Software Requirements Specification

Telematics Data Acquisition Application

*Config Id: NAV\_SRS\_N2DAQ*

Table of Contents

[1 References 3](#_Toc97054153)

[2 Scope 3](#_Toc97054154)

[3 Introduction 3](#_Toc97054155)

[3.1 Requirements Terms 3](#_Toc97054156)

[3.2 Subject Abbreviations 3](#_Toc97054157)

[3.3 Requirements Structure 5](#_Toc97054158)

[4 Overview 5](#_Toc97054159)

[5 Assumptions 7](#_Toc97054160)

[6 Software Requirements 8](#_Toc97054161)

[6.1 High Level Software Requirements 8](#_Toc97054162)

[6.2 Low Level Software Requirements 24](#_Toc97054163)

[7 Version Summary Table 53](#_Toc97054164)

# References

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Type** | **Document Number** | **Title** | **Rev** | **Date** |
| 1 | Cummins | N/A | Next Gen Data Ingestion Interface | 1.1.0 | 01-29-2019 |
| 2 | Cummins | N/A | NGDI File Upload Process | 1.1.1 | 05-17-2019 |
| 3 | Cummins | 14640 | Telematics Partner Specification | 001 | N/A |
| 4 | SRE-SDK | N/A | Stoneridge Linux Application Programming Interface | 6.6.2 | N/A |
| 5 | Cummins | N/A | OEM-TSP Cloud Specific for Engineering Data | 1.1.1 | 06-03-2019 |
| 6 | Navistar cloud | N/A | Device Communication APIs | 0.0.1 | N/A |

# Scope

This document defines functional requirements applicable to the Telematics Data Acquisition Application software developed by or for Navistar.

# Introduction

## Requirements Terms

The following words are used as following in the requirements provided by this document:

* “Shall” is used to define a mandatory requirement. Such requirements are to be verifiable.
* “Must” is used to state requirements placed upon other system components (including other software components) which are not defined in this document.
* “Will” is used to describe conditions resulting from adherence to other requirements and known behavior
* “Are” and “Is” are used in definitions and to describe facts.

## Subject Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| DAQ App | Data Acquisition Application |
| DAQ | Data Acquisition over XCP |
| EAL | Engineering Access Level |
| EDL | Engineering Data Logging |
| LAL | Learning Access Level |
| UDS | Unified Diagnostics Services |
| DDID | Dynamically Defined Data Identifier |
| NRC | Negative Response Code |
| XCP | Universal Measurement and Calibration Protocol |
| APP | Application Manager |
| CFGMGR | Configuration Manager |
| CMM | Client Manager Model |
| CSM | Cloud Services Model |
| EVNT | Simpler Event, Composite event model and Custom Event model |
| EVMGR | Events Manager Model |
| DAM | Data Access Model |
| DCM | Data Connectivity Model |
| DLM | Data Logger Model |
| DSM | DTC Sampler Model |
| FHM | File Handler and Writer Model |
| GPS | GPS Sampler |
| J1939SMP | J1939 Sampler and Requester |
| CFGPRSR | JSON Configuration Parser Model |
| SMM | Samplers Manager Model |
| TU | Timer Utilities |
| UDSSMP | UDS Sampler |
| XCPSMP | XCP Sampler |

## Requirements Structure

Requirements in the following sections follow this pattern:

Requirement Identifier

List of Derived-from Requirements

Requirement Description

Where *Requirement Identifier* is of the form “DAQ-{H|L}-<subj\_abbrev>\_nnn

* “DAQ” indicates the requirement applies to the DAQ Application Project
* *H* or *L* identify the requirement as either High *Level(Functional)* or *Low Level* (Component) requirement. This is referring to the description level of the requirement itself.
* *subj\_abbrev* is as defined in earlier section
* *nnn* is a numerical identifier

Example requirement:

DAQ-H-APP\_01

Derived from [0] Requirement 42, [6] Section 7.1.1

The ECU software shall …

# Overview

The DAQ application provides a configurable remote data acquisition mechanism for capturing vehicle network data over J1939 and internal software data via UDS and XCP interfaces .

This document describes the software requirements for the data exchange processes that are specific to the implementation of the Data Acquisition Application for cases where trip data and/or engineering data needs to be logged .

A high-level overview of the data acquisition application architecture is provided below :

Graphical user interface, timeline

Description automatically generated

Figure 1:Architecture Overview

The application consists of the following components. Each component is described in detail in the subsequent sections in the document.

1. Configuration Manager Model
2. File Writer Model
3. Events Manager Model
4. Sampler Manager Model
5. J1939 Sampler and requester
6. XCP Sampler
7. UDS Sampler
8. GPS Sampler
9. Data Logger Model
10. Data Access Model
11. Cloud Services Model
12. File Handling Model
13. System State Receiver
14. DTC Sampler
15. Static Data Sampler
16. Time Utilities
17. Client Manager Model
18. Validator Model
19. Translator Model

# Assumptions

* DAQ App assumes that the XCP is enabled on the target ECU. DAQ App does not contain the logic to verify whether XCP is enabled or not on the target ECU and will fail in case XCP is not enabled on target production ECUs.

# Software Requirements

## High Level Software Requirements

### Generic Application Software Requirements

DAQ-H-GEN\_1

DAQ application shall provide a mechanism to remotely configure data acquisition via J1939, XCP and UDS data links via a configuration file that defines the data collection and data content configuration specification.

DAQ-H-GEN\_2

DAQ application shall verify that the configuration specification is compatible with the target ECU before configuring the data acquisition operation.

DAQ-H-GEN\_3

DAQ application shall return an appropriate response to the remote configuration server if the application is compatible / not compatible with the target ECU.

DAQ-H-GEN\_4

DAQ application shall transmit the collected data to the remote server as specified in the configuration file.

DAQ-H-GEN\_5

DAQ application shall provide common functionality to check whether input string is number or not.

DAQ-H-GEN\_6

DAQ application shall provide common functionality to convert hex to integer value.

DAQ-H-GEN\_7

DAQ application shall provide common functionality to convert integer to hex value.

DAQ-H-GEN\_8

DAQ application shall provide common functionality to convert numeric string to integer value.

DAQ-H-GEN\_9

DAQ application shall provide common functionality to check whether input string is hex or not.

### Configuration Manager High Level Software Requirements

DAQ-H-CFGMGR\_01

DAQ App shall support single or multiple configuration file(s) with multiple samplers sampling simultaneously.

DAQ-H-CFGMGR\_02

If a new configuration file has been received, the Configuration manager shall check whether the newly received file is valid or not using ValidationModel.

DAQ-H-CFGMGR\_03

If there is not any configuration file in Active directory, ConfigurationManager shall parse the default configuration file.

DAQ-H-CFGMGR\_04

If a new configuration file has been received, the Configuration manager shall apply the latest configuration file on the next key cycle.

DAQ-H-CFGMGR\_05

Derived from Ref [1] Section IV.A,

DAQ app shall parse the UDS (EAL, TripData) configuration file according to the schema defined in Ref [1].

DAQ-H-CFGMGR\_06

DAQ app shall parse the XCP and J1939 Configuration file(s) according to the schema definition for Navistar ECMs.

DAQ-H-CFGMGR\_07

DAQ App shall use Data Priority which indicates the priority of the Data Sampling Configuration(s) that are being activated.

DAQ-H-CFGMGR\_08

Configuration Manager shall send the parsed J1939 message to Translator Model for mapping into respective SPN, PGN, Topic, Signals and IsCyclic parameters.

### File Writer Model High Level Software Requirements

*DAQ-H-FWM\_01*

File writer model shall write the logged buffer data into a file, once threshold set by configuration parameters

*DAQ-H-FWM\_02*

File writer model shall write all the data into .csv format file except only for trip data. For trip data File writer model shall write data into .txt format file

*DAQ-H-FWM\_03*

File writer model shall ensure to have unique name for each output file

*DAQ-H-FWM\_04*

File writer model shall maintain the priority of the output file

*DAQ-H-FWM\_05*

File writer model shall write GPS data only for supported protocols.

*DAQ-H-FWM\_06*

File writer model shall write specific sources in output file only for J1939 & EAL.

*DAQ-H-FWM\_07*

File writer model shall generate .txt file if there is an error in XCP signals.

*DAQ-H-FWM\_08*

File writer model shall not generate output file if identifying information is not available.

### Events Manager Model High Level Software Requirements

*DAQ-H-EVMGR\_01*

Events Manager model shall be able to sample data by using events that are defined via event configuration elements as triggers to initiate data acquisition.

*DAQ-H-EVMGR\_02*

*Derived from Ref[1] Section 3*

Events Manager model shall support two different types of events namely Simple Events and Composite events via event configuration elements.

*DAQ-H-EVMGR\_03*

Events Manager model shall support monitoring the J1939 data link for events as defined in the data acquisition configuration.

*DAQ-H-EVMGR\_04*

*Derived from Ref[1] Section 3.4*

Events Manager model shall support the following categories of simple events.

* + - ActiveFault
    - InactiveFault
    - PendingFault
    - ParameterCompare

*DAQ-H-EVMGR\_05*

*Derived from Ref[1] Section 3.1*

Simple events in one of the active/inactive/pending fault categories shall use a single argument to define which faults to monitor , defined by the following attributes :

* + - Protocol : The communication protocol to be monitored
    - NetworkID : Network identifier
    - DeviceID : ECU identifier
    - ParameterID: Identifier for the fault parameter
    - FailureModeID : Failure Mode Identifier

*DAQ-H-EVMGR\_06*

*Derived from Ref[1] Section 3.5*

Events Manager model shall support the simple events of type ‘ParameterCompare’ where the provided arguments are compared using the operator defined in the event definition. ‘ParameterCompare’ is supported for dynamic events for J1939, XCP.

*DAQ-H-EVMGR\_07*

*Derived from Ref[1] Section 3.5*

Simple events in the parameter comparecategory shall use the first argument to define which faults to monitor , defined by the following attributes :

* + - Protocol : The communication protocol to be monitored
    - NetworkID : Network identifier
    - DeviceID : ECU identifier
    - ParameterID: Identifier for the fault parameter

*DAQ-H-EVMGR\_08*

*Derived from Ref[1] Section 3.7*

Events Manager model shall use the operator from the parameter compare event definition to determine how the first argument would be compared to the second argument in the Parameter Compare event definition.

*DAQ-H-EVMGR\_09*

*Derived from Ref[1] Section 3.8*

Events Manager model shall support the use of the second argument provided in the parameter compare simple event definition as a parameter or constant value.

*DAQ-H-EVMGR\_10*

*Derived from Ref[1] Section 3.9*

Events Manager model shall detect the occurrence of a parameter compare simple event whenever the first argument meets the specified comparative relationship with the second argument.

*DAQ-H-EVMGR\_11*

*Derived from Ref[1] Section 3.11 to Section 3.15,*

Events Manager model shall support composite events which are the logical combination of two or more simple events that are already defined in the data acquisition configuration.

*DAQ-H-EVMGR\_12*

Events Manager model shall support events management for system level events for the synchronization between different components.

### Sampler Manager Model High Level Software Requirements

*DAQ-H-SMPLMGR\_01*

Sampler Manager model shall have the infrastructure to create all the required samplers which will sample the required data by subscribing to specific topics.

### J1939 Sampler and Requester High Level Software Requirements

*DAQ-H-J1939\_01*

DAQ App shall sample data from the target ECU over the J1939 data link as defined in the configuration.

*DAQ-H-J1939\_02*

DAQ App shall acquire periodic J1939 data from the target ECU via listening to the broadcast PGNs.

*DAQ-H-J1939\_02*

DAQ App shall retrieve the non-periodic J1939 PGN data from the target ECU by sending a request message to the target ECU.

*DAQ-H-J1939\_03*

J1939 Sampler shall receive the J1939 data acquisition configuration data structure from the Sampler Manager Model.

*DAQ-H-J1939\_04*

J1939 Sampler shall subscribe to the appropriate J1939 topic to receive data and store the received data in the container.

*DAQ-H-J1939\_05*

In case of exceptions , J1939 Sampler shall write an appropriate warning / error message using the SDK interface to log messages and terminate the process.

*DAQ-H-J1939\_06*

J1939 requester shall periodically send PGN requests to initiate transmission from the target ECU so that the sampler will receive the desired PGN data for data acquisition.

*DAQ-H-J1939\_07*

J1939 Requestor shall use an access token which is only valid for one single request for a specific ECU.

### XCP Sampler High Level Software Requirements

*DAQ-H-XCP\_01*

DAQ App shall configure the telematics device as the XCP Master.

*DAQ-H-XCP\_02*

DAQ App shall configure the communication with the target ECU over the XCP data link using the parameters specified by the configuration file.

*DAQ-H-XCP\_03*

Before Configuring the data acquisition over XCP datalink , the application shall verify that if the data content specification is compatible with the target ECU.

*DAQ-H-XCP\_04*

DAQ App shall establish a secure connection with over XCP before executing data acquisition configuration and data sampling commands if the target ECU is a production unit as defined in the configuration file.

*DAQ-H-XCP\_05*

If the target ECU is a development unit , the DAQ app shall perform the data acquisition configuration and data sampling without XCP secure access .

*DAQ-H-XCP\_06*

DAQ App shall support data collection from the target ECU over XCP by synchronous DAQ and periodic short uploads.

*DAQ-H-XCP\_07*

DAQ App shall store the data acquired over the XCP data link *<data format>*.

*DAQ-H-XCP\_08*

DAQ App shall store the physical values of the acquired signals values from the target ECU to the output file.

*DAQ-H-XCP\_09*

DAQ app shall set ECU properties.

*DAQ-H-XCP\_10*

DAQ App shall configure the target ECU to execute synchronous DAQ over the XCP data link according to the data content specification from the configuration data.

*DAQ-H-XCP\_11*

DAQ app shall write data to Data Access layer for received ODT.

*DAQ-H-XCP\_12*

DAQ app shall setup the list of parameters to be sampled along with the information required for ECU.

*DAQ-H-XCP\_13*

DAQ app shall stop the DAQ list when application is terminating.

*Connection Management*

*Secure Access*

*Sampling*

### UDS Sampler High Level Software Requirements

*DAQ-H-UDS\_01*

*Derived from Ref[3]*

DAQ App shall provide telematics features capabilities with Cummins ECMs using the UDS Protocol if configured in the data acquisition configuration file.

*DAQ-H-UDS\_02*

*Derived from Ref[3] , section 4.5.2.3*

DAQ App shall support UDS Transport Protocol to send and receive UDS messages that has a payload greater than 8 bytes.

*DAQ-H-UDS\_03*

DAQ App shall establish Engineering Access Level with the ECM and start Engineering Data Logging (EDL) from the ECM over UDS on the CAN 500K port.

*DAQ-H-UDS\_04*

*Derived from Ref[1] , section IV.C*

DAQ App shall establish UDS Default Session with the ECM to read standard parameters over the UDS data link if the data content specification contains parameters with Protocol = UDS.

*DAQ-H-UDS\_05*

*Derived from Ref[1] section IV.C , Ref[3] section 4.5.2*

DAQ App shall establish the telematics diagnostics session , Engineering Access Level with the ECM to collect engineering data from the ECM over UDS data link if the data content specification contains parameters with Protocol = EAL.

*DAQ-H-UDS\_06*

*Derived from Ref[1] , section IV.C and Ref[3] section 4.5.3*

DAQ App shall establish the telematics diagnostics session , Engineering Access Level with the ECM for Embedded Trip Data acquisition to read the stored trend parameters in the ECM over UDS if the data content specification contains parameters with

Protocol = TripData.

*DAQ-H-UDS\_07*

*Derived from Ref[1] , section IV.C*

DAQ App shall implement the secure access mechanism over UDS data link to get the EAL access from the ECU.

*DAQ-H-UDS\_08*

*Derived from Ref[1] section 1*

DAQ App shall initiate a Learning Session , if the telematics device does not have a valid Access Key for EAL , if the data content specification contains parameters with

Protocol = TripData or Protocol = EAL.

*DAQ-H-UDS\_09*

*Derived from Ref[3] . section 4.5.2.1*

DAQ app shall start the telematics diagnostic session using the UDS DiagnosticsSessionControl Service (0x10) Sub-function Telematics Session (0x40) to start the telematics diagnostics session.

|  |  |
| --- | --- |
| ID | Sub-Function |
| 0x40 | diagnosticSessionType |

*DAQ-H-UDS\_10*

*Derived from Ref[3] . section 4.5.2.2*

DAQ app shall perform the local seed and key algorithm in order to enable the Telematics Diagnostic Session and the Engineering Access Level using the UDS SecurityAccessService (0x27).

*DAQ-H-UDS\_11*

*Derived from Ref[3] . section 4.5.2.2*

The application shall use UDS SecurityAccessService (0x27) Sub-Function Engineering Access Level Seed (0x11) to request the seed from the ECU.

*DAQ-H-UDS\_12*

*Derived from Ref[3] . section 4.5.2.2*

The application shall use UDS SecurityAccessService (0x27) Sub-Function Engineering Access Level Key (0x12) to send the session key to the ECU.

|  |  |
| --- | --- |
| ID | Sub-Function |
| 0x11 | securityAccessRequestSeed |
| 0x12 | securityAccessSendKey |

*LAL seed (0x13 ?)*

*LAL Key (0x14 ?)*

*DAQ-H-UDS\_13*

*Derived from Ref[3] . section 4.5.2.3*

The application shall use the service DynamicallyDefineDataIdentifier (0xBA) sub-function defineByIdentifier (0x2) in order to define the set of data identifiers retrieved from the ECU.

|  |  |
| --- | --- |
| ID | Sub-Function |
| 0x02 | defineByIdentifier |

*DAQ-H-UDS\_14*

*Derived from Ref[3] . section 4.5.2.4*

The application shall use the service ReadDataByPeriodicIdentifier (0xBB) in order to periodically retrieve the set of data identifiers retrieved from the ECU.

*DAQ-H-UDS\_14*

*Derived from Ref[3] . section 4.5.2.5*

The application shall use the service ReadDataByIdentifier (0x22) in order to retrieve the set of data identifiers retrieved from the ECU.

*ReadDataByPeriodicidentifier (0xBB) with Transmission Mode Parameters (slow , medium,fast,stop)*

*DAQ-H-UDS\_15*

*Derived from Ref[1] . section 1*

DAQ App shall receive and process the NRCs received from the ECU , and take appropriate action for the corresponding NRC when a negative response message is received.

### GPS Sampler High Level Software Requirements

*DAQ-H-GPSSMPLM\_01*

*Derived from Ref[1] Section 1.5*

GPS Sampler shall subscribe to obtain the GPS data snapshot

*DAQ-H-CLMGR\_02*

GPS Sampler shall consider GPS data valid only if GPS fix is available, else discard the snapshot

### Data Logger Model High Level Software Component Requirements

*DAQ-H-DLM-01*

Data Logger shall log the data to different locations according to the priority of the data specified in the configuration file.

*DAQ-H-DLM-02*

Data Logger shall support the periodic logging of the data snapshot with sampling rate specified in the configuration.

*DAQ-H-DLM-03*

Data Logger shall start logging data when event specified with starting event ID is raised.

*DAQ-H-DLM-04*

Data Logger shall stop logging data when event specified with ending event ID is raised.

*DAQ-H-DLM-05*

*Derived from Ref[1] Section 2.2*

Data Logger shall create the file once logged buffer data is equal to threshold. Threshold shall be set to MaxTransmitPeriod or MaxSetSize whichever is met first.

*DAQ-H-DLM-06*

Data Logger shall record start & end date-timestamp for each instance of starting & ending event id.

*DAQ-H-DLM-07*

Data Logger shall stop sampling if there is invalid signal address found in json for XCP.

### Data Access Model High Level Software Requirements

*DAQ-H-DAM\_01*

Data Access Model shall have the infrastructure to maintain pool of the sampled data updated frequently by different samplers. Data containers for following protocol samplers are required to be maintained.

“J1939 Sampler”

“DTC Sampler”

“XCP Sampler”

“UDS Sampler”

*DAQ-H-DAM\_02*

Data Access Model shall provide the infrastructure to have read and write access into data sample’s pool

### Cloud Services Model High Level Software Requirements

*DAQ-H-CSM\_01*

Cloud Services Model shall always try to keep cloud connectivity with the backend server.

*DAQ-H-CSM\_02*

Cloud services Model shall upload a file submitted for cloud upload if cloud connectivity is available.

*DAQ-H-CSM\_03*

*Derived from Ref[6], section File Upload*

Cloud services Model shall comply client provided file upload procedure

*DAQ-H-CSM\_04*

Cloud services Model shall covey to Daq Content Handler whenever message from cloud is published.

*DAQ-H-CSM\_05*

*Derived from Ref[6], section Cummins Token API*

Cloud services Model shall comply client provided Cummins Token API for UDS.

### DAQ Content Handler Model High Level Software Requirements

*DAQ-H-CH\_01*

DAQ Content handler Model shall subscribe to cloud over dedicated content ID 112.

*DAQ-H-CH\_02*

DAQ Content handler Model shall receive the payload message which is published from Backend.

*DAQ-H-CH\_03*

DAQ Content handler Model shall download payload message into compressed file.

*DAQ-H-CH\_04*

DAQ Content handler Model shall notify Config manager model once received new file.

*DAQ-H-CH\_05*

DAQ Content handler Model shall capture file information which is required for Backend notification.

*DAQ-H-CH\_06*

DAQ Content handler Model shall notify File Handler model once received new file.

### Data Connectivity Handler Model High Level Software Requirements

*DAQ-H-DC\_01*

Data Connectivity Model shall keep on checking the modem connectivity once requested for internet connectivity.

*DAQ-H-DC\_02*

Data Connectivity Model shall be first point of contact before making any call to cloud REST API.

### File Handling Model High Level Software Requirements

*DAQ-H-FHM\_01*

Files Handling Model shall ensure that entire directory structure is available throughout lifecycle of the App.

*DAQ-H-FHM\_02*

Files Handling Model shall store the output log files in compressed format, files should be available in compressed format for upload.

*DAQ-H-FHM\_03*

Files Handling Model shall activate a newly arrived configuration file only if it contains ‘activateDataCollection’ message and only after successful validation.

*DAQ-H-FHM\_04*

Files Handling Model shall delete currently activated configuration file if ‘deactivateDataCollection’ message is received for the configuration specified into it.

*DAQ-H-FHM\_05*

Files Handling Model shall apply the newly arrived configurations after ignition cycle.

*DAQ-H-FHM\_06*

Files Handling Model shall ensure memory availability at any point of time.

*DAQ-H-FHM\_07*

Files Handling Model shall upload files whenever cloud connectivity is established.

*DAQ-H-FHM\_08*

File Handling Model shall send the file path and correlationId to Cloud Services Model for uploading the logged file.

*DAQ-H-FHM\_9*

Files Handling Model shall decompress newly downloaded configuration file and store into **Miscellaneous** directory.

### System State Receiver High Level Software Requirements

*DAQ-H-SYSSTRC\_01*

*Derived from Ref[4] Classes::ecu::lapi::pb::EcuShutdownstatus*

System State Receiver shall subscribe to receive shutdown signal from system interface.

*DAQ-H-SYSSTRC\_02*

*Derived from Ref[4] Classes::ecu::lapi::pb::EcuShutdownstatus*

System State Receiver shall create an infrastructure to postpone the shutdown signal on the request of any component.

### DTC Sampler High Level Software Requirements

*DAQ-H-DTCS\_01*

DTC Sampler shall sample data from the target ECU for the DM messages defined in the configuration.

### Static Data Sampler High Level Software Requirements

*DAQ-H-STDS\_01*

Static Data Sampler shall sample the ECU serial number as a static data.

*DAQ-H-STDS\_02*

Static Data Sampler shall sample the ECU telematics device ID as a static data.

*DAQ-H-STDS\_03*

Static Data Sampler shall sample the ECU vehicle identification number as a static data.

*DAQ-H-STDS\_04*

Static Data Sampler shall sample the ECU DM20 ignition counter as a static data.

*DAQ-H-STDS\_05*

Static Data Sampler shall save the ECU vehicle identification number from previous key cycle.

*DAQ-H-STDS\_06*

Static Data Sampler shall save the ECU serial number from previous key cycle.

*DAQ-H-STDS\_07*

Static Data Sampler shall save the ECU telematics device identification number from previous key cycle.

*DAQ-H-STDS\_08*

Static Data Sampler shall check the device identification number not empty, not matching with existing one and save new device ID to file for next key cycle.

*DAQ-H-STDS\_09*

Static Data Sampler shall use dummy 000000-0000 values for device ID when not stored in file.

### Time Utilities High Level Software Requirements

*DAQ-H-TMUTL\_01*

Time Utility shall have a central infrastructure for the components to create threads for a functionality to be executed in periodic manner.

*DAQ-H-TMUTL\_02*

Time Utility shall have the infrastructure to periodically sample ECU parameter values for events specified in configuration file.

### Client Manager model High Level Software Requirements

*DAQ-H-CLMGR\_01*

Client Manager Model shall create one and only one transport client for the communication to LAPI interface over MQTT.

*DAQ-H-CLMGR\_02*

Client Manager Model shall create one system client for the communication to LAPI interface for system param related information.

*DAQ-H-CLMGR\_03*

Client Manager Model shall configure CAN2 baud rate.

*DAQ-H-CLMGR\_04*

Client Manager Model shall stop and disconnect transport client on application termination.

### Validation model High Level Software Requirements

*DAQ-L-VLDM\_01*

*Derived from DAQ-H-VLDM\_01*

ValidationModel shall validate each newly received configuration file whether it contains at least one json schema from the following:

"defineDataContentSpec",

"defineDataSamplingSpec",

"defineDataSamplingConfig",

"activateDataCollection",

“deactivateDataCollection”

*DAQ-L-VLDM\_02*

*Derived from DAQ-H-VLDM\_01*

ValidationModel shall validate “**defineDataContentSpec**” json config for **J1939** protocol and set the config reason code with description for sending the config status “**ACCEPTED**” or “**REJECTED**” to CloudServiceModel if the configuration is valid or invalid.

*DAQ-L-VLDM\_03*

*Derived from DAQ-H-VLDM\_01*

ValidationModel shall validate “**defineDataSamplingSpec**” json config for both triggerTypes “**periodic**” and “**eventDriven**” for all the protocols except “**TripData**” and set the config reason code with description to notify CloudServiceModel if the configuration is invalid.

*DAQ-L-VLDM\_04*

*Derived from DAQ-H-VLDM\_01*

ValidationModel shall validate SimpleEvent configuration and set the config reason code with description if the configuration is invalid to notify CloudServiceModel.

*DAQ-L-VLDM\_05*

*Derived from DAQ-H-VLDM\_01*

ValidationModel shall validate the json config commands “**activateDataCollection**", and “**deactivateDataCollection**”.

### Translator Model High Level Software Requirements

*DAQ-H-TRNSM\_01*

Translator Model shall create an infrastructure to map received J1939 message parameters into respective SPN, PGN, Source Address, Topic, Signals and IsCyclic parameters.

## Low Level Software Requirements

### Configuration Manager Low Level Software Requirements

*DAQ-L-CFGMGR\_01*

*Derived from DAQ-H-CFGMGR\_01*

Once a JSON configuration file is getting downloaded from cloud, configuration manager shall receive a signal to validate the file.

*DAQ-L-CFGMGR\_02*

Derived from DAQ-H-CFGMGR\_02

ConfigurationManager set the config status “ACCEPTED” or “REJECTED” if the configuration is valid or invalid for sending config status notification to Cloud.

*DAQ-L-CFGMGR\_03*

*Derived from DAQ-H-CFGMGR\_01*

ConfigurationManager shall parse maximum 4 configuration files for multi config support.

*DAQ-L-CFGMGR\_04*

Derived from DAQ-H-CFGMGR\_05

Configuration Manager shall parse the below configuration items from the downloaded configuration data in order to configure data acquisition for Cummins ECMs over UDS (EAL) protocol.

* + - defineDataContentSpec
    - defineDataSamplingSpec
    - defineDataSamplingConfig
    - activateDataCollection

*DAQ-L-CFGMGR\_05*

Derived from DAQ-H-CFGMGR\_05

Configuration Manager shall parse the below configuration items from the downloaded configuration data in order to configure data acquisition for Cummins ECMs over UDS (TripData) protocol.

* + - defineDataContentSpec
    - defineDataSamplingConfig
    - activateDataCollection

*DAQ-L-CFGMGR\_06*

Derived from DAQ-H-CFGMGR\_06

Configuration Manager shall parse the below configuration items from the downloaded configuration data in order to configure data acquisition for J1939 protocol.

* + - defineDataContentSpec
    - defineDataSamplingSpec
    - defineSimpleEvent
    - defineCompositeEvent
    - defineDataSamplingConfig
    - activateDataCollection

*DAQ-L-CFGMGR\_07*

Derived from DAQ-H-CFGMGR\_06

Configuration Manager shall parse the below configuration items from the downloaded configuration data in order to configure data acquisition for XCP protocol.

* + - defineECUParameterSpecification
    - defineDataContentSpec
    - defineDataSamplingSpec
    - defineSimpleEvent
    - defineDataSamplingConfig
    - activateDataCollection

### File Writer Model Low Level Software Requirements

*DAQ-L-FWM\_01*

*Derived from Ref[2].section 2.3.4*

File Writer Model shall follow following syntax for J1939 and XCP output files.

If json is supported for Navistar then file name would be:

“<TSPName>\_<BoxID>\_<VIN>\_SCxxxx\_<FileCloseDateTimestamp\_in\_ISO8601\_format>\_FileCounter.csv”

If json is supported for Cummins then file name would be:

“<TSPName>\_<BoxID>\_<ESN>\_SCxxxx\_<FileCloseDateTimestamp\_in\_ISO8601\_format>\_FileCounter.csv”

*DAQ-L-FWM\_02*

*Derived from Ref[2].section 5*

File Writer Model shall follow following syntax for EAL output filenames

“<TSPName>\_<BoxID>\_<ESN>\_<SCxxxx>\_<FileCloseDateTimestamp\_in\_ISO8601\_format>-<KeyCycleCounter>\_<DDID>\_<SA\_EALSourceAddress>.csv”

*DAQ-L-FWM\_03*

*Derived from Ref[2].section 5*

File Writer Model shall follow following syntax for output filenames for trip data files

“<BoxID>-<TSPName >-<FileCloseDatestamp\_in\_YYYYMMDD> <FileCloseTimestamp\_in\_HHMMSSsss>-<KeyCycleCounter>.txt”

*DAQ-L-FWM\_04*

File Writer Model shall write following static data in the header of the .csv files only.

“messageFormatVersion”

“dataEncryptionSchemeId”

“telematicsBoxId”

“componentSerialNumber”

“dataSamplingConfigId”

“startingEventDateTimestamp”

“endingEventDateTimestamp”

*DAQ-L-FWM\_05*

*Derived from DAQ-H-FWM\_03*

If filename already exists File Writer Model will replace the old file with new file.

*DAQ-L-FWM\_06*

*Derived from DAQ-H-FWM\_04*

File Writer Model shall create the output file into specific directory as per it’s priority. For e.g. Low priority data output file will be created inside “LowPriority” directory.

*DAQ-L-FWM\_07*

*Derived from DAQ-H-FWM\_05*

File Writer Model shall write data for GPS parameters for ex. Latitude, Longitude etc in output files for J1939/XCP/EAL only if it is present in configuration file.

*DAQ-L-FWM\_08*

*Derived from DAQ-H-FWM\_06*

File Writer Model shall write source as "converted~J1939~CAN1~0~" in output file only if configuration is supported for Cummins & protocol is J1939.

*DAQ-L-FWM\_09*

*Derived from DAQ-H-FWM\_06*

File Writer Model shall write source as "raw~EDL~CAN2~0~" in output file only if configuration is supported for Cummins & protocol is EAL.

*DAQ-L-FWM\_10*

*Derived from DAQ-H-FWM\_07*

File Writer Model shall create "XCPInvalidSignalErrorLog.txt" file in respective priority folder if there is invalid address present for signals in configuration file for XCP protocol.

*DAQ-L-FWM\_11*

*Derived from DAQ-H-FWM\_08*

File Writer Model shall not generate output file if identifying information like Telematics Device Id, VIN and ECU serial number is not available.

### Events Manager Model Low Level Software Requirements

*DAQ-L-EVMGR\_01*

*Derived from DAQ-H-EVMGR\_05*

Events Manager Model shall support special network identifier called ‘allNetworks” as defined in the configuration for the Simple events that are defined in one of the active/inactive/pending fault categories.

*DAQ-L-EVMGR\_02*

*Derived from DAQ-H-EVMGR\_05*

Events Manager Model shall support special device identifier called ‘allDevices” as defined in the configuration for the Simple events that are defined in one of the active/inactive/pending fault categories.

*DAQ-L-EVMGR\_03*

*Derived from DAQ-H-EVMGR\_05*

Events Manager Model shall support special parameter identifier called ‘allParameters” as defined in the configuration for the Simple events that are defined in one of the active/inactive/pending fault categories.

*DAQ-L-EVMGR\_04*

*Derived from DAQ-H-EVMGR\_05*

Events Manager Model shall support special failure mode identifier called ‘allFMIs” as defined in the configuration for the Simple events that are defined in one of the active/inactive/pending fault categories.

*DAQ-L-EVMGR\_05*

*Derived from DAQ-H-EVMGR\_03 , DAQ-H-EVMGR\_05*

Events Manager Model shall use the source address of the ECU as the deviceID if the protocol used is J1939 in the simple event definition.

*DAQ-L-EVMGR\_08*

*Derived from DAQ-H-EVMGR\_03, DAQ-H-EVMGR\_05*

Events Manager Model shall use the SPN as the parameterID if the protocol used is J1939 in the simple event definition

*DAQ-L-EVMGR\_09*

*Derived from DAQ-H-EVMGR\_03, DAQ-H-EVMGR\_05*

Events Manager Model shall use the FMI as the FailureModeID if the protocol used is J1939 in the simple event definition.

*DAQ-L-EVMGR\_10*

*Derived from DAQ-H-EVMGR\_03, DAQ-H-EVMGR\_05*

Events Manager Model shall subscribe to the appropriate topic in order to retrieve the DM1 message from the target ECU in order to retrieve the active fault information.

*DAQ-L-EVMGR\_11*

*Derived from DAQ-H-EVMGR\_03, DAQ-H-EVMGR\_05*

Events Manager Model shall request the DM2 message from the target ECU in order to retrieve the previously active fault information.

*DAQ-L-EVMGR\_12*

*Derived from DAQ-H-EVMGR\_04*

The Event Manager shall detect the occurrence of a simple event even one of the specified fault(s) becomes active , pending or inactive according to the event definition in the configuration.

*DAQ-L-EVMGR\_13*

*Derived from Ref[1] Section 3.7*

Events Manager Model shall support following comparison operators for comparing first argument with second argument.

“Equal to =”

“Not Equal to !=”

“Greater than >”

“Greater than or equal to >=”

“Lest than <”

“Less than or equal to <=”

*DAQ-L-EVMGR\_14*

*Derived from Ref[1] Section 3.8.1 and 3.8.2*

Events Manager Model shall support second argument of type “float”, “Integer” and “string” and constant value.

*DAQ-L-EVMGR\_15*

*Derived from DAQ-H-FHM\_11*

Events Manager Model shall support combination of two or more simple events combined with following logical operators.

“Logical AND”

“logical OR”

*DAQ-L-EVMGR\_16*

*Derived from DAQ-H-FHM\_12*

Events Manager Model shall make infrastructure for the components to be able to add events and notify accordingly.

*DAQ-L-EVMGR\_17*

*Derived from DAQ-H-FHM\_12*

Events Manager Model shall make infrastructure for the components to be able to subscribe to a particular event

*DAQ-L-EVMGR\_18*

*Derived from DAQ-H-FHM\_12*

Events Manager Model shall make infrastructure for the components to be able to unsubscribe from a particular event.

### Sampler Manager Model Low Level Software Requirements

*DAQ-L-SMPLMGR\_01*

*Derived from DAQ-H- SMPLMGR \_01*

Sampler Manager Model shall set up all the required samplers specified into configuration message. Following sampler would be created

“J1939 Sampler”

“DTC Sampler”

“XCP Sampler”

“UDS Sampler”

“GPS Sampler”

“Static Data Sampler”

(Note : Multiple samplers supported for “J1939 Sampler” & “DTC Samplers” only)

### J1939 Sampler and Requester Low Level Software Requirements

*DAQ-L-J1939\_01*

*Derived from DAQ-H-J1939\_06*

J1939 Requester shall use PGN request API which is required for acquiring a request token for a Single, ECU specific, PGN request topic.

*DAQ-L-J1939\_02*

*Derived from DAQ-H-J1939\_07*

If the application needs to send multiple requests to the target ECU , the J1939 requester shall get the access token again.

*DAQ-L-J1939\_03*

*Derived from DAQ-H-J1939\_07*

J1939 Requestor shall explicitly release the access token when you acquire the token but not used.

*DAQ-L-J1939\_04*

*Derived from DAQ-H-J1939\_07*

J1939 Requestor should not require release access token explicitly when real-time partition has processed the PGN Request successfully.

### XCP Sampler Low Level Software Requirements

*DAQ-L-XCP\_01*

*Derived from DAQ-H-XCP\_01*

DAQ app shall create the XCP Samplers only after the application successfully verifies the compatibility of the configuration with the target ECU.

*DAQ-L-XCP\_02*

*Derived from DAQ-H-XCP\_01*

DAQ App shall read the EEPROM Identifier string from the target ECU and compare the value retrieved from the target ECU with the EEPROM identifier value defined in the configuration file in order to ensure the compatibility of the configuration with the target ECU before configuring data acquisition.

*DAQ-L-XCP\_03*

*Derived from DAQ-H-XCP\_06*

XCP Sampler shall use a linear equation along with the slope and offset defined in the XCP parameter definition from the configuration file to convert the raw values received from the target ECU to physical values before providing the signal values to the data access layer.

*DAQ-L-XCP\_04*

*Derived from DAQ-H-XCP\_04*

The application shall use Cryptolibrary APIs from the Shared Object integrated with the DAQ app to perform secure access to the target ECU over XCP data link.

*DAQ-L-XCP\_05*

*Derived from DAQ-H-XCP\_04*

The application shall securely unlock the target ECU at secure access level 3 if the target is a production unit , before executing DAQ configuration / Memory access commands .

*<List of commands that require secure access>*

*DAQ-L-XCP\_06*

*Derived from DAQ-H-XCP\_02*

DAQ app shall create XCP client & shall allocate XCP interface which is used for subsequent XCPRequests. The sampler shall provide the following parameters to the API to create and allocate the XCP interface.

* + - * + CAN Port (CAN1/CAN2), CAN baud rate, master ECU CAN id, slave ECU CAN id, timeout in milliseconds.

*DAQ-L-XCP\_07*

*Derived from DAQ-H-XCP\_09*

XCP Sampler shall provide the following ECU properties which is defined in configuration, to the base software after allocating the interface successfully.

* + - * + AddressGranularity, ByteOrder, slave block mode name, MaxDaq, MinDaq, MaxEventChannel, MaxOdtEntrySize, DaqConfigType, MaxCTO, MaxDTO.

*DAQ-L-XCP\_08*

*Derived from DAQ-H-XCP\_10*

XCP Sampler shall populate the DAQLists according to the content specification from the configuration.

*DAQ-L-XCP\_09*

*Derived from DAQ-H-XCP\_10*

XCP Sampler shall use the daq\_add() API to send each DAQLists to the slave ECU to configure DAQ. Following parameters are passed for daq\_add()

* + - * + DAQList, a guard timeout in ms to make sure the call is not blocking forever.

*DAQ-L-XCP\_10*

*Derived from DAQ-H-XCP\_10*

XCP Sampler shall clear the DAQ from ECU memory using daq\_clear(), if it is not currently running. Following parameters are passed for daq\_clear()

* + - * + Daq id, a guard timeout in ms to make sure the call is not blocking forever.

*DAQ-L-XCP\_11*

*Derived from DAQ-H-XCP\_10*

Once the DAQ list is added to ECU’s memory, XCP Sampler shall start all configured list of the specified ECU by using daq\_start() interface. Following parameters are passed for daq\_start()

* + - * + Daq id, a guard timeout in ms to make sure the call is not blocking forever.

*DAQ-L-XCP\_12*

*Derived from DAQ-H-XCP\_10*

The XCP Sampler shall provide an interface to stop DAQ by using the daq\_stop() by providing the following arguments :

* + - * + Daq id, a guard timeout in ms to make sure the call is not blocking forever.

*DAQ-L-XCP\_13*

*Derived from DAQ-H-XCP\_11*

XCP Sampler shall invoke on\_daq\_data() callback interface which will receive DAQ ODT(Object Descriptor Table) data.

*DAQ-L-XCP\_14*

*Derived from DAQ-H-XCP\_11*

Using ODT data, for each XCP signals its equivalent value is extracted & XCP Sampler shall write that XCP signal name, its equivalent value, protocol type(i.e. XCP) to data access layer.

*DAQ-L-XCP\_15*

*Derived from DAQ-H-XCP\_06, DAQ-H-XCP\_12*

DAQ app shall create list of DAQEntry from configuration file & it is added in DAQList.

1. The master driver shall receive the following parameters from the application to set up periodic short uploads form the target ECU :
   1. Daqid (Data Acquisition unique number), DAQ mode (POLLING), Rate, Addresses which contains data size , ECU memory address extension and ECU memory address for all the elements in each POLLING list
2. This is used for both setting up a DAQ in the XcpServer as well as describing currently registered DAQ.

*DAQ-L-XCP\_16*

*Derived from DAQ-H-XCP\_06, DAQ-H-XCP\_12*

1. DAQ app shall create list of DAQEntry from configuration file & it is added in DAQList.
2. The master driver shall receive the following parameters from the application for each configured DAQ lists:
   1. Daqid (Data Acquisition unique number), Event channel identifier, Event channel priority, MAX\_DAQ\_LIST, Pre-scaler, DAQ Mode (SAMPLE), Addresses which contains data size , ECU memory address extension and ECU memory address for all the DAQ elements in each DAQ list
3. This is used for both setting up a DAQ in the XcpServer as well as describing currently registered DAQ.

*DAQ-L-XCP\_17*

*Derived from DAQ-H-XCP\_09*

Based on “datatype” from data content specification of the configuration file, conversion of received raw signals values are performed & written to data access layer.

*DAQ-L-XCP\_18*

*Derived from DAQ-H-XCP\_13*

DAQ app shall stop the DAQ list using daq\_stop() when application is terminating.

### UDS Sampler Low Level Software Requirements

*DAQ-L-UDS-01*

*Derived from DAQ-H-UDS\_04 , DAQ-H-UDS\_12*

If the data content specification contains parameters with Protocol = UDS , the DAQ App shall use ReadDataByIdentifier service ($22) to read DIDs specified in the data content specification when prescribed in the data sampling configuration from the configuration file.

*DAQ-L-UDS-02*

*Derived from Ref[3] , DAQ-H-UDS\_08*

DAQ App shall use the UDS configuration data and initiate LAL (Learning Access Level – Access level 13 ?) process over UDS .

*DAQ-L-UDS-03*

*Derived from DAQ-H-UDS\_01*

Upon successfully establishing the LAL with the ECM , the DAQ App shall initiate Engineering Access and request for engineering data over UDS on CAN 500kbaud bus.

*DAQ-L-UDS-05*

*Derived from DAQ-H-UDS\_04*

If the data content specification contains parameters with Protocol = EAL , DAQ App shall establish a default session with the ECM as the initial step.

*DAQ-L-UDS-06*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C*

If the data content specification contains parameters with Protocol = EAL , DAQ App shall initiate a learning session if the telematics device does not have a valid access key for Engineering Access Level.

*DAQ-L-UDS-07*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C*

If the data content specification contains parameters with Protocol = EAL , DAQ App shall not initiate a learning session if the telematics device already had a valid access key for Engineering Access Level.

*DAQ-L-UDS-08*

*Derived from DAQ-H-UDS\_04 and Ref[3] <Add section>*

If the data content specification contains parameters with Protocol = EAL , DAQ App shall initiate a learning session only if the telematics device has good over-the -air coverage.

<Add requirements for seed and key access for LAL according to TPS and derived from DAQ-H-UDS\_06>

*DAQ-L-UDS-09*

*Derived from DAQ-H-UDS\_04 and Ref[3] <Add section>*

If the data content specification contains parameters with Protocol = EAL , DAQ App shall complete the learning access process, from the time the ECM receives the seed request , until the ECM receives the session key, must be completed in 5 minutes or less.

*DAQ-L-UDS-10*

*Derived from DAQ-H-UDS\_04 and Ref[3] <Add section>*

If the data content specification contains parameters with Protocol = EAL and the DAQ App does not the learning access process in 5 minutes or less , the process shall be retried until it is completed within the time limit.

*DAQ-L-UDS-11*

*Derived from DAQ-H-UDS\_04 and Ref[3] <Add section>*

If the data content specification contains parameters with Protocol = EAL and Learning Access Process is initiated , DAQ App shall only send the ‘Tester Present ($3E) ‘ service to the ECM until the learning process is completed.

*DAQ-L-UDS-12*

*Derived from DAQ-H-UDS\_04 and Ref[3] <Add section>*

If the data content specification contains parameters with Protocol = EAL , and the DAQ Application receives NRC -35 (Invalid Key) from the ECM during the learning access process , the DAQ application shall retry the learning access process once.

*DAQ-L-UDS-13*

*Derived from DAQ-H-UDS\_04 and Ref[3] <Add section>*

If the data content specification contains parameters with Protocol = EAL , and the DAQ Application receives NRC -35 (Invalid Key) from the ECM during two consecutive tries of the learning access process , the DAQ application shall abort the process .

*DAQ-L-UDS-14*

*Derived from DAQ-H-UDS\_04 and Ref[3] <Add section>*

If the data content specification contains parameters with Protocol = EAL and if the telematics device had a valid access key for EAL , the application shall initiate a telematics session with the ECU and request Engineering Access Level.

*DAQ-L-UDS-15*

*Derived from Ref[3] . Section 4.5*

*<Add requirements for Engineering Access level request from the Telematics Supplier Spec>*

*DAQ-L-UDS-16*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C - 1.5.1*

DAQ App shall abort the Engineering Access Level Process if the ECM denies the Engineering Access Level Request from the Telematics device.

*DAQ-L-UDS-17*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C – 1.5.2*

If the data content specification contains parameters with Protocol = EAL , and the DAQ Application receives NRC -35 (Invalid Key) from the ECM during the Engineering Access Level Request process , the DAQ application shall retry the learning access process once.

*DAQ-L-UDS-18*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C – 1.5.2*

If the data content specification contains parameters with Protocol = EAL , and the DAQ Application receives NRC -35 (Invalid Key) from the ECM during two consecutive tries of the Engineering Access Level Request process, the DAQ application shall abort the process .

*DAQ-L-UDS-19*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C – 1.6*

If the data content specification contains parameters with Protocol = EAL and the EAL access is established with the ECM , DAQ App shall define data groups as specified in the data content specification using the DynamicallyDefineDataIdentifier service .

*<Add requirements for DDID from the Telematics Supplier Spec>*

*DAQ-L-UDS-XX*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C – 1.6.1*

If the data content specification contains parameters with protocol = EAL , while sampling engineering data after obtaining successful Engineering Access to the ECM, DAQ App shall define each data group separately.

*DAQ-L-UDS-XX*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C – 1.6.2*

If the data content specification contains parameters with protocol = EAL , while sampling engineering data after establishing successful Engineering Access with the ECM , if a Negative Response Code is received from the ECM , DAQ app shall implement the appropriate action for the NRC.

*DAQ-L-UDS-XX*

*Derived from <add high level req traceability> , Ref[1] Section IV.C – 1.7*

If the data content specification contains parameters with protocol = EAL ,after the DDIDs are established , DAQ App shall request the data using ReadDataByIdentifier (0x22) service if the Trigger Type is not ‘periodic’ in the data sampling specification.

*DAQ-L-UDS-XX*

*Derived from <add high level req traceability> , Ref[1] Section IV.C – 1.7*

If the data content specification contains parameters with protocol = EAL ,after the DDIDs are established , DAQ App shall request the data using ReadDataByPeriodicIdentifier (0xBB) service if the Trigger Type is ‘periodic’ in the data sampling specification.

*DAQ-L-UDS-XX*

*Derived from <add high level req traceability> , Ref[1] Section IV.C – 1.7.1*

*<Add requirements for slow , fast ,medium, stop for service 0xBB>*

*DAQ-L-UDS-XX*

*Derived from <add high level req traceability> , Ref[1] Section IV.C – 1.7.2*

If the data content specification contains parameters with protocol = EAL and the DDIDs are established , DAQ app shall request each DDID separately from the ECM.

*DAQ-L-UDS-XX*

*Derived from DAQ-H-UDS\_04 , Ref[1] Section IV.C – 1.6.2*

If the data content specification contains parameters with protocol = EAL and DDIDs are established with the ECM , if a Negative Response Code is received from the ECM while reading the DDIDs, DAQ app shall implement the appropriate action for the NRC.

*DAQ-L-UDS-XX*

*Derived from <add high level req traceability> , Ref[1] Section IV.C – 1.7.6*

If the data content specification contains parameters with protocol = EAL and the DDIDs are reveived successfully by the ECM , DAQ app shall pass the data to the Cummins cloud unchanged and each sample shall be formatted as below :

“<DDID>” : “<Data\_Block\_Received>”

<Add Low level requirements for Trip data>

*DAQ-L-UDS-XX*

*Derived from DAQ-H-UDS\_05, Ref[1] Section IV.C – 1.8*

If the data content specification contains parameters with Protocol = TripData , DAQ App shall establish a default session with the ECM as the initial step.

*DAQ-L-UDS-XX*

*Derived from DAQ-H-UDS\_05 , Ref[1] Section IV.C – 1.9*

If the data content specification contains parameters with Protocol = TripData, DAQ App shall initiate a learning session if the telematics device does not have a valid access key for Engineering Access Level.

*DAQ-L-UDS-XX*

*Derived from DAQ-H-UDS\_05 , Ref[1] Section IV.C – 1.9.1*

If the data content specification contains parameters with Protocol = TripData, DAQ App shall not initiate a learning session if the telematics device already had a valid access key for Engineering Access Level.

*DAQ-L-UDS-0xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.9.1*

If the data content specification contains parameters with Protocol = TripData, DAQ App shall initiate a learning session only if the telematics device has good over-the -air coverage.

<Add requirements for seed and key access for LAL according to TPS and derived from DAQ-H-UDS\_06 – communize reqs for LAL>

*DAQ-L-UDS-09*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.9.1*

If the data content specification contains parameters with Protocol = TripData, DAQ App shall complete the learning access process, from the time the ECM receives the seed request , until the ECM receives the session key, must be completed in 5 minutes or less.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.9.1*

If the data content specification contains parameters with Protocol = TripData and the DAQ App does not the learning access process in 5 minutes or less , the process shall be retried until it is completed within the time limit.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.9.1*

If the data content specification contains parameters with Protocol = TripData and Learning Access Process is initiated , DAQ App shall only send the ‘Tester Present ($3E) ‘ service to the ECM until the learning process is completed.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.9.2*

If the data content specification contains parameters with Protocol = TripData , and the DAQ Application receives NRC -35 (Invalid Key) from the ECM during the learning access process , the DAQ application shall retry the learning access process once.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.9.2*

If the data content specification contains parameters with Protocol = TripData , and the DAQ Application receives NRC -35 (Invalid Key) from the ECM during two consecutive tries of the learning access process , the DAQ application shall abort the process .

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.10*

If the data content specification contains parameters with Protocol = TripData and if the telematics device had a valid access key for EAL , the application shall initiate a telematics session with the ECU and request Engineering Access Level.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.10.1*

If the data content specification contains parameters with Protocol = TripData and the ecu denies Engineering Access Level, the application shall abort the process.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.10.2*

If the data content specification contains parameters with Protocol = TripData and a ‘Negative response code’ NRC-35 (“Invalid Key(IK)”) is received from the ECU while establishing the session or granting the access, , the application shall retry once.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.10.2*

If the data content specification contains parameters with Protocol = TripData and a ‘Negative response code’ NRC-35 (“Invalid Key(IK)”) is received from the ECU while establishing the session or granting the access, the application shall abort the process after two tries.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , the application shall receive trip data using the routine control service per section 4.5.3 of Telematics Partner Specification.

<Add requirements on how to receive trip data as per TPS>

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.1*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , the application shall receive trip data using the routine control service per section 4.5.3 of Telematics Partner Specification.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.1*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , the application shall include the RoutineControlOption = “Partial” if data content specification defines “Partial” in the parameters section.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.1*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , the application shall include the RoutineControlOption = “Full” if data content specification defines “Full” in the parameters section.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.2 and 1.11.3*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , the application shall repeat the data request if a positive response with the data is received from the ECU after the initial request.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.2 and 1.11.3*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , the application shall repeat the data request-response (data retrieval) sequence until the ECU returns the ‘no more data’ positive response (0xFF 0xFF) .

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.4*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , if a Negative Response Code is received from the ECM while reading trip data, DAQ app shall implement the appropriate action for the NRC .

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.5, 1.11.6*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , the application shall pass the received block of data [[1]](#footnote-1)unchanged to the Cummins cloud , omitting the ‘no more data (0xFF 0xFF) indication.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.6*

If the data content specification contains parameters with Protocol = TripData and Engineering Access Level is established with the ECU , the application shall return each block of trip data received from the ECU as a separate sample.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.6*

If the data content specification contains parameters with Protocol = TripData , the application shall return the trip data block retrieved from the ECU in the following format :

“<partial/full>” : “<DATA\_BLOCK\_RETRIEVED>”

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.7*

If the data content specification contains parameters with Protocol = TripData , the application shall choose to request the trip data four times , transmit those four blocks of data , and then continue with the data retrieval sequence , in order to minimize the stored data.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_05 and Ref[3] Section IV.C – 1.11.7 and the req above*

If the data content specification contains parameters with Protocol = TripData and if the application is requesting the trip data four times , the trip data must be passed to the Cummins cloud in groups of four data blocks.

*DAQ-L-UDS-xx*

*Derived from DAQ-H-UDS\_15*

The application shall retry the requested operation three times if a negative response is received from the ECU with NRC : 0x22 – conditionsNotCorrect (CNC).

### GPS Sampler Low Level Software Requirements

*DAQ-L-GPSSMPLM\_01*

*Derived from DAQ-H-GPSSMPLM\_01*

GPS Sampler shall always maintain a last valid GPS data snapshot.

### Data Logger Model Low Level Software Component Requirements

*DAQ-L-DLM-01*

*Derived from DAQ-H-DLM-01*

The data logger shall log high priority data to the folder named ‘High’.

*DAQ-L-DLM-02*

*Derived from DAQ-H-DLM-01*

The data logger shall log normal priority data to the folder named ‘Normal’.

*DAQ-L-DLM-03*

*Derived from DAQ-H-DLM-01*

The data logger shall log low priority data to the folder named ‘Low’.

*DAQ-L-DLM-04*

*Derived from DAQ-H-DLM-02*

The Data logger model shall ensure that for periodic sampling settings do not have events settings specified.

*DAQ-L-DLM-05*

*Derived from DAQ-H-DLM-02*

The Data logger model shall ensure that for periodic sampling, sampling rate cannot be zero.

*DAQ-L-DLM-06*

*Derived from DAQ-H-DLM-03*

The Data logger model shall ensure that for event sampling setting Starting and ending event are not same.

*DAQ-L-DLM-07*

*Derived from DAQ-H-DLM-03*

The Data logger model shall ensure that for event sampling setting, post buffer settings are specified only if ending event specified.

*DAQ-L-DLM-08*

*Derived from DAQ-H-DLM-04*

The Data logger model shall ensure that for event sampling setting, pre buffer settings are specified only if starting event specified.

*DAQ-L-DLM-09*

*Derived from DAQ-H-DLM-04*

The Data logger model shall ensure that prebuffer is not greater than MaxSetSize or MaxTransmitPeriod.

*DAQ-L-DLM-10*

*Derived from DAQ-H-DLM-05*

The Data logger model shall ensure that MaxTransmitPeriod and MaxSetSize are not equal.

*DAQ-L-DLM-11*

The Data logger model shall ensure to store all the currently logged data into a file if shutdown signal is raised.

*DAQ-L-DLM-12*

*Derived from DAQ-H-DLM-06*

The Data Logger shall ensure for event trigger type then captures start & end DateTimestamps of each instance of the starting event and ending event id.

*DAQ-L-DLM-13*

*Derived from DAQ-H-DLM-07*

The Data Logger shall stop sampling when invalid signal address found in json for XCP send data to write the error output file.

### Data Access Model Low Level Software Requirements

*DAQ-L-DAM\_01*

*Derived from DAQ-H-DAM\_01*

Data Access Model shall store the data in a key-value pair container where key will be unique

*DAQ-L-DAM\_02*

*Derived from DAQ-H-DAM\_02*

Data Access Model shall provide functionality to fetch the latest snapshot of all the sampled configuration parameter’s values

*DAQ-L-DAM\_03*

Data Access Model shall provide functionality to flush all the container data

### Cloud Services Model Low Level Software Requirements

*DAQ-L-CSM\_01*

*Derived from DAQ-H-CSM\_01*

Cloud service Model shall use client provided APIs for using services offered by backend server.

*DAQ-L-CSM\_02*

Cloud service Model shall comply with the authentication mechanism required by backend server.

*DAQ-L-CSM\_03*

*Derived from DAQ-H-CSM\_02*

Cloud service Model shall try to upload a file twice, in case of failure it shall delayed trying to upload for 1 minute.

*DAQ-L-CSM\_04*

*Derived from Ref[6], section File Upload*

Cloud service Model shall upload the compressed files using REST API with following three steps:   
1] Initialize File upload API  
2] Upload Blob API  
3] Device Notification API.   
In order to use above API's an authorization token required for emulator which has some validity period.

*DAQ-L-CSM\_05*

*Derived from Ref[6], section File Upload*

Cloud service Model shall call device notify message even file upload gets failed at initialize API or Upload Blob API.

*DAQ-L-CSM\_06*

*Derived from Ref[6], section File Upload*

Cloud service Model shall use different notify file upload message for Cummins and for Navistar.

*DAQ-L-CSM\_07*

*Derived from DAQ-H-CSM\_04*

Cloud service Model shall notify the configurations status response of the downloaded configuration file to the backend server.

*DAQ-L-CSM\_08*

*Derived from DAQ-H-CSM\_05*

Cloud service Model shall call Cummins Token API to fetch the token from Cummins endpoint for UDS connection.

### DAQ Content Handler Model Low Level Software Requirements

*DAQ-L-CH\_01*

*Derived from DAQ-H-CH\_03*

DAQ Content Handle shall download a file into miscellaneous directory and notify Configuration manager model to validate the same

*DAQ-L-CH\_02*

*Derived from DAQ-H-CH\_03*

DAQ Content handler Model shall download new configuration file whenever message from cloud is published over the content ID 112.

*DAQ-L-CH\_03*

*Derived from DAQ-H-CH\_04*

DAQ Content handler Model shall notify to configuration Manager model for validating received configuration.

*DAQ-L-CH\_04*

*Derived from DAQ-H-CH\_05*

DAQ Content handler Model shall notify Cloud Services Model to maintain Correlation Identifier.

*DAQ-L-CH\_05*

*Derived from DAQ-H-CH\_03*

DAQ Content handler Model shall download new configuration file by extracting FileURL from payload message.

*DAQ-H-CH\_07*

*Derived from DAQ-H-CH\_03*

DAQ Content handler Model shall download new configuration file content by extracting payload message and stores into Configuration.json.zip.

*DAQ-L-CH\_08*

*Derived from DAQ-H-CH\_06*

DAQ Content handler Model shall notify File Handling Model to maintain Correlation Identifier into text file.

### Data Connectivity Model Low Level Software Requirements

*DAQ-L-DC\_01*

*Derived from DAQ-H-DC\_01*

Cloud service Model shall try to establish internet connection with following server URL

<https://developer.sandbox.oncommandconnection.com> after getting modem connection.

### File Handling Model Low Level Software Requirements

*DAQ-L-FHM\_01*

*Derived from DAQ-H-FHM\_01*

File Handling Model shall maintain Configuration files into **ActiveConfigFiles** and **InActiveConfigFiles** configuration directories as mentioned below.

“common/ConfigFiles/ActiveConfigFiles”: for storing currently activated configuration file.

“common/ConfigFiles/InActiveConfigFiles”: For storing currently inactive configuration files.

*DAQ-L-FHM\_02*

*Derived from DAQ-H-FHM\_01*

File Handling Model shall maintain output Log files into **LowPriority**, **NormalPriority** and **HighPriority** log directories as per output file’s priority set by received configuration. Directory paths should follow below mentioned path.

“common/LogFiles/LowPriority”: For storing low priority data

“common/LogFiles/NormalPriority”: For storing medium priority data

“common/LogFiles/HighPriority”: For storing high priority data

*DAQ-L-FHM\_03*

*Derived from DAQ-H-FHM\_01*

File Handling Model shall maintain create a **Miscellaneous** directory for storing miscellaneous and temporary files, directory shall be created inside “common” directory.

"common/Miscellaneous”: For storing miscellaneous and temporary files

*DAQ-L-FHM\_04*

*Derived from DAQ-H-FHM\_01*

File Handling Model shall not create the directory if already exists.

*DAQ-L-FHM\_05*

*Derived from DAQ-H-FHM\_01*

File Handling Model shall not delete any directory in the lifetime of the App.

*DAQ-L-FHM\_06*

*Derived from DAQ-H-FHM\_03*

File Handling Model on valid activateDataCollection message, shall move the newly downloaded file from **Miscellaneous** to **InActiveConfigFiles** directory and then move to **ActiveConfigFiles** directory.

*DAQ-L-FHM\_07*

*Derived from DAQ-H-FHM\_04*

File Handling Model on valid deactivateDataCollection message shall search the file in **ActiveConfigFiles** or **InActiveConfigFiles** configuration directory and delete it if found.

*DAQ-L-FHM\_08*

*Derived from DAQ-H-FHM\_06*

File Handling Model shall start deletion of files if low memory space is detected less than minimum threshold value. File Handling Model shall delete the Low priority log files first, then Medium priority log files and then High priority log files. File Handling Model shall delete the oldest file present.

*DAQ-L-FHM\_9*

*Derived from DAQ-H-FHM\_06*

File Handling Model shall stop deleting old files once sufficient memory is available which is greater than max threshold value.

*DAQ-L-FHM\_10*

*Derived from DAQ-H-FHM\_07*

File Handling Model shall initiate logged files for uploading from highest priority to lowest priority order once cloud connectivity is established.

*DAQ-L-FHM\_11*

*Derived from DAQ-H-FHM\_07*

File Handling Model shall delete the file from respective directory once it is uploaded successfully to the cloud.

*DAQ-L-FHM\_12*

*Derived from DAQ-H-FHM\_08*

File Handling Model extract ConfigId from logged file name if the protocol is J1939 or XCP.

*DAQ-L-FHM\_13*

*Derived from DAQ-H-FHM\_08*

File Handling Model extract ConfigId from UDSConfigMessage if the protocol is TripData.

*DAQ-L-FHM\_14*

*Derived from DAQ-H-FHM\_09*

File Handling Model returns Config file path from **ActiveConfigFiles** directory or returns default config if there is not any config file present in **ActiveConfigFiles** directory.

*DAQ-L-FHM\_15*

*Derived from DAQ-H-FHM\_09*

File Handling Model returns latest downloaded configuration file path from **Miscellaneous** directory.

*DAQ-L-FHM\_16*

*Derived from DAQ-H-FHM\_09*

File Handling Model shall move the validated config file from **Miscellaneous** directory to **InActiveConfigFiles** directory and if the file is not valid then it shall delete the file from the directory.

*DAQ-L-FHM\_17*

*Derived from DAQ-H-FHM\_08*

File Handling Model shall create **.txt** output file which contains **correlationId** as per the specified file name and directory path.

*DAQ-L-FHM\_18*

*Derived from DAQ-H-FHM\_09*

File Handling Model returns all the config files available in **ActiveConfigFiles** directory.

### System State Receiver Low Level Software Requirements

*DAQ-L-SYSSTRC\_01*

*Derived from DAQ-H-SYSSTRC\_01*

The shutdown signal will be notified to subscribed components by SystemStateReceiver model.

*DAQ-L-SYSSTRC\_02*

*Derived from DAQ-H-SYSSTRC\_02*

System State Receiver shall postpone the ‘SHUTDOWN\_PENDING’ signal if any pending task is required to be finished by it.

*DAQ-L-SYSSTRC\_03*

*Derived from DAQ-H-SYSSTRC\_02*

System State Receiver shall not request system interface to postpone system shutdown if ‘SHUTDOWN\_EMMINENT’ signal is received.

### DTC Sampler Low Level Software Requirements

*DAQ-L-DTCS\_01*

*Derived from DAQ-H-DTCS\_01*

DTC Sampler shall subscribe to the appropriate J1939 topic to receive DM codes

*DAQ-L-DTCS\_02*

*Derived from DAQ-H-DTCS\_01*

DTC Sampler shall store the DM message data into a container maintained by DataAccessModel. Following DM messages shall be supported

“DM1”: active fault codes

“DM2”: Inactive fault codes

“DM27 “: Pending fault codes

### Static Data Sampler Low Level Software Requirements

*DAQ-L-STDS\_01*

*Derived from DAQ-H-STDS\_01*

Static Data Sampler shall subscribe to the predefined topic “rt/telcan/ci” for accessing ECU serial number.

*DAQ-L-STDS\_02*

*Derived from DAQ-H-STDS\_01*

Static Data Sampler shall store the ECU serial number in DataAccessModel under predefined SPN “588”

*DAQ-L-STDS\_03*

*Derived from DAQ-H-STDS\_03*

Static Data Sampler shall subscribe to the predefined topic “rt/telcan/e\_vi ” for accessing ECU vehicle identification number

*DAQ-L-STDS\_04*

*Derived from DAQ-H-STDS\_04*

Static Data Sampler shall subscribe to the predefined topic “rt/telcan/dm20” for accessing ECU DM20 ignition counter number.

*DAQ-L-STDS\_05*

*Derived from DAQ-H-STDS\_02*

Static Data Sampler shall capture telematics device number from SysParamsClient interface.

### Time Utilities Low Level Software Requirements

*DAQ-L-TMUTL\_01*

*Derived from DAQ-H-TMUTL\_01*

Time utility shall provide a utility to callback a functionality after provided delay

*DAQ-L-TMUTL\_02*

*Derived from DAQ-H-TMUTL\_01*

Time utility shall provide a utility to infinitely loop a callback to a functionality with provided delay.

*DAQ-L-TMUTL\_03*

*Derived from DAQ-H-TMUTL\_01*

Time utility shall release the requested functionality from the loop infinitely running, if requested by the corresponding component.

*DAQ-L-TMUTL\_04*

*Derived from DAQ-H-TMUTL\_01*

Timer utility shall provide system time in ISO 8601 format for DataAccessModel to be used when generating a record.

*DAQ-L-TMUTL\_05*

*Derived from DAQ-H-TMUTL\_01*

Timer utility provides time/date in a specific format for FilesWriterModel to be used when generating output files

Format: “YYYY-MM-DDTHH:MM:SS.sssZ”

### Client Manager Model Low Level Software Requirements

*DAQ-L- CLMGR \_01*

*Derived from DAQ-H- CLMGR \_01*

Client Manager Model shall share the transport client for creating samplers for all protocols like J1939, XCP and UDS.

*DAQ-L- CLMGR \_02*

*Derived from DAQ-H- CLMGR \_02*

Client Manager Model shall share the system client for capturing static information from ECU like engine serial number, telematics device ID, vehicle identification number.

*DAQ-L- CLMGR \_03*

*Derived from DAQ-H- CLMGR \_03*

Client Manager Model shall use the system parameter OTAID “41089” that controls CAN2 baud rate.

*DAQ-L- CLMGR \_04*

*Derived from DAQ-H- CLMGR \_03*

Client Manager Model shall save the configured CAN2 baud rate to system client for future use.

*DAQ-L- CLMGR \_05*

*Derived from DAQ-H- CLMGR \_03*

Client Manager Model shall match the configured CAN2 baud rate with toggle request, if not matched then write to system and if matched then use saved value from last ignition cycle.

### Validation Model Low Level Software Requirements

*DAQ-L-VLDM\_01*

*Derived from DAQ-H-VLDM\_01*

Validation Model shall validate each newly received configuration file whether it contains the following mandatory blocks

“defineECUParameterSpecification",

"defineDataContentSpec",

"defineDataSamplingSpec",

"defineDataSamplingConfig",

"activateDataCollection"

*DAQ-L-VLDM\_02*

*Derived from DAQ-H-VLDM\_01*

Validation Model shall only parse the validated configuration file else discard the same.

### Translator Model High Level Software Requirements

*DAQ-L-TRNSM\_01*

*Derived from DAQ-H-TRNSM\_01*

Translator Model shall refer ‘Topic List’ file provided, which will have the mapping of J1939 configuration parameters

*DAQ-L-TRNSM\_02*

*Derived from DAQ-H-TRNSM\_01*

Translator Model shall store J1939 parameters obtained through Topic List file in a container with J1939 SPN as a ‘key’ mapping J1939 configuration parameters as ‘value’.

*DAQ-L-TRNSM\_03*

*Derived from DAQ-H-TRNSM\_01*

Translator Model shall allow duplicates SPNs be present in the container.

# Version Summary Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Description** | **Revised by** |
| 0.1 | 21 February 2020 | Initial Software Requirement Specification document provided by KPIT | Vijay Kumar  Rohini Kokate  Laxmikant Pawar  Neeta Jawale  Pooja Adake |
| 0.2 | 5 March 2020 | Initial version of the Software Requirement Specification with XCP requirements | Sinaj Madappallil |
| 0.3 | 10 September 2020 | Initial draft. Consolidated requirements from the previous versions of the Software Requirement Documentation | Sinaj Madappallil |
| 0.4 | 22 October | Modified draft, Added remaining software requirements | Aniruddha Kulkarni |
| 0.5 | 22 Oct 2021 | Modified architecture overview figure, added DAQ content Handler and Data Connectivity requirements | Rohini Kokate |
| 0.6 | 22 Dec 2021 | Modified File Writer & CCP component requirements. | Neeta Jawale |
| 0.7 | 23 Dec 2021 | Modified and added requirements for Static Data Sampler and Client Manager Model. | Rohini Kokate |
| 1.0 | 06 Jan 2022 | Modified and added requirements for File Handling Model. | Vijay Kumar |
| 1.1 | 19 Jan 2022 | Modified and added requirements for Generic Application Software, Events Manager Model, Sampler Manager Model, XCP Sampler, Data Logger Model, Data Access Model. | Neeta Jawale |
| 19 Jan 2022 | Modified and added the requirements for ValidationModel | Vijay Kumar |
| 1.2 | 01 March 2022 | Removed CCP requirements | Neeta Jawale |

1. The ECU shall return all the parameter values in a given request as a single data block. Separate requests will return separate data blocks. [↑](#footnote-ref-1)