Set\_Settings1** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **NewProfileSettings** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RawProfileData** | This signal contains a package of profile data that is not in vehicle-readable format. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ReturnProfileSetting** | The current value of a profile setting for the active portable profile. This data is sent to the Profile Management Server to update the user's portable profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Cabin comfrot settings** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Skip\_Portability = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **QueryProfile\_Rq** | A request sent from Profile Management Server to the Profile Setting Servers to collect all profile data when a new profile is being created. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Setting Values** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Login HMI** | HMI to allow the user to input their user credentials. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Remove Request** | Remove request coming from Enhanced Memory. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Remove Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **AssociatedVIN** | Signal provides all of the VINs that are associated to the user's portable profile to the Off-Board Interface Client. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Create Other Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ApplyName** | Applies downloaded profile name during in vehicle profile creation. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Trigered Events data** | Target 2 Newtons | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Auth\_Status = Valid AND Export\_Confirmation = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Connection Status** | Sync Bluetooth tells PPP which mobile device is actively paired to the vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import HMI** | HMI to allow the user to import a profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Capabilities** | Vehicle providing information to PPP about what personalized features it has. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Export\_Rq** | A request from the Profile Interface Client to the Profile Management Server based on the user's input. This signal determines if profile data should be exported to the cloud when a profile is linked to a FordPass account. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Provided Data** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import\_Rq** | A request from the Profile Interface Client to the Profile Management Server based on the user's input. This signal determines if profile data should be imported into the vehicle when a profile is linked to a FordPass account. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Positional Data Points** | Autosave notifying PPP which positional data point should be part of the profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **WelcomeScreen** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall Event Confirmation** | Enhanced Memory telling PPP that there has been a Profile Recall Event. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RS Settings** | Pre-determined group of settings that Rocket Setup will use to populate a newly created Portable Profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Portable Profile Options** | The feature providing the user with profile options for their settings. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PortableProflile\_Available** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PromptPosID** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Autosave Inhibit** | Portable Profiles inhibiting the auto save functionality due to profile protection status. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **VersionUpdate\_Rq** | A request sent from a vehicle to the cloud to check if there is an updated version of a user's portable profile stored in the cloud. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import\_Request = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ID Response** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Guest Profile Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key On** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Disassociate Profile from VIN** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **information** | Target 5000 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Auth\_Status = Invalid** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Update status** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPPSettings\_Rq** | Pushing settings from Profile Management Server to all Profile Setting Servers that contain profile settings | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall Other Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ApplyPhoto** | Applies downloaded profile photo during in vehicle profile creation. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RemoveVIN\_Rq** | Request to remove a VIN association from a user's portable profile storage. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Feature\_St** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall | Import | Create & Synch Enabled** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RemoveProfile\_Rsp** | Response to the RemoveProfile\_Cmd that determines if the profile removal was a success or failure. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Ignition Status** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ProfileCredential\_Value** | Credential values that are collected by Profile Interface Client and sent to the Profile Management Server. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Authorization Status** | User is authenticated by the cloud account manager. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PositiveID\_Rq** | A request to the Identification Client to attempt to establish positive ID for the active profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RemoteRemove\_Rq** | A request sent from the Off-Board Interface Client to the CV Preference Management Server when the user requests to remove their portable profile from a specific vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Information** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Manipulated Settings** | Target 2 Newtons | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ProcessFeatureID** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Create Profile Request** | Request to create a profile that is send from Enhanced Memory to the Profile Management Server of PPP. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Create Portable Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RVSPayload\_St** | An internal vehicle flag to determine if an RVS profile has been delivered to the vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key Off** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Available Profile Settings** | Defined list of profile parameters defined by the Profile Attribute. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PromptUser\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Login Credentials** | User providing login credentials to the system. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PositiveID\_St** | The positive identification status of a user that is in the host vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |

### 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)

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Description** | **Details** |
| **Create\_PP = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Phone Selection for Protection** | User deciding which phone to user for profile protection. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPPSettings\_Rsp** | Success or Failure response that the Profile Settings Server provides to the Profile Management Server after profile settings are applied. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Unlink Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Customer Input** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Connection Trigger** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **TranslatedProfileData** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle ON/OFF input** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ProfilePayload\_Rq** | This signal will request the CV Preference Management System to collect user profile in order to send it to the vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall | Import | Create** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Driver Door Status** | Vehicle tells PPP when the driver door goes from closed to ajar. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **SeatData** | All of the information needed to perform anthropometric translation. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Change\_Setting\_Rq** | User's input to changes to their Portable Profile settings. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Mobile Device Authentication** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Preference Status** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **TCU info** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **LinkProfile\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Profile\_Creation\_Confirmation = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Personal Profile Status** | Profile active/inactive status. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **DisplayWelcome\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Search Response** | Mobile devices in vicinity of host vehicle responding to a search request. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Available Profiles** | Available profiles as a result of an identification search | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Cloud Authentication** | PPP requesting the cloud to authenticate a user's credentials. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RetrieveData\_Rq** | This signal is a request sent to the CV Profile Storage to retrieve a user's profile data from storage. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **User Detected** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **User Credentials** | User credentials being sent from PPP to cloud to be verified. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Setting Categorization** | Profile Attribute deciding how to categorize profile data. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RemoveProfile\_Cmd** | This signal is a sent from the cloud to a specific vehicle that requests the removal of a user's portable profile from the specified vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPP Creation Confirmation** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PositiveID\_enabled** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Imported Profile** | Notify Enhanced Memory that there is a new imported profile in the vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Profile Data** | Target 5000 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Identification Search** | Portable Profiles searching for available profiles. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ID Request** | A request to collect credentials from the user to establish positive identification. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **AuthStatusIndicator** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Auth\_Status = Profile\_Exists** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPP\_Set\_Settings2** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Notification** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ActivePersonality\_St** | Indicates the active vehicle profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import Other Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Feature Enable Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **DownloadSuccess** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Synch\_Flag** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Disable Feature Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **SynchEnable\_St** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Remote Remove Request** | Remove profile request coming from outside of the vehicle will be sent from the Profile Management Server to Enhanced Memory so that Enhanced Memory can perform a delete profile operation. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Setting Selections** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **FordPass Account information** | Target 5000 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **VINTargetedRemove\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Link Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPP\_Set\_Settings** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key On and Profile PP Enabled** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Display\_Cred\_Error\_Timeout = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Data** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **BeginPrimaryAuth** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ButtonPairing\_St** | Status to indicate the memory seat button that is paired to a profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Passcode\_Rq** | Signal is sent to the Profile Interface client when the primary authentication method failed to identify the user. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **BLEM info** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Changes to Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PushSettings\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **CredentialValidation\_St** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Ignition Status** | Vehicle tells PPP what the ignition status of the vehicle is. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Changed Settings** | Target 2 Newtons | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPP\_Set\_Settings1** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **NewProfileSettings** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RawProfileData** | This signal contains a package of profile data that is not in vehicle-readable format. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ReturnProfileSetting** | The current value of a profile setting for the active portable profile. This data is sent to the Profile Management Server to update the user's portable profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Cabin comfrot settings** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Skip\_Portability = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **QueryProfile\_Rq** | A request sent from Profile Management Server to the Profile Setting Servers to collect all profile data when a new profile is being created. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Setting Values** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Login HMI** | HMI to allow the user to input their user credentials. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Remove Request** | Remove request coming from Enhanced Memory. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Remove Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **AssociatedVIN** | Signal provides all of the VINs that are associated to the user's portable profile to the Off-Board Interface Client. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Create Other Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ApplyName** | Applies downloaded profile name during in vehicle profile creation. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Trigered Events data** | Target 2 Newtons | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Auth\_Status = Valid AND Export\_Confirmation = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Connection Status** | Sync Bluetooth tells PPP which mobile device is actively paired to the vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import HMI** | HMI to allow the user to import a profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Capabilities** | Vehicle providing information to PPP about what personalized features it has. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Export\_Rq** | A request from the Profile Interface Client to the Profile Management Server based on the user's input. This signal determines if profile data should be exported to the cloud when a profile is linked to a FordPass account. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Provided Data** | Target 50 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import\_Rq** | A request from the Profile Interface Client to the Profile Management Server based on the user's input. This signal determines if profile data should be imported into the vehicle when a profile is linked to a FordPass account. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Positional Data Points** | Autosave notifying PPP which positional data point should be part of the profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **WelcomeScreen** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall Event Confirmation** | Enhanced Memory telling PPP that there has been a Profile Recall Event. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RS Settings** | Pre-determined group of settings that Rocket Setup will use to populate a newly created Portable Profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Portable Profile Options** | The feature providing the user with profile options for their settings. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PortableProflile\_Available** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PromptPosID** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Autosave Inhibit** | Portable Profiles inhibiting the auto save functionality due to profile protection status. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **VersionUpdate\_Rq** | A request sent from a vehicle to the cloud to check if there is an updated version of a user's portable profile stored in the cloud. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import\_Request = TRUE** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ID Response** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Guest Profile Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key On** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Disassociate Profile from VIN** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **information** | Target 5000 ms | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Auth\_Status = Invalid** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Update status** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PPPSettings\_Rq** | Pushing settings from Profile Management Server to all Profile Setting Servers that contain profile settings | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall Other Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ApplyPhoto** | Applies downloaded profile photo during in vehicle profile creation. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RemoveVIN\_Rq** | Request to remove a VIN association from a user's portable profile storage. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Feature\_St** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall | Import | Create & Synch Enabled** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Recall Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RemoveProfile\_Rsp** | Response to the RemoveProfile\_Cmd that determines if the profile removal was a success or failure. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Ignition Status** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ProfileCredential\_Value** | Credential values that are collected by Profile Interface Client and sent to the Profile Management Server. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Authorization Status** | User is authenticated by the cloud account manager. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Import Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PositiveID\_Rq** | A request to the Identification Client to attempt to establish positive ID for the active profile. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RemoteRemove\_Rq** | A request sent from the Off-Board Interface Client to the CV Preference Management Server when the user requests to remove their portable profile from a specific vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Information** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Manipulated Settings** | Target 2 Newtons | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **ProcessFeatureID** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Create Profile Request** | Request to create a profile that is send from Enhanced Memory to the Profile Management Server of PPP. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Create Portable Profile** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RVSPayload\_St** | An internal vehicle flag to determine if an RVS profile has been delivered to the vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key Off** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Available Profile Settings** | Defined list of profile parameters defined by the Profile Attribute. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PromptUser\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Login Credentials** | User providing login credentials to the system. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PositiveID\_St** | The positive identification status of a user that is in the host vehicle. | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |

#### 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.

#### 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

<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) .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CAN ID** | **Transmission Mode** | **Period** | **Signal Names** | **Transmitter(s)** | **Receiver(s)** |
|  |  |  |  |  |  |
|  |
|  |
|  |

#### 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)

<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)

#### ExportProfile\_Rsp

A response to the ExportProfile\_Rq that indicates if the export was a success or failure.

#### Export Cmd - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| YES |  |
| NO |  |

#### HWCategory

FSRCategory is a value type that defines literals used for specifying the "Category" of a Functional Safety Requirement.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Design Verification |  |
| External Fault Detection |  |
| External Fault Tolerance |  |
| General |  |
| Internal Fault Detection |  |
| Metrics |  |
| Production/Service/Decommissioning |  |

#### UserID\_Rsp

User ID information sent by the Identification Client to the Profile Management Server that indicates who is identified in the vehicle.

#### ErrorMessage\_DISPLAY

#### Result

Result is an enumeration that defines literals for specifying the outcome of a review.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| passed review |  |
| did not pass review |  |

#### SecondaryAuth\_Cmd

A command to display the secondary authentication screen on the Profile Interface Client.

#### Primary Authentication Status - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Active |  |
| Inactive |  |

#### RetrieveProfile\_Rsp

CV Profile Storage providing the requested profile data to the CV Profile Management Server

#### PrimaryAuth\_Rq

Signal from Profile Interface Client to Profile Management Server to store Primary Authentication for the active user.

#### SetupSecondaryAuth\_Rsp

Response to creating the passcode.

#### CloudProfileData\_Rsp

A success/failure response when profile data is sent to the CV Profile Storage to be saved.

#### X Setting Type

#### 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.

#### PositionalSetting\_USER INPUT

#### Apply Setting Status

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Active |  |
| Inactive |  |

#### CustomizeProfileOptions\_DISPLAY

#### Download Rsp - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| YES |  |
| NO |  |

#### Download\_Cmd

Command sent from the vehicle to the cloud to download a portable profile.

#### FordPassLogin

#### 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.

#### DisplayAccountLink\_Cmd

#### FeatureValue

Carries the setting configuration for a Feature Number. Feature Numbers and Feature Values are always paired together.

#### ExportProfile\_Cmd

This signal will send user profile data from the vehicle to the cloud to be stored.

#### Start Prim Auth Rq - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| YES |  |
| NO |  |

#### RemovedVIN\_Cmd

A command from the Off-Board Interface Client (mobile app/web) that requests a profile to be removed from a specific VIN.

#### Authentication Rsp - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| YES |  |
| NO |  |

#### PrimaryAuth\_Rsp

Signal from Profile Management Server to Profile Interface Client providing the status of the Primary Authentication linking to the active profile.

#### ScanQR\_USER INPUT

#### UserID\_Rq

Request to the authentication manager to ID the user of the vehicle.

#### CollectProfile\_Rq

Internal request in the Profile Management Server that requests the Collect Profile Settings function to initiate and begin collecting profile settings from the Profile Setting Servers.

#### UpdateConfirmation\_Rq

Request to the Profile Interface Client to display a profile update confirmation to the user.

#### DeleteProfile\_Rsp

Profile Management Server response to the DeleteProfile\_Rq. Indicates Success or Failure.

#### AvatarName\_USER INPUT

#### Passcode\_Rq

Signal from Profile Management Server to Profile Interface Client to display the passcode screen.

#### Download\_Error

Message sent to the Profile Interface Client indicating an error in the profile download so that a message can be displayed to the user.

#### Assumption Categories

Assumption Categories is a value type that defines literals used for specifying the "Category" of an Assumptions element.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| <UNSPECIFIED> |  |
| Behavioral |  |
| Controllability |  |
| Vehicle |  |
| Other Systems |  |

#### ProfileSettings

Sends profile settings to be stored in the Profile Management Server.

#### SecondaryAuthComplete\_Status

A status sent by the Profile Management Server to indicate that the Secondary Authentication has been successfully linked to a profile.

#### DisplayProfile\_Cmd

#### VINAvailableOffline

#### Selected Login Method - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Method 1 |  |
| Method 2 |  |
| Method 3 |  |

#### RemoteRemoveProfile\_Rq

A request sent from the cloud to the vehicle to remotely remove a profile from a vehicle.

#### UserIDRecall\_Rq

Recall request based on a user ID.

#### Portable Profile Attribute

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
|  |  |

#### UnlinkProfile\_Rsp

Signal sent from the CV Profile Management Server tot he vehicle with the status of the Unlink profile command.

#### LinkPersIndex\_Cmd

#### PositionalExportValues

#### StoredVehicleProfileSettings

Internal signal within the Profile Management Server that provides other functions with profile settings stored on the vehicle.

#### DeleteProfile\_Rq

A command to the Profile Management Server to delete a specific profile. The profile identifier shall be a parameter of the DeleteProfile\_Cmd

#### ExportError\_Rq

#### Download Cmd - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| YES |  |
| NO |  |

#### V&VMethod

V&VMethod is a value type that defines literals for specifying the V&V Method(s) used to verify proper implementation of a Functional or Technical Safety Requirement.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Vehicle test |  |
| Peer review |  |
| System test |  |
| Software test |  |

#### Unlink\_Rq

Request from the Profile Interface Client to the Profile Management Server to unlink a profile from its linked FordPass account.

#### User Selection- logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Opition 1 |  |
| Option 2 |  |
| Option 3 |  |
| Option 4 |  |

#### Authentication Cmd - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| YES |  |
| NO |  |

#### Z Setting Type

#### Y Settings Type

#### RecallMethod\_USER INPUT

#### RVS\_Cmd

A command from Profile Management Server to Profile Interface Client to indicate a RVS payload has been received by the vehicle.

#### PrimaryAuthenticationSetupComplete

#### DisassociateVIN\_Rsp

A response to the DisassociateVIN\_Rq that indicates if the request was a success or failure.

#### Delete\_Rq

Request from Profile Management Server to Enhanced Memory system to delete a profile.

#### ExportTrigger\_Rq

Signal received by the Profile Management Server that triggers an export of profile settings to the CV Profile Management Server.

#### Authentication\_Status

Internal vehicle status that broadcasts if a profile is authenticated or not.

#### SettingChanged

#### Authentication\_Cmd

Profile credentials sent to the cloud to be authenticated.

#### CloudProfileData\_Cmd

Profile data that is either sent from the Off-Board Interface Client or the vehicle to the CV Profile Storage.

#### ProfileCredentials

Credential values that are collected by Profile Interface Client and sent to the Profile Management Server.

#### CollectPreferences\_Cmd

#### TranslatedPositional\_Values

#### Unlink\_Rsp

Response to the Unlink\_Rq indicating a Success or Fail.

#### QRValue

#### CustomizeProfileSelection

#### HardwareV&VMethods

HardwareV&VMethods is a value type that defines literals used for specifying the V&V Method(s) used for to verify proper implementation of a Hardware Safety Requirement.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Durability Test |  |

#### DownloadComplete

Signal to indicate to the Profile Interface Client that the profile download/creation is complete and the screen no longer needs to be locked out.

#### SetupSecondaryAuth\_Rq

Passcode values sent from Profile Interface Client to Profile Management Server during initial passcode creation.

#### LinkIgnitionDevice\_Cmd

#### PositionalSettingValues

#### ApplySettingStatus

Determines if a profile setting had successfully been set by the Profile Setting Server.

#### VINAssociations

#### QuerySettingStatus

#### PasscodeValues

Passcode values received by the user and passed to Profile Management Server.

#### UnlinkProfile\_USER INPUT

#### DisassociateVIN\_Rq

A request from vehicle to cloud to disassociate a VIN from a user's cloud profile.

#### Authentication\_Rsp

A success or failure response for the Authentication\_Cmd. The response tells the vehicle if the provided credentials are authenticated or not.

#### SRSCategory

SRSCategory is a value type that defines literals used for specifying the "Category" of a Technical Safety Requirement.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| General |  |
| Safety Related Function |  |
| Internal Fault Handling |  |
| External Fault Handling |  |
| Latent Fault Handling |  |
| Metric |  |
| Reduced Functionality |  |
| User Information |  |
| Maintain Safe State / Recovery |  |

#### Export Rsp - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| YES |  |
| NO |  |

#### Message\_Rq

Generic signal from Profile Management Server to Profile Interface Client to request a message to the user.

#### Absolute Positioning

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
|  |  |
|  |  |

#### SecondaryPasscode\_USER INPUT

#### Display login Rq - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| YES |  |
| NO |  |

#### SecondAuth Status - logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Complete |  |
| Not Complete |  |

#### Operation Code - logic

#### DownloadLockout\_Rq

A command sent to the Profile Interface Client when the Profile Management Server is actively applying profile data.

#### DisplayLogin\_Rq

#### SoftwareV&VMethods

SoftwareV&VMethods is a value type that defines literals used for specifying the V&V Method(s) used to verify proper implementation of a Software Safety Requirement.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Inspection |  |

#### NoPositionalData

#### UnlinkProfile\_Cmd

A command sent from the vehicle to the cloud to unlink a FordPass account from the requesting vehicle profile.

#### LoginStatus\_DISPLAY

#### VaVStatus

VaVStatus is a value type that defines literals used for specifying that status of a FS V&V review activity.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| not yet started |  |
| review in progress |  |
| review complete |  |

#### AuthenticatedCredentials

Internal signal in the Profile Management Server that passes received Authenticated Credentials that are stored with the newly linked vehicle profile.

#### Download\_Rsp

A response to the Download\_Cmd. The response imports profile data for the requesting profile.

#### VehicleContent\_Rq

A request to gather the personalizable vehicle content for a specific VIN.

#### FSRCategory

FSRCategory is a value type that defines literals used for specifying the "Category" of a Functional Safety Requirement.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| General |  |
| Safety Related Function |  |
| Maintain Safe State/Recovery |  |
| User Information |  |
| Reduced Functionality |  |

#### UserCredentials\_USER INPUT

#### VehicleContent\_Rsp

A response to VehicleContent\_Rq that contains personalizable vehicle content for a specific VIN.

#### Feature Value - logic

#### Feature Number - logic

#### UserCredentials

#### Person Index - Logic

#### Dwld Lockout status- logic

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| Active |  |
| Inactive |  |

#### StoreVINCloud\_Rq

#### asil

asil is a value type that defines literals used for specifying the automotive safety integrity level for requirements, functions, and architecture elements.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| QM |  |
| A |  |
| B |  |
| C |  |
| D |  |
| A(B) |  |
| A(C) |  |
| A(D) |  |
| B(C) |  |
| B(D) |  |
| C(D) |  |
| A(A) |  |
| B(B) |  |
| C(C) |  |
| D(D) |  |
| QM(A) |  |
| QM(B) |  |
| QM(C) |  |
| QM(D) |  |

#### AccountID

Identifier assigned to a user's Ford Account in a vehicle that is used to link profiles with the Ford Account.

#### PersIndex

Determines the profile number that the Profile Setting Server needs to execute an operation on.

#### DisplayCustomizeScreen\_Rq

#### NormalizedPositionalData

#### FeatureNumber

Signal that carries the hexadecimal values assigned to each setting that is stored in the profile.

#### SWCategory

FSRCategory is a value type that defines literals used for specifying the "Category" of a Functional Safety Requirement.

##### Encoding values

|  |  |
| --- | --- |
| **Enumeration Values** | **Enumeration Value Description** |
| General Software Safety |  |
| Safety Related Function |  |
| Internal Fault Handling |  |
| External Fault Handling |  |
| Latent Fault Handling |  |
| Reduced Functionality |  |
| User Information |  |
| Maintain Safe State/Recovery |  |
| Off-Board Tests and Coordination |  |
| Production and Service Modification of Software |  |

#### RetrieveProfile\_Rq

CV Profile Management Server requests to retrieve profile data from the CV Profile Storage.

#### PushProfile\_Rq

Internal Profile Management Server signal to determine when settings should be pushed to the Profile Setting Servers.

#### CustomizeProfileOptions\_USER INPUT

Document ends here.