



## French C-ITS Deployment Coordination committee

# CataLog

---

### 2.4.1.3\_H

## Activity 2.4

Sub Activity: 2.4.1.3\_H

Version 0.3020

Publication Date: 29/12/202205/06/2023



Co-financed by the Connecting Europe  
Facility of the European Union

*The contents of this publication are the sole responsibility of the SCOOP@F project consortium, C-ROADS France project consortium, InterCor project consortium (French beneficiaries only) and InDiD project consortium and do not necessarily reflect the opinion of the European Union.*



## Information on the document

Document: 2.4.1.3.H CataLog

Responsible, Entity: Hasnaâ ANISS, Université Gustave Eiffel

## Distribution

Date	Version	Author(s)	Updates & changes	Diffusion
	0.01	Hasnaâ Aniss, IFSTTAR URCA		
11/12//2019	0.02	Hasnaâ Aniss, IFSTTAR	Editorial corrections	
27/01/2020	0.03	Hasnaâ Aniss, UGE	Editorial corrections	
19/11/2020	0.04	Hasnaâ Aniss, UGE	Add information on log files	
25/11/2020	0.05	Hasnaâ Aniss, UGE Antoine Foulquié, Viveris AMO DIT	Minor corrections	COCSIC
18/03/2021	0.06	Hasnaâ ANISS	Prise en compte des remarques du COCSIC de <del>Décembre</del> décembre 2020	
18/03/2021	0.10	Antoine Foulquié, Viveris AMO DIT	Document approved	COCSIC
21/09/2022	0.11	Hasnaâ Aniss	Add information on log files	Cocsic
15/11/2022	0.12	Hasnaâ Aniss	Add information on file name and annex	
23/11/2022	0.13	Hasnaâ Aniss	Added list of annexes	
29/12/2022	0.20	Antoine Foulquié	Document approved	COCSIC
25/04/2023	0.21	Hasnaâ ANISS	Expérimentation <del>ermet</del> CEREMA enlevé - <del>Doublons</del> Doublons de citation des annexes enlevés (suite de la prise en compte des remarques cocsic-études de janvier 2023)	COCSIC
<u>05/06/2023</u>	<u>0.30</u>	<u>Thiwiza BELLACHE</u>	<u>Document approved</u>	<u>COCSIC</u>



## Quality rules

### Reference to the version administration

Version number to be composed of 3 digits > vR.XY

- **R** corresponds to the release number : it is upgraded each time SC Studies validates the diffusion of a new release,
  - **X** is the major version number: it is upgraded each time SC Studies validates the deliverable,
  - **Y** is the minor version number: it is upgraded each time a contributor changes anything.
- Once the deliverable is approved, its version number is upgraded from vR.XY to vR.(X+1)0  
Once the deliverable is release, its version number is upgraded from vR.XY to v(R+1).00

As illustration :

- 0.03 > Work in progress version
- 0.10 > Del. Approved by SC Studies but not released
- 2.00 > Del. approved & released (in release 2)
- 2.05 > Del. Updated - in progress version

### Requirements identification & traceability

In this document, the following verbal forms are used to indicate requirements: **Shall / Shall not**

Recommendations shall be indicated by the verbal forms: **Should / Should not**

Permissions shall be indicated by the verbal forms: **May / May not**

Possibility and capability shall be indicated by the verbal forms: **Can / Cannot**

Inevitability used to describe behavior of systems beyond of the scope of this del. shall be indicated by: **Will / Will not**

Facts shall be indicated by the verbal forms: **Is / Is not**

In the table here below:

2.4.X.XX > is the number given to the deliverable (e.g. 2.4.4.8)

YYYY > for digit are given to identifying which component/entity the requirement is addressing (e.g. LTCA for long terme certificate authority)

ZZZ > is the numeration of the requirement

ID	2.4.X.XX-YYYY-ZZZ
Component(s)	(e.g) ITSS-VU, ITSS-VRO, ITSS-R, PKI
Requirement	(e.g) An ITS station <b>SHALL</b> be able to request and get a Long term Certificate (LTC) from the SCOOP Public Key Infrastructure (PKI).
Acceptance	(e.g) CA1 : ITSS-VU sends a LTC request to the LTCA CA2 : ITSS-R relays the LTC request CA3 : The LTCA verifies the request and sends a response CA4 : The ITSS-R relays the response CA5 : The response is received by the ITSS-VU and is valid
Additional information	

# Table of Contents

Quality rules .....	<u>54</u>
Table of Contents .....	<u>65</u>
1. Introduction .....	<u>87</u>
2. Baseline .....	<u>98</u>
3. Storage .....	<u>98</u>
4. Data collection .....	<u>98</u>
5. Message Format .....	<u>108</u>
6. ANNEX 1: Data definitions .....	<u>119</u>
6.1. Structure of log items .....	<u>119</u>
6.2. Layers .....	<u>119</u>
6.3. Communication Logging .....	<u>1240</u>
6.3.1. Definition of event models and actions .....	<u>1344</u>
6.3.2. Logging events and actions .....	<u>1442</u>
6.3.3. C-ITS messages .....	<u>1644</u>
6.4. Definition of Event Models .....	<u>1745</u>
6.5. Logging Events .....	<u>2047</u>
6.6. Logging EventActions .....	<u>2148</u>

## Tables

<a href="#"><u>Table 1: Mandatory data elements of log items</u></a>	8
<a href="#"><u>Table 2 : Mandatory and conditional data elements of communication log items</u></a>	9
<a href="#"><u>Table 3 – Mandatory data elements to define the EventModels for an EventType</u></a>	10
<a href="#"><u>Table 4 – Mandatory data elements to log detected events</u></a>	11
<a href="#"><u>Table 5 – Mandatory data elements to log eventactions per event type</u></a>	12
<a href="#"><u>Table 6 : ITS-G5 message data elements to uniquely identify a message</u></a>	12
<a href="#"><u>Table 7: Event Types for application and HMI logging</u></a>	13
<a href="#"><u>Table 8: DENM EventModels of event models and event actions for the DENM event type</u></a>	13
<a href="#"><u>Table 9: SECURITY EventModels of event models for the SECURITY event type</u></a>	14
<a href="#"><u>Table 10: Mandatory data elements to define detected events per event type</u></a>	15

## Table of figures

<a href="#"><u>Figure 1 :Vehicle ITS-Station simplified architecture</u></a>	9
<a href="#"><u>Figure 2: Road side ITS-Station simplified architecture</u></a>	9
<a href="#"><u>Figure 3: RWW example of events on a time line</u></a>	10

## 1. Introduction

This document is a catalog which provides the requirement of data logging for: ~~following TLOG.~~

- CAM
  - DENM
  - IVIM
  - MAPEM
- TLM
- SPATEM
  - POIM

They are based on log data defined in InterCor and C-Mobile.

~~These logs~~ These logs are defined for ITSG5 and cellular communications for all C-ITS stations and for the smartphone.

Communication of other message standards, such as DATEX2, can also be logged in similar schemes, but these standards are not included in this document.

Log data is provided as log items directly in the form of a database (dump or CSV) or in one of the file formats that can be converted into the same database structure (ASN1 Upper). Directly in database format is preferred.

A log item consists of meta data and the standardized message contents.

Excel files for logs are provided in annex:

Annex	Comments
<u>2.4.1.3_H_LogFormat_Communication_Vehicle_v-0.10</u>	Data at lower layer for V-ITS-S
<u>2.4.1.3_H_LogFormat_Communication_Coopits_v-0.10</u>	Data at lower layer for Coopits
<u>2.4.1.3_H_LogFormat_Communication_R-ITS-S_v-0.10</u>	Data at lower layer for R-ITS-S
<u>2.4.1.3_H_LogFormat_Communication_Serveurs_v-0.10</u>	Data at lower layer for servers
<u>COCSIC_LogFormat_Application_CITS_v-0.10</u>	Data definition at upper layer for all experimentation

~~COCSIC\_LogFormat\_Application\_CITS.xlsx (Nota: All logs from CAM event and CAM action are not used until UC with CAM is developed)~~

~~COCSIC\_LogFormat\_Communication.xlsx~~

~~For each message sent or received, the two format are to be filled in for V-ITS-S, R-ITS-S and smartphones.~~

~~This document is not applicable for all equipment even using SCOOP log format (TLog and ULog).~~

~~For each message sent or received, the two format are to be filled in for V-ITS-S, R-ITS-S and smartphones.~~

~~This document is applicable for all equipment.~~

~~for DENM and CAM but is applicable for equipment sending or receiving IVIM, SPATEM and MAPEM,~~

### Organisation of logging by C-ITS

V-ITS-S, R-ITS-S and smartphone application shall log all bold data of the annex



Nap-Serv, Nfr-ITS-S shall log all data send and received from COCSIC\_LogFormat\_Communication.xlsx. No need to log data described in COCSIC\_LogFormat\_Application.xlsx

Contents of communicated messages need only be logged once, if:

- messages are uniquely identified and traceable, and
- the consistency of message contents has been verified/validated, and availability of the single source of logging is ensured

For example, if the relevant DENM contents is logged by the originator (e.g. RITS-S) then receivers (e.g. V-ITS-S) may log only the identifying fields in bold upon reception.

## 2. Baseline

In order to be able to define if C-ITS system bring novelties in driving action, a baseline is required. This means that the system is ON and monitored but no information is giving to drivers. ~~Baseline process is still under discussion. Some ideas are to have a baseline done by COCSIC partners during 3 months. They will have the application without any message display. We ask them to turn on the app each time they drive.~~

We point out the fact that to be efficient and relevant, the baseline has to be enough populated and in theoretical context should represent half of the studied population. The way to process the baseline strongly affect the performance of the upcoming analysis.

## 3. Storage

Université Gustave Eiffel rents an OVH server so each ~~V-ITS-S~~, smartphones and servers can send its log. ~~The frequency has to be high so there is no loss of information.~~  
Connection information to OVH servers is provided to all actors by bilateral discussions.

## 4. Data collection

Data are collected:

- every second during an event (HMI display) – 30 seconds before and 30 seconds after for application Log
- every message sent or received for communication log.

~~In annex, several files described data collected for Coopits, for Ro-ITS-S, servers and Cerema experimentation. See Annex 2: Data files~~  
Annex 2: Data files for the detailed list of these files.

~~For Cerema experimentation on driver's behavior using coopits on their smartphone, following data element are mandatory in i-viaction and DENM action sheet:~~

1. ~~These data element are used only for this experimentation.~~

## 5. Message Format

Log will be sent using binary file. One file can contain several messages. Each station can have its own rule regarding the file size but cannot exceed 24h of logging.

File shall have one line per message logged.

A file is dedicated to an ego station and a LogType.

It's not useful to generate an empty file. Only file with data logged has to be sent to the server.

Only csv files (not zipped files) has to be sent to the OVH server.

Name of files ~~shall~~can be like:

Timestamp\_Type\_StationID\_LogType

### Where

Type: string

Unavailable (0)

V-ITS-S (1)

Smartphone (2)

App-serveur (3)

Nfr-ITS-S (4)

R-ITS-S (5), ...

Timestamp: AnneeMoisJourheureminsecondemilliseconde

LogType:

ApplicationLog (A)

CommunicationLog (C)

The naming is not mandatory as it is explained previously but the file name shall be unique.

Example: iviaction\_182913802\_20200921T085201.csv

The file name shall be unique.

## 6. ANNEX 1: Data definitions

Extract from InterCor\_CommonLogFormatDescription deliverable from WP 4 evaluation lead By Bart Netten from TNO.

### 6.1. Structure of log items

This section defines the structure of log items needed to organize and manage the data from multiple applications, stations, use cases and pilots. This structure is independent of the organization of log data in files, experiments, test sessions or test units.

Every log item shall be prefixed with the data elements in [table 1](#)~~Erreur ! Source du renvoi introuvable.~~. The prefix "log\_" is used to denote that the data element is added to the log item contents.

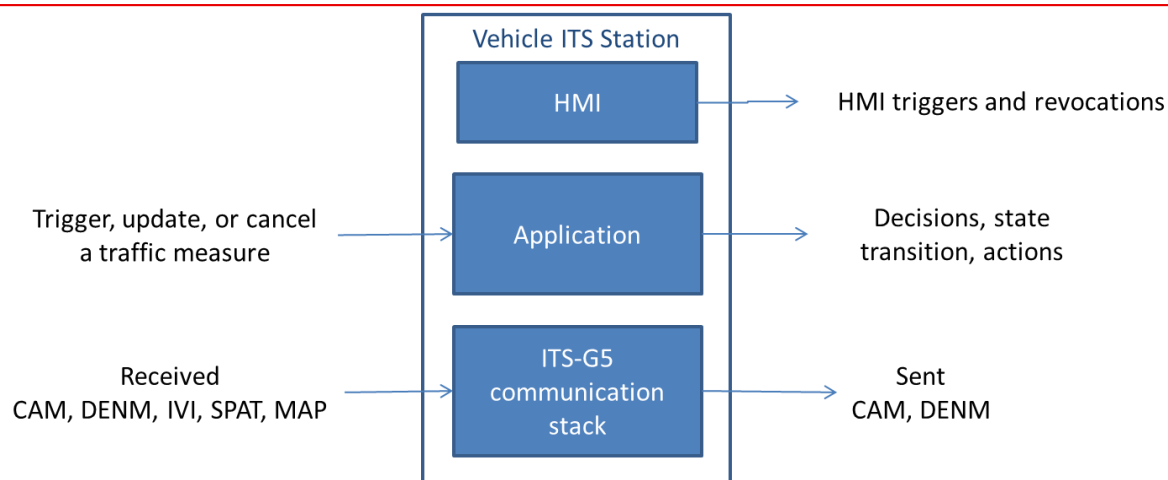
**Table 1: Mandatory data elements of log items**

Data element	Description
<b>log_stationid</b>	Identifier of the log station that logs the log item. The log_stationid should be unique within the project.
<b>log_applicationid</b>	Identifier of the log application (in the log station) that logs the log item. The log_applicationid is at least unique within the log station (log_stationid).
<b>log_timestamp</b>	Timestamp at which the log application logs the log item.
<b>log_action</b>	Enumerated value identifying the action in the data flow to, in or from the application unit (log_applicationid): { SENT, RECEIVED, ...}

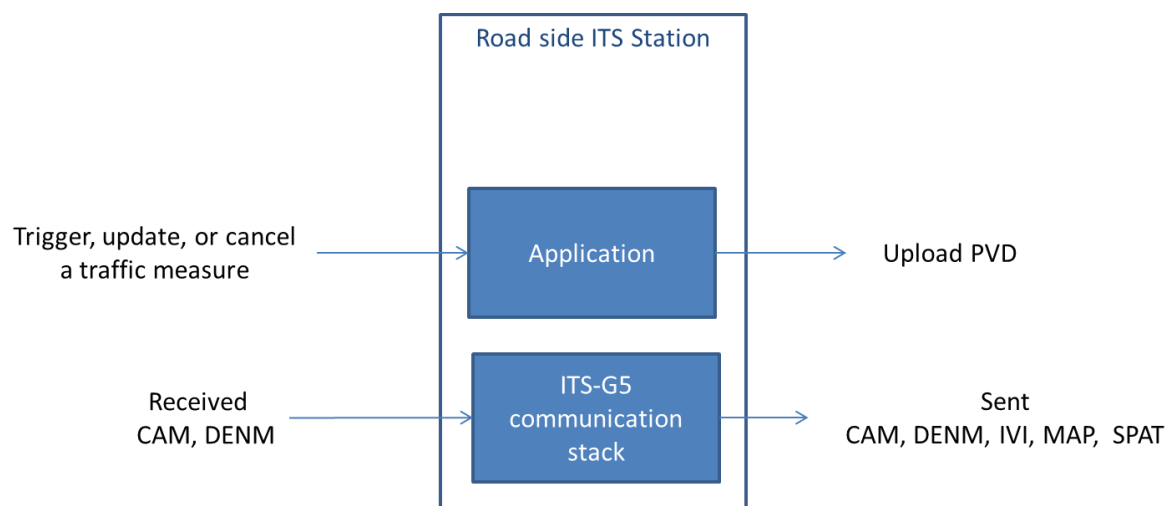
### 6.2. Layers

Logging is also structured in layers for communication, application logic and HMI events as sketched in Figure 1 and Figure 2. At the communication layer, logging is collected on the contents and timing of messages that are sent and received. At the application layer, events are logged on the application logic such as decisions, state transitions, control or advice actions. At the HMI level, presentation and revocation events on the driver or device display are logged.

The layered approach enables the selection of specific layers for logging and analysis in function of the objectives for testing, verification, validation or evaluations. The selection may be refined by message type or service. A project may decide for example to collect only logging for piloting and evaluation at the application and HMI layers, and only for specific services, and switch ~~off~~ the logging of communication that was used during verification and validation TESTFESTs.



**Figure 1 :Vehicle ITS-Station simplified architecture**



**Figure 2: Road side ITS-Station simplified architecture**

## 6.3.Communication Logging

Communication Logging is the logging of the messages that are sent or received by a station via any communication medium, path or channel.

The log\_action label in the log item, defined in Table 1, denotes whether the message was 'SENT' or 'RECEIVED'.

An alternative communication channel may be used to communicate the same messages. The log item header as defined in [table 1](#) ~~Erreur ! Source du renvoi introuvable.~~ should be extended with the data element from Table 2 to identify the communication channel used. In parallel the log\_applicationid may also differ per communication channel.

**Table 2 : Mandatory and conditional data elements of communication log items**

Data element	Description
<b>Data elements from Table 1: Mandatory data elements of log items</b>	
<b>log_communicationprofile</b>	Identifier of the communication channel or profile used: {ITS_G5, CELLULAR, UWB, LTE_V2X, ...}
<b>log_messagetype</b>	Type of standardised message. The enum fields define to the <standardisation organisation>.<message type>.
<b>log_messageuuid</b>	Universal Unique Identifier of the message. This is an alternative for the identification of messages from the message contents.

### 6.3.1. Definition of event models and actions

The functions and services can be regarded as state machines or event models. The applications and HMI implementing the functions and services receive or trigger external events, such as the inputs from sensors and communication, or internal events such as the decisions from internal application components about geographical relevance or severity of environmental objects. The events can be logically grouped by use case, service, communication protocol or interaction protocol. In this document, such a logical group of events is called a type of events or **Event Type**. Examples are given for the processing of DENM and IVI messages in section 6.4.

Per event type, several external and internal events can be distinguished and characteristic for the functionality and performance of the implemented applications. Figure 3 gives an example of event models for the event type for a road works warning service in a Vehicle ITS-Station (VIS):

1. Traffic Control Centre or Central Unit sets a RWW traffic measure, for which a series of road side units generate DENM and IVI messages. A VIS passing the road side units receives the messages for a period of time (vertical arrows). At this level, the event model described the processing of the series of repeated DENM or IVI messages, where each reception is an event.
2. During this period, the vehicle application triggers events when entering the relevance area and the awareness area for the road works.
3. For another period of time, a road works warning is triggered and later revoked on the HMI.

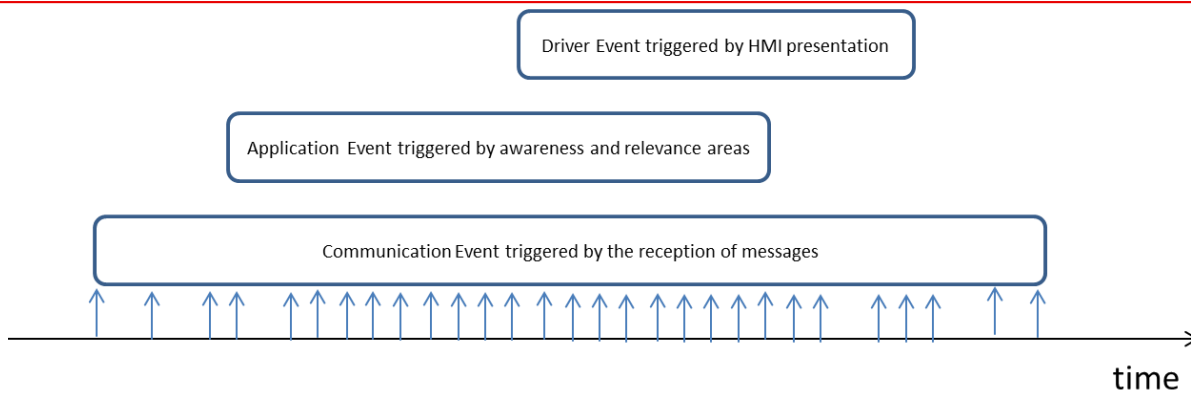


Figure 3: RWW example of events on a time line

Table 3 – Mandatory data elements to define the EventModels for an EventType

Name	Description
<b>eventmodelid</b>	unique identifier (enum) of the EventModel within the set of models for the EventType
<b>eventactionid</b>	unique identifier (enum) of the EventAction within the EventModel

The events per event type are modelled in so called **Event Models** using the two mandatory identifiers defined in Table 3. Every event model has a unique **eventmodelid** within the event type. Choosing an event model returns the design of a template with pre-allocated set of available actions, ie it designs the message according to the set of expected actions. Within an event model, multiple actions can be taken. Event model 1 can be modelled with actions for sending messages, and classifications of received messages, as triggers for new messages, reception of repeated messages, updates and terminations of the sequence of messages. Event model 2 can distinguish actions for determining the relevance of the host position relative to the event position of the road works, as well as the matching on the trace and event history, or the detection and relevance zones of an IVI. An event model can be defined by a list of relevant actions, called **Event Actions**. Every event action has a unique **eventactionid** within the event model; i.e. every <eventtypeid, eventmodelid, eventactionid> uniquely defines the action within an event type.

Without considering the implementation and hierarchy of applications, or the nesting of state machines, the relevant actions can be grouped into sets of actions that are logically or causally related to a specific event models per event type. The relationships need not be modelled to evaluate the occurrence, ordering or timing of the actions taken by a specific implementation.

### 6.3.2. Logging events and actions

This section describes the dynamics of logging events using the models defined in the previous section.

Every occurrence or detection of a specific type of events is considered as an **event**. In the example of Figure 3 above, the RWW event is first detected when the first DENM or IVI is received by the vehicle ITS-Station. The event should be defined and logged upon reception of the message. All subsequent DENMs received for the same RWW belong to the same event.

Logging events is done in two activities:

1. First the station detects, defines and logs the event with the mandatory parameters from Table 4.
2. Then the station logs every eventaction with reference to the related event.

**Table 4 – Mandatory data elements to log detected events**

Name	Description
<b>eventtypeid</b>	eventtypeid
<b>eventid</b>	unique id defined by the stationid. Note that this cannot be specific to an application within a station
<b>log_action</b>	Action to detect the event that triggered the logging event. (Enum: 'SENT', 'RECEIVED')

Upon detection of a new event in step 1, the station or application should generate a unique identifier for the event: the **eventid**. The station should ensure that every eventid is unique within the logging of the station and its applications. Preferably, eventids are generated as a monotonic sequence.

The station or application should also detect the type of event, e.g. from the message type of the received message. Additional parameters that identify the event from external information is type specific. The data that need to be logged to uniquely identify the event is specific for the event type (see section 6.5).

After step 1, applications log every action related to the event with the mandatory fields of Table 5. The action is called '**eventaction**' to identify that the action is associated to a specific eventmodel. An eventaction is identified by the <eventmodelid, eventactionid> of the event type as defined in Table 3.

The purpose of the eventid is to ~~trace-track~~ all actions across all applications within the station that are related to a single event. Eventactions should be uniquely associated to the single event by the eventid. The eventid should be passed on to all relevant applications that log actions related to this event.

The action is logged with the eventid, eventmodelid and eventactionid. Additional parameters to quantify the action can also be logged with the action.

**Table 5 – Mandatory data elements to log eventactions per event type**

Name	Description
------	-------------

<b>eventid</b>	unique id defined by the station for the related event.
<b>eventmodelid</b>	id of the event model from the EventModels sheet, e.g. as defined in Table 8: DENM EventModels of event models and event actions for the DENM event type
<b>eventactionid</b>	id of the event action from the EventModels sheet, e.g. as defined in Table 8: DENM EventModels of event models and event actions for the DENM event type

### 6.3.3. C-ITS messages

the data elements to uniquely identify messages in the logging are used as defined in Table 6. These data elements are always mandatory for all sender and receiver stations.

**Table 6 : ITS-G5 message data elements to uniquely identify a message**

Message Type	Data elements
<b>CAM</b>	CAM.stationid
	CAM.generationdeltatime
	Generationtimestamp of the CAM, converted by the logging station from the CAM.generationdeltatime in TAI
<b>DENM</b>	DENM.originatingstationid
	DENM.sequencenumber
	DENM.referencetime in TAI
<b>IVI</b>	IVI.stationid
	IVI.serviceprovideridentifier
	IVI.iviidentificationnumber
	IVI.timestamp in TAI



<b>MAP</b>	<p>ItsPduHeader.stationid of the sender  MapData.timestamp (if available) in UTC  MapData.intersections[1].id.region  MapData.intersections[1].id.id  MapData.intersections[1].revision  <i>When multiple intersections are specified in a MAP message, each intersection should be logged separately.</i></p>
<b>SPAT</b>	<p>ItsPduHeader.stationid of the sender  SPAT.timestamp (if available) in UTC  SPAT.intersections[1].id.region  SPAT.intersections[1].id.id  SPAT.intersections[1].revision  SPAT.intersections[1].moy and SPAT.intersections[1].timestamp (if available) in UTC  <i>When multiple intersections are specified in a SPAT message, each intersection should be logged separately.</i></p>

Table 7: Event Types for application and HMI logging

Event Type	eventtypeid	Description
<b>DENM</b>	2002	all events and actions related to the processing of DENM for safety and hazard services
<b>IVI</b>	2006	all events and actions related to IVI messages and in-vehicle signage services
<b>TLM</b>	2004	all events and actions related to MAP and SPAT messages and controlled intersection services
<b>SECURITY</b>	35	all events and actions related to the communication security and PKI such as signing and verification of certificates.

## 6.4. Definition of Event Models

The actions that can be logged for an event type are defined by one or more models of events, or 'event models' according to Table 3. An event model defines the relevant actions of an aggregated application processes, a decision model, interaction protocol or state machine.

Multiple event models could be distinguished in each of the application and HMI layers. Examples of event models that can be distinguished are the processing of a received message, the classification of the relevance of the received message, requesting warnings and information for display on the HMI, and presentation or revocation of information on the HMI display. HMI and application events are closely related within a service, and are therefore defined as event models within the same event type.

Event models for the C-ITS event types are defined in Annex (excel file called COCSIS\_Logformat application ) in tables called **<TYPE> EventModels**, where <TYPE> is the name of the event type.

For the DENM event type, the eventmodels and actions are defined on sheet "**DENM EventModels**" (Table 8). The eventmodels are identified by a unique **eventmodelid**. The eventactions within every eventmodel are identified by an **eventactionid** that is unique within the eventmodel.

**Table 8: DENM EventModels of event models and event actions for the DENM event type**

event model id	Event Model description	event action id	Event Action description
<b>1</b>	Generating DENM	<b>1</b>	DENM trigger
		<b>2</b>	DENM update
		<b>3</b>	DENM cancelation
		<b>4</b>	DENM negation
<b>2</b>	Receiving DENM	<b>1</b>	DENM trigger
		<b>2</b>	DENM update
		<b>3</b>	DENM cancelation
		<b>4</b>	DENM negation
<b>3</b>	Relevance of received DENM	<b>1</b>	relevant = assumes all of the following criteria are checked as relevant
		<b>2</b>	not relevant: not in time validity duration
		<b>3</b>	not relevant: not in relevance traffic direction
		<b>4</b>	not relevant: cause code will not be processed; e.g. not implemented or recognised
		<b>5</b>	not relevant: sub cause code will not be processed; e.g. not implemented or recognised
<b>4</b>	Matching of received DENM	<b>1</b>	not in awareness or relevance distance yet from event, trace or event history
		<b>2</b>	in relevance distance of event, trace or event history, while off trace and off event history path

		<b>3</b>	on trace
		<b>4</b>	on event history path
<b>5</b>	Awareness of received DENM	<b>1</b>	trigger presentation request to HMI
		<b>2</b>	trigger update request to HMI (only specify the updated information)
		<b>3</b>	trigger revocation request to HMI
<b>6</b>	Presentation on HMI of received information	<b>1</b>	first presented on HMI
		<b>2</b>	updated on HMI
		<b>3</b>	revoked from HMI

Event models and actions for the IVI and TLM event types are defined similarly on sheets "IVI EventModels" and "TLM EventModels" respectively.

Note that the current event types for DENM, IVI and TLM have separate event models for the application requests to presentation of information to the HMI and for the confirmation of the HMI display unit that the information is actually presented or revoked. This may be relevant if the HMI is not integrated with the applications in a single unit and if the HMI unit has its own internal logic to prioritise, organise and present information on the display. Consequently an application request may not always be executed and presented to the driver.

Table 9 shows the event models for the SECURITY event type which are defined on sheets "SECURITY EventModels" and "securityaction". The event models are different from the DENM, IVI and TLM types because of the nature of the type of events.

**Table 9: SECURITY EventModels of event models for the SECURITY event type**

eventmodel id	Event Model description	event action id	Event Action description
<b>1</b>	Sender assigns or changes a pseudonym certificate	<b>1</b>	assign GN address
		<b>2</b>	assign or change an Authorisation Ticket (AT)
<b>2</b>	Sender signs a message	<b>1</b>	message is not signed
		<b>2</b>	failed signature creation
		<b>3</b>	successful signature creation

3	Presentation on HMI of received information	1	validation SUCCESS and payload delivered
		2	validation FAILED but payload delivered (non-strict verification)
		3	validation FAILED and payload discarded

The sheets and models can be easily adapted and extended for other applications by adapting or extending the lists on the sheets.

Note: at the beginning of the project No need to log actions in CAM Event Model for 'regular' CAMs without any event information. When UC with CAM

## 6.5. Logging Events

This section describes how a station or application should generate logging for detected events as defined in section 6.3.2.

Upon detection of a new event, the station or application should generate a unique identifier for the event: the **eventid**. The station or application should also detect the type of event, e.g. from the message type of the receive message. A new event must be logged with the mandatory parameters from Table 3; i.e. the newly generated eventId, eventtypeid, and additional parameters that uniquely identify the event as an instance of a particular event type.

**Additional parameters that identify the event from external information are type specific. The data that need to be logged to uniquely identify the event is specific for the event type. Table 10 defines the mandatory items to log in addition to Table 3 for each event type. The items are similar to those in the communication logging (Table 2 : Mandatory and conditional data elements of communication log items). Note that timestamp information of the messages is not mandatory as an event can span the life time covering all repetitions, updates and negations of messages.**

**Table 10: Mandatory data elements to define detected events per event type**

Event Type	Data elements
<b>DENM</b>	DENM.originatingstationid
	DENM.sequencenumber
<b>IVI</b>	IVI.stationid
	IVI.serviceprovideridentifier
	IVI.iviidentificationnumber

<b>TLM</b>	ItsPduHeader.stationid of a MAP or SPAT
	MapData.intersection.id
	MapData.intersection.revision
<b>SECURITY</b>	ITS Application ID (ITS_AID) or the BTP destination port of the message type to be signed or verified

## 6.6.Logging EventActions

This section describes how a station or application should generate logging for eventactions.

Eventactions must be logged with the mandatory parameters from Table 5. An eventaction can only be logged if the eventid is received as logged in section 6.5. Every action related to the same event should be logged with the eventid. Eventactions must be logged with the eventmodelid and the eventactionid. Additional parameters can be logged for specific event actions.

## **7. Annex 2: Data files**

~~COCSIC\_LogFormat\_Application\_CITS.xlsx~~

~~COCSIC\_LogFormat\_Communication.xlsx~~