DEUTSCHE NORM September 2015

# DIN EN 61158-6-5

ICS 35.100.70; 35.160

Ersatz für
DIN EN 61158-6-5:2008-09
Siehe Anwendungsbeginn

# Industrielle Kommunikationsnetze – Feldbusse – Teil 6-5: Protokollspezifikation des Application Layer (Anwendungsschicht) – Typ 5-Elemente (IEC 61158-6-5:2014); Englische Fassung EN 61158-6-5:2014, nur auf CD-ROM

Industrial communication networks –
Fieldbus specifications –
Part 6-5: Application layer protocol specification – Type 5 elements
(IEC 61158-6-5:2014);
English version EN 61158-6-5:2014, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 6-5: Spécification du protocole de la couche application – Éléments de type 5
(CEI 61158-6-5:2014);
Version anglaise EN 61158-6-5:2014, seulement en CD-ROM

Gesamtumfang 119 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

## Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2014-09-23 angenommene Europäische Norm als DIN-Norm ist 2015-09-01.

Für DIN EN 61158-6-5:2008-09 besteht eine Übergangsfrist bis 2017-09-23.

## Nationales Vorwort

*Vorausgegangener Norm-Entwurf: E DIN EN 61158-600:2013-08.*

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Feldbus“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom SC 65C „Industrial networks“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „http://webstore.iec.ch“ zu dieser Publikation angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation
– bestätigt,
– zurückgezogen,
– durch eine Folgeausgabe ersetzt oder
– geändert.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden Deutschen Dokumenten ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für die vorliegende Norm erfüllt.

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 61158-6-5 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt, und Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

### Einführung

Ein Feldbus ist ein digitaler serieller Datenbus für die Kommunikation zwischen Geräten der industriellen Automatisierungstechnik wie beispielsweise Messeinrichtungen, Stellgliedern, Reglern und speicherprogrammierbaren Steuerungen. Die Normen der Reihe DIN EN 61158 legen eine Anzahl von Feldbus-Protokolltypen fest. Die vorliegende Norm legt das Protokoll des Application Layer Typ 5 fest.



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

**Änderungen**

Gegenüber DIN EN 61158-6-5:2008-09 wurden folgende Änderungen vorgenommen:

a) Auffüllen von Telegrammen hinzugefügt;

b) Kodierungsregeln verbessert;

c) Open Session Service (Dienst zum Eröffnen einer Sitzung) verbessert;

d) Zeitsynchronisation wird jetzt mit Hilfe der Annunciation Message durchgeführt;

e) zusätzliche Redundanzmöglichkeiten in der Annunciation Message durchgeführt.

**Frühere Ausgaben**

DIN V 19245-1: 1988-01
DIN 19245-1: 1988-01, 1991-04
DIN 19245-2: 1991-04
DIN EN 50170/1: 1997-07
DIN EN 50170/2: 1997-07
DIN EN 50170/3: 1997-07
DIN EN 50254: 1999-07
DIN EN 50170/A2: 2000-08
DIN EN 50170/A1: 2002-11
DIN EN 50170/A3: 2002-11
DIN EN 61158-6: 2005-05
DIN EN 61158-6-5: 2008-09

Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

# Nationaler Anhang NA
(informativ)

## 1 Anwendungsbereich[N1)]

### 1.1 Allgemeines

Der Feldbus-Application Layer (FAL) liefert Benutzerprogramme mit Hilfsmitteln für den Zugriff auf die Feldbuskommunikationsumgebung. In dieser Hinsicht, kann die FAL als ein Fenster zwischen zugehörigen Anwendungsprogrammen betrachtet werden.

Diese Norm stellt für Typ 5-Feldbusse maßgebliche allgemeine Elemente für zeitkritischen und nicht zeitkritischen Datentransfer zwischen Anwendungsprogrammen in einer Automatisierungsumgebung bereit. Der Begriff „zeitkritisch“ (en: time-critical) wird verwendet um die Präsenz eines Zeitfensters darzustellen, in dem eine oder mehrere festgelegte Aktivitäten mit einem vorherbestimmten Grad der Gewissheit erledigt werden. Das Scheitern der Erledigung der festgelegten Aktivität innerhalb des Zeitfensters, trägt das Risiko des Betriebsausfalls der Anwendung mit, und somit das aufgeforderte Eingreifen mit dazugehörigem Risiko für Geräte, Anlage und möglicherweise Menschenleben.

Diese Norm definiert auf eine abstrakte Weise das von außen sichtbare Verhalten, das vom Typ 5 des Feldbus-Application Layer bereitgestellt wird, in Form von:

a) einer abstrakten Syntax zur Festlegung der Application Layer-Protokoll-Dateneinheiten die zwischen kommunizierenden Anwendungseinheiten übermittelt werden,

b) der übertragenen Syntax zur Festlegung der Application Layer-Protokoll-Dateneinheiten die zwischen kommunizierenden Anwendungseinheiten übermittelt werden,

c) des Anwendungskontext der Zustandsmaschine zur Festlegung des sichtbaren Dienstverhalten zwischen kommunizierenden Anwendungseinheiten,

d) der Anwendungsbeziehung der Zustandsmaschine zur Festlegung des sichtbaren Kommunikationsverhalten zwischen kommunizierenden Anwendungseinheiten.

Der Zweck dieser Norm ist, das Protokoll zu definieren, welches bereitgestellt wird zur

- Festlegung der Leitungsdarstellung der Dienstprimitiven, welche in IEC 61158-5-5 beschrieben sind und
- zur Festlegung des von außen sichtbaren Verhaltens verbunden mit ihrer Übertragung.

Diese Norm beschreibt das Protokoll des Typ 5-Feldbus-Application Layer in Übereinstimmung mit dem OSI-Basis-Referenzmodell (ISO/IEC 7498-1) und der OSI-Application Layer-Struktur (ISO/IEC 9545).

FAL-Dienste und Protokolle sind von FAL-Applikations-Einheiten (AE) bereitgestellt, welche innerhalb von Anwendungsprozessen enthalten sind. Die FAL-AE besteht aus einer Menge von objekt-orientierten Anwendungs-Dienst-Elementen (ASEs) und Layer-Management-Einheiten (LME) welche die AE leiten. Die ASEs liefern Kommunikationsdienste, die auf einer Menge von zugehörigen Applikationsprozessobjekt (APO) -Klassen arbeitet. Eine der FAL-ASEs ist eine Management ASE, welche eine allgemeine Menge von Diensten für das Management der Instanzen der FAL-Klassen bereitstellt.

Obwohl diese Dienste, aus dem Blickwinkel der Anwendung, beschreiben wie Anfrage und Antwort ausgegeben und abgegeben werden, beinhalten sie keine Festlegung was die anfragende und antwortende Anwendungen mit diesen vorzunehmen haben. Die Verhaltensaspekte der Anwendungen sind nicht festgelegt, nur eine Definition von welchen Anfragen und Antworten diese senden/empfangen können, ist beschrieben. Dies erlaubt den FAL-Benutzern eine größere Flexibilität, um solch ein Objektverhalten zu standardisieren. Zusätzlich zu diesen Diensten, sind in dieser Norm ebenfalls einige Dienste zur Unterstützung festgelegt, um den Zugriff zur FAL bereitzustellen und somit bestimmte Aspekte des Betriebs zu steuern.

---

[N1)] Die Nummerierung aus IEC 61158-6-5 wurde beibehalten.



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

## 1.2 Spezifikationen

Das hauptsächliche Ziel dieser Norm, ist das Festlegen der Syntax und des Verhaltens des Application Layer-Protokolls, welcher die in IEC 61158-5-5 beschriebenen Application Layer-Dienste, vermittelt.

Ein zweites Ziel, ist das Bereitstellen von Migrationspfade von bisherigen bestehenden industriellen Kommunikationsprotokollen. Es ist dieses letztere Ziel, welches zur Steigerung von unterschiedlichen Protokollen, die in der IEC 61158-6 Reihe genormt werden, führt.

## 1.3 Konformität

Diese Norm legt keine individuellen Implementierungen oder Produkte fest, noch beschränkt sie die Implementierungen von Application Layer-Einheiten innerhalb eines industriellen Automatisierungssystems. Konformität wird erreicht, durch das Implementieren dieser Application Layer-Protokoll-Spezifikation.

## 3.2 Begriffe aus IEC 61158-1

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 61158-1.

**3.2.1**
**Anwendung**
(en: application)
Funktion oder Datenstruktur, für die Daten konsumiert oder produziert werden

**3.2.2**
**Interoperabilität auf Application Layer**
**(en: application layer interoperability)**
Fähigkeit von Anwendungsentitäten, mithilfe der Dienste des FAL koordinierte und kooperative Operationen auszuführen

**3.2.3**
**Anwendungsobjekt**
(en: application object)
Objektklasse, die zur Laufzeit für den Nachrichtenaustausch im Netzwerk und innerhalb des Netzwerkgerätes sorgt und diesen verwaltet

Anmerkung 1 zum Begriff: Es können mehrere Typen von Anwendungsobjektklassen definiert werden.

**3.2.4**
**Anwendungsprozess**
(en: application process)
Teil einer verteilten Anwendung in einem Netzwerk, der sich auf einem Gerät befindet und eindeutig adressiert wird

**3.2.5**
**Anwendungsprozesskennung**
(en: application process identifier)
unterscheidet mehrere in einem Gerät verwendete Anwendungsprozesse

**3.2.6**
**Anwendungsprozessobjekt**
(en: application process object)
Komponente eines Anwendungsprozesses, die über eine FAL-Anwendungsbeziehung identifizierbar und zugänglich ist

Anmerkung 1 zum Begriff: Definitionen von Anwendungsprozessobjekten setzen sich aus einem Satz von Werten für die Attribute deren Klasse zusammen (siehe Definition von „Anwendungsprozessobjektklasse“). Auf die Definitionen von Anwendungsprozessobjekten kann mithilfe der Dienste des Objektverwaltungs-ASE der FAL von einem entfernten Standort aus zugegriffen werden. Objektverwaltungsdienste der FAL können zum Laden und Aktualisieren von Objektdefinitionen, zum Lesen von Objektdefinitionen und zur dynamischen Erstellung und Löschung von Anwendungsobjekten und deren Definitionen genutzt werden.

Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

**3.2.7**
**Anwendungsprozessobjektklasse**
(en: application process object class)
Klasse von Anwendungsprozessobjekten, die in Bezug auf den Satz der über das Netzwerk zugänglichen Attribute und Dienste der Anwendungsprozessobjekte definiert ist

**3.2.8**
**Anwendungsbeziehung**
(en: application relationship)
kooperative Verbindung zwischen zwei oder mehr Aufrufen von Anwendungsentitäten zum Zweck des Informationsaustausches und der Koordination des gemeinsamen Betriebs

Anmerkung 1 zum Begriff: Diese Beziehung wird entweder durch den Austausch von Protokolldateneinheiten der Anwendung oder als Ergebnis von Vorkonfigurationsaktivitäten aktiviert.

**3.2.9**
**Anwendungsdienstelement für Anwendungsbeziehungen**
(en: application relationship application service element)
Anwendungsdienstelement, das die alleinigen Mittel für den Aufbau und die Beendigung aller Anwendungsbeziehungen bereitstellt

**3.2.10**
**Endpunkt der Anwendungsbeziehung**
(en: application relationship endpoint)
Kontext und Verhalten einer Anwendungsbeziehung entsprechend der Sicht und Verwaltung durch einen der Anwendungsprozesse innerhalb der Anwendungsbeziehung

Anmerkung 1 zum Begriff: Jeder Anwendungsprozess, der in die Anwendungsbeziehung eingeschlossen ist, verwaltet seinen eigenen Endpunkt der Anwendungsbeziehung.

**3.2.11**
**Attribut**
(en: attribute)
Beschreibung eines von außen sichtbaren Merkmals oder einer von außen sichtbaren Funktion eines Objekts

Anmerkung 1 zum Begriff: Die Attribute eines Objekts enthalten Informationen zu veränderlichen Teilen eines Objekts. Attribute liefern üblicherweise Statusinformationen oder steuern den Betrieb eines Objekts. Sie können auch das Verhalten eines Objekts beeinflussen. Attribute werden in Klassenattribute und Instanzattribute unterteilt.

**3.2.12**
**Verhalten**
(en: behaviour)
Anzeige dafür, wie das Objekt auf bestimmte Ereignisse reagiert

Anmerkung 1 zum Begriff: Seine Beschreibung schließt die Beziehung zwischen Attributwerten und Diensten ein.

**3.2.13**
**Klasse**
(en: class)
Menge von Objekten, die allesamt den gleichen Typ von Systemkomponenten repräsentieren

Anmerkung 1 zum Begriff: Eine Klasse ist eine Verallgemeinerung des Objekts; eine Vorlage für die Festlegung von Variablen und Methoden. Alle Objekte in einer Klasse sind in ihrer Form und in ihrem Verhalten identisch, enthalten aber üblicherweise unterschiedliche Daten in ihren Attributen.

**3.2.14**
**Klassenattribute**
(en: class attributes)
Attribut, welches allen Objekten innerhalb derselben Klasse gemeinsam ist



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

**3.2.15**
**Klassencode**
(en: class code)
eindeutige Kennung, die jeder Objektklasse zugewiesen ist

**3.2.16**
**klassenspezifischer Dienst**
(en: class specific service)
durch eine bestimmte Objektklasse definierter Dienst, um eine erforderliche Funktion auszuführen, welche nicht als allgemeiner Dienst ausgeführt wird

Anmerkung 1 zum Begriff: Ein klassenspezifisches Objekt ist für die Objektklasse, welche es definiert, eindeutig.

**3.2.17**
**Client**
(en: client)
(a) Objekt, das die Dienste eines anderen (Server-)Objekts zur Ausführung einer Aufgabe nutzt;

(b) Initiator einer Nachricht, auf die ein Server reagiert, wie z. B. die Rolle eines AR-Endpunktes, in der dieser bestätigte Dienstanforderungs-APDUs an einen einzelnen AR-Endpunkt, welcher als Server agiert, ausgibt

**3.2.18**
**Übertragungspfad**
(en: conveyance path)
unidirektionaler Strom von APDUs in einer Anwendungsbeziehung

**3.2.19**
**zyklisch**
(en: cyclic)
Benennung zur Beschreibung von Ereignissen, die sich regelmäßig wiederholen

**3.2.20**
**zugeordnete AR**
(en: dedicated AR)
Anwendungsbeziehung, die direkt vom FAL-Benutzer verwendet wird

Anmerkung 1 zum Begriff: Bei zugeordneten ARs werden nur der FAL-Header und die Benutzerdaten übertragen.

**3.2.21**
**Gerät**
(en: device)
physische Hardware-Verbindung zum Verbindungszweig

Anmerkung 1 zum Begriff: Ein Gerät kann mehr als einen Knoten umfassen.

**3.2.22**
**Geräteprofil**
(en: device profile)
Zusammenstellung geräteabhängiger Informationen und Funktionalitäten, um Konsistenz zwischen ähnlichen Geräten desselben Gerätetyps herzustellen

**3.2.23**
**dynamische AR**
(en: dynamic AR)
Anwendungsbeziehung, die nur unter Verwendung der Prozeduren zum Aufbau von ARs in einen aktiven Zustand versetzt werden kann

**3.2.24**
**Endpunkt**
(en: endpoint)
eine der Daten übermittelnden Entitäten, die an einer Verbindung beteiligt ist

AVA Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

**3.2.25**
**Fehler**
(en: error)
Nichtübereinstimmung zwischen einem berechneten, beobachteten oder gemessenen Wert oder Zustand und dem festgelegten oder theoretisch richtigen Wert oder Zustand

**3.2.26**
**Fehlerklasse**
(en: error class)
allgemeine Gruppierung für Fehlerdefinitionen

Anmerkung 1 zum Begriff: Innerhalb einer Fehlerklasse sind Fehlercodes für spezifische Fehler festgelegt.

**3.2.27**
**Fehlercode**
(en: error code)
Kennzeichnung eines bestimmten Fehlertyps innerhalb einer Fehlerklasse

**3.2.28**
**FAL-Subnetz**
(en: FAL subnet)
Netzwerke, die aus einem oder mehreren Data-Link-Layer-Segmenten bestehen

Anmerkung 1 zum Begriff: Sie dürfen Bridges, jedoch keine Router umfassen. FAL-Subnetze werden durch eine Untermenge der Netzwerkadresse identifiziert.

**3.2.29**
**logisches Gerät**
(en: logical device)
legt eine bestimmte FAL-Klasse fest, die eine Softwarekomponente oder eine Firmwarekomponente als eigenständige Einrichtung eines Automatisierungsgerätes abstrahiert

**3.2.30**
**Verwaltungsinformationen**
(en: management information)
über das Netzwerk zugängliche Informationen, die die Verwaltung des Betriebs des Feldbussystems, einschließlich des Application Layer, unterstützen

Anmerkung 1 zum Begriff: Zur Verwaltung gehören Funktionen wie die Steuerung, die Überwachung und die Diagnose.

**3.2.31**
**Netzwerk**
(en: network)
Reihe von Knoten, die über ein bestimmtes Kommunikationsmedium miteinander verbunden sind

Anmerkung 1 zum Begriff: Die Verbindungspfade zwischen beliebigen Paaren von Knoten können Repeater, Router und Gateways einschließen.

**3.2.32**
**Peer**
(en: peer)
Rolle eines AR-Endpunktes, in der dieser in der Lage ist, sowohl als Client als auch als Server zu agieren

**3.2.33**
**vordefinierter AR-Endpunkt**
(en: pre-defined AR endpoint)
AR-Endpunkt, der lokal innerhalb eines Gerätes ohne Nutzung des Erstellungsdienstes festgelegt ist

Anmerkung 1 zum Begriff: Vordefinierte ARs, die nicht bereits aktiviert sind, werden vor ihrer Nutzung hergestellt.



AVA Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

**3.2.34**
**vorher aktivierter AR-Endpunkt**
(en: pre-established AR endpoint)
AR-Endpunkt, der während der Konfiguration der AEs, die ihre Endpunkte steuern, in einen aktiven Zustand versetzt wird

**3.2.35**
**Publisher**
(en: publisher)
Rolle eines AR-Endpunktes, in der dieser APDUs auf den Feldbus für den Konsum durch einen oder mehrere Subscriber überträgt

Anmerkung 1 zum Begriff: Der Publisher muss die Identität oder Anzahl der Subscriber nicht kennen und kann seine APDUs mithilfe einer zugeordneten AR veröffentlichen. In dieser Norm sind zwei Typen von Publishern festgelegt, die jeweils einzeln definiert sind: Pull-Publisher und Push-Publisher.

**3.2.36**
**Server**
(en: server)
a) Rolle eines AREP, in der dieser eine bestätigte Dienstantwort-APDU an den Client, der die Anforderung initiiert hat, zurückgibt;

b) Objekt, das einem anderen (Client-)Objekt Dienste bereitstellt

**3.2.37**
**Dienst**
(en: service)
Operation oder Funktion, die ein Objekt und/oder eine Objektklasse auf Anforderung eines anderen Objekts und/oder einer anderen Objektklasse hin ausführt

Anmerkung 1 zum Begriff: Es sind eine Reihe von allgemeinen Diensten definiert, und es sind Richtlinien für die Festlegung objektspezifischer Dienste vorgesehen. Objektspezifische Dienste sind Dienste, die durch eine bestimmte Objektklasse festgelegt sind, um eine erforderliche Funktion auszuführen, die nicht von einem allgemeinen Dienst ausgeführt wird

**3.2.38**
**Subscriber**
(en: subscriber)
Rolle eines AREP, in der dieser von einem Publisher produzierte APDUs empfängt

Anmerkung 1 zum Begriff: In dieser Norm sind zwei Typen von Subscribern festgelegt, die jeweils einzeln definiert sind: Pull-Subscriber und Push-Subscriber

Documents
Document Sharing Hub

– Leerseite –



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

# EN 61158-6-5

October 2014

ICS 25.040.40; 35.100.70; 35.110

Supersedes EN 61158-6-5:2008

English Version

# Industrial communication networks - Fieldbus specifications - Part 6-5: Application layer protocol specification - Type 5 elements (IEC 61158-6-5:2014)

Réseaux de communication industriels - Spécifications des bus de terrain - Partie 6-5: Spécification du protocole de la couche application - Éléments de type 5 (CEI 61158-6-5:2014)

Industrielle Kommunikationsnetze - Feldbusse - Teil 6-5: Protokollspezifikation des Application Layer (Anwendungsschicht) - Typ 5-Elemente (IEC 61158-6-5:2014)

This European Standard was approved by CENELEC on 2014-09-23. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

**CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels**

© 2014 CENELEC All rights of exploitation in any form and by any means reserved worldwide for CENELEC Members.

Ref. No. EN 61158-6-5:2014 E

AVA Documents Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

## Foreword

The text of document 65C/764/FDIS, future edition 2 of IEC 61158-6-5, prepared by SC 65C "Industrial networks" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61158-6-5:2014.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2015-06-23
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2017-09-23

This document supersedes EN 61158-6-5:2008.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

## Endorsement notice

The text of the International Standard IEC 61158-6-5:2014 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

| | | |
|---|---|---|
| IEC 61158-3-1 | NOTE | Harmonized as EN 61158-3-1. |
| IEC 61158-4-1 | NOTE | Harmonized as EN 61158-4-1. |
| IEC 61784-1 | NOTE | Harmonized as EN 61784-1. |
| IEC 61784-2 | NOTE | Harmonized as EN 61784-2. |



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

# CONTENTS

INTRODUCTION ..... 8
1 Scope ..... 9
1.1 General ..... 9
1.2 Specifications ..... 10
1.3 Conformance ..... 10
2 Normative references ..... 10
3 Terms, definitions, symbols, abbreviations and conventions ..... 10
3.1 Terms and definitions from other ISO/IEC standards ..... 11
3.2 IEC 61158-1 terms ..... 11
3.3 Abbreviations and symbols ..... 15
3.4 Conventions ..... 16
3.5 Conventions used in state machines ..... 17
4 Protocol ..... 18
4.1 Overview ..... 18
4.2 FAL syntax description ..... 18
4.3 Transfer syntax ..... 18
4.4 FAL protocol state machine structure ..... 70
4.5 SMK state machine ..... 70
4.6 VCR state machine ..... 86
4.7 FAL service protocol machine (FSPM) ..... 87
4.8 Application relationship protocol machines (ARPMs) ..... 88
4.9 DLL mapping protocol machine (DMPM) ..... 103
Bibliography ..... 108
Annex ZA (normative) Normative references to international publications with their corresponding European publications ..... 109

Figure 1 – State transition diagram for SMK ..... 72
Figure 2 – State transition diagram of client / server ARPM ..... 91
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 ..... 17
Table 2 – Data types ..... 19
Table 3 – Data types ..... 19
Table 4 – APDU header format ..... 20
Table 5 – FDA address use ..... 21
Table 6 – FDA address header field APDUs sent by a client VCR endpoint ..... 22
Table 7 – FDA address header field APDUs sent by a server VCR endpoint ..... 22
Table 8 – FDA address header field APDUs sent by a publisher VCR endpoint ..... 23
Table 9 – FDA address header field APDUs sent by a report source VCR endpoint ..... 23
Table 10 – APDU trailer fields ..... 24
Table 11 – Request APDU parameters ..... 26
Table 12 – SMK FDA address values ..... 28



AVA Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

Table 13 – SMK FDA address values ..... 29
Table 14 – Request APDU parameters ..... 29
Table 15 – SMK FDA address values for SM identify ..... 31
Table 16 – SMK FDA address values for SMK set assignment info request APDUs ..... 31
Table 17 – SMK clear address request APDU parameters ..... 32
Table 18 – SMK FDA address values for SMK set assignment info request APDUs ..... 32
Table 19 – SMK set assignment info request APDU parameters ..... 32
Table 20 – SMK set assignment info response APDU parameters ..... 34
Table 21 – SMK FDA address values for SMK device clear assignment Info APDUs ..... 34
Table 22 – SMK clear assignment info request APDU parameters ..... 34
Table 23 – SMK FDA address values for SMK device annunciation request APDUs ..... 35
Table 24 – SMK device annunciation request APDU parameters ..... 36
Table 25 – Initiate request APDU parameters ..... 38
Table 26 – Initiate response APDU parameters ..... 38
Table 27 – Abort request APDU parameters ..... 39
Table 28 – Get response APDU parameters ..... 39
Table 29 – Identify response APDU parameters ..... 39
Table 30 – Get OD request APDU parameters ..... 40
Table 31 – Get OD response APDU parameters ..... 40
Table 32 – Initiate put OD request APDU parameters ..... 40
Table 33 – Put OD request APDU parameters ..... 41
Table 34 – Generic initiate download sequence request APDU parameters ..... 41
Table 35 – Generic download segment request APDU parameters ..... 42
Table 36 – Generic terminate download sequence request APDU parameters ..... 42
Table 37 – Response APDU parameters ..... 43
Table 38 – Initiate download sequence request APDU parameters ..... 43
Table 39 – Download segment request APDU parameters ..... 43
Table 40 – Download segment response APDU parameters ..... 44
Table 41 – Terminate download sequence request APDU parameters ..... 44
Table 42 – Initiate upload sequence request APDU parameters ..... 45
Table 43 – Upload segment request APDU parameters ..... 45
Table 44 – Upload segment response APDU parameters ..... 45
Table 45 – Terminate upload sequence request APDU parameters ..... 46
Table 46 – Request domain download request APDU parameters ..... 46
Table 47 – Request domain upload request APDU parameters ..... 46
Table 48 – Create program invocation request APDU parameters ..... 47
Table 49 – Create program invocation response APDU parameters ..... 47
Table 50 – Delete program invocation request APDU parameters ..... 48
Table 51 – Start request APDU parameters ..... 48
Table 52 – Stop request APDU parameters ..... 48
Table 53 – Resume request APDU parameters ..... 49
Table 54 – Reset request APDU parameters ..... 49



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

Table 55 – Kill request APDU parameters ..... 49
Table 56 – Read request APDU parameters ..... 50
Table 57 – Read response APDU parameters ..... 50
Table 58 – Read with subindex request APDU parameters ..... 50
Table 59 – Read with subindex response APDU parameters ..... 50
Table 60 – Write request APDU parameters ..... 51
Table 61 – Write with subindex request APDU parameters ..... 51
Table 62 – Define variable list request APDU parameters ..... 52
Table 63 – Define variable list response APDU parameters ..... 52
Table 64 – Delete variable list request APDU parameters ..... 52
Table 65 – Information report request APDU parameters ..... 52
Table 66 – Information report with subindex request APDU parameters ..... 53
Table 67 – Information report on change request APDU parameters ..... 53
Table 68 – Information report on change with subindex request APDU parameters ..... 54
Table 69 – Event notification request APDU parameters ..... 54
Table 70 – Alter event condition monitoring request APDU parameters ..... 54
Table 71 – Acknowledge event notification request APDU parameters ..... 54
Table 72 – LAN redundancy diagnostic message request APDU parameters ..... 55
Table 73 – LAN redundancy get information response APDU parameters ..... 57
Table 74 – LAN redundancy get statistics request APDU parameters ..... 59
Table 75 – Object description header ..... 60
Table 76 – Null object ..... 61
Table 77 – Structure of the list of object descriptions ..... 61
Table 78 – Structure of a load region in the S-OD ..... 62
Table 79 – Structure of a function invocation in the DP-OD ..... 62
Table 80 – Structure of an event in the S-OD ..... 63
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 ..... 64
Table 83 – Structure of a simple variable in the S-OD ..... 65
Table 84 – Structure of an array in the S-OD ..... 65
Table 85 – Structure of a record in the S-OD ..... 66
Table 86 – Structure of a variable list in the DV-OD ..... 66
Table 87 – Common error parameters ..... 67
Table 88 – PI error parameters ..... 67
Table 89 – OD error parameters ..... 68
Table 90 – Error class and error code values ..... 68
Table 91 – SMKPM service primitives ..... 71
Table 92 – SMKPM states ..... 72
Table 93 – SMKPM state table – initialization ..... 72
Table 94 – SMKPM state table – receive transitions ..... 73
Table 95 – SMKPM state table – internal events ..... 78
Table 96 – HseRepeatTimerExpires () ..... 79



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

Table 97 – RcvNewNetworkAddress (interface, address) ..... 79
Table 98 – RcvMsg () ..... 79
Table 99 – SntpSyncLost () ..... 79
Table 100 – AddressToClear (sm_svc) ..... 80
Table 101 – AssignmentInfo_Set () ..... 80
Table 102 – ConfigurationSessionActive () ..... 80
Table 103 – DeviceRedundancyState () ..... 80
Table 104 – DevId_Match (sm_svc) ..... 80
Table 105 – DuplicateQueryIdMatch (sm_svc) ..... 81
Table 106 – DuplicatePdTagDetected () ..... 81
Table 107 – FdaAddressType (sm_svc) ..... 81
Table 108 – IsValid (sm_svc) ..... 81
Table 109 – NetworkAddressChange (interface, address) ..... 82
Table 110 – NumberOfAssignedAddresses () ..... 82
Table 111 – OperationalRestore () ..... 82
Table 112 – PdTag_Match (sm_svc) ..... 82
Table 113 – PdTagDeviceIndex_Check (sm_svc) ..... 82
Table 114 – Query_Match (sm_svc) ..... 83
Table 115 – QueryType (sm_svc) ..... 83
Table 116 – SmCacheEntry (sm_svc) ..... 83
Table 117 – Clear_Address (interface_to_clear) ..... 83
Table 118 – Clear_DuplicatePdTagFlag () ..... 84
Table 119 – Get_AddlCode () ..... 84
Table 120 – New_Address (interface, address) ..... 84
Table 121 – Restart_HseRepeatTimer () ..... 84
Table 122 – Restore_Defaults () ..... 84
Table 123 – Send_SM_CommonErrorRsp (sm_service_type, svc_spec_params) ..... 85
Table 124 – Send_SM_ReqRspMessage (sm_svc) ..... 85
Table 125 – Set_Assignment_Data (sm_svc) ..... 85
Table 126 – Set_DuplicatePdTagFlag () ..... 85
Table 127 – SvcType (sm_svc) ..... 85
Table 128 – Additional code used by error class and code ..... 86
Table 129 – Additional code parameter IDs ..... 86
Table 130 – Primitives issued by FSPM to ARPM ..... 88
Table 131 – Primitives issued by ARPM to FSPM ..... 88
Table 132 – Parameters used with primitives exchanged between FSPM and ARPM ..... 89
Table 133 – Client / Server ARPM states ..... 91
Table 134 – Client / server ARPM state table – sender transitions ..... 92
Table 135 – Client / server ARPM state table – receiver transitions ..... 93
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 ..... 96



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

Table 139 – Publisher / subscriber ARPM states ..... 98
Table 140 – MulticastARPM state table – sender transitions ..... 98
Table 141 – MulticastARPM state table – receiver transitions ..... 99
Table 142 – BuildFAL-ErrPDU() ..... 100
Table 143 – BuildFAL-ReqRspPDU() ..... 100
Table 144 – GetArepId() ..... 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_GetVcrId() ..... 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



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

# 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.



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

# 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



AVA Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

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>

## 3 Terms, definitions, symbols, abbreviations and conventions

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

[1] Withdrawn



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

## 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.



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

**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



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

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.



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

**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.



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

**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 |



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

| | |
|---|---|
| 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.



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

### 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<br>/ condition<br>=> action | Next state |
|---|---|---|---|
| Name of this transition. | The current state to which this state transition applies | Events or conditions that trigger this state transaction.<br>=><br>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



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

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.



Documents
Document Sharing Hub

Full document details: DIN EN 61158-6-5:2015-09

### 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.



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

**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.<br>1 = Version 1 |
| Options | 1 | Unsigned8 | 1 | Bit 8: 1 = APDU Number present in the Trailer<br>Bit 7: 1 = Invoke Id present in the Trailer (this bit shall be set for all client / server sessions)<br>Bit 6: 1 = Time Stamp present in the Trailer<br>Bit 5: Reserved, set to 0<br>Bit 4: 1 = Extended Control Field present in the Trailer<br>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:<br>0 = Unused<br>1 = AR ASE<br>2 = SMK ASE<br>3 = Type 9 ASEs/Type 5 VFD ASE<br>4 = LAN Redundancy ASE<br>5-63 : Reserved<br>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:<br>0 = Request APDU<br>1 = Response APDU<br>2 = Error APDU<br>3 = Reserved |
| Service | 3 | Unsigned8 | 1 | Specifies the service within the specified protocol.<br>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.<br>Bits 7-1 Service Id of the service (7-bit unsigned integer).<br>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.



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

**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 | Link Id<br>Set to 0 if the destination SMK is located in the Type 5 device receiving the APDU.<br>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. |
| | | 16-1 | Node.Selector<br>The node value is the high order 8 bits and the selector value is the low order 8 bits, when taken as a16-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 | Link Id<br>If the server S-VFD is located in the Type 5 device, then this is 0.<br>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. |
| | | 16-1 | Type 5 Selector<br>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.<br>Initiate Response: The Type 5 FAL AE returns the VCR Id of the initiated VCR.<br>All Others: The Selector returned in the Initiate Response. |
| Type 9 | Publisher / Subscriber | 32-17 | Link Id<br>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.<br>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. |
| | | 16-1 | Dlcep Selector<br>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.<br>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 | Link Id<br>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.<br>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. |
| | | 16-1 | S-VFD Context Id or Dlsap Selector<br>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 Source VFD.<br>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. | | | |



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

#### 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<br>Connect Option 1 | Non-MIB VFD of Type 5 device | Bits 32-17<br>Bits 16-1 | 0 (Local Type 5 Device)<br>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<br>Bits 16-1 | Link Id of Type 9 Link<br>Client Type 9 VCR Index |
| Type 5 Initiate Request APDU<br>Connect Option 2 | MIB VFD of Type 5 device | Bits 32-17<br>Bits 16-1 | 0 (Local Type 5 Device)<br>0.0 |
| | MIB VFD of Linking Device Type 9 Link Interface | Bits 32-17<br>Bits 16-1 | Link Id of Type 9 Link<br>0.0 |
| | MIB VFD of Type 9 device connected to Linking Device | Bits 32-17<br>Bits 16-1 | Link Id of Type 9 Link<br>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<br>Connect Option 1 | Non-MIB VFD of Type 5 device | Bits 32-17<br>Bits 16-1 | 0 (Local Type 5 Device)<br>Type 5 VCR Id[1] |
| | Non-MIB VFD of Type 9 device connected to Linking Device | Bits 32-17<br>Bits 16-1 | Link Id of Type 9 Link<br>Type 5 VCR Id[1] |



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

| Message type | Destination AP | FDA address | |
|---|---|---|---|
| | | Bits | Value |
| Type 5 Initiate Response APDU<br>Connect Option 2 | MIB VFD of Type 5 device | Bits 32-17<br>Bits 16-1 | 0 (Local Type 5 Device)<br>Type 5 VCR Id[1] |
| | MIB VFD of Linking Device Type 9 Link Interface | Bits 32-17<br>Bits 16-1 | Link Id of Type 9 Link Interface<br>Type 5 VCR Id[1] |
| | MIB VFD of Type 9 device connected to Linking Device | Bits 32-17<br>Bits 16-1 | Link Id of Type 9 Link<br>Type 5 VCR Id[1] |
| All Other Response Messages | MIB VFD of Type 5 device | Bits 32-1 | Same as Request Message<br>Same as Request Message |
| | Non MIB VFD of Type 5 device | Bits 32-1 | Same as Request Message<br>Same as Request Message |
| | MIB VFD of Linking Device Type 9 Link | Bits 32-1 | Same as Request Message<br>Same as Request Message |
| | MIB VFD of Type 9 device connected to Linking Device | Bits 32-1 | Same as Request Message<br>Same as Request Message |
| | Non-MIB VFD of Type 9 device connected to Linking Device | Bits 32-1 | Same as Request Message<br>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 Type 9 link of a Linking Device | Bits 32-17<br>Bits 16-1 | Link Id of Type 9 Link<br>Node.Selector DIcep 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 Type 9 link of a Linking Device | Bits 32-17<br>Bits 16-1 | Link Id of Type 9 Link<br>Node.Selector DIsap of the source of the report |



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

### 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.



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

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.



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09

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 an 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 is 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 is 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. |



Documents
Document Sharing Hub
Full document details: DIN EN 61158-6-5:2015-09