

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 6-5: Application layer protocol specification – Type 5 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 6-5: Spécification du protocole de la couche application – Éléments
de type 5**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 6-5: Application layer protocol specification – Type 5 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 6-5: Spécification du protocole de la couche application – Éléments
de type 5**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XE

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-1759-7

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	7
INTRODUCTION.....	9
1 Scope.....	10
1.1 General.....	10
1.2 Specifications.....	11
1.3 Conformance.....	11
2 Normative references	11
3 Terms, definitions, symbols, abbreviations and conventions	12
3.1 Terms and definitions from other ISO/IEC standards	12
3.2 IEC 61158-1 terms	13
3.3 Abbreviations and symbols.....	16
3.4 Conventions	17
3.5 Conventions used in state machines	18
4 Protocol.....	19
4.1 Overview	19
4.2 FAL syntax description	19
4.3 Transfer syntax	19
4.4 FAL protocol state machine structure	71
4.5 SMK state machine	71
4.6 VCR state machine.....	88
4.7 FAL service protocol machine (FSPM).....	89
4.8 Application relationship protocol machines (ARPMs)	89
4.9 DLL mapping protocol machine (DMPM).....	103
Bibliography.....	108
Figure 1 – State transition diagram for SMK.....	73
Figure 2 – State transition diagram of client / server ARPM	92
Figure 3 – State transition diagram of the publisher / subscriber ARPM	98
Figure 4 – State transition diagram of DMPM.....	104
Table 1 – Conventions used for state machines	18
Table 2 – Data types.....	20
Table 3 – Data types.....	20
Table 4 – APDU header format	21
Table 5 – FDA address use.....	22
Table 6 – FDA address header field APDUs sent by a client VCR endpoint.....	23
Table 7 – FDA address header field APDUs sent by a server VCR endpoint	24
Table 8 – FDA address header field APDUs sent by a publisher VCR endpoint.....	24
Table 9 – FDA address header field APDUs sent by a report source VCR endpoint	25
Table 10 – APDU trailer fields.....	25
Table 11 – Request APDU parameters.....	28
Table 12 – SMK FDA address values	30
Table 13 – SMK FDA address values	30

Table 14 – Request APDU parameters.....	31
Table 15 – SMK FDA address values for SM identify	32
Table 16 – SMK FDA address values for SMK set assignment info request APDUs	33
Table 17 – SMK clear address request APDU parameters.....	33
Table 18 – SMK FDA address values for SMK set assignment info request APDUs	33
Table 19 – SMK set assignment info request APDU parameters	34
Table 20 – SMK set assignment info response APDU parameters.....	35
Table 21 – SMK FDA address values for SMK device clear assignment Info APDUs.....	36
Table 22 – SMK clear assignment info request APDU parameters	36
Table 23 – SMK FDA address values for SMK device annunciation request APDUs.....	36
Table 24 – SMK device annunciation request APDU parameters.....	37
Table 25 – Initiate request APDU parameters	39
Table 26 – Initiate response APDU parameters.....	40
Table 27 – Abort request APDU parameters	40
Table 28 – Get response APDU parameters.....	40
Table 29 – Identify response APDU parameters.....	41
Table 30 – Get OD request APDU parameters	41
Table 31 – Get OD response APDU parameters.....	41
Table 32 – Initiate put OD request APDU parameters	42
Table 33 – Put OD request APDU parameters.....	42
Table 34 – Generic initiate download sequence request APDU parameters.....	43
Table 35 – Generic download segment request APDU parameters.....	43
Table 36 – Generic terminate download sequence request APDU parameters	44
Table 37 – Response APDU parameters	44
Table 38 – Initiate download sequence request APDU parameters.....	44
Table 39 – Download segment request APDU parameters	45
Table 40 – Download segment response APDU parameters.....	45
Table 41 – Terminate download sequence request APDU parameters	45
Table 42 – Initiate upload sequence request APDU parameters	46
Table 43 – Upload segment request APDU parameters.....	46
Table 44 – Upload segment response APDU parameters	47
Table 45 – Terminate upload sequence request APDU parameters.....	47
Table 46 – Request domain download request APDU parameters	47
Table 47 – Request domain upload request APDU parameters	48
Table 48 – Create program invocation request APDU parameters.....	48
Table 49 – Create program invocation response APDU parameters	49
Table 50 – Delete program invocation request APDU parameters	49
Table 51 – Start request APDU parameters	49
Table 52 – Stop request APDU parameters.....	50
Table 53 – Resume request APDU parameters	50
Table 54 – Reset request APDU parameters.....	50
Table 55 – Kill request APDU parameters	51
Table 56 – Read request APDU parameters.....	51

Table 57 – Read response APDU parameters	51
Table 58 – Read with subindex request APDU parameters.....	52
Table 59 – Read with subindex response APDU parameters	52
Table 60 – Write request APDU parameters.....	52
Table 61 – Write with subindex request APDU parameters.....	52
Table 62 – Define variable list request APDU parameters	53
Table 63 – Define variable list response APDU parameters	53
Table 64 – Delete variable list request APDU parameters	53
Table 65 – Information report request APDU parameters	54
Table 66 – Information report with subindex request APDU parameters	54
Table 67 – Information report on change request APDU parameters	54
Table 68 – Information report on change with subindex request APDU parameters	55
Table 69 – Event notification request APDU parameters	55
Table 70 – Alter event condition monitoring request APDU parameters.....	55
Table 71 – Acknowledge event notification request APDU parameters	56
Table 72 – LAN redundancy diagnostic message request APDU parameters	56
Table 73 – LAN redundancy get information response APDU parameters	58
Table 74 – LAN redundancy get statistics request APDU parameters.....	59
Table 75 – Object description header.....	61
Table 76 – Null object	61
Table 77 – Structure of the list of object descriptions	62
Table 78 – Structure of a load region in the S-OD	62
Table 79 – Structure of a function invocation in the DP-OD.....	63
Table 80 – Structure of an event in the S-OD.....	64
Table 81 – Structure of a data type in the ST-OD	64
Table 82 – Structure of a data type structure description in the ST-OD	65
Table 83 – Structure of a simple variable in the S-OD.....	65
Table 84 – Structure of an array in the S-OD	66
Table 85 – Structure of a record in the S-OD	66
Table 86 – Structure of a variable list in the DV-OD	67
Table 87 – Common error parameters.....	68
Table 88 – PI error parameters	68
Table 89 – OD error parameters	68
Table 90 – Error class and error code values	69
Table 91 – SMKPM service primitives	72
Table 92 – SMKPM states.....	73
Table 93 – SMKPM state table – initialization	73
Table 94 – SMKPM state table – receive transitions.....	74
Table 95 – SMKPM state table – internal events	79
Table 96 – HseRepeatTimerExpires ()	80
Table 97 – RcvNewNetworkAddress (interface, address)	80
Table 98 – RcvMsg ().....	80
Table 99 – SntpSyncLost ().....	80

Table 100 – AddressToClear (sm_svc)	81
Table 101 – AssignmentInfo_Set ()	81
Table 102 – ConfigurationSessionActive ()	81
Table 103 – DeviceRedundancyState ()	81
Table 104 – DevId_Match (sm_svc)	82
Table 105 – DuplicateQueryIdMatch (sm_svc)	82
Table 106 – DuplicatePdTagDetected ()	82
Table 107 – FdaAddressType (sm_svc)	82
Table 108 – IsValid (sm_svc)	83
Table 109 – NetworkAddressChange (interface, address)	83
Table 110 – NumberOfAssignedAddresses ()	83
Table 111 – OperationalRestore ()	83
Table 112 – PdTag_Match (sm_svc)	83
Table 113 – PdTagDeviceIndex_Check (sm_svc)	84
Table 114 – Query_Match (sm_svc)	84
Table 115 – QueryType (sm_svc)	84
Table 116 – SmCacheEntry (sm_svc)	84
Table 117 – Clear_Address (interface_to_clear)	85
Table 118 – Clear_DuplicatePdTagFlag ()	85
Table 119 – Get_AddlCode ()	85
Table 120 – New_Address (interface, address)	85
Table 121 – Restart_HseRepeatTimer ()	86
Table 122 – Restore_Defaults ()	86
Table 123 – Send_SM_CommonErrorRsp (sm_service_type, svc_spec_params)	86
Table 124 – Send_SM_ReqRspMessage (sm_svc)	86
Table 125 – Set_Assignment_Data (sm_svc)	86
Table 126 – Set_DuplicatePdTagFlag ()	87
Table 127 – SvcType (sm_svc)	87
Table 128 – Additional code used by error class and code	87
Table 129 – Additional code parameter IDs	87
Table 130 – Primitives issued by FSPM to ARPM	89
Table 131 – Primitives issued by ARPM to FSPM	90
Table 132 – Parameters used with primitives exchanged between FSPM and ARPM	90
Table 133 – Client / Server ARPM states	92
Table 134 – Client / server ARPM state table – sender transitions	93
Table 135 – Client / server ARPM state table – receiver transitions	94
Table 136 – Primitives issued by FSPM to ARPM	96
Table 137 – Primitives issued by ARPM to FSPM	96
Table 138 – Parameters used with primitives exchanged between FSPM and ARPM	97
Table 139 – Publisher / subscriber ARPM states	98
Table 140 – MulticastARPM state table – sender transitions	99
Table 141 – MulticastARPM state table – receiver transitions	99
Table 142 – BuildFAL-ErrPDU()	100

Table 143 – BuildFAL-ReqRspPDU()	100
Table 144 – GetArepld()	100
Table 145 – ConfigurationArCheckOK()	100
Table 146 – FAL_Pdu_BufferSize()	100
Table 147 – FAL_Pdu_Confirmed()	101
Table 148 – FAL_Pdu_DuplicateMsg ()	101
Table 149 – FAL_Pdu_GetVcrlid()	101
Table 150 – FAL_Pdu_InactivityCloseTime()	101
Table 151 – FAL_Pdu_TransmitDelayTime()	101
Table 152 – FAL_Pdu_SvcType()	101
Table 153 – FAL_Pdu_RemoteAddress()	102
Table 154 – FAL_Pdu_TrailerFields()	102
Table 155 – FAL_Pdu_ServiceSpecificParameters()	102
Table 156 – FAL_Pdu_Valid()	102
Table 157 – MaxOutstandingReached()	102
Table 158 – StartInactivityCloseTimer()	102
Table 159 – Primitives issued by ARPM to DMPM	103
Table 160 – Primitives issued by DMPM to ARPM	103
Table 161 – Parameters used with primitives exchanged between ARPM and DMPM	103
Table 162 – Primitives exchanged between the socket model and DMPM	104
Table 163 – Parameters of DMPM/socket model primitives	104
Table 164 – DMPM state descriptions	104
Table 165 – DMPM state table – sender transitions	105
Table 166 – DMPM state table – receiver transitions	106
Table 167 – ConnectionOriented	106
Table 168 – GetBufferedData	106
Table 169 – GetConnectionId	106
Table 170 – LoadBuffer	107
Table 171 – RemainingBufferSizeCheck	107
Table 172 – StartTransmitDelayTimer	107

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 6-5: Application layer protocol specification –
Type 5 elements**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

Attention is drawn to the fact that the use of the associated protocol type is restricted by its intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by its intellectual-property-right holders.

NOTE Combinations of protocol types are specified in IEC 61784-1 and IEC 61784-2.

International Standard IEC 61158-6-5 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2007. This edition constitutes a technical revision. The main change with respect to the previous edition is listed below:

- Add support for message padding
- Clarified encoding rules
- Clarified open session service
- Time synchronization now present in annunciation message
- Additional redundancy options in annunciation message

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/764/FDIS	65C/774/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with ISO/IEC Directives, Part 2.

A list of all the parts of the IEC 61158 series, under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under <http://webstore.iec.ch> in the data related to the specific publication. At this date, the publication will be:

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC 61158-1.

The application protocol provides the application service by making use of the services available from the data-link or other immediately lower layer. The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer application entities (AEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- as a guide for implementors and designers;
- for use in the testing and procurement of equipment;
- as part of an agreement for the admittance of systems into the open systems environment;
- as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-5: Application layer protocol specification – Type 5 elements

1 Scope

1.1 General

The fieldbus Application Layer (FAL) provides user programs with a means to access the fieldbus communication environment. In this respect, the FAL can be viewed as a “window between corresponding application programs.”

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 5 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible behavior provided by the Type 5 fieldbus Application Layer in terms of

- a) the abstract syntax defining the application layer protocol data units conveyed between communicating application entities,
- b) the transfer syntax defining the application layer protocol data units conveyed between communicating application entities,
- c) the application context state machine defining the application service behavior visible between communicating application entities; and
- d) the application relationship state machines defining the communication behavior visible between communicating application entities; and.

The purpose of this standard is to define the protocol provided to

- 1) define the wire-representation of the service primitives defined in IEC 61158-5-5, and
- 2) define the externally visible behavior associated with their transfer.

This standard specifies the protocol of the Type 5 IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

FAL services and protocols are provided by FAL application-entities (AE) contained within the application processes. The FAL AE is composed of a set of object-oriented Application Service Elements (ASEs) and a Layer Management Entity (LME) that manages the AE. The ASEs provide communication services that operate on a set of related application process object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can

send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

1.2 Specifications

The principal objective of this standard is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-5.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of protocols standardized in IEC 61158-6 series.

1.3 Conformance

This standard does not specify individual implementations or products, nor does it constrain the implementations of application layer entities within industrial automation systems. Conformance is achieved through implementation of this application layer protocol specification.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE All parts of the IEC 61158 series, as well as IEC 61784-1 and IEC 61784-2 are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

IEC 61158-1, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-5-5, *Industrial communication networks – Fieldbus specifications – Part 5-5: Application layer service definition – Type 5 elements*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model – Part 1: The Basic Model*

ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*¹

ISO/IEC 9545, *Information technology – Open Systems Interconnection – Application Layer structure*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

IETF RFC 791, *Internet Protocol*; available at <<http://www.ietf.org>>

¹ Withdrawn

3 Terms, definitions, symbols, abbreviations and conventions

For the purposes of this document, the following terms, definitions, symbols, abbreviations and conventions apply.

3.1 Terms and definitions from other ISO/IEC standards

3.1.1 Terms and definitions from ISO/IEC 7498-1

- a) abstract syntax
- b) application entity
- c) application process
- d) application protocol data unit
- e) application service element
- f) application entity invocation
- g) application process invocation
- h) application transaction
- i) presentation context
- j) real open system
- k) transfer syntax

3.1.2 Terms and definitions from ISO/IEC 9545

- a) application-association
- b) application-context
- c) application context name
- d) application-entity-invocation
- e) application-entity-type
- f) application-process-invocation
- g) application-process-type
- h) application-service-element
- i) application control service element

3.1.3 Terms and definitions from ISO/IEC 8824

- a) object identifier
- b) type
- c) value
- d) simple type
- e) structured type
- f) component type
- g) tag
- h) Boolean type
- i) true
- j) false
- k) integer type
- l) bitstring type
- m) octetstring type
- n) null type
- o) sequence type
- p) sequence of type
- q) choice type
- r) tagged type
- s) any type
- t) module
- u) production

3.1.4 Terms and definitions from ISO/IEC 8825

- a) encoding (of a data value)
- b) data value
- c) identifier octets (the singular form is used in this standard)
- d) length octet(s) (both singular and plural forms are used in this standard)
- e) contents octets

3.2 IEC 61158-1 terms

For the purposes of the present document, the following IEC 61158-1 terms apply.

3.2.1

application

function or data structure for which data is consumed or produced

3.2.2

application layer interoperability

capability of application entities to perform coordinated and cooperative operations using the services of the FAL

3.2.3

application object

object class that manages and provides the run time exchange of messages across the network and within the network device

Note 1 to entry: Multiple types of application object classes may be defined.

3.2.4

application process

part of a distributed application on a network, which is located on one device and unambiguously addressed

3.2.5

application process identifier

component that distinguishes multiple application processes used in a device

3.2.6

application process object

component of an application process that is identifiable and accessible through an FAL application relationship

Note 1 to entry: Application process object definitions are composed of a set of values for the attributes of their class (see the definition for Application Process Object Class Definition). Application process object definitions may be accessed remotely using the services of the FAL Object Management ASE. FAL Object Management services can be used to load or update object definitions, to read object definitions, and to dynamically create and delete application objects and their corresponding definitions.

3.2.7

application process object class

class of application process objects defined in terms of the set of their network-accessible attributes and services

3.2.8

application relationship

cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation

Note 1 to entry: This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities.

3.2.9

application relationship application service element

application-service-element that provides the exclusive means for establishing and terminating all application relationships

3.2.10

application relationship endpoint

context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

Note 1 to entry: Each application process involved in the application relationship maintains its own application relationship endpoint.

3.2.11

attribute

description of an externally visible characteristic or feature of an object

Note 1 to entry: The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behaviour of an object. Attributes are divided into class attributes and instance attributes.

3.2.12

behaviour

indication of how the object responds to particular events

Note 1 to entry: Its description includes the relationship between attribute values and services.

3.2.13

class

set of objects, all of which represent the same kind of system component

Note 1 to entry: A class is a generalisation of the object; a template for defining variables and methods. All objects in a class are identical in form and behaviour, but usually contain different data in their attributes.

3.2.14

class attributes

attribute that is shared by all objects within the same class

3.2.15

class code

unique identifier assigned to each object class

3.2.16

class specific service

service defined by a particular object class to perform a required function which is not performed by a common service.

Note 1 to entry: A class specific object is unique to the object class which defines it.

3.2.17

client

(a) object which uses the services of another (server) object to perform a task

(b) initiator of a message to which a server reacts, such as the role of an AR endpoint in which it issues confirmed service request APDUs to a single AR endpoint acting as a server

3.2.18

conveyance path

unidirectional flow of APDUs across an application relationship

3.2.19

cyclic

term used to describe events which repeat in a regular and repetitive manner

3.2.20**dedicated AR**

AR used directly by the FAL User

Note 1 to entry: On Dedicated ARs, only the FAL Header and the user data are transferred.

3.2.21**device**

physical hardware connection to the link

Note 1 to entry: A device may contain more than one node.

3.2.22**device profile**

collection of device dependent information and functionality providing consistency between similar devices of the same device type

3.2.23**dynamic AR**

AR that requires the use of the AR establishment procedures to place it into an established state

3.2.24**endpoint**

one of the communicating entities involved in a connection

3.2.25**error**

a discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.2.26**error class**

general grouping for error definitions

Note 1 to entry: Error codes for specific errors are defined within an error class.

3.2.27**error code**

identification of a specific type of error within an error class

3.2.28**FAL subnet**

networks composed of one or more data link segments

Note 1 to entry: They are permitted to contain bridges, but not routers. FAL subnets are identified by a subset of the network address.

3.2.29**logical device**

certain FAL class that abstracts a software component or a firmware component as an autonomous self-contained facility of an automation device

3.2.30**management information**

network-accessible information that supports managing the operation of the fieldbus system, including the application layer

Note 1 to entry: Managing includes functions such as controlling, monitoring, and diagnosing.

3.2.31

network

series of nodes connected by some type of communication medium

Note 1 to entry: The connection paths between any pair of nodes can include repeaters, routers and gateways.

3.2.32

peer

role of an AR endpoint in which it is capable of acting as both client and server

3.2.33

pre-defined AR endpoint

AR endpoint that is defined locally within a device without use of the create service

Note 1 to entry: Pre-defined ARs that are not pre-established are established before being used.

3.2.34

pre-established AR endpoint

AR endpoint that is placed in an established state during configuration of the AEs that control its endpoints

3.2.35

publisher

role of an AR endpoint in which it transmits APDUs onto the fieldbus for consumption by one or more subscribers

Note 1 to entry: The publisher may not be aware of the identity or the number of subscribers and it may publish its APDUs using a dedicated AR. Two types of publishers are defined by this standard, Pull Publishers and Push Publishers, each of which is defined separately.

3.2.36

server

- a) role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request
- b) object which provides services to another (client) object

3.2.37

service

operation or function than an object and/or object class performs upon request from another object and/or object class

Note 1 to entry: A set of common services is defined and provisions for the definition of object-specific services are provided. Object-specific services are those which are defined by a particular object class to perform a required function which is not performed by a common service.

3.2.38

subscriber

role of an AREP in which it receives APDUs produced by a publisher

Note 1 to entry: Two types of subscribers are defined by this standard, pull subscribers and push subscribers, each of which is defined separately.

3.3 Abbreviations and symbols

AE	Application Entity
AL	Application Layer
ALME	Application Layer Management Entity
ALP	Application Layer Protocol
APO	Application Object
AP	Application Process

APDU	Application Protocol Data Unit
API	Application Process Identifier
AR	Application Relationship
AREP	Application Relationship End Point
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
Cnf	Confirmation
DL-	(as a prefix) data-link-
DLC	Data-link Connection
DLCEP	Data-link Connection End Point
DLL	Data-link layer
DLM	Data-link-management
DLSAP	Data-link Service Access Point
DLSDU	DL-service-data-unit
FAL	Fieldbus Application Layer
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
LME	Layer Management Entity
OSI	Open Systems Interconnect
QoS	Quality of Service
Req	Request
Rsp	Response
SAP	Service Access Point
SDU	Service Data Unit
SMIB	System Management Information Base
SMK	System Management Kernel
VFD	Virtual Field Device

3.4 Conventions

3.4.1 General concept

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of three parts: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5 series. The protocol specification for each of the ASEs is defined in this standard.

The class definitions define the attributes of the classes supported by each ASE. The attributes are accessible from instances of the class using the Management ASE services specified in IEC 61158-5 standard. The service specification defines the services that are provided by the ASE.

This standard uses the descriptive conventions given in ISO/IEC 10731.

3.4.2 Conventions for class definitions

The data-link layer mapping definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is defined in IEC 61158-5-5.

3.4.3 Abstract syntax conventions

When the "optionalParametersMap" parameter is used, a bit number which corresponds to each OPTIONAL or DEFAULT production is given as a comment.

3.5 Conventions used in state machines

The state machines are described in Table 1.

Table 1 – Conventions used for state machines

#	Current state	Event / condition => action	Next state
Name of this transition.	The current state to which this state transition applies	Events or conditions that trigger this state transaction. => The actions that are taken when the above events or conditions are met. The actions are always indented below events or conditions	The next state after the actions in this transition is taken

The conventions used in the state machines are as follows:

:= Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event.

xxx A parameter name.

Example:

- Identifier := reason
1) means value of a 'reason' parameter is assigned to a parameter called 'Identifier.'

"xxx" Indicates fixed value.

Example:

- Identifier := "abc"
2) means value "abc" is assigned to a parameter named 'Identifier.'
3)

= A logical condition to indicate an item on the left is equal to an item on the right.

< A logical condition to indicate an item on the left is less than the item on the right.

> A logical condition to indicate an item on the left is greater than the item on the right.

<> A logical condition to indicate an item on the left is not equal to an item on the right.

&& Logical "AND"

|| Logical "OR"

This construct allows the execution of a sequence of actions in a loop within one transition. The loop is executed for all values from start_value to end_value.

Example:

- for (Identifier := start_value to end_value)
actions
4) endfor

This construct allows the execution of alternative actions depending on some condition (which might be the value of some identifier or the outcome of a previous action) within one transition.

```
Example:
  If (condition)
    actions
  else
    actions
5)   endif
```

Readers are strongly recommended to refer to the subclauses for the AREP attribute definitions, the local functions, and the FAL-PDU definitions to understand protocol machines. It is assumed that readers have sufficient knowledge of these definitions, and they are used without further explanations.

4 Protocol

4.1 Overview

The Type 5 services and protocols are provided by the Type 5 AE, referred to as the *Field Device Access (FDA) Agent* throughout this specification.

The FDA Agent conveys Type 9 APDUs using Type 9 Application Relationships, referred to as *FDA Sessions*, or more simply as *sessions* throughout this Type 5 specification.

The FDA Agent may provide gateway functions that map between Type 5 services and corresponding services that operate over a Type “X” DLL. Networks that employ the Type “X” DLL are referred to as *Type “X” networks*.

4.2 FAL syntax description

4.2.1 PDU abstract syntax

The abstract syntax of APDUs is combined with their transfer syntax and is specified in 4.3.

4.2.2 Abstract syntax of data types

The abstract syntax of data types is combined with their transfer syntax and is specified in 4.3.

4.3 Transfer syntax

4.3.1 General

The transfer syntax combines the specification of the abstract syntax and their encodings as a set of fixed format APDUs. Each APDU contains a header and an APDU body. An optional trailer is also defined. The presence of the trailer is indicated in the APDU header.

The format of the APDU header and trailer is specified in 4.3.3 and 4.3.4. The APDU header contains the fields common to all messages that are mandatory. The APDU trailer contains the fields common to all APDUs that are optional.

The formats of the APDU bodies are specified in 4.3.5. If a service request or response APDU has no parameters, it is stated in the subclause that describes the APDU body. If the service has no response or error APDU body associated with it, then the corresponding subclause is not present. Common error parameters are used in many service-specific APDUs. Their definitions are specified in 4.3.6.1.

User data, when present in an APDU, is defined as a service-specific APDU body parameter. The content and length of the user data are dependent upon the service and the object being accessed. Therefore, neither is specified as part of the APDU body. However, the user data length can be calculated from the APDU length contained in the APDU header.

The APDU header, trailer, and bodies are composed of a subset of data types defined in IEC 61158-5. The encodings for these data types are defined next, followed by the definitions of the header, trailer, and APDU bodies.

4.3.2 Data type encodings

The following data types are used in APDUs as listed in Table 2. Other data types may be conveyed as user data.

Table 2 – Data types

Data Type	Octet Length	Comment
Boolean	1	0 = False, Non-Zero = TRUE
Integer8	1	
Integer16	2	
Unsigned8	1	
Unsigned16	2	
Unsigned32	4	
VisibleString	-	The length of a VisibleString is defined by the field definition.
OctetString	-	The length of an OctetString is defined by the field definition.
Time value	8	

For all data types, the bit and octet order of transmission is as defined in Type 9. The unsigned integer value for each bit position is shown in Table 3.

NOTE The resulting encoding is the same as that the encoding specified in Appendix B of RFC 791 – Internet Protocol. The bit and byte order specified in RFC 791 Appendix B is the big endian octet ordering (most significant byte first for integers) and bit ordering within bytes is most significant bit first. Although the bit numbering is opposite of that specified in FMS, both specify the high order bit (the leftmost bit) as the most significant bit. In the descriptions that follow, bit 1 is the most significant bit in each byte. The unsigned integer value for each bit position is shown in the table below.

Table 3 – Data types

Bit Position	Hex Value	Decimal Value
8	0x80	128
7	0x40	64
6	0x20	32
5	0x10	16
4	0x08	8
3	0x04	4
2	0x02	2
1	0x01	1

4.3.3 APDU header

4.3.3.1 APDU header format

The header is the same for each APDU and has a fixed length.

All header fields are mandatory.

The format of the header is shown in Table 4.

Table 4 – APDU header format

Parameter name	Octet offset	Data type	Octet length	Description
Type 5 APDU Version	0	Unsigned8	1	Specifies the Version number of the Type 5 APDU format. The version number of the Type 5 APDUs specified in this document is 1. 1 = Version 1
Options	1	Unsigned8	1	Bit 8: 1 = APDU Number present in the Trailer Bit 7: 1 = Invoke Id present in the Trailer (this bit shall be set for all client / server sessions) Bit 6: 1 = Time Stamp present in the Trailer Bit 5: Reserved, set to 0 Bit 4: 1 = Extended Control Field present in the Trailer Bit 3-1: Pad Length = Specifies the number of pad bytes that are inserted between the user data and the trailer fields. Each pad byte is set to binary zero.
ASE Id And Confirmed Msg Type	2	Unsigned8	1	Bits 8–3: ASE Id – Used to specify the ASE whose service is being conveyed by the APDU. They are interpreted as a 6-bit unsigned integer with the following values: 0 = Unused 1 = AR ASE 2 = SMK ASE 3 = Type 9 ASEs/Type 5 VFD ASE 4 = LAN Redundancy ASE 5-63 : Reserved Bits 1, 2: Confirmed Msg Type – Used to differentiate between request, response, and error APDUs. Not used with unconfirmed services. These bits are interpreted as a 2-bit unsigned integer with the following values: 0 = Request APDU 1 = Response APDU 2 = Error APDU 3 = Reserved
Service	3	Unsigned8	1	Specifies the service within the specified protocol. Bit 8: Confirmed Flag (Confirmed=1) and (Unconfirmed=0). When this flag is set for a request APDU, the corresponding response (either a response APDU or an error APDU) is always returned after the request has been processed to indicate the result of the processing. Bits 7-1 Service Id of the service (7-bit unsigned integer). The value used for each Service Id is specified in its APDU Body specification below. It is the number in parenthesis that follows the service name in the header for each service. All other values are reserved.
FDA Address	4	Unsigned32	4	The format and use of the FDA Address is dependent on the ASE Id and the type of S-VFD Context. The next subclause describes this field.
APDU Length	8	Unsigned32	4	Specifies the number of octets contained in the entire APDU, including the header and the trailer.

4.3.3.2 FDA address use

4.3.3.2.1 FDA address use summary

Subclause 4.3.3.2.1 specifies the values for the FDA Address field of the header. See Table 5 for details. In the descriptions that follow, a Type 9 device is a device with the Type 9 AL and

a DLL whose addressing can be represented as a Type 1 DLL address. A Type 9 SMK is the SMK in a Type 9 device.

Table 5 – FDA address use

ASE	VCR	Bits	Use
AR	N/A	32-1	Service-specific.
LAN Redundancy	N/A	32-1	Not Used, set to 0.
SMK	N/A	32-17 16-1	Link Id Set to 0 if the destination SMK is located in the Type 5 device receiving the APDU. If the device is a Linking Device, and the destination SMK is in a Type 9 device attached to the Linking Device, then the FDA Address field contains the link id used to access the Type 9 device. Node.Selector The node value is the high order 8 bits and the selector value is the low order 8 bits, when taken as a 16-bit unsigned integer value. When used to identify a Type 5 SMK (when the Link Id is zero), the node value is set to 0 and the selector value is set to 2. When used to identify a Type 9 SMK (when the Link Id is non-zero), this is the individual or group Type 9 node.selector address. See the description of the Find Tag Query APDU for a detailed description of its use of the FDA Address.
Type 9	Client/Server	32-17 16-1	Link Id If the server S-VFD is located in the Type 5 device, then this is 0. If the device is a Linking Device, and the destination VFD is in a Type 9 device attached to the Linking Device, then this is the Link Id of the link to which the Type 9 device is attached. Type 5 Selector Initiate Request: The use of this field is dependent on the Connect Option selected for the request. See the description of the Initiate APDU for the values used. Initiate Response: The Type 5 FAL AE returns the VCR Id of the initiated VCR. All Others: The Selector returned in the Initiate Response.
Type 9	Publisher / Subscriber	32-17 16-1	Link Id If the publisher VFD is located in the Type 5 device sending the APDU, then this is combined with the Selector bits to form a 32-bit flat Dlcep address. If the device is a Linking Device, and the publisher is in a Type 9 device attached to the Linking Device, then this is the Link Id of the link to which the publishing device is attached. Dlcep Selector If the publisher is located in the Type 5 device sending the APDU, then this is part of the 32-bit flat Dlcep address assigned to the published data. If the device is a Linking Device, and the publisher is in a Type 9 device attached to the Linking Device, then this is the Dlcep Selector of the publisher buffer.
Type 9	Report Distribution	32-17 16-1	Link Id If the report source VFD is located in the Type 5 device sending the APDU, then this is combined with the Selector bits to form a 32-bit flat DLSAP address. If the device is a Linking Device, and the report source is in a Type 9 device attached to the Linking Device, then this is the Link Id of the link to which the reporting device is attached. S-VFD Context Id or Dlsap Selector If the report source VFD is located in the Type 5 device sending the APDU, then this is part of the 32-bit flat Dlsap address assigned to the Report

ASE	VCR	Bits	Use
			Source VFD. If the device is a Linking Device, and the report source is in a Type 9 device attached to the Linking Device, then this is the Dlsap Selector of the report source.
NOTE The Link Id, Node, DLCEP Selector, and DLSAP Selectors are all Type 1 DLL address components.			

4.3.3.2.2 APDU sent by a client VCR endpoint

Table 6 illustrates the use of the FDA Address header field in APDUs sent by a client VCR endpoint.

Table 6 – FDA address header field APDUs sent by a client VCR endpoint

Message type	Destination AP	FDA address	
		Bits	Value
Type 5 Initiate Request APDU Connect Option 1	Non-MIB VFD of Type 5 device	Bits 32-17 Bits 16-1	0 (Local Type 5 Device) Generic Selector, Returned by Find Tag Query, or obtained by some other means
	Non-MIB VFD of Type 9 device connected to Linking Device	Bits 32-17 Bits 16-1	Link Id of Type 9 Link Client Type 9 VCR Index
Type 5 Initiate Request APDU Connect Option 2	MIB VFD of Type 5 device	Bits 32-17 Bits 16-1	0 (Local Type 5 Device) 0.0
	MIB VFD of Linking Device Type 9 Link Interface	Bits 32-17 Bits 16-1	Link Id of Type 9 Link 0.0
	MIB VFD of Type 9 device connected to Linking Device	Bits 32-17 Bits 16-1	Link Id of Type 9 Link Node.0
All Other Request Messages	MIB VFD of Type 5 device	Bits 32-1	FDA Address returned in Initiate Response message
	Non-MIB VFD	Bits 32-1	FDA Address returned in Initiate Response message
	MIB VFD of Linking Device Type 9 Link Interface	Bits 32-1	FDA Address returned in Initiate Response message
	MIB VFD of Type 9 device connected to Linking Device	Bits 32-1	FDA Address returned in Initiate Response message
	Non-MIB AP of Type 9 device connected to Linking Device	Bits 32-1	FDA Address returned in Initiate Response message

4.3.3.2.3 APDUs sent by a server AR VCR endpoint

Table 7 illustrates the use of the FDA Address header field in APDUs sent by a server VCR endpoint.

Table 7 – FDA address header field APDUs sent by a server VCR endpoint

Message type	Destination AP	FDA address	
		Bits	Value
Type 5 Initiate Response APDU Connect Option 1	Non-MIB VFD of Type 5 device	Bits 32-17	0 (Local Type 5 Device)
		Bits 16-1	Type 5 VCR Id ¹
	Non-MIB VFD of Type 9 device connected to Linking Device	Bits 32-17	Link Id of Type 9 Link
		Bits 16-1	Type 5 VCR Id ¹
Type 5 Initiate Response APDU Connect Option 2	MIB VFD of Type 5 device	Bits 32-17	0 (Local Type 5 Device)
		Bits 16-1	Type 5 VCR Id ¹
	MIB VFD of Linking Device Type 9 Link Interface	Bits 32-17	Link Id of Type 9 Link Interface
		Bits 16-1	Type 5 VCR Id ¹
	MIB VFD of Type 9 device connected to Linking Device	Bits 32-17 Bits 16-1	Link Id of Type 9 Link Type 5 VCR Id ¹
All Other Response Messages	MIB VFD of Type 5 device	Bits 32-1	Same as Request Message Same as Request Message
	Non MIB VFD of Type 5 device	Bits 32-1	Same as Request Message Same as Request Message
	MIB VFD of Linking Device Type 9 Link	Bits 32-1	Same as Request Message Same as Request Message
	MIB VFD of Type 9 device connected to Linking Device	Bits 32-1	Same as Request Message Same as Request Message
	Non-MIB VFD of Type 9 device connected to Linking Device	Bits 32-1	Same as Request Message Same as Request Message

NOTE The selector value (bits 17-32) returned is the Type 5 VCR Index of the dynamically created VCR.

4.3.3.2.4 APDUs sent by a publisher VCR endpoint

Table 8 illustrates the use of the FDA Address in messages sent by a publisher VCR endpoint.

Table 8 – FDA address header field APDUs sent by a publisher VCR endpoint

Message type	Publisher AP	FDA address	
		Bits	Value
Type 9 Request Message	AP in the Type 5 device	Bits 32-1	Flat 32-bit Type 9 address assigned to the published data buffer
	VFD of Type 9 device connected through a	Bits 32-17	Link Id of Type 9 Link
	Type 9 link of a Linking Device	Bits 16-1	Node.Selector Dlcep of the published data buffer

4.3.3.2.5 Messages sent by a report source VCR endpoint

Table 9 illustrates the use of the FDA Address in APDUs sent by a report source VCR endpoint.

Table 9 – FDA address header field APDUs sent by a report source VCR endpoint

Message type	Report source AP	FDA address	
		Bits	Value
Type 9 Request Message	AP in the Type 5 device	Bits 32-1	Flat 32-bit Type 9 address assigned to the report source VFD.
	VFD of Type 9 device connected through a	Bits 32-17	Link Id of Type 9 Link
	Type 9 link of a Linking Device	Bits 16-1	Node.Selector Disap of the source of the report

4.3.4 Trailer

FDA Sessions define which trailer fields are defined for the Type 9 APDUs that they convey. The use of trailer fields by SMK ASE and LAN Redundancy APDUs is service specific. See the SMK ASE and LAN Redundancy APDU definitions for their use of trailer fields.

The presence of trailer fields is negotiated for Client / Server sessions and configured for Publisher / Subscriber and Report Distribution sessions. Their presence is indicated in the Message Header Options field of the each message. The fields that are present are positioned after the user data in the order shown in Table 10.

Table 10 – APDU trailer fields

Field name	Order	Data type	Octet length	Description
APDU Number	1	Unsigned32	4	Monotonically increasing by 1, 0 based, number of the APDU within a specific session.
Invoke Id	2	Unsigned32	4	Request/response identifier used to match responses with requests. It is required for all request / response exchanges. Invoke Ids are unique within the context of an Application Relationship.
Time Stamp	3	OctetString	8	System Time at which sender created APDU (in Time value format). It can be used by receivers to calculate the one-way transfer time from the sender.
Extended Control Field	4	Unsigned32	4	Reserved For Future Use.

4.3.5 APDU Body

4.3.5.1 General

Each service has a set of APDU bodies that are conveyed in a fixed format in each APDU body. The length of the APDU body formats may vary only if the last field of the APDU body repeats or its length is variable. Variable length fields are supported only as the last field of a APDU body (before the APDU Trailer) so that their offset from the beginning of the APDU body is constant.

The number in parenthesis that follows the name of the service in each subclause header below is the value that is used for the Service Id field in the Type 5 APDU Header.

4.3.5.2 FDA Session ASE Services

4.3.5.2.1 Open Session Service (confirmed service Id = 1)

4.3.5.2.1.1 Service overview

This service is used to open a Client/Server AR between a client AR endpoint and a server AR endpoint. The source and destination address pair identifies each AR endpoint.

The Version field in the Type 5 APDU Header in the request message indicates the version of the FDA Agent that is being requested for the AR. The responder may negotiate the version down.

The Options field in the Type 5 APDU Header in the request indicates the options that are being requested for the AR. The Invoke Id bit is always set. The responder can refuse all other options. Returning 0 in the bit position representing the option indicates refusal.

If the message number option is requested, then the trailer field contains the initial value to be used by both endpoints of the session. This value is returned in the response message. If the requester does not receive a response to the request, and elects to resend the request, it increments the message number in the trailer, while keeping the invoke id the same as the original. This causes the request to be delivered to the AR endpoint created if the first request was received and a positive response was returned. If the original request was not received, or it caused a negative response to be returned (that was not received), then the new request is treated like a new request because there is no existing AR endpoint to process the message.

The FDA Address in the APDU Header is not used and is set to 0.

If the server receives an Open Session request message for a session on which it has already sent an Open Session response message, it closes the session. If not, the following applies.

The Version field in the FDA Message Header in the request message indicates the version of the FDA that is being requested for the session. The responder may negotiate the version down.

The Open Session request message conveys session attributes for validation and use by the responder. If the responder is able to support certain of these, it is permitted to negotiate them, as defined in the Request Message Parameters table below. If the requester is not prepared to operate with the negotiated attributes returned by the responder, it is free to not open the session. It may then reissue an Open Session request with different session attributes, or if using TCP, close the associated TCP connection to close the session, as appropriate.

The Options field in the Message Header in the request message indicates the options that are being requested for the session. The Invoke Id bit is always set. The responder can refuse all other options. Returning 0 in the bit position representing the option indicates refusal.

If the message number option is requested, then the trailer field contains the initial value to be used by both endpoints of the session. This value is returned in the response message. If the requester does not receive a response to the request, and elects to resend the request, it increments the message number in the trailer, while keeping the invoke id the same as the original. This causes the request to be delivered to the session endpoint created if the first request was received and a positive response was returned. This will cause the session to close. If the original request was not received, or it caused a negative response to be returned (that was not received), then the new request is treated like a new request because there is no existing session endpoint to process the message.

For the PadLength option, the requester specifies the alignment boundary for the message. This is done by setting the PadLength to the maximum number of padding bytes that can be inserted and by padding this message accordingly. If this value requests 4-or 8-byte padding and the receiver does not support padding, it may reject the request using “invalid options”, or it may accept the request and return “000” in this field.

000 = No padding

011 = pad to a 4-byte boundary. The maximum number of pad bytes that can be inserted is three. Three bytes of pad are inserted into this message.

111 = pad to 8 byte boundary. The maximum number of pad bytes that can be inserted is seven. Seven bytes of pad are inserted into this message.

The server receives requests at a generic endpoint address and creates a new endpoint at a previously unused address to process each request and to return the response. The newly created endpoint is used for all subsequent APDUs sent and received on the AR.

If a request is received with invalid parameter or unsupported values, a negative response is returned with an error class of “service” and an error code of “parameter-inconsistent” for the offending parameter. This negative response also uses the additional code value of 1, and the additional description to propose acceptable values for each of the parameters indicated below that can be used to open the session. To convey these values the 16 octets of the additional description is interpreted as 4 Unsigned32 integers instead of as a VisibleString. The location of each parameter value in the 16 octets is specified in the descriptions below.

If the responder does not have an available Session Index for the new session, then a negative response is returned with error class = “resource” and error code = “max sessions exceeded”.

If the responder does not have the resources to support the new session, then a negative response is returned with error class = “resource” and error code = “object creation failure” or some other appropriate error code within the “resource” error class.

Successful exchange of Open Session request and response messages results in the establishment of a session that may be used to send FMS request and response messages. If UDP is used, the server returns the response from a previously unused UDP port. This port is then used to send and receive all subsequent messages for the session.

ARs opened for MIB VFD Configuration Use are referred to as configuration ARs. Only one configuration AR is allowed at a time. If a request is received to open a configuration AR and one is already open, or if there is an Type 9 VCR to any MIB VFD (Type 5 or Type 9) with update services supported already open, then the configuration AR is refused. The AR ASE returns a negative response with error class = “service” and error code = “config-access-already-open”.

Opening Publisher / Subscriber and Report Distribution ARs is local, and does not result in the exchange of Open Session Service APDUs. As a result, the Open Session Request APDU parameters for these AR types are preconfigured instead of negotiated.

4.3.5.2.1.2 Request APDU parameters

Table 11 lists the request APDU parameters.

Table 11 – Request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
AR Index	0	Unsigned32	4	This parameter contains the numeric identifier of the client AR endpoint, if one. If not, 0 is used. For the response, this parameter contains the numeric identifier of the newly opened AR.
Max Buffer Size	4	Unsigned32	4	This parameter defines the maximum buffer length in octets (the buffer may contain concatenated Type 9 APDUs) that the sender of this APDU may send or receive on this AR. This parameter can be negotiated down (but not up) by the responder using the Max Buffer Size parameter. If the negotiation fails for any parameter, an acceptable value for this parameter is returned in the additional description field as an Unsigned32 at offset 0. If the requested value is supported, but another parameter failed the negotiation, then the requested value is returned at offset 0.
Max Message Length	8	Unsigned32	4	This parameter defines the maximum length in octets of APDUs to be sent on this AR by the sender of this APDU. It is used by the receiving AR endpoint as its Max Message Length attribute. This parameter cannot be negotiated. The server may reject the value supplied by the client if it cannot support it using error class = "access" and error code = "unsupported max APDU length". If the negotiation fails for any parameter, an acceptable value for this parameter is returned in the additional description field as an Unsigned32 at offset 4. If the requested value is supported, but another parameter failed the negotiation, then the requested value is returned at offset 4.
Reserved	12	Unsigned8	1	Unused, set to 0.
Configuration Use	13	Unsigned8	1	0 = Configuration Not Permitted 1 = Configuration Permitted
Inactivity Close Time	14	Unsigned16	2	This parameter identifies how long, in seconds, that the AR stays open without receiving an APDU. After experiencing inactivity on the AR for this period of time, the AREP is closed, along with all VCR activity associated with the AREP. The responder can negotiate this value down. The value 0 is not permitted. If the negotiation fails for any parameter, an acceptable value for this parameter is returned in the additional description field as an Unsigned32 at offset 8. If the requested value is supported, but another parameter failed the negotiation, then the requested value is returned at offset 8.
Transmit Delay Time	16	Unsigned32	4	This parameter is used to set the Transmit Delay Time attribute in the receiving AREP. Its value may not be negotiated. Its value is expressed in milliseconds. If the negotiation fails for any parameter, an acceptable value for this parameter is returned in the additional description field

Parameter name	Octet offset	Data type	Octet length	Description
				as an Unsigned32 at offset 12. If the requested value is supported, but another parameter failed the negotiation, then the requested value is returned at offset 12.
SMK PD Tag	20	VisibleString	32	SMK PD Tag of the server. If the SMK PD Tag in the request message does not match the PD Tag of the server, then the server rejects the request with error class "access" and error code "object-access-denied".

4.3.5.2.1.3 Response APDU parameters

Same as Request APDU

4.3.5.2.1.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.2.2 Idle (confirmed service Id = 3)

4.3.5.2.2.1 Service overview

This APDU is used to inform the receiver that the sender is present.

4.3.5.2.2.2 Request APDU parameters

None.

4.3.5.2.2.3 Response APDU parameters

None.

4.3.5.3 SMK ASE services

4.3.5.3.1 SM find tag query (unconfirmed service Id = 1)

4.3.5.3.1.1 Service overview

The network address used to send the FIND_TAG_QUERY message and the FDA Address in the Type APDU Header together indicate which SMKs are to receive and process the message.

The Find Tag Query message may be sent to one or more SMKs in Type 5 and/or Type 9 devices. When sent to a linking device, the linking device Type 5 SMK responds if the linking device contains the queried object. The Type 9 interface SMKs in the linking device are used only to forward the query to Type 9 links. They never respond to the queries.

Although this service is unconfirmed, it uses the Invoke Id in the Message Trailer to allow it to be matched to the returned SM Find Tag Reply message.

Table 12 describes the distribution that may be requested for this service. The FDA Addresses in the table are shown in hexadecimal.

Table 12 – SMK FDA address values

FDA address meaning	FDA address		Destination address	Distribution
	LLLL	NN.SS		
Type 5 Device SMK	00 00	00.02	SMK Multicast	All Type 5 SMKs
			Individual	Individual Type 5 SMK
Type 5 and Type 9 SMK Local Link Group Address	00 00	01.09 ¹	SMK Multicast	All Type 5 SMKs and all Type 9 SMKs
			Individual	Individual LD SMK and all Type 9 SMKs accessible through the LD. This does not include the SMKs of the linking device's Type 9 interfaces
Type 9 SMK Local Link Group Address	LL LL	01.09 ¹	SMK Multicast	All Type 9 SMKs on Type 9 link LL LL
			Individual	All Type 9 SMKs on Type 9 link LL LL accessible through the LD. This does not include the SMK of the linking device's Type 9 LL LL interface
Individual Type 9 Address	LL LL	NN.02	Individual	Individual Type 9 SMK accessible through the LD
NOTE Specified in IEC 61158-4-1 as the Type 1 link specific address for "the SMAEs of all DLEs on the link".				

4.3.5.3.1.2 Request APDU parameters

Table 13 lists the SMK FDA address values.

Table 13 – SMK FDA address values

Parameter name	Octet offset	Data type	Octet length	Description
Query Type	0	Unsigned8	1	Indicates the type of query: 0 = PD Tag query for primary or non-redundant device 1 = VFD tag query 2 = Function-Block tag query 3 = Element Id query 4 = PD Tag/VFD Reference query 5 = Device Index query 6 = PD Tag query for secondary or member of redundant set All other values are reserved.
Reserved	1	Octetstring	3	Reserved, set to 0
Element Id Or VFD Reference	4	Unsigned32	4	Service Parameter When not used, set to binary 0.
PD Tag Or Object Tag	8	VisibleString	32	Service Parameter When not used, set to binary 0.
VFD Tag	40	VisibleString	32	Service Parameter When not used, set to binary 0.

4.3.5.3.2 SM find tag reply (unconfirmed service Id = 2)

4.3.5.3.2.1 Service overview

This message is used to reply to the SM Find Tag request message.

Although this service is unconfirmed, it uses the Invoke Id in the Type 5 APDU Trailer to allow it to be matched to the corresponding SM Find Tag Query message.

4.3.5.3.2.2 Request APDU parameters

Table 14 lists the request APDU parameters.

Table 14 – Request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Query Type	0	Unsigned8	1	Indicates the type of query: 0 = PD Tag query for primary or non-redundant device 1 = VFD tag query 2 = Function-Block tag query 3 = Element Id query 4 = PD Tag/VFD Reference query 5 = Device Index query 6 = PD Tag query for secondary or member of redundant set All other values are reserved.
Type 9 Node Address	1	Unsigned8	1	Service Parameter Set to 0 if the Reply is for a Type 5 device
Type 9 Link Id	2	Unsigned16	2	Service Parameter Set to 0 if the Reply is for a Type 5 device.
VFD Reference	4	Unsigned32	4	Service Parameter
Queried Object Numeric Id	8	Unsigned32	4	Service Parameter
Queried Object Network Address	12	Octetstring	16	Service Parameter
Queried Object OD Version Number	28	Unsigned32	4	Service Parameter
Queried Object Device ID	32	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
Queried Object PD Tag	64	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
Duplicate Detection State	96	Unsigned8	1	Service Parameter Encoded as follows. Bits 8-3: Reserved, set to 0. Bit 2: 1 = Duplicate PD Tag Detected 0 = Duplicate PD Tag Not Detected Bit 1: 1 = Duplicate Device Index Detected 0 = Duplicate Device Index Not Detected
Reserved	97	Unsigned8	1	Reserved, set to 0.
Number in List of FDA Addresses	98	Unsigned16	2	This parameter indicates how many FDA Address Selectors are contained in the List of FDA Address Selectors parameter. This value of this attribute is 0 if the Query Type is "physical-device-index".

Parameter name	Octet offset	Data type	Octet length	Description
List of FDA Addresses	100	Unsigned16	2x number in list	Service Parameter Each Selector is an Unsigned32 that immediately follows its predecessor in the series (no spaces or padding between them).

4.3.5.3.3 SM Identify (confirmed service Id = 3)

4.3.5.3.3.1 Service overview

This service is used to request the identity of a single Type 5 or Type 9 device or of a group of Type 5 devices.

Table 15 specifies the network addresses and FDA Addresses used with the SM Identify request message.

Table 15 – SMK FDA address values for SM identify

Identification of	FDA address LLLL.NN.SS	Network address	Request message delivered to
Type 5 SMK device, including	00 00.00.02	SM Multicast	All Type 5 SMKs
Live List Version Number List ^a		Individual	Individual Type 5 SMK
Linking Device Node Version Number List for Specific Type 9 Interface	LL LL.00.02	SM Multicast	All Linking Device Type 5 SMKs – the one with the requested Link Id responds
	LL LL.00.02	Individual	Linking Device Type 5 SMK
Type 9 device SMK	LL LL.NN.02	Individual	Linking Device Type 5 SMK -> Type 9 SMK

^a The Live List Version Number List is returned by Linking Device SMKs.

4.3.5.3.3.2 Request APDU parameters

None.

4.3.5.3.3.3 Response APDU parameters

Same as SM Device Annunciation Request APDU parameters.

4.3.5.3.3.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.3.4 SMK clear address (confirmed service Id = 12)

4.3.5.3.4.1 Service overview

This service is always confirmed and, therefore, always uses the Invoke Id in the Message Trailer.

Table 16 specifies the Network Address and FDA Address used with the SM Clear Address request and response APDUs.

Table 16 – SMK FDA address values for SMK set assignment info request APDUs

Use	FDA address LLLL.NN.SS	Network address	Request path
Type 5 Clear Address	00 00.00.02	Individual	Type 5 Requester -> Type 5 SMK
Type 9 Clear Address	LL LL.NN.02	Individual	Type 5 Requester -> LD FDA Agent -> Type 9 SMK

4.3.5.3.4.2 Request APDU parameters

Table 17 specifies the request APDU parameters.

Table 17 – SMK clear address request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Device Id	0	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
Physical Device Tag	32	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
InterfaceToClear	64	Unsigned8	1	Service Parameter Encoded as follows: 0 = Interface A 1 = Interface B 255 = All interfaces
Reserved	65	Octetstring	3	Reserved, set to 0.

4.3.5.3.4.3 Response APDU parameters

None.

4.3.5.3.4.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.3.5 SMK set assignment info (confirmed service Id = 14)**4.3.5.3.5.1 Service overview**

This service is always confirmed and, therefore, always uses the Invoke Id in the Message Trailer.

Table 18 specifies the Network Address and FDA Address used with the SM Set Assignment Info request and response APDUs.

Table 18 – SMK FDA address values for SMK set assignment info request APDUs

Use	FDA address LLLL.NN.SS	Network address	Request path
Type 5 Set Assignment Info	00 00.00.02	Individual	Type 5 Requester -> Type 5 SMK
Type 9 Set Assignment Info	LL LL.NN.02	Individual	Type 5 Requester -> LD FDA Agent -> Type 9 SMK

4.3.5.3.5.2 Request APDU parameters

Table 19 specifies the request APDU parameters.

Table 19 – SMK set assignment info request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Device Id	0	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
Physical Device Tag	32	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
Type 9 New Address	64	Unsigned8	1	Service Parameter The field contains the new Type 9 address for the Type 9 device identified in the FDA Address of the Type 5 APDU Header. The value of this field is 0 if the FDA Address does not identify a Type 9 device or this APDU is not being used to change the Type 9 device's address.
Device Redundancy State	65	Unsigned8	2	Service Parameter Unused and set to 0 if the FDA Address identifies a Type 9 device. The value 0 indicates that the device is not participating in device redundancy. If it is to participate in device redundancy, the value of bit pair 2 and 1 is non-zero. The value of bit pair 4 and 3 is non-zero whenever bits 2 & 1 indicate "External Switchover Control Device". When the bit pair 1 and 2 indicate "External Switchover Control Device", bits 3 and 4 must be set to "2" to indicate that the current primary is to become the secondary, and one of the secondaries is to become the primary. Bit 8 = Reserved, set to 0 Bit 7 = Reserved, set to 0 Bit 6 = Reserved, set to 0 Bit 5 = Reserved, set to 0 Bits 4 & 3 – External Switchover Control Device Redundancy Role 0 = Not used 1 = Primary 2 = Secondary Bits 2 & 1 – Assigned Device Redundancy Type 0 = Non-redundant 1 = External Switchover Control 2 = Internal Switchover Control
LAN Redundancy Socket Address	66	Unsigned16	2	Service Parameter Unused and set to 0 if the FDA Address identifies a Type 9 device. Set to the Reserved LAN Redundancy port if the FDA Address identifies Type 5 device.
Annunciation Repeat Time	68	Unsigned32	4	Service Parameter Unused and set to 0 if the FDA Address identifies a Type 9 device.
Device Index	72	Unsigned16	2	Service Parameter Encoded as follows.

Parameter name	Octet offset	Data type	Octet length	Description
				0 = Index not assigned 1 to Max Device Index = Device Index
Max Device Index	74	Unsigned16	2	Service Parameter Unused and set to 0 if the FDA Address identifies a Type 9 device.
Operational Network Address	76	Octetstring	16	Service Parameter Unused and set to 0 if the FDA Address identifies an Type "X" device.
Reserved	92	Octetstring	3	Reserved, set to 0.
Clear Duplicate Detection State	95	Unsigned8	2	Service Parameter Unused and set to 0 if the FDA Address identifies a Type 9 device. Encoded as follows: Bits 8-3: Reserved, set to 0. Bit 2: 1 = Do not clear Duplicate PD Tag Detected 0 = Clear Duplicate PD Tag Detected Bit 1: 1 = Do not clear Duplicate Device Index Detected 0 = Clear Duplicate Device Index Detected

4.3.5.3.5.3 Response APDU parameters

Table 20 specifies the response APDU parameters.

Table 20 – SMK set assignment info response APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Reserved	0	Unsigned16	2	Reserved, set to 0.
Max Device Index	2	Unsigned16	2	Service Parameter Unused and set to 0 if the FDA Address identifies a Type 9 device.
Annunciation Repeat Time	4	Unsigned32	4	Service Parameter Unused and set to 0 if the FDA Address identifies a Type 9 device.

4.3.5.3.5.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.3.6 SMK clear assignment info (confirmed service Id = 15)

4.3.5.3.6.1 Service overview

This service is always confirmed and, therefore, always uses the Invoke Id in the Message Trailer.

Table 21 specifies the Network Address and FDA Address used with the SM Clear Assignment Info request and response APDUs.

Table 21 – SMK FDA address values for SMK device clear assignment Info APDUs

Use	FDA address LLLL.NN.SS	Network address	Request path
Type 5 Clear Assignment Info	00 00.00.02	Individual	Type 5 Requester -> Type 5 SMK
Type 9 Clear Assignment Info	LL LL.NN.02	Individual	Type 5 Requester -> LD FDA Agent -> Type 9 SMK

4.3.5.3.6.2 Request APDU parameters

Table 22 defines the request APDU parameters.

Table 22 – SMK clear assignment info request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Device Id	0	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
Physical Device Tag	32	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.

4.3.5.3.6.3 Response APDU parameters

None.

4.3.5.3.6.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.3.7 SMK device annunciation (unconfirmed service Id = 16)**4.3.5.3.7.1 Service overview**

The request APDU body format defined below is used for the SM Device Annunciation service and also for the SM Identify service.

This service is always unconfirmed and no APDU Trailer fields are used.

Table 23 specifies the Network Address and FDA Address used with the SMK Device Annunciation request APDUs.

Table 23 – SMK FDA address values for SMK device annunciation request APDUs

Use	FDA address LLLL.NN.SS	Network address	Request path
Device Annunciation	00 00.00.02	SM Multicast	Type 5 SMK -> Configuration Applications

4.3.5.3.7.2 Request APDU parameters

Table 24 defines the request APDU parameters.

Table 24 – SMK device annunciation request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
SMK State	0	Unsigned8	1	Service Parameter Encoded as follows. Bits 2-8: 0 = Reserved 1 = NO_TAG 2 = OPERATIONAL 3-127 = Reserved Bit 1: 0 = Not Synchronized with SNTP Time Server 1 = Synchronized with SNTP Time Server
Device Type	1	Unsigned8	1	Service Parameter Encoded as follows. Each bit identifies a different device type. It is the destination device type for request messages and the source device for response messages. A combination of bits 8-6 and 2,1 may be set for devices. Bit 5 is used only for Type "X" devices, and only in SM Identify responses. Non Devices participating in device assignment set the value of bits 8 through 3 of this parameter to 0. Bit 8 = Linking Device Bit 7 = I/O Gateway Bit 6 = Field Device Bit 5 = Type "X" Device Bit 4 = Reserved Bits 3-1 – Redundant Device Type Capability 0 = Non-redundant 1 = External Switchover Control 2 = Internal Switchover Control 3 = External and Internal Switchover Control 4 = Not used 5 = Internal Switchover Control and Non-redundant device 6 = External Switchover Control and Non-redundant device 7 = External and Internal Switchover Control and Non-redundant device
Device Redundancy State	2	Unsigned8	2	Service Parameter Encoded as follows. Encoded as follows. Bit 8 = Reserved, set to 0 Bit 7 = Reserved, set to 0 Bit 6 = Reserved, set to 0 Bit 5 = Reserved, set to 0 Bits 4 & 3 – Device Redundancy Role 0 = Non-redundant 1 = Primary 2 = Secondary Bits 2 & 1 – Assigned Redundant Device Type 0 = Non-redundant

Parameter name	Octet offset	Data type	Octet length	Description
				1 = External Switchover Control 2 = Internal Switchover Control
Duplicate Detection State	3	Unsigned8	2	Service Parameter Encoded as follows. Bits 8-3: Reserved, set to 0. Bit 2: 1 = Duplicate PD Tag Detected 0 = Duplicate PD Tag Not Detected Bit 1: 1 = Duplicate Device Index Detected 0 = Duplicate Device Index Not Detected
Device Index	4	Unsigned16	2	Service Parameter Encoded as follows. 0 = Index not assigned 1 to Max Device Index = Device Index
Max Device Index	6	Unsigned16	2	Service Parameter
Operational Network Address	8	Octetstring	16	Service Parameter
Device Id	24	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
Physical Device Tag	56	VisibleString	32	Service Parameter Unused octets in this field are set to ASCII space characters.
Annunciation Repeat Time	92	Unsigned32	4	Service Parameter
LAN Redundancy Socket Address	96	Unsigned16	2	Service Parameter
Reserved	98	Unsigned16	2	Service Parameter
APDU Version Number	100	Unsigned32	4	Service Parameter
Device Version Number	100	Unsigned32	4	Service Parameter
Number of Entries in Version Number List	104	Unsigned32	4	This parameter indicates how many version numbers are contained in the Version Number List field below. Its value is 0 if the device is not a linking device or a gateway.
Version Number List	108	Unsigned32	4 x Number of Entries	Service Parameter This field is present only for linking devices and I/O gateways. This field contains a list of numbers. Each is an Unsigned32 number. The number of entries in the list is specified by the field above "Number of Entries in Version Number List". This field supports two types of lists, as determined by the Link Id portion of the FDA Address, as follows: Link Id = 0 For linking devices, this is a list of Type 9 live list version numbers, one for each Type 9 interface attached to the linking device. Each element in the list is constructed as follows: Bits 32-17: Type 9 Link Id Bits 16-9: Reserved, set to 0

Parameter name	Octet offset	Data type	Octet length	Description
				Bits 8-1: Version Number The list is ordered by Type 9 bridge interface number with the Live List Version for the first interface occurring first in the list. The value 0 is used for the Link Id bits if the interface has not been assigned a link Id. Link Id > 0 This is a List of Type "X" Node Address Version Numbers, one for each node id currently in the live list for the specified Type "X" link, including the LD Type "X" interface. The List of Type "X" Node Address Version Numbers is used only when this message format is used for SM Identify Responses that describe an LD Type "X" interface. The list begins with the version number of the specified Type "X" Interface. The remainder of the list is sorted in ascending order by Type "X" node address, with two version numbers packed into each Unsigned32 value, as follows: Bits 33-25: Type "X" Node Address Bits 24-17: Version Number Bits 16-9: Type "X" Node Address Bits 8-1: Version Number

4.3.5.4 Type 9 ASE Services

4.3.5.4.1 General note

NOTE Unless specified otherwise, all parameters of the Type 9 Service APDUs and their values are defined in IEC 61158-5 subseries.

4.3.5.4.2 Reject

The Type 9 Reject Service is used only to indicate to the requester that the response is too long for Type 9 to convey it. Instead of defining a separate APDU here to convey this information, the Common Error APDU conveys it, using the Error Class "Reject" and the Reject Code as defined in the Type 9 abstract syntax for the Error Code.

4.3.5.4.3 Initiate (confirmed service Id = 96)

4.3.5.4.3.1 Request APDU parameters

Table 25 defines the request APDU parameters.

Table 25 – Initiate request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Connect Option	0	Unsigned8	1	This field indicates which option is to be used to open the S-VFD Context. Its values are: 1 = VCR Selector 2 = MIB Access 3 = FBAP Access
Access Protection Supported Calling	8	Boolean	1	Service Parameter
Password and Access Groups Calling	2	Unsigned16	2	Service Parameter
Version OD Calling	4	Integer16	2	Service Parameter

Parameter name	Octet offset	Data type	Octet length	Description
Profile Number Calling	6	Unsigned16	2	Service Parameter
PD Tag	8	OctetString	32	The SMK Physical Device Tag of the device containing the destination VFD.

4.3.5.4.3.2 Response APDU parameters

Table 26 lists the response APDU parameters.

Table 26 – Initiate response APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Version OD Called	0	Integer16	2	Service Parameter
Profile Number Called	2	Unsigned16	2	Service Parameter

4.3.5.4.3.3 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1. See Type 9 Initiate specific error codes.

4.3.5.4.4 Abort (unconfirmed service Id = 112)

4.3.5.4.4.1 Request APDU parameters

Table 27 specifies the request APDU parameters.

Table 27 – Abort request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Abort Detail	0	OctetString	16	Supplied by the sender of the APDU.
Abort Identifier	16	Unsigned8	1	Service parameter. Values defined for Type 9 PDUs.
Reason Code	17	Unsigned8	1	Service Parameter
Reserved	18	Unsigned16	2	Reserved, set to 0

4.3.5.4.5 Get status (confirmed service Id = 0)

4.3.5.4.5.1 Request APDU parameters

None.

4.3.5.4.5.2 Response APDU parameters

Table 28 defines the response APDU parameters.

Table 28 – Get response APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Logical Status	0	Unsigned8	1	Service Parameter
Physical Status	1	Unsigned8	1	Service Parameter
Reserved	2	Unsigned16	2	Reserved, set to 0
Local Detail	4	OctetString	4	Service Parameter. The high order octet is reserved and set to 0. The low order three octets contain the 3-octet Local Detail value.

4.3.5.4.6 Status notification (unconfirmed service Id = 1)**4.3.5.4.6.1 Request APDU parameters**

Same as Get Status Response APDU

4.3.5.4.7 Identify (confirmed service Id = 1)**4.3.5.4.7.1 Request APDU parameters**

None.

4.3.5.4.7.2 Response APDU parameters

Table 29 specifies the response APDU parameters.

Table 29 – Identify response APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Vendor Name	0	VisibleString	32	Service Parameter
Model Identifier	32	VisibleString	32	Service Parameter
Vendor Revision	64	VisibleString	32	Service Parameter

4.3.5.4.8 Get OD (confirmed service Id = 4)**4.3.5.4.8.1 Request APDU parameters**

Table 30 specifies the request APDU parameters.

Table 30 – Get OD request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
All Attributes	0	Boolean	1	Service Parameter
Start Index Flag	1	Boolean	1	Service Parameter
Reserved	2	Unsigned16	2	Reserved, set to 0
Index	4	Unsigned32	4	Service Parameter

4.3.5.4.8.2 Response APDU parameters

Table 31 specifies the response APDU parameters.

Table 31 – Get OD response APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
More Follows	0	Boolean	1	Service Parameter
Number of Object Descriptions	1	Unsigned8	1	Service Parameter
Reserved	2	Unsigned16	2	Reserved, set to 0
List of Object Descriptions	4	OctetString	N	Defined in 4.3.5.6.2

4.3.5.4.8.3 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.9 Initiate Put OD (confirmed service Id = 28)**4.3.5.4.9.1 Request APDU parameters**

Table 32 specifies the request APDU parameters.

Table 32 – Initiate put OD request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Reserved	0	Unsigned16	2	Reserved, set to 0
Consequence	2	Unsigned16	2	Service Parameter. Values defined for Type 9 PDUs

4.3.5.4.9.2 Response APDU parameters

None

4.3.5.4.9.3 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.10 Put OD (confirmed service Id = 29)**4.3.5.4.10.1 Request APDU parameters**

Table 33 specifies the request APDU parameters.

Table 33 – Put OD request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Number of Object Descriptions	0	Unsigned8	1	Number of entries in List of Object Descriptions
Reserved	1	OctetString	3	Reserved, set to 0
List of Object Descriptions	4	OctetString	N	Defined in 4.3.5.6.2

4.3.5.4.10.2 Response APDU parameters

None

4.3.5.4.10.3 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.11 Terminate put OD (confirmed service Id = 30)**4.3.5.4.11.1 Request APDU parameters**

None.

4.3.5.4.11.2 Response APDU parameters

None.

4.3.5.4.11.3 Error APDU parameters

Defined by the OD Error APDU parameters in 8.3.6.

4.3.5.4.12 Generic initiate download sequence (confirmed service Id = 31)

4.3.5.4.12.1 Service overview

This service corresponds to the Type 9 Initiate Load Service with the following parameter settings:

- Load Region Index Called: Load region of the responder (Server)
- Load Type: DOWNLOAD
- Load Service to Use: Push Segment (Generic Download Segment)
- Load Service Initiator: TRUE (Initiator (Client) initiates the load service (The Push Segment service)).

4.3.5.4.12.2 Request APDU parameters

Table 34 specifies the request APDU parameters.

Table 34 – Generic initiate download sequence request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter

4.3.5.4.12.3 Response APDU parameters

None

4.3.5.4.12.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.13 Generic download segment (confirmed service Id = 32)

4.3.5.4.13.1 Service overview

This service corresponds to the Type 9 Generic Download Segment Service.

4.3.5.4.13.2 Request APDU parameters

See Table 35.

Table 35 – Generic download segment request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter
More Follows	4	Boolean	1	Service Parameter
Reserved	5	OctetString	3	Set to 0.
Load Data	8	OctetString	N	Service Parameter

4.3.5.4.13.3 Response APDU parameters

None.

4.3.5.4.13.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.14 Generic terminate download sequence (confirmed service Id = 33)

4.3.5.4.14.1 Service overview

This service corresponds to the Type 9 Generic Terminate Download Sequence Service.

4.3.5.4.14.2 Request APDU parameters

See Table 36.

Table 36 – Generic terminate download sequence request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter

4.3.5.4.14.3 Response APDU parameters

See Table 37.

Table 37 – Response APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Terminate Reason	0	Boolean	1	Service Parameter
Reserved	1	OctetString	3	Reserved, set to 0

4.3.5.4.14.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.15 Initiate download sequence (confirmed service Id = 9)

4.3.5.4.15.1 Service overview

This service corresponds to the Type 9 Initiate Download Sequence Service with the following parameter settings:

- Load Region Index Called: Load region of the responder (Server)
- Load Type: DOWNLOAD
- Load Service to Use: Pull Segment (Download Segment)
- Load Service Initiator: FALSE (Remote AP (Server) initiates the load service (The Pull Segment service)).

4.3.5.4.15.2 Request APDU parameters

See Table 38.

Table 38 – Initiate download sequence request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter

4.3.5.4.15.3 Response APDU parameters

None.

4.3.5.4.15.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.16 Download segment (confirmed service Id = 10)**4.3.5.4.16.1 Service overview**

This service corresponds to the Type 9 Download Segment Service.

4.3.5.4.16.2 Request APDU parameters

See Table 39.

Table 39 – Download segment request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter

4.3.5.4.16.3 Response APDU parameters

See Table 40.

Table 40 – Download segment response APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
More Follows	0	Boolean	1	Service Parameter
Reserved	2	OctetString	3	Reserved, each octet is set to binary 0
Load Data	4	OctetString	N	Service Parameter

4.3.5.4.16.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.17 Terminate download sequence (confirmed service Id = 11)**4.3.5.4.17.1 Service overview**

This service corresponds to the Type 9 Terminate Download Sequence Service.

4.3.5.4.17.2 Request APDU parameters

See Table 41.

Table 41 – Terminate download sequence request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter
Reserved	4	Octetstring	3	Reserved, each octet is set to binary 0
Terminate Reason	7	Boolean	1	Service Parameter

4.3.5.4.17.3 Response APDU parameters

None.

4.3.5.4.17.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.18 Initiate upload sequence (confirmed service Id = 12)

4.3.5.4.18.1 Service overview

This service corresponds to the Type 9 Initiate Upload Sequence Service with the following parameter settings:

- Load Region Index Called: Load region of the responder (Server)
- Load Type: UPLOAD
- Load Service to Use: Pull Segment (Upload Segment)
- Load Service Initiator: TRUE (Initiator (Client) initiates the load service (The Pull Segment service))

4.3.5.4.18.2 Request APDU parameters

See Table 42.

Table 42 – Initiate upload sequence request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter

4.3.5.4.18.3 Response APDU parameters

None.

4.3.5.4.18.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.19 Upload segment (confirmed service Id = 13)

4.3.5.4.19.1 Service overview

This service corresponds to the Type 9 Upload Segment Service.

4.3.5.4.19.2 Request APDU parameters

See Table 43.

Table 43 – Upload segment request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter

4.3.5.4.19.3 Response APDU parameters

See Table 44.

Table 44 – Upload segment response APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
More Follows	0	Boolean	1	Service Parameter
Reserved	2	OctetString	3	Reserved, each octet is set to binary 0
Load Data	4	OctetString	N	Service Parameter

4.3.5.4.19.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.20 Terminate upload sequence (confirmed service Id = 14)**4.3.5.4.20.1 Service overview**

This service corresponds to the Type 9 Terminate Upload Sequence Service.

4.3.5.4.20.2 Request APDU parameters

See Table 45.

Table 45 – Terminate upload sequence request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter

4.3.5.4.20.3 Response APDU parameters

None

4.3.5.4.20.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.21 Request domain download (confirmed service Id = 15)**4.3.5.4.21.1 Service overview**

This service corresponds to the Type 9 Request Domain Download Service.

4.3.5.4.21.2 Request APDU parameters

See Table 46 for definitions.

Table 46 – Request domain download request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter
Additional Information	4	VisibleString	N	Service Parameter

4.3.5.4.21.3 Response APDU parameters

None.

4.3.5.4.21.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.22 Request domain upload (confirmed service Id = 16)

4.3.5.4.22.1 Service overview

This service corresponds to the Type 9 Create Program Invocation Service.

4.3.5.4.22.2 Request APDU parameters

See Table 47 for definitions.

Table 47 – Request domain upload request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Index	0	Unsigned32	4	Service Parameter
Additional Information	4	VisibleString	N	Service Parameter

4.3.5.4.22.3 Response APDU parameters

None.

4.3.5.4.22.4 Error APDU parameters

Defined by the Common Error APDU parameters in 8.3.6.1.

4.3.5.4.23 Create program invocation (confirmed service Id = 17)

4.3.5.4.23.1 Service overview

The Create Program Invocation corresponds to to the Type 9 Create Program Invocation service.

4.3.5.4.23.2 Request APDU parameters

See Table 48 for definitions.

Table 48 – Create program invocation request APDU parameters

Parameter name	Octet offset	Data type	Octet length	Description
Reusable	0	Boolean	1	Service Parameter
Reserved	1	Unsigned8	1	Reserved, set to 0
Number of Domain Indexes	2	Unsigned16	2	The number of domain indexes in this service
List Of Domain Indexes	4	Unsigned32	4 X Number of Domain Indexes	Service Parameter

4.3.5.4.23.3 Response APDU parameters

See Table 49 for definitions.