

EUCARIS

XML Message Specification Mileage

Drawn up by:	Sjaak Kempe, Business Consultant & Information Analyst, RDW Netherlands.
Date:	28-04-2015
Version:	8.0.0
Status:	Final

Document control

Version	Date	Remarks
0.1	22-04-2009	First Draft.
0.2	05-05-2009	Changes after review RDW.
0.3	22-07-2009	More changes after review RDW.
0.4	06-08-2009	Changes after meeting with Belgian organisation responsible for mileages recordings.
0.5	25-02-2010	Adaptation because of mileage recording project in NL, besides recording date also recording time.
0.6	14-04-2010	Change in request message: Include the mileage that was read from the odometer. Error in structure of response message corrected.
0.7	25-05-2010	Remark from RDW test team, error in structure of response message corrected.
0.8	03-06-2010	Remark from RDW end users. In some of the processes the mileage query is to used, the current mileage from the vehicle is not available. Change: current mileage optional instead of mandatory.
0.9	20-10-2010	Functional test stage, defect 137: Mismatch between XML message specs and XSD, concerning MileageUnityDesc, corrected.
1.0	26-10-2010	Functional test stage, defect 98: Table of MileageRecordings optional, since it is possible that there are mileage recordings, but none of them is useful; Addition of element stating the reliability of a mileage recordings table.
6.0	05-01-2011	Final version Eucaris release 6.0
7.0.0	16-10-2014	A few small errors corrected. Adapted to 2014 standards. Final release version
7.0.1	11-03-2015	Nesting change MileageReliability
8.0.0	28-04-2015	Final release version

Related Documentation

ID	Version	Date	Name
[DOC-1]		29-04-1999	Council Directive 1999/37/EC on the registration documents for vehicles

Contents

1. INTRODUCTION	4
1.1 GENERAL	4
1.2 DEFINITIONS.....	4
2. MILEAGE QUERY USING EUCARIS	5
2.1 DESCRIPTION OF A MILEAGE QUERY	5
2.2 ASSUMPTIONS AND REQUIREMENTS	5
3. MILEAGE MESSAGE	7
3.2 HEADER	8
3.3 REQUEST MESSAGE	9
3.4 RESPONSE MESSAGE	9
4. ANNEX A – NODES AND ELEMENTS	11
5. ANNEX B – DEFINED MESSAGES	16
5.1 SYSTEM MESSAGES	16
5.2 APPLICATION MESSAGES	16
5.3 MISCELLANEOUS MESSAGES.....	16
6. ANNEX C – XSD VALIDATION	18
7. ANNEX D – COUNTRY CODES	19
7.1 EUCARIS COUNTRY CODE CONVENTION	19
7.2 CODING SYSTEMS.....	19
7.3 ARTIFICIAL COUNTRY CODES	20

1. Introduction

1.1 General

Trigger for a mileage query is the import of a car into a Eucaris Member State, coming from another Eucaris Member State.

Using Eucaris, and using the message described in this document, a Member State queries mileage records in another Member State. The input for the query is the Vehicle Identification Number (VIN). The query yields basic vehicle data, and a table with recorded mileages.

1.2 Definitions

MS Import	Member State of Import. The Member State that wants to know the mileage of a car because of import.
MS Export	Member State of Export. The Member State in which the imported car was previously registered, also the Member State where registration of recorded mileages is queried.

2. Mileage query using Eucaris

When a car is imported into a Eucaris member state (MS Import), it is possible to query recorded car mileages in the former Member State – MS Export, when the following requirements are met:

- There is a legal framework allowing mileage queries;
- In the former country, a mileage registration exists;
- The former country of registration is a Eucaris Member State.

There are two reasons for MS Import to query recorded mileages in MS Export:

- Check if the mileage on the odometer is consistent with recorded mileages, as a countermeasure against odometer rollover;
- Use the recorded data in MS Export as a starting point for further mileage recording in MS Import, the state where the car now will be registered.

Some Member States issue a mileage certificate. In such a case, the query to MS Export can be used to copy previous mileage recordings to the mileage registration in MS Import, in order to print all mileage recordings for the particular vehicle on the mileage certificate.

2.1 Description of a mileage query

A mileage query consists of the following steps:

- A car is imported into MS Import; The car was previously registered in another Eucaris Member State.
- Using Eucaris, the national Registration Authority sends a mileage query to the Registration Authority of the Export MS;
- MS Export receives the query and organises a query to its national register of mileage recordings to provide an answer;
- The answer to the query is sent to and received by MS Import;
- If the received mileage recordings suggest odometer rollover, measures can be taken in MS Import;
- Possibly, the national Registration Authority itself is not the organisation responsible for checking and/or registering mileages. In that case, the Registration Authority uses the query result to send a notification to the organisation responsible for recording mileages;
- If applicable, recorded mileages in export MS will be printed on a mileage certificate, issued by import MS.

2.2 Assumptions and requirements

In designing the Eucaris Mileage message, the following assumptions were made:

- Since the query is associated with the event of a car being imported, the Vehicle Identification Number of the vehicle will be known;
- There is no need for a Multi Country Inquiry, since the origin of the car will be determined during import, the target country for the mileage query will be known;
- There is no need for a multiple request; one request involves one VIN of one specific vehicle, meant to be imported;
- The communication between organisation responsible for mileage recordings and national Registration Authority (transport medium and format) should be developed nationally by MS Import;
- If the national format differs from the Eucaris format, Member States are responsible for conversions;
- MS Export should develop a gateway between Eucaris and the legacy register that stores mileages;
- Eucaris mileage exchange is interactive, a request sent by MS import is immediately answered by MS Export;
- It is possible that for mileage recordings, one Eucaris country uses miles and another uses kilometres. The query does not convert kilometres to miles and vv. In the message, MS Export mentions the unity of the mileages.

- A successful query consists of a table with recorded mileages. Where possible, basic vehicle data are provided to allow identification of the vehicle (and reveal query input errors).
- A query on VIN might yield multiple hits. The message supports multiple hits.

3. Mileage message

3.1.1 Used conventions

The tables used to describe the XML messages provide the following information:

- NestingLevel
- Item
- Occ (occurrence)
- Type
- Remarks

3.1.2 Nesting level

Describes how the XML nodes and elements are nested in the message.

3.1.3 Item

The first column indicates the item name:

- An ***XML node*** is indicated in bold & italic
- An element is indicated by a normal appearance

In Annex A, for every node and element, its functional definition and its use is described.

3.1.4 Occ

The second column indicates the occurrence of the node or element.

Value	Description
1	A mandatory item.
0-1	An optional item but if present, the item must be unique.
0-n	An optional item. When present, it may appear more than once.
1-n	A mandatory item. The item may also appear more than once

Remark:

If an optional item is absent, leave it out of the message entirely (do not send empty tags).

3.1.5 Type

This column indicates the data type of the element.

Type	Description
Text	A sequence of characters.
DT	Date and Time. Can be given in UTC format (Co-ordinated Universal Time) as 'CCYY-MM-DDThh:mm:ssZ' or 'CCYY-MM-DDThh:mm:ss±00.00', or can be given in local time with the offset to UTC as 'CCYY-MM-DDThh:mm:ssZ±nn.nn'. For more information see http://en.wikipedia.org/wiki/ISO_8601 .
Date	Date. Can be given as a date in format 'CCYY-MM-DD', or as a date with a time zone denomination, i.e. UTC 'CCYY-MM-DDZ' or local time with the offset to UTC as 'CCYY-MM-DD±nn.nn'.

Type	Description
DateInc	Date that can be incomplete. Format 'CCYYMMDD'. See Annex A for further information
Int	Numeric, integer values only.
Dec	Numeric, fractional numbers possible.
Boolean	An element that either has a 'true' value or a 'false' value.
Choice	Allows one and only one of the nodes or elements contained in the selected group to be present within the containing node (exclusive choice).
Enum	Enumeration; the element has a specified set of values. The possible values are described in Annex A. In XSD validation, the value of the element in the message will be checked against the possible values.
UUID	Universally Unique Identifier, version 4. For more information see http://en.wikipedia.org/wiki/Universally_Unique_Identifier and http://en.wikipedia.org/wiki/Universally_Unique_Identifier#Implementations

3.1.6 Remarks

Contains information about situations in which a node or element applies or not. Also contains information about relations between separate elements. Note: The functional definition and the use of each node or element is described in Annex A.

3.2 Header

Nesting Level	Item	Occ	Type	Remarks
1	Header	1		
2	MessageID	1	UUID	
2	MessageRefID	0-1	UUID	
2	MessageVersion	1	Text	Request message: Use the MessageVersion that is supported by MS Import. Response message: Use the MessageVersion that the legacy system of MS Export supports. After downgrading: Change MessageVersion to the version number of the downgraded message. Default value 1.0, subsequent versions will be 2.0, 3.0 etc.
2	ServiceExecutionReason	1		Provided in the request message, returned unaltered in the response message
3	ServiceExecutionReasonCode	1	Enum	Fixed value
3	ServiceExecutionReasonDesc	1	Text	Fixed value
2	ServiceFileNumber	0-1	Text	If present in the request message, it has to be returned in the response message, unaltered.
2	RecipientCountry	1	Enum	Country receiving the message. In a response message, the sender/recipient countries are reversed, compared to the original request.
2	SenderCountry	1	Enum	Country sending the message. In a response message, the sender/recipient countries are reversed, compared to the original request.
2	SenderOrganisation	1		Provided in the request message, returned unaltered in the response message
3	SenderOrganisationCode	1	Enum	
3	SenderOrganisationDesc	1	Text	Enum

Nesting Level	Item	Occ	Type	Remarks
2	SenderName	1	Text	Provided in the request message, returned unaltered in the response message
2	SenderOrganisationName	0-1	Text	If present in the request message, it has to be returned in the response message, unaltered.
2	TimeStamp	1	DT	Creation date and time in UTC
2	TimeOut	1	Int	Is not used. Provide value 0.

3.3 Request message

The following message describes the request for mileage recordings by Vehicle Identification Number (VIN)

Nesting Level	Item	Occ	Type	Remarks
2	MileageReqByChassis	1		
2	Header	1		
2	Body	1		
3	Request	1		
4	VehIdentificationNumberReq	1	Text	
4	CurrentMileage	0-1	Int	
4	CurrentMileageUnity	0-1	Text	

3.4 Response message

The following message describes the response to the request for mileage recordings.

Nesting Level	Item	Occ	Type	Remarks
1	MileageResponse	1		
2	Header	1		
2	Body	1		
3	BodyInformationReply	0-1	Choice	This node is present when the query yielded information.
4	VehIdentificationNumber	1	Text	= VehIdentificationNumberReq
4	MileageReplies	1		This node contains 1 mileage reply for each vehicle found.
5	MileageReply	1-n		The node occurs for every vehicle found. It contains recorded mileages (MileageRecordings) and, optionally, vehicle information (Vehicle)
6	Vehicle	0-1		This node contains basis vehicle data. It is possible to leave out this node, but note that a reply containing multiple hits should also contain vehicle data, so that the Import MS can choose the right vehicle.
7	VehRegistrationNumber	0-1		
8	VehRegistrationNumberPart1	1	Text	
8	VehRegistrationNumberPart2	0-1	Text	
8	VehRegistrationNumberPart3	0-1	Text	

Nesting Level	Item	Occ	Type	Remarks
8	VehRegistrationNumberPart4	0-1	Text	
7	VehMake	0-1	Text	
7	VehCommercialName	0-1	Text	
7	VehFirstRegistrationDate	0-1	DateInc	
7	VehStartRegistrationDate	0-1	DateInc	
7	VehEndRegistrationDate	0-1	DateInc	
6	<i>MileageRecordings</i>	0-1		This node contains the recorded mileages
7	<i>MileageRecording</i>	1-n		This node occurs for each individual recording. One recording consists of either a <i>Mileage</i> (possibly together with a status) or a status (<i>MileageRecordingStatusCode</i> + <i>MileageRecordingStatusDesc</i>).
8	MileageRecordingDate	1	DateInc	
8	MileageRecordingTime	0-1	Int	
8	Mileage	1	Int	
8	<i>MileageUnity</i>	0-1		
9	MileageUnityCode	1	Enum	
9	MileageUnityDesc	1	Enum	
8	<i>MileageStatus</i>	0-1		
9	MileageStatusCode	1	Enum	
9	MileageStatusDesc	1	Enum	
8	MileageRecordingOrganisation	0-1	Text	
8	<i>MileageRecordingCountry</i>	0-1		
9	MileageRecordingCountryCode	1	Enum	
9	MileageRecordingCountryDesc	1	Text	
7	<i>MileageRecordingReliability</i>	0-1		
8	MileageRecordingReliabilityCode	1	Enum	
8	MileageRecordingReliabilityDesc	1	Enum	
3	<i>BodyMessageReply</i>	0-1	Choice	This node is present when the query yielded no information or an error situation occurred.
4	<i>Messages</i>	1		This node contains 1 or more explanatory messages.
5	<i>Message</i>	1-n		This node contains 1 individual message.
6	MessageTypeCode	1	Enum	
6	MessageTypeDesc	1	Enum	
6	MessageClassCode	1	Enum	
6	MessageClassDesc	1	Enum	
6	MessageCode	1	Enum	
6	MessageDesc	1	Enum	
6	MessageVariable	0-1	Text	

4. Annex A – Nodes and elements

In Alphabetical order this annex describes in detail all the nodes and elements used in the messages. The following information is provided:

- Type
- Len
- Description

Item	Type	Len	Description
Body			The body contains all the nodes and elements of the actual request, reply or message.
BodyInformationReply			This part of the message is used when the body consists of information, i.e. the mileage query was successful.
BodyMessageReply			This part of the message is used when the response to the request consists of an informational or an error message.
CurrentMileage	Int	1-7	Contains the mileage from the odometer, of the vehicle that is queried for mileages. The value is input in the request message, by the user making the request.
CurrentMileageUnity	Enum		Is used in combination with CurrentMileage. Possible values: km (kilometre); m (mile); xx (not specified).
Header			The header is used to control the process flow. The header is sent with each message between client and server.
Message			This node contains one single message. A Message always consist of the following elements: MessageTypeCode MessageTypeDesc, MessageClassCode MessageClassDesc, MessageCode MessageDesc. The element MessageVariable is optional.
Messages			This node is used to inform the user/client system for errors or other informational issues. If present, the node contains one or more <i>Message</i>
MessageClassCode	Text	1-5	Contains the classification of a message. Always used in combination with MessageClassDesc. ERR = ERROR INF = INFORMATION WARN = WARNING
MessageClassDesc	Text	1-50	Contains the description. Always used in combination with MessageClassCode. See for the notation MessageClassCode.
MessageCode	Int	1-5	Contains the message code related to the message. Always used in combination with MessageDesc. See Annex B for the possible values.
MessageDesc	Text	1-25	Contains the description. Always used in combination with MessageCode and MessageTypeCode.
MessageID	UUID	36	A unique identifier for this message.

Item	Type	Len	Description
MessageRefID	UUID	36	Only applicable to response messages. The element contains the MessageId of the request message, so that request and response message can be correlated.
MessageTypeCode	Text	1-5	Indicates the type of message. Always used in combination with MessageTypeDesc. SYST = SYSTEM MESSAGE APPL = APPLICATION MESSAGE MISC = MISCELLANEOUS MESSAGE
MessageTypeDesc	Text	1-50	Contains the description. Always used in combination with MessageTypeCode. See for the notation MessageTypeCode.
MessageVariable	Text	1- unbou nded	Can be used to supply additional information regarding the message. Example: Server down in country X, where X is contained in the message variable. If used then in combination with MessageCode and MessageDesc.
MessageVersion	Text	3	Specifies the version number of the message. This version number can be used to make distinctions between multiple versions of messages. Name conventions: First MessageVersion is 1.0, subsequent versions 2.0, 3.0 etc. The client (in the request) and the legacy system (in the response), mention the MessageVersion that is supported. A Eucaris core system downgrades a response message to the MessageVersion of the client in case the response message contains a higher MessageVersion than the request message.
Mileage	Int	1-7	The mileage recorded at a certain MileageRecordingDate, in a certain MileageUnity
MileageRecording			This node contains one mileage recording
MileageRecordingCountry			This node contains information about the country where the mileage was recorded.
MileageRecordingCountryCode	Text	1-5	Code of the country where the mileage was recorded. Coding system: DS code. See Annex D for a list of possible values.
MileageRecordingCountryDesc	Text	1-80	Description of the country where the mileage was recorded. Is always used in combination with MileageRecordingCountryCode. For possible values, see there.
MileageRecordingDate	DateInc	8	Date at which the mileage has been recorded. Format CCYYMMDD. If the date is only partly known, fill the missing parts with 99.
MileageRecordingOrganisation	Text	1-50	Organisation responsible for the mileage recording
MileageRecordingReliability			This node contains information on the reliability of the mileage recordings.
MileageRecordingReliabilityCode	Enum		Code giving an indication of the reliability of the mileage recordings table, on the basis of a trend analysis of recordings. Value list: 1 = No suspicion 2 = Too few data for trend analysis 3 = Trend suspicious 4 = Trend highly suspicious
MileageRecordingReliabilityDesc	Text	1-50	See <i>MileageRecordingReliabilityCode</i>

Item	Type	Len	Description
MileageRecordingTime	Time	3-4	Time at which the mileage has been recorded, in format hhmm.
MileageRecordings			This node contains a table of mileage recordings
MileageReplies			This node contains the reply to the query, which contains (successful query) 1 or more nodes MileageReply and (unsuccessful reply) a node Messages.
MileageReply			This node contains the reply regarding one individual vehicle.
MileageReqByChassis			This node contains the complete request for the search of mileages by Vehicle Identification Number (VIN)
MileageStatus			This node contains information on the status of a mileage.
MileageStatusCode	Enum		Code denoting the status of mileage recording, it is possible that a recording consists of a status only. Value list: 1 = No odometer 2 = Odometer unreadable 3 = Not recorded 4 = Under investigation 5 = Suspect 99 = Not specified
MileageStatusDesc	Text	1-50	See <i>MileageRecordingStatusCode</i>
MileageUnity			This node contains the unity of the mileage.
MileageUnityCode	Enum		The unity of the mileage recording, or the unity of the odometer of the vehicle. The element is always used in combination with MileageUnityDesc. Value list: km = kilometre; m = mile; xx = not specified.
MileageUnityDesc	Text	1-20	See <i>MileageUnityCode</i>
OdometerMileage	Int	1-7	always equal to <i>OdometerMileageReq</i> . The value of this item is echoed in the response message for logging purposes (the logging of the response message suffices to record the value of this item)
OdometerMileageReq	Int	1-7	Before querying previous mileages of a vehicle, the requester reads the current mileage from the odometer and puts the value in the request message.
RecipientCountry	Text	1-5	RecipientCountry specifies the country that receives the message, i.e. the Member State of registration. The coding system used should correspond with the URL at which the request is submitted. See further Annex D.
Request			The request contains the information with which the target country will perform the query or undertake some other action.
SenderCountry	Text	1-5	SenderCountry specifies the name of the country that is invoking the request. The notation of SenderCountry is in accordance to the notation of the country on vehicles. See for notation RecipientCountry.
SenderName	Text	1-50	Originator of the request. The organisation in the Import MS that is responsible for mileage recordings.
SenderOrganisation			Specifies the type of organisation making the request, usually

Item	Type	Len	Description
			the registration office of the Enforcing Member State.
SenderOrganisationCode	Enum		Is used in combination with <i>SenderOrganisationDesc</i> . 0 = NOT SPECIFIED 1 = REGISTRATION OFFICE 2 = POLICE 3 = CUSTOMS
SenderOrganisationDesc	Text	1-50	Is used in combination with <i>SenderOrganisationCode</i> . See for the notation <i>SenderOrganisationCode</i> .
SenderOrganisationName	Text	1-50	Name of the organisation making the request.
ServiceExecutionReason			Specifies the reason for the request.
ServiceExecutionReasonCode	Enum		Is used in combination with <i>ServiceExecutionReasonDesc</i> . Value list 0 = Not specified (for all regular requests) 3 = Composing statistics (used when monitoring EU CARIS availability) 5 = Test
ServiceExecutionReasonDesc	Text	1-25	Is used in combination with <i>ServiceExecutionReasonCode</i> . See for the notation <i>ServiceExecutionReasonCode</i> .
ServiceFileNumber	Text	1-36	An identification associated with the inquiry, provided by the sender of the request. The information can be used later for reference and auditing. The originator of a request is completely free in using this element.
TimeOut	Int	1-7	Is not used in the message exchange. Provide value 0.
TimeStamp	DT		The full date and time stamp of the request/response.
VehCommercialName	Text	1-50	Document abbreviation [doc-1]: D.3 The commercial description / type of the vehicle. For example Focus, Astra, Megane.
VehEndRegistrationDate	DateInc	8	Date when registration has been ended in the Export MS. Format CCYYMMDD. If the date is only partly known, fill the missing parts with 99.
VehFirstRegistrationDate	DateInc	8	Document abbreviation [doc-1]: B Date of first registration of the vehicle (somewhere in the world). Format CCYYMMDD. If the date is only partly known, fill the missing parts with 99.
VehIdentificationNumber	Text	1-25	Document abbreviation [doc-1]: E The identification number (VIN) of the vehicle as registered by Export MS.
VehIdentificationNumberReq	Text	1-25	Document abbreviation [doc-1]: E The identification number (VIN) of the vehicle to search for as requested by Import MS.
VehMake	Text	1-50	Document abbreviation [doc-1]: D.1 The make of the car. For example Ford, Opel, Renault etc.
VehRegistrationNumber			Document abbreviation [doc-1]: A This node contains the registration number (licence number) of the vehicle in the notation as registered by the Export MS.
VehRegistrationNumberPart1	Text	1-15	Contains the full RegistrationNumber if a registration number can not be split up in more than one way. In Germany this part will contain the so-called "Ortcode or Districtcode"
VehRegistrationNumberPart2	Text	1-15	Applicable when the registration number can be split up in

Item	Type	Len	Description
			<p>more than one way. For Germany this part will contain the rest of the registration number.</p>
VehRegistrationNumberPart3	Text	1-15	<p>Contains the third part of the registration number (if applicable).</p>
VehRegistrationNumberPart4	Text	1-15	<p>Contains the fourth part of the registration number (if applicable).</p>
VehStartRegistrationDate	DateInc	8	<p>Date at which the vehicle was first registered in the Export MS. Format CCYYMMDD. If the date is only partly known, fill the missing parts with 99.</p>

5. Annex B – Defined messages

It is possible to give more than one message to clarify (error) situations. For all messages, the element MessageVariable can be used for further explanation.

5.1 System Messages

Message Code	MessageDesc	MessageClassCode	MessageClassDesc
0	UNKNOWN		
1	SYSTEM NOT AVAILABLE	ERR	ERROR
2	SYSTEM RETURNED ERROR	ERR	ERROR
3	AUTHORISATION ERROR	ERR	ERROR
4	TIMEOUT OCCURRED	INF	INFORMATION

5.2 Application Messages

Message Code	MessageDesc	MessageClassCode	MessageClassDesc
103	TOO MANY RECORDS FOUND	INF	INFORMATION
105	SEARCH INPUT NOT CORRECT	INF	INFORMATION
113	VEHICLE NOT FOUND	INF	INFORMATION
114	MILEAGE RECORDINGS NOT FOUND	INF	INFORMATION

5.3 Miscellaneous Messages

Message Code	MessageDesc	MessageClassCode	MessageClassDesc	MessageVariable
501	EUCARIS SERVER	INF	INFORMATION	
502	REGISTER SERVER	INF	INFORMATION	
503	GATEWAY SERVER	INF	INFORMATION	
504	LOCAL WEB SERVER	INF	INFORMATION	
505	CONTACT SERVICEDESK	INF	INFORMATION	<Contact data Eucaris help desk> ¹
507	PLEASE TRY LATER	INF	INFORMATION	

¹ Listed in the document “How To Reach The Eucaris Help Desk”

6. Annex C – XSD validation

Basis for validation applied in the XSD specifications is Annex A. The restrictions that are mentioned in the description of elements will also be implemented in the XSD specifications (e.g. data type, minimum length, maximum length, enumeration).

Data in request and response messages will be validated against these XSD specifications. If a request or response message fails XSD validation, the sender of the message will receive a response message with an explanatory error message.

In XSD validation, the following principles apply:

- For items that are made up of a combination of code and descriptions, only the codes are validated, not the descriptions.
- Interdependence of XML elements is not validated, since this is not possible in XML v1.0. Because it is too complex, there is no validation on a valid date for elements that contain a date in format CCYYMMDD (VehFirstRegistrationDate, and many others). Any value of 8 digits will pass schema validation.

7. Annex D – Country codes

7.1 EUCARIS country code convention

For a request message, the SenderCountry is the country where the request originated, and the RecipientCountry is the country that is to receive the request.

In response messages, SenderCountry/RecipientCountry are reversed. Now, the SenderCountry is the country where the response originated, and the RecipientCountry is the country that is to receive the response.

7.2 Coding systems

EUCARIS supports the following coding system for country codes:

- EUCARIS country codes. The coding is based on a United Nations agreement on vehicle country codes (hereafter named DS code) <http://www.unece.org/trans/conventn/Distsigns.pdf>. However, for countries where there is more than one organisation that is National Contact Point for EUCARIS, to the country code, a sequence number might be added.
- ISO 3166-1 alpha-2

A client submitting a request, can choose the coding systems it wants to use, via the URL at which it submits the request. Each supported country coding system has a different URL.

In a consolidated response, the responding countries will be denoted in the coding system that was chosen while submitting the the request.

The table below lists the countries and the coded values in various coding systems:

Country name (english)	ISO 3166-1 alpha-2	ISO 3166-1 alpha-3	DS code
Alderney	--	--	GBA
Austria	AT	AUT	A
Belgium	BE	BEL	B
Bulgaria	BG	BGR	BG
Croatia	HR	HRV	HR
Cyprus	CY	CYP	CY
Czech Republic	CZ	CZE	CZ
Denmark	DK	DNK	DK
Estonia	EE	EST	EST
Finland	FI	FIN	FIN
France	FR	FRA	F
Germany	DE	DEU	D
Gibraltar	GI	GIB	GBZ
Greece	GR	GRC	GR
Guernsey	GG	GGY	GBG
Hungary	HU	HUN	H
Iceland	IS	ISL	IS
Ireland	IE	IRL	IRL
Isle of Man	IM	IMN	GBM
Italy	IT	ITA	I
Jersey	JE	JEY	GBJ
Latvia	LV	LVA	LV
Liechtenstein	LI	LIE	FL

Country name (english)	ISO 3166-1 alpha-2	ISO 3166-1 alpha-3	DS code
Lithuania	LT	LTU	LT
Luxembourg	LU	LUX	L
Malta	MT	MLT	M
The Netherlands	NL	NLD	NL
Netherlands Antilles	AN	ANT	NA
Norway	NO	NOR	N
Poland	PL	POL	PL
Portugal	PT	PRT	P
Romania	RO	ROU	RO
Slovakia	SK	SVK	SK
Slovenia	SI	SVN	SLO
Spain	ES	ESP	E
Sweden	SE	SWE	S
Switzerland	CH	CHE	CH
United Kingdom	GB	GBR	GB

7.3 Artificial country codes

FIN2	Finland has two organisations hosting EUCARIS. FIN = Haltik (ICT agency providing technical data and communication services). FIN2 = The Finnish Transport Safety Agency (TRAFI). TRAFI is responsible for Prüm and CBE, Haltik is responsible for other EUCARIS services supported by Finland.
L2	Luxembourg has two organizations hosting EUCARIS: The SNCT (hosting the EUCARIS treaty services) and the Police (hosting the Prüm services). The SNCT uses 'L' as country code while the Luxembourg Police is using 'L2' as country code.
RO2	The 'Registrul Auto Român' (R.A.R.) was the organisation initial hosting EUCARIS and supplying data for the EUCARIS treaty using the RO country code. Mid 2010 the 'Romanian Driving License and Vehicle Registration Directorate' took over the responsibility for supplying both EUCARIS- and Prüm treaty data. Since the R.A.R is still using EUCARIS for their outbound requests the country code RO2 was assigned to the 'Romanian Driving License and Vehicle Registration Directorate'.
F2	France has two organizations hosting EUCARIS: The ministry of interior which is supplying Prüm treaty data using country code F, and the DPICA (Direction Programme Interministérielle Contrôle Automatisé) which is in charge of the automated control program for road safety in France, using code F2. The DPICA exchanges data regarding traffic fines.