Commission Implementing Regulation (EU) 2016/799 of 18 March 2016 implementing Regulation (EU) No 165/2014 of the European Parliament and of the Council laying down the requirements for the construction, testing, installation, operation and repair of tachographs and their components (Text with EEA relevance)

ANNEX I C

Requirements for construction, testing, installation, and inspection

Appendix 7

DATA DOWNLOADING PROTOCOLS

1. INTRODUCTION

This appendix specifies the procedures to follow in order to perform the different types of data download to an External Storage Medium, together with the protocols that must be implemented to assure the correct data transfer and the full compatibility of the downloaded data format to allow any controller to inspect these data and be able to control their authenticity and their integrity before analysing them.

[F1 1.1. Scope

Data may be downloaded to an ESM:

— from a Vehicle Unit by an Intelligent Dedicated Equipment (IDE) connected to the VU,

— from a tachograph card by an IDE fitted with a card interface device (IFD),

— from a tachograph card via a vehicle unit by an IDE connected to the VU.

To give the possibility to verify the authenticity and integrity of downloaded data stored on an ESM, data is downloaded with a signature appended in accordance with Appendix 11 Common Security Mechanisms. The source equipment (VU or card) identification and its security certificates (Member state and equipment) are also downloaded. The verifier of the data must possess independently a trusted European public key.

Data downloaded from a VU are signed using Appendix 11 Common Security Mechanisms Part B (Second-generation tachograph system), except when drivers' control is performed by a non EU control authority, using a first generation control card, in which case data are signed using Appendix 11 Common Security Mechanisms Part A (First-generation tachograph system), as requested by Appendix 15 Migration, requirement MIG_015.

This Appendix specifies therefore two types of data downloads from the VU:

— Generation 2 type of VU data download, providing the generation 2 data structure, signed using Appendix 11 Common Security Mechanisms Part B,

— Generation 1 type of VU data download, providing the generation 1 data structure, signed using Appendix 11 Common Security Mechanisms Part A.

Similarly, there are two types of data downloads from second generation driver cards inserted in a VU, as specified in paragraphs 3 and 4 of this Appendix.]

Textual Amendments

F1 Substituted by Commission Implementing Regulation (EU) 2018/502 of 28 February 2018 amending Implementing Regulation (EU) 2016/799 laying down the requirements for the construction, testing, installation, operation and repair of tachographs and their components (Text with EEA relevance).

1.2. Acronyms and notations

The following acronyms are used in this appendix:

AID	Application Identifier
ATR	Answer To Reset
CS	Checksum byte
DF	Dedicated File
DS_	Diagnostic Session
EF	Elementary File

ESM	External Storage Medium
FID	File Identifier (File ID)
FMT	Format Byte (first byte of message header)
ICC	Integrated Circuit Card
IDE	Intelligent Dedicated Equipment: The equipment used to perform data downloading to the ESM (e.g. Personal Computer)
IFD	Interface Device
KWP	Keyword Protocol 2000
LEN	Length Byte (last byte of message header)
PPS	Protocol Parameter Selection
PSO	Perform Security Operation
SID	Service Identifier
SRC	Source byte
TGT	Target Byte
TLV	Tag Length Value
TREP	Transfer Response Parameter
TRTP	Transfer Request Parameter
VU	Vehicle Unit

2. V.U. DATA DOWNLOADING

2.1. **Download procedure**

In order to carry on a VU data download, the operator must perform the following operations:

— Insert his tachograph card inside a card slot of the VU[(1)];
— Connect the IDE to the VU download connector;
— Establish the connection between the IDE and the VU;
— Select on the IDE the data to download and send the request to the VU;
— Close the download session.

2.2. **Data download protocol**

The protocol is structured on a master-slave basis, with the IDE playing the master role and the VU playing the slave role.

The message structure, types and flow are principally based on the Keyword Protocol 2000 (KWP) (ISO 14230-2 Road vehicles — Diagnostic systems — Keyword protocol 2000 — Part2: Data link layer).

The application layer is principally based on the current draft to date of ISO 14229-1 (Road vehicles — Diagnostic systems — Part 1: Diagnostic services, version 6 of 22 February 2001).

2.2.1 *Message structure*

DDP_002All the messages exchanged between the IDE and the VU are formatted with a structure consisting of three parts:

— Header composed by a Format byte (FMT), a Target byte (TGT), a Source byte (SRC) and possibly a Length byte (LEN),
— Data field composed by a Service Identifier byte (SID) and a variable number of data bytes, which can include an optional diagnostic session byte (DS_) or an optional transfer parameter byte (TRTP or TREP).
— Checksum composed by a Checksum byte (CS).

Header	**Data field**	**Checksum**

FMT	**TGT**	**SRC**	**LEN**	**SID**	**DATA**	**…**	**…**	**…**	**CS**
4 bytes				Max 255 bytes					1 byte

The TGT and SRC byte represent the physical address of the recipient and originator of the message. Values are F0 Hex for the IDE and EE Hex for the VU.

The LEN byte is the length of the Data field part.

The Checksum byte is the 8 bit sum series modulo 256 of all the bytes of the message excluding the CS itself.

FMT, SID, DS_, TRTP and TREP bytes are defined later in this document.

DDP_003 In the case where the data to be carried by the message is longer than the space available in the data field part, the message is actually sent in several sub messages. Each sub message bears a header, the same SID, TREP and a 2-byte sub message counter indicating the sub message number within the total message. To enable error checking and abort the IDE acknowledges every sub message. The IDE can accept the sub message, ask for it to be re-transmitted, request the VU to start again or abort the transmission.

DDP_004 If the last sub message contains exactly 255 bytes in the data field, a final sub message with an empty (except SID TREP and sub message counter) data field must be appended to show the end of the message.

Example:

Header	**SID**	**TREP**	**Message**	**CS**
4 Bytes	Longer than 255 Bytes			

Will be transmitted as:

Header	**SID**	**TREP**	**00**	**01**	**Sub message 1**	**CS**
4 Bytes	255 Bytes					
Header	**SID**	**TREP**	**00**	**02**	**Sub message 2**	**CS**
4 Bytes	255 Bytes					

…

Header	**SID**	**TREP**	**xx**	**yy**	**Sub message n**	**CS**
4 Bytes	Less than 255 Bytes					

or as:

Header	**SID**	**TREP**	**00**	**01**	**Sub message 1**	**CS**
4 Bytes	255 Bytes					
Header	**SID**	**TREP**	**00**	**02**	**Sub message 2**	**CS**
4 Bytes	255 Bytes					

…

Header	**SID**	**TREP**	**xx**	**yy**	**Sub message n**	**CS**
4 Bytes	255 Bytes					

Header	**SID**	**TREP**	**xx**	**yy + 1**	**CS**
4 Bytes	4 bytes				

2.2.2 *Message types*

The communication protocol for data download between the VU and the IDE requires the exchange of 8 different message types.

The following table summarises these messages.

[F1Message Structure		**Max 4 Bytes Header**				**Max 255 Bytes Data**			**1 Byte CheckSum**
IDE ->	<- VU	**FMT**	**TGT**	**SRC**	**LEN**	**SID**	**DS_ / TRTP**	**DATA**	**CS**
Start Communication Request		81	EE	F0		81			E0
Positive Response Start Communication		80	F0	EE	03	C1		EA, 8F	9B
Start Diagnostic Session Request		80	EE	F0	02	10	81		F1
Positive Response Start Diagnostic		80	F0	EE	02	50	81		31
Link Control Service									
Verify Baud Rate (stage 1)									
9 600 Bd		80	EE	F0	04	87		01,01,01	EC

19 200 Bd	80	EE	F0	04	87		01,01,02	ED
38 400 Bd	80	EE	F0	04	87		01,01,03	EE
57 600 Bd	80	EE	F0	04	87		01,01,04	EF
115 200 Bd	80	EE	F0	04	87		01,01,05	F0
Positive Response Verify Baud Rate	80	F0	EE	02	C7		01	28
Transition Baud Rate (stage 2)	80	EE	F0	03	87		02,03	ED
Request Upload	80	EE	F0	0A	35		00,00,00,00,00,FF,FF,FF,FF	
Positive Response Request Upload	80	F0	EE	03	75		00,FF	D5
Transfer Data Request								
Overview	80	EE	F0	02	36	01 or 21		97
Activities	80	EE	F0	06	36	02 or 22	Date	CS
Events & Faults	80	EE	F0	02	36	03 or 23	Date	99
Detailed Speed	80	EE	F0	02	36	04 or 24	Date	9 A
Technical Data	80	EE	F0	02	36	05 or 25	Date	9B
Card download	80	EE	F0	02	36	06	Slot	CS
Positive Response Transfer Data	80	F0	EE	Len	76	TREP	Data	CS
Request Transfer Exit	80	EE	F0	01	37			96
Positive Response Request Transfer Exit	80	F0	EE	01	77			D6
Stop Communication Request	80	EE	F0	01	82			E1
Positive Response Stop Communication	80	F0	EE	01	C2			21

Acknowledge sub message	80	EE	F0	Len	83		Data	CS
Negative responses								
General reject	80	F0	EE	03	7F	Sid Req	10	CS
Service not supported	80	F0	EE	03	7F	Sid Req	11	CS
Sub function not supported	80	F0	EE	03	7F	Sid Req	12	CS
Incorrect Message Length	80	F0	EE	03	7F	Sid Req	13	CS
Conditions not correct or Request sequence error	80	F0	EE	03	7F	Sid Req	22	CS
Request out of range	80	F0	EE	03	7F	Sid Req	31	CS
Upload not accepted	80	F0	EE	03	7F	Sid Req	50	CS
Response pending	80	F0	EE	03	7F	Sid Req	78	CS
Data not available	80	F0	EE	03	7F	Sid Req	FA	CS**]**

Notes:

— **[**F2TRTP 21 to 25 are used for Generation 2 type of VU data download requests, TRTP 01 to 05 are used for Generation 1 type of VU data download requests, which can only be accepted by the VU in the frame of drivers' control performed by a non EU control authority, using a first generation control card.

— TRTP 11 to 19 and 31 to 39 are reserved for manufacturer specific download requests.**]**

— Sid Req = the Sid of the corresponding request.

— TREP = the TRTP of the corresponding request.

— Dark cells denote that nothing is transmitted.

— The term upload (as seen from the IDE) is used for compatibility with ISO 14229. It means the same as download (as seen from the VU).

— Potential 2-byte sub message counters are not shown in this table.

— Slot is the slot number, either “1” (card on driver slot) or “2” (card on co-driver slot)

— In case the slot is not specified, the VU shall select slot 1 if a card is inserted in this slot and it shall select slot 2 only in case it is specifically selected by the user.

2.2.2.1 *Start Communication Request (SID 81)*

DDP_005 This message is issued by the IDE to establish the communication link with the VU. Initial communications are always performed at 9 600 baud (until baud rate is eventually changed using the appropriate Link control services).

2.2.2.2 *Positive Response Start Communication (SID C1)*

DDP_006 This message is issued by the VU to answer positively to a start communication request. It includes the 2 key bytes 'EA''8F' indicating that the unit supports protocol with header including target source and length information.

2.2.2.3 *Start Diagnostic Session Request (SID 10)*

DDP_007 The Start Diagnostic Session request message is issued by the IDE in order to request a new diagnostic session with the VU. The sub function 'default session' (81 Hex) indicates a standard diagnostic session is to be opened.

2.2.2.4 *Positive Response Start Diagnostic (SID 50)*

DDP_008 The Positive Response Start Diagnostic message is sent by the VU to answer positively to Diagnostic Session Request.

2.2.2.5 *Link Control Service (SID 87)*

DDP_052 The Link Control Service is used by the IDE to initiate a change in baud rate. This takes place in two steps. In step one the IDE proposes the baud rate change, indicating the new rate. On receipt of a positive message from the VU the IDE sends out confirmation of the baud rate change to the VU (step two). The IDE then changes to the new baud rate. After receipt of the confirmation the VU changes to the new baud rate

2.2.2.6 *Link Control Positive Response (SID C7)*

DDP_053 The Link Control Positive Response is issued by the VU to answer positively to Link Control Service request (step one). Note that no response is given to the confirmation request (step two).

2.2.2.7 *Request Upload (SID 35)*

DDP_009 The Request Upload message is issued by the IDE to specify to the VU that a download operation is requested. To meet the requirements of ISO14229 data is included covering address, the size and format details for the data requested. As these are not known to the IDE prior to a download, the memory address is set to 0, format is unencrypted and uncompressed and the memory size is set to the maximum.

2.2.2.8 *Positive Response Request Upload (SID 75)*

DDP_010 The Positive Response Request Upload message is sent by the VU to indicate to the IDE that the VU is ready to download data. To meet the requirements of ISO 14229 data is included in this positive response message, indicating to the IDE that further Positive Response Transfer Data messages will include 00FF hex bytes maximum.

2.2.2.9 *Transfer Data Request (SID 36)*

[F1 DDP_011 The Transfer Data Request is sent by the IDE to specify to the VU the type of data that are to be downloaded. A one byte Transfer Request Parameter (TRTP) indicates the type of transfer.

There are six types of data transfer. For VU data download, two different TRTP values can be used for each transfer type:

Data transfer type	TRTP value for generation 1 type of VU data download	TRTP value for generation 2 type of VU data download
Overview	01	21
Activities of a specified date	02	22
Events and faults	03	23
Detailed speed	04	24
Technical data	05	25

Data transfer type	TRTP value
Card download	06**]**

[[F1]DDP_054 It is mandatory for the IDE to request the overview data transfer (TRTP 01 or 21) during a download session as this only will ensure that the VU certificates are recorded within the downloaded file (and allow for verification of digital signature).

In the second case (TRTP 02 or 22) the Transfer Data Request message includes the indication of the calendar day (`TimeReal` format) to be downloaded.**]**

2.2.2.10 *Positive Response Transfer Data (SID 76)*

DDP_012 The Positive Response Transfer Data is sent by the VU in response to the Transfer Data Request. The message contains the requested data, with a Transfer Response Parameter (TREP) corresponding to the TRTP of the request.

[[F1]DDP_055 In the first case (TREP 01 or 21), the VU will send data helping the IDE operator to choose the data he wants to download further. The information contained within this message is:

— Security certificates,
— Vehicle identification,
— VU current date and time,
— Min and Max downloadable date (VU data),
— Indication of cards presence in the VU,
— Previous download to a company,
— Company locks,
— Previous controls.**]**

2.2.2.11 *Request Transfer Exit (SID 37)*

DDP_013 The Request Transfer Exit message is sent by the IDE to inform the VU that the download session is terminated.

2.2.2.12 *Positive Response Request Transfer Exit (SID 77)*

DDP_014 The Positive Response Request Transfer Exit message is sent by the VU to acknowledge the Request Transfer Exit.

2.2.2.13 *Stop Communication Request (SID 82)*

DDP_015 The Stop Communication Request message is sent by the IDE to disconnect the communication link with the VU.

2.2.2.14 *Positive Response Stop Communication (SID C2)*

DDP_016 The Positive Response Stop Communication message is sent by the VU to acknowledge the Stop Communication Request.

2.2.2.15 *Acknowledge Sub Message (SID 83)*

DDP_017 The Acknowledge Sub Message is sent by the IDE to confirm receipt of each part of a message that is being transmitted as several sub messages. The data field contains the SID received from the VU and a 2-byte code as follows:

— MsgC+1 Acknowledges correct receipt of sub message number MsgC.

Request from the IDE to the VU to send next sub message

— MsgC indicates a problem with the receipt of sub message number MsgC.

Request from the IDE to the VU to send the sub message again.

— FFFF requests termination of the message.

This can be used by the IDE to end the transmission of the VU message for any reason.

The last sub message of a message (LEN byte < 255) may be acknowledged using any of these codes or not acknowledged.

The VU responses that will consist of several sub messages are:

— Positive Response Transfer Data (SID 76)

2.2.2.16 *Negative Response (SID 7F)*

DDP_018 The Negative Response message is sent by the VU in response to the above request messages when the VU cannot satisfy the request. The data fields of the message contains the SID of the response (7F), the SID of the request, and a code specifying the reason of the negative response. The following codes are available:

— 10 general reject

The action cannot be performed for a reason not covered below.

— 11 service not supported

The SID of the request is not understood.

— 12 sub function not supported

The DS_ or TRTP of the request is not understood, or there are no further sub messages to be transmitted.

— 13 incorrect message length

The length of the received message is wrong.

— 22 conditions not correct or request sequence error

The required service is not active or the sequence of request messages is not correct.

— 31 Request out of range

The request parameter record (data field) is not valid.

— 50 upload not accepted

The request cannot be performed (VU in a non appropriate mode of operation or internal fault of the VU).

— 78 response pending

The action requested cannot be completed in time and the VU is not ready to accept another request.

— **[**[F1]FA data not available

The data object of a data transfer request are not available in the VU (e.g. no card is inserted, generation 1 type of VU data download requested outside the frame of a driver's control by a non EU control authority…).**]**

2.2.3 *Message flow*

A typical message flow during a normal data download procedure is the following:

IDE		**VU**
Start Communication Request	#	
	#	Positive Response
Start Diagnostic Service Request	#	
	#	Positive Response
Request Upload	#	
	#	Positive Response
Transfer Data Request Overview	#	
	#	Positive Response
Transfer Data Request #2	#	
	#	Positive Response #1
Acknowledge Sub Message #1	#	
	#	Positive Response #2
Acknowledge Sub Message #2	#	
	#	Positive Response #m
Acknowledge Sub Message #m	#	
	#	Positive Response (Data Field < 255 Bytes)
Acknowledge Sub Message (optional)	#	
…		
Transfer Data Request #n	#	
	#	Positive Response

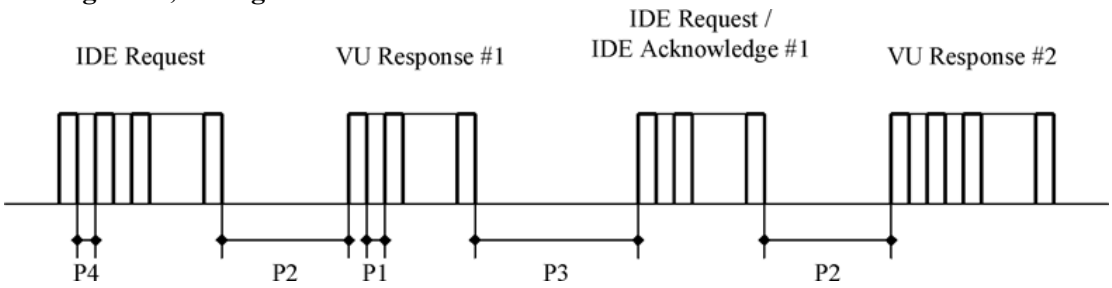
Request Transfer Exit	#	
	#	Positive Response
Stop Communication Request	#	
	#	Positive Response

2.2.4 *Timing*

DDP_019 During normal operation the timing parameters shown in the following figure are relevant:

Figure 1

Message flow, timing



Where:

P1 = Inter byte time for VU response.
P2 = Time between end of IDE request and start of VU response, or between end of IDE acknowledge and start of next VU response.
P3 = Time between end of VU response and start of new IDE request, or between end of VU response and start of IDE acknowledge, or between end of IDE request and start of new IDE request if VU fails to respond.
P4 = Inter byte time for IDE request.
P5 = Extended value of P3 for card downloading.

The allowed values for the timing parameters are showed in the following table (KWP extended timing parameters set, used in case of physical addressing for faster communication).

Timing Parameter	Lower limitValue (ms)	Upper limitValue (ms)
P1	0	20
P2	20	1 000[a]
P3	10	5 000
P4	5	20
P5	10	20 minutes

a If the VU responds with a Negative Response containing a code meaning ‘request correctly received, response pending’, this value is extended to the same upper limit value of P3.

2.2.5 *Error handling*

If an error occurs during the message exchange, the message flow scheme is modified depending on which equipment has detected the error and on the message generating the error.

In figure 2 and figure 3 the error handling procedures for the VU and the IDE are respectively shown.

2.2.5.1 *Start Communication phase*

DDP_020 If the IDE detects an error during the Start Communication phase, either by timing or by the bit stream, then it will wait for a period P3 min before issuing again the request.

DDP_021 If the VU detects an error in the sequence coming from the IDE, it shall send no response and wait for another Start Communication Request message within a period P3 max.

2.2.5.2 *Communication phase*

Two different error handling areas can be defined:

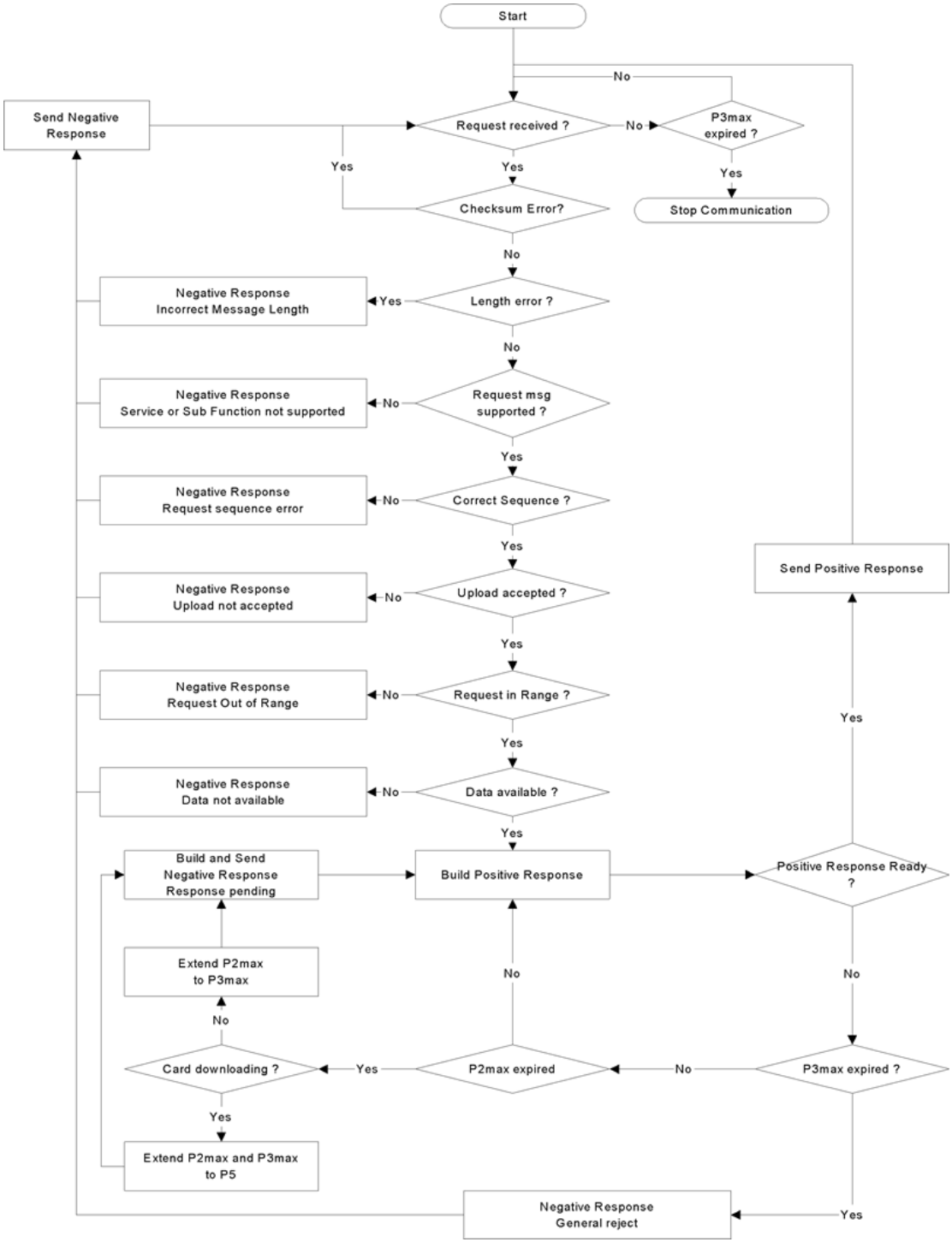
1. **The VU detects an IDE transmission error.**

DDP_022 For every received message the VU shall detect timing errors, byte format errors (e.g. start and stop bit violations) and frame errors (wrong number of bytes received, wrong checksum byte).

DDP_023 If the VU detects one of the above errors, then it sends no response and ignores the message received.

DDP_024 The VU may detect other errors in the format or content of the received message (e.g. message not supported) even if the message satisfies the length and checksum requirements; in such a case, the VU shall respond to the IDE with a Negative Response message specifying the nature of the error.
Figure 2

VU error handling



2. **The IDE detects a VU transmission error.**

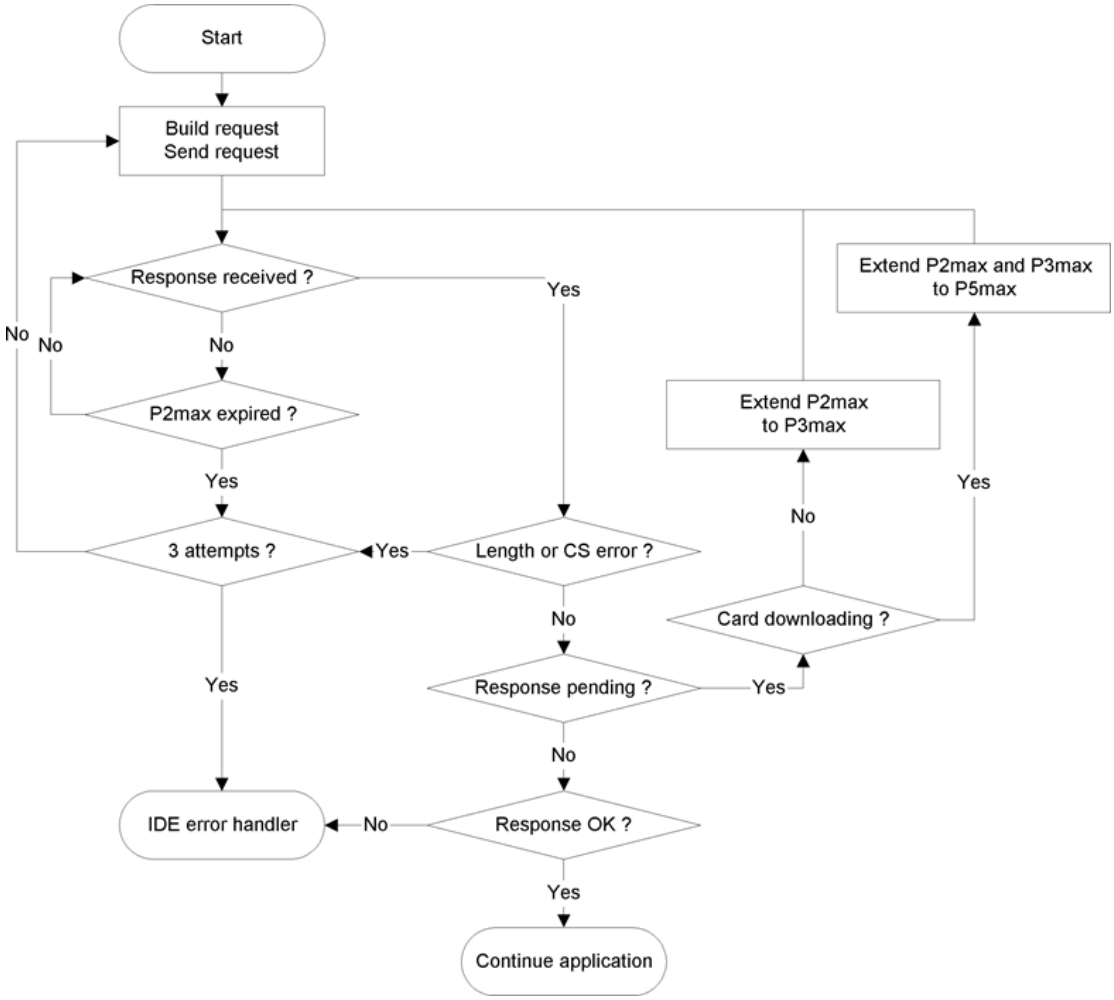
DDP_025 For every received message the IDE shall detect timing errors, byte format errors (e.g. start and stop bit violations) and frame errors (wrong number of bytes received, wrong checksum byte).

DDP_026 The IDE shall detect sequence errors, e.g. incorrect sub message counter increments in successive received messages.

DDP_027 If the IDE detects an error or there was no response from the VU within a P2 max period, the request message will be sent again for a maximum of three transmissions in total. For the purposes of this error detection a sub message acknowledge will be considered as a request to the VU.

DDP_028 The IDE shall wait at least for a period of P3 min before beginning each transmission; the wait period shall be measured from the last calculated occurrence of a stop bit after the error was detected.
Figure 3

IDE error handling



2.2.6 *Response Message content*

This paragraph specifies the content of the data fields of the various positive response messages.

Data elements are defined in Appendix 1 data dictionary.

Remark: For generation 2 downloads, each top-level data element is represented by a record array, even if it contains only one record. A record array starts with a header; this header contains

the record type, the record size and the number of records. Record arrays are named by '… RecordArray' (with header) in the following tables.

2.2.6.1 *Positive Response Transfer Data Overview*

DDP_029 **[**[F1]The data field of the 'Positive Response Transfer Data Overview' message shall provide the following data in the following order under the SID 76 Hex, the TREP 01 or 21 Hex and appropriate sub message splitting and counting:**]**

[[F1]Data structure generation 1 (TREP 01 Hex)**]**

Data element		Comment
MemberStateCertificate	VUCertificate	VU Security certificates
VehicleIdentificationNumber	VehicleRegistrationIdentification	Vehicle identification
CurrentDateTime		VU current date and time
VuDownloadablePeriod		Downloadable period
CardSlotsStatus		Type of cards inserted in the VU
VuDownloadActivityData		Previous VU download
VuCompanyLocksData		All company locks stored. If the section is empty, only noOfLocks = 0 is sent.
VuControlActivityData		All control records stored in the VU. If the section is empty, only noOfControls = 0 is sent
Signature		RSA signature of all data (except certificates) starting from VehicleIdentificationNumber down to last byte of last VuControlActivityData.

[[F1]Data structure generation 2 (TREP 21 Hex)**]**

Data element		Comment
MemberStateCertificateRecordArray		Member state certificate
VUCertificateRecordArray		VU certificate
VehicleIdentificationNumberRecordArray		Vehicle identification
VehicleRegistrationNumberRecordArray		Vehicle registration number
CurrentDateTimeRecordArray		VU current date and time
VuDownloadablePeriodRecordArray		Downloadable period
CardSlotsStatusRecordArray		Type of cards inserted in the VU

[F1Data structure generation 2 (TREP 21 Hex)**]**		
VuDownloadActivityDataRecordArray		Previous VU download
VuCompanyLocksRecordArray		All company locks stored. If the section is empty, an array header with noOfRecords = 0 is sent
VuControlActivityRecordArray		All control records stored in the VU. If the section is empty, an array header with noOfRecords = 0 is sent
SignatureRecordArray		ECC signature of all preceding data except the certificates.

2.2.6.2 *Positive Response Transfer Data Activities*

DDP_030 **[**F1The data field of the ‘Positive Response Transfer Data Activities’ message shall provide the following data in the following order under the SID 76 Hex, the TREP 02 or 22 Hex and appropriate sub message splitting and counting:**]**

[F1Data structure generation 1 (TREP 02 Hex)**]**		
Data element		**Comment**
TimeReal		Date of day downloaded
OdometerValueMidnight		Odometer at end of downloaded day
VuCardIWData		Cards insertion withdrawal cycles data. — If this section contains no available data, only noOfVuCardIWRecords = 0 is sent. — When a VuCardIWRecord lies across 00:00 (card insertion on previous day) or across 24:00 (card withdrawal the following day) it shall appear in full within the two days involved.
VuActivityDailyData		Slots status at 00:00 and activity changes recorded for the day downloaded.

[F1 Data structure generation 1 (TREP 02 Hex)**]**

VuPlaceDailyWorkPeriodData		Places related data recorded for the day downloaded. If the section is empty, only noOfPlaceRecords = 0 is sent.
VuSpecificConditionData		Specific conditions data recorded for the day downloaded. If the section is empty, only noOfSpecificConditionRecords=0 is sent
Signature		RSA signature of all data starting from TimeReal down to last byte of last specific condition record.

[F1 Data structure generation 2 (TREP 22 Hex)**]**

Data element		**Comment**
DateOfDayDownloadedRecordArray		Date of day downloaded
OdometerValueMidnightRecordArray		Odometer at end of downloaded day
VuCardIWRecordArray		Cards insertion withdrawal cycles data. — If this section contains no available data, an array header with noOfRecords = 0 is sent. — When a VuCardIWRecord lies across 00:00 (card insertion on previous day) or across 24:00 (card withdrawal the following day) it shall appear in full within the two days involved.
VuActivityDailyRecordArray		Slots status at 00:00 and activity changes recorded for the day downloaded.
VuPlaceDailyWorkPeriodRecordArray		Places related data recorded for the day downloaded. If the section is empty, an array

[F1Data structure generation 2 (TREP 22 Hex)]		
		header with noOfRecords = 0 is sent.
[F1VuGNSSADRecordArray		GNSS positions of the vehicle when the accumulated driving time of the vehicle reaches a multiple of three hours. If the section is empty, an array header with noOfRecords = 0 is sent.]
VuSpecificConditionRecordArray		Specific conditions data recorded for the day downloaded. If the section is empty, an array header with noOfRecords =0 is sent
SignatureRecordArray		ECC signature of all preceding data.

2.2.6.3 *Positive Response Transfer Data Events and Faults*

DDP_031 [F1The data field of the 'Positive Response Transfer Data Events and Faults' message shall provide the following data in the following order under the SID 76 Hex, the TREP 03 or 23 Hex and appropriate sub message splitting and counting:]

[F1Data structure generation 1 (TREP 03 Hex)]		
Data element		**Comment**
VuFaultData		All faults stored or on-going in the VU. If the section is empty, only noOfVuFaults = 0 is sent.
VuEventData		All events (except over speeding) stored or on-going in the VU. If the section is empty, only noOfVuEvents = 0 is sent.
VuOverSpeedingControlData		Data related to last over speeding control (default value if no data).
VuOverSpeedingEventData		All over speeding events stored in the VU. If the section is empty, only noOfVuOverSpeedingEvents = 0 is sent.

[[F1]Data structure generation 1 (TREP 03 Hex)**]**		
VuTimeAdjustmentData		All time adjustment events stored in the VU (outside the frame of a full calibration). If the section is empty, only noOfVuTimeAdjRecords = 0 is sent.
Signature		RSA signature of all data starting from noOfVuFaults down to last byte of last time adjustment record
[[F1]Data structure generation 2 (TREP 23 Hex)**]**		
Data element		**Comment**
VuFaultRecordArray		All faults stored or on-going in the VU. If the section is empty, an array header with noOfRecords = 0 is sent.
VuEventRecordArray		All events (except over speeding) stored or on-going in the VU. If the section is empty, an array header with noOfRecords = 0 is sent.
VuOverSpeedingControlDataRecordArray		Data related to last over speeding control (default value if no data).
VuOverSpeedingEventRecordArray		All over speeding events stored in the VU. If the section is empty, an array header with noOfRecords = 0 is sent.
VuTimeAdjustmentRecordArray		All time adjustment events stored in the VU (outside the frame of a full calibration). If the section is empty, an array header with noOfRecords = 0 is sent.
[[F3]**]**		
SignatureRecordArray		ECC signature of all preceding data.

2.2.6.4 *Positive Response Transfer Data Detailed Speed*

DDP_032 **[**F1 The data field of the 'Positive Response Transfer Data Detailed Speed' message shall provide the following data in the following order under the SID 76 Hex, the TREP 04 or 24 Hex and appropriate sub message splitting and counting:**]**

[F1 Data structure generation 1 (TREP 04)**]**

Data element		Comment
VuDetailedSpeedData		All detailed speed stored in the VU (one speed block per minute during which the vehicle has been moving) 60 speed values per minute (one per second).
Signature		RSA signature of all data starting from noOfSpeedBlocks down to last byte of last speed block.

[F1 Data structure generation 2 (TREP 24)**]**

Data element		Comment
VuDetailedSpeedBlockRecordArray		All detailed speed stored in the VU (one speed block per minute during which the vehicle has been moving) 60 speed values per minute (one per second).
SignatureRecordArray		ECC signature of all preceding data.

2.2.6.5 *Positive Response Transfer Data Technical Data*

DDP_033 **[**F1 The data field of the 'Positive Response Transfer Data Technical Data' message shall provide the following data in the following order under the SID 76 Hex, the TREP 05 or 25 Hex and appropriate sub message splitting and counting:**]**

[F1 Data structure generation 1 (TREP 05)**]**

Data element		Comment
VuIdentification		
SensorPaired		
VuCalibrationData		All calibration records stored in the VU.
Signature		RSA signature of all data starting from vuManufacturerName down to last byte of last VuCalibrationRecord.

[[F1]Data structure generation 2 (TREP 25)**]**

Data element		Comment
VuIdentificationRecordArray		
VuSensorPairedRecordArray		All MS pairings stored in the VU
VuSensorExternalGNSSCoupledRecordArray		All external GNSS facility couplings stored in the VU
VuCalibrationRecordArray		All calibration records stored in the VU.
VuCardRecordArray		All card insertion data stored in the VU.
VuITSConsentRecordArray		
VuPowerSupplyInterruptionRecordArray		
SignatureRecordArray		ECC signature of all preceding data.

2.3. **ESM File storage**

DDP_034 When a download session has included a VU data transfer, the IDE shall store within one single physical file all data received from the VU during the download session within Positive Response Transfer Data messages. Data stored excludes message headers, sub-message counters, empty sub-messages and checksums but include the SID and TREP (of the first sub-message only if several sub-messages).

3. TACHOGRAPH CARDS DOWNLOADING PROTOCOL

3.1. **Scope**

This paragraph describes the direct card data downloading of a tachograph card to an IDE. The IDE is not part of the secure environment; therefore no authentication between the card and the IDE is performed.

3.2. **Definitions**

Download session : Each time a download of the ICC data is performed. The session covers the complete procedure from the reset of the ICC by an IFD until the deactivation of the ICC (withdraw of the card or next reset).

Signed Data File : A file from the ICC. The file is transferred to the IFD in plain text. On the ICC the file is hashed and signed and the signature is transferred to the IFD.

3.3. **Card Downloading**

[[F1]DDP_035 The download of a tachograph card includes the following steps:

— Download the common information of the card in the EFs ICC and IC This information is optional and is not secured with a digital signature.

— (for first and second generation tachograph cards) Download EFs within Tachograph DF:

— Download the EFs Card_Certificate and CA_Certificate This information is not secured with a digital signature.

It is mandatory to download these files for each download session.

— Download the other application data EFs (within Tachograph DF) except EF Card_Download. This information is secured with a digital signature, using Appendix 11 Common Security Mechanisms Part A.

— It is mandatory to download at least the EFs Application_Identification and Identification for each download session.

— When downloading a driver card it is also mandatory to download the following EFs:

— (for second generation tacograph cards only) Except when a download of a driver card inserted in a VU is performed during drivers' control by a non EU control authority, using a first generation control card, download EFs within Tachograph_G2 DF:

— Download the EFs CardSignCertificate, CA_Certificate and Link_Certificate (if present). This information is not secured with a digital signature.

It is mandatory to download these files for each download session.

— Download the other application data EFs (within Tachograph_G2 DF) except EF Card_Download. This information is secured with a digital signature, using Appendix 11 Common Security Mechanisms Part B.

— It is mandatory to download at least the EFs Application_Identification and Identification for each download session.

— When downloading a driver card it is also mandatory to download the following EFs:

— When downloading a driver card, update the LastCardDownload date in EF Card_Download, in the Tachograph and, if applicable, Tachograph_G2 DFs.

— When downloading a workshop card, reset the calibration counter in EF Card_Download in the Tachograph and, if applicable, Tachograph_G2 DFs.

— When downloading a workshop card the EF `Sensor_Installation_Data` in the `Tachograph` and, if applicable, `Tachograph_G2` DFs shall not be downloaded.**]**

3.3.1 *Initialisation sequence*

DDP_036 The IDE shall initiate the sequence as follows:

Card	Direction	IDE/IFD	Meaning/Remarks
	#	Hardware reset	
ATR	#		

It is optional to use PPS to switch to a higher baud rate as long as the ICC supports it.

3.3.2 *Sequence for un-signed data files*

DDP_037 **[**[F1] The sequence to download EFs ICC, IC, Card_Certificate (or CardSignCertificate for DF Tachograph_G2), CA_Certificate and Link_Certificate (for DF Tachograph_G2 only) is as follows:**]**

Card	Direction	IDE/IFD	Meaning/Remarks
	#	**Select File**	Select by File identifiers
OK	#		
	#	**Read Binary**	If the file contains more data than the buffer size of the reader or the card the command has to be repeated until the complete file is read.
File Data OK	#	Store data to ESM	according to **3.4** Data storage format

Note 1: Before selecting the Card_Certificate (or CardSignCertificate) EF, the Tachograph Application must be selected (selection by AID).

Note 2: Selecting and reading a file may also be performed in one step using a Read Binary command with a short EF identifier.

3.3.3 *Sequence for Signed data files*

DDP_038 The following sequence shall be used for each of the following files that has to be downloaded with their signature:

[[F1] **Card**	**Dir**	**IDE / IFD**	**Meaning / Remarks**
	⇦	**Select File**	

OK	⇨		
	⇦	**Perform Hash of File**	— Calculates the hash value over the data content of the selected file using the prescribed hash algorithm in accordance with Appendix 11, part A or B. This command is not an ISO-Command.
Calculate Hash of File and store Hash value temporarily			
OK	⇨		
	⇦	**Read Binary**	If the file contains more data than the buffer of the reader or the card can hold, the command has to be repeated until the complete file is read.
File Data **OK**	⇨	Store received data to ESM	according to **3.4** Data storage format
	⇦	**PSO: Compute Digital Signature**	
Perform Security Operation ‘Compute Digital Signature’ using the temporarily stored Hash value			
Signature **OK**	⇨	Append data to the previous stored data on the ESM	according to **3.4** Data storage format**]**

Note: Selecting and reading a file may also be performed in one step using a Read Binary command with a short EF identifier. In this case the EF may be selected and read before the command Perform Hash of File is applied.

3.3.4 *Sequence for resetting the calibration counter.*

DDP_039 The sequence to reset the `NoOfCalibrationsSinceDownload` counter in the EF `Card_Download` in a workshop card is the following:

Card	Dir	IDE/IFD	Meaning/Remarks
	#	**Select File** EF Card_Download	Select by File identifiers
OK	#		
	#	**Update Binary** NoOfCalibrationsSinceDownload = ‘00 00’	
resets card download number			
OK	#		

Note: Selecting and updating a file may also be performed in one step using an Update Binary command with a short EF identifier.

3.4. **Data storage format**

3.4.1 *Introduction*

DDP_040 The downloaded data has to be stored according to the following conditions:

— The data shall be stored transparent. This means that the order of the bytes as well as the order of the bits inside the byte that are transferred from the card has to be preserved during storage.

— All files of the card downloaded within a download session are stored in one file on the ESM.

3.4.2 *File format*

DDP_041 The file format is a concatenation of several TLV objects.

DDP_042 The tag for an EF shall be the FID plus the appendix „00“.

DDP_043 The tag of an EF's signature shall be the FID of the file plus the appendix „01“.

DDP_044 The length is a two byte value. The value defines the number of bytes in the value field. The value „FF FF“ in the length field is reserved for future use.

DDP_045 When a file is not downloaded nothing related to the file shall be stored (no tag and no zero length).

[F1DDP_046 A signature shall be stored as the next TLV object directly after the TLV object that contains the data of the file.

Definition	Meaning	Length
FID (2 Bytes) || ‘00’	Tag for EF (FID) in the `Tachograph` or for	3 Bytes

	common information of the card	
FID (2 Bytes) || '01'	Tag for Signature of EF (FID) in the Tachograph DF	3 Bytes
FID (2 Bytes) || '02'	Tag for EF (FID) in the Tachograph_G2 DF	3 Bytes
FID (2 Bytes) || '03'	Tag for Signature of EF (FID) in the Tachograph_G2 DF	3 Bytes
xx xx	Length of Value field	2 Bytes

Example of data in a download file on an ESM:

Tag	**Length**	**Value**
00 02 00	00 11	— Data of EF ICC
C1 00 00	00 C2	— Data of EF Card_Certificate
		— …
05 05 00	0A 2E	Data of EF Vehicles_Used (in the Tachograph DF)
05 05 01	00 80	Signature of EF Vehicles_Used (in the Tachograph DF)
05 05 02	0A 2E	Data of EF Vehicles_Used in the Tachograph_G2 DF
05 05 03	xx xx	Signature of EF Vehicles_Used in the Tachograph_G2 DF]

4. DOWNLOADING A TACHOGRAPH CARD VIA A VEHICLE UNIT.

DDP_047 The VU must allow for downloading the content of a driver card inserted to a connected IDE.

DDP_048 The IDE shall send a ‘Transfer Data Request Card Download’ message to the VU to initiate this mode (see 2.2.2.9).

[F1 DDP_049 First generation driver cards: Data shall be downloaded using the first generation data download protocol, and downloaded data shall have the same format as data downloaded from a first generation vehicle unit.

Second generation driver cards: the VU shall then download the whole card, file by file, in accordance with the card downloading protocol defined in paragraph 3, and forward all data received from the card to the IDE within the appropriate TLV file format (see 3.4.2) and encapsulated within a ‘Positive Response Transfer Data’ message.]

DDP_050 The IDE shall retrieve card data from the ‘Positive Response Transfer Data’ message (stripping all headers, SIDs, TREPs, sub message counters, and checksums) and store them within one single physical file as described in paragraph 2.3.

DDP_051 The VU shall then, as applicable, update the `Control_Activity_Data` or the `Card_Download` file of the driver card.

(**1**) The card inserted will trigger the appropriate access rights to the downloading function and to the data. It shall, however, be possible to download data from a driver card inserted into one of the VU slots when no other card type is inserted in the other slot.

Status:
Point in time view as at 31/12/2020.

Changes to legislation:
There are outstanding changes not yet made to Commission Implementing Regulation (EU) 2016/799. Any changes that have already been made to the legislation appear in the content and are referenced with annotations.