

DIN EN 61158-5-5

ICS 35.100.70; 35.160

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

**Industrielle Kommunikationsnetze –
Fieldbusse –
Teil 5-5: Dienstfestlegungen des Application Layer
(Anwendungsschicht) – Typ 5-Elemente
(IEC 61158-5-5:2014);
Englische Fassung EN 61158-5-5:2014, nur auf CD-ROM**

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

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

Gesamtumfang 323 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2014-09-22 angenommene Europäische Norm als DIN-Norm ist 2015-09-01.

Für DIN EN 61158-5-5:2008-09 besteht eine Übergangsfrist bis 2017-09-22.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 61158-500:2013-08.

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 eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

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

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen 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 61158-5-5 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 die Dienste des Application Layer für Typ 5 fest.

Änderungen

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

- a) Auffüllen von Telegrammen hinzugefügt.

Frühere Ausgaben

DIN V 19245-1: 1988-01
DIN 19245-1: 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 61158-5: 2005-05
DIN EN 61158-5-5: 2008-09

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 der 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 vorbestimmten Grad an 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 Art und Weise den von außen sichtbaren Dienst, der vom Typ 5 des Feldbus-Application Layer bereitgestellt wird, in Form von:

- a) einem abstrakten Modell für das Beschreiben von Anwendungsressourcen (Objekten), die von Benutzern mittels FAL-Diensten verstellt werden können,
- b) Aktivitäten und Ereignissen der Dienstprimitiven,
- c) den mit den Aktivitäten und Ereignissen der Dienstprimitiven verbundenen Parametern, sowie der Form, die sie einnehmen und
- d) dem Zusammenhang zwischen diesen Aktivitäten und Ereignissen und ihren gültigen Abläufen.

Die Zielsetzung dieser Norm ist es, Dienste zu definieren, die bereitgestellt werden für:

- den FAL-Benutzer an der Grenze zwischen dem Benutzer und dem Application Layer des Feldbus-Referenzmodells und
- das Systemmanagement an der Grenze zwischen dem Application Layer und dem Systemmanagement des Feldbus-Referenzmodells.

Diese Norm beschreibt die Struktur und die Dienste 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-Anwendungseinheiten (AE) bereitgestellt, welche innerhalb von Anwendungsprozessen enthalten sind. Die FAL AE besteht aus einer Menge von objektorientierten Anwendungsdienstelementen (ASEs) und Layer-Management-Einheiten (LME), welche die AE verwalten. Die ASEs liefern Kommunikationsdienste, die auf einer Menge von zugehörigen Anwendungsprozessobjekt(APO-) Klassen arbeitet. Eine der FAL ASEs ist eine Management-ASE, welche eine Menge gemeinsamer Dienste 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 anfragenden und antwortenden Anwendungen mit diesen vorzunehmen haben. Die Verhaltensaspekte der Anwendungen sind nicht festgelegt, nur eine Definition, welche Anfragen und Antworten diese senden/empfangen können, ist beschrieben. Dies erlaubt den FAL-Benutzern eine größere Flexibilität, um solch ein Objektverhalten zu

^{N1)} Die Nummerierung aus IEC 61158-5-5 wurde beibehalten.

standardisieren. Zusätzlich zu diesen Diensten sind in dieser Norm ebenfalls einige unterstützende Dienste festgelegt, um den Zugriff zum FAL bereitzustellen und somit bestimmte Aspekte des Betriebs zu steuern.

1.2 Spezifikationen

Das hauptsächliche Ziel dieser Norm ist es, für zeitkritische Kommunikation geeignete Eigenschaften der konzeptionellen Application Layer-Dienste festzulegen und folglich die Ergänzung des OSI-Basisreferenzmodells mit der Anleitung der Entwicklung der Application Layer-Protokolle für die zeitkritische Kommunikation.

Ein zweites Ziel ist das Bereitstellen von Migrationspfaden von bisherigen bestehenden industriellen Kommunikationsprotokollen. Es ist dieses letztere Ziel, welches zur Steigerung von unterschiedlichen genormten Diensten, als verschiedene Typen der IEC 61158, führt.

Diese Spezifikation darf als Basis für formelle Application-Programmierschnittstellen verwendet werden. Nichtsdestotrotz ist es keine formelle Programmierschnittstelle, und jede derartige Schnittstelle hat Implementierungsanforderungen zu erfüllen, die nicht von dieser Spezifikation abgedeckt werden und Folgendes beinhalten:

- a) die Größen und Oktett-Folgen von verschiedenen Multi-oktett-Dienstparametern,
- b) die Korrelation von gepaarten Anfrage- und Indikationsprimitiven oder Anweisungs- und Antwortprimitiven.

1.3 Konformität

Diese Norm legt keine individuellen Implementierungen oder Produkte fest, auch beschränkt sie die Implementierungen von Application Layer-Einheiten innerhalb eines industriellen Automatisierungssystems nicht.

Es gibt keine Konformität von Geräten zu dieser Application Layer-Dienstdefinitionsnorm. Stattdessen wird Konformität durch die Implementierung der entsprechenden Application Layer-Protokolle erreicht, welche den Typ 5 der Application Layer-Dienste, wie in dieser Norm beschrieben, erfüllen.

3.6 Spezielle Begriffe für den Feldbus-Application Layer Typ 5

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.6.1

Zugriffsschutz

(en: access protection)

Beschränkung der Nutzung eines Anwendungsobjekts auf einen Client

3.6.2

ACCO

(en: active connection control object)

Instanz einer bestimmten FAL-Klasse, die die Verbindungseinrichtung eines Automatisierungsgerätes (als Konsument und Provider) abstrahiert

3.6.3

Adresszuweisungstabelle

(en: address assignment table)

Zuordnung des internen E/A-Datenobjektspeichers des Clients zu den dezentralen Eingabe- und Ausgabedatenobjekten

3.6.4

zuweisen

(en: allocate)

eine Ressource aus einem allgemeinen Bereich für die ausschließliche Nutzung durch eine spezifische Entität zuordnen

3.6.5

Anwendung

(en: application)

Funktion oder Datenstruktur, für die Daten konsumiert oder produziert werden

3.6.6

Application Layer-Interoperabilität

(en: application layer interoperability)

Fähigkeit von Anwendungsentitäten, mithilfe der Dienste des FAL koordinierte und kooperative Operationen auszuführen

3.6.7

Anwendungsobjekte

(en: application objects)

mehrere Objektklassen, die zur Laufzeit für einen Nachrichtenaustausch im Netzwerk und innerhalb des Netzwerkgerätes sorgen und diesen verwalten

3.6.8

Anwendungsprozess

(en: application process)

Teil einer verteilten Anwendung in einem Netzwerk, der sich auf einem Gerät befindet und eindeutig adressiert wird

3.6.9

Anwendungsprozesskennung

(en: application process identifier)

unterscheidet mehrere in einem Gerät verwendete Anwendungsprozesse

3.6.10

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 von 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.

3.6.11

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

3.6.12

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.

3.6.13

Anwendungsdienstelement für Anwendungsbeziehungen

(en: application relationship application service element)

Anwendungsdienstelement, das die alleinigen Mittel für den Aufbau und die Beendigung aller Anwendungsbeziehungen bereitstellt

3.6.14**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.

3.6.15**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. Sie liefern üblicherweise Statusinformationen oder steuern den Betrieb eines Objekts. Attribute können auch das Verhalten eines Objekts beeinflussen. Attribute werden in Klassenattribute und Instanzattribute unterteilt.

3.6.16**Verhalten**

(en: behaviour)

Anzeige dafür, wie das Objekt auf bestimmte Ereignisse reagiert

3.6.17**Bitnummer**

(en: bit-no)

kennzeichnet die Nummer eines Bits in einer Bitfolge oder einem Oktett

3.6.18**Kanal**

(en: channel)

einzelner physischer oder logischer Verbindungsweig eines Eingabe- oder Ausgabe-Anwendungsobjektes eines Servers zum Prozess

3.6.19**kanalbezogene Diagnose**

(en: channel related diagnosis)

Informationen in Bezug auf ein spezifisches Element eines Eingabe- oder Ausgabe-Anwendungsobjektes, die zu Wartungszwecken bereitgestellt werden

BEISPIEL Gültigkeit von Daten.

3.6.20**Klasse**

(en: class)

Menge von Objekten, die allesamt den gleichen Typ von Systemkomponenten repräsentieren

Anmerkung 1 zum Begriff: Eine Klasse ist eine Verallgemeinerung eines 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.

3.6.21**Klassenattribute**

(en: class attributes)

Attribut, welches allen Objekten innerhalb derselben Klasse gemeinsam ist

3.6.22**Klassencode**

(en: class code)

eindeutige Kennung, die jeder Objektklasse zugewiesen ist

3.6.23

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.

3.6.24

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

3.6.25

Konfigurationsprüfung

(en: configuration check)

Vergleich der erwarteten Strukturierung des E/A-Datenobjekts des Clients mit der tatsächlichen Strukturierung des E/A-Datenobjekts beim Server in der Anlaufphase

3.6.26

Konfigurationsdatenbank

(en: configuration data base)

Verbindungsinformationen, die von dem ACCO-ASE verwaltet werden

3.6.27

Konfigurationsfehler

(en: configuration fault)

durch den Server erkannter unzulässiger Unterschied zwischen der erwarteten Strukturierung des E/A-Datenobjekts und der tatsächlichen Strukturierung des E/A-Datenobjekts

3.6.28

Konfigurationskennung

(en: configuration identifier)

Repräsentation eines Teils von E/A-Daten eines einzelnen Eingabe- und/oder Ausgabemoduls eines Servers

3.6.29

Verbindung

(en: connection)

logische Verknüpfung zwischen Anwendungsobjekten, die sich innerhalb eines Gerätes oder in unterschiedlichen Geräten befinden können

Anmerkung 1 zum Begriff: Verbindungen können entweder Punkt-zu-Punkt-Verbindungen oder Mehrpunktverbindungen sein.

3.6.30

Verbindungskanal

(en: connection channel)

Beschreibung einer Verbindung zwischen einer Senke und einer Quelle von Datenelementen

3.6.31

Verbindungs-ID

(en: Connection ID; CID)

Kennung, die einer Übertragung im Zusammenhang mit einer bestimmten Verbindung zwischen Produzenten und Konsumenten zugewiesen ist und die einen Namen für einen bestimmten Teil von Anwendungsinformationen bereitstellt

3.6.32

Verbindungspfad

(en: connection path)

Oktettenstrom, der das Anwendungsobjekt definiert, auf das sich eine Verbindungsinstanz bezieht

3.6.33

Verbindungspunkt

(en: connection point)

Puffer, der als Unterinstanz eines Assembly-Objekts dargestellt ist

3.6.34

konsumieren

(en: consume)

Vorgang des Empfangs von Daten von einem Produzenten

3.6.35

Konsument

(en: consumer)

Knoten oder Senke, der/die Daten von einem Produzenten empfängt

3.6.36

konsumierende Anwendung

(en: consuming application)

Anwendung, die Daten konsumiert

3.6.37

Steuerbefehle

(en: control commands)

vom Client zum Server übertragene Aktionsaufrufe, um Ausgänge freizugeben, Eingänge zu blockieren (Freeze) und/oder Ausgänge zu synchronisieren

3.6.38

Übertragungspfad

(en: conveyance path)

unidirektionaler Strom von APDUs in einer Anwendungsbeziehung

3.6.39

zyklisch

(en: cyclic)

sich regelmäßig wiederholend

3.6.40

Datenkonsistenz

(en: data consistency)

Mittel für die kohärente Übertragung des Eingabe- oder Ausgabedatenobjekts zwischen Client und Server und für den kohärenten Zugriff auf das Eingabe- oder Ausgabedatenobjekt innerhalb von Client und Server

3.6.41

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.

3.6.42

Gerät

(en: device)

an den Verbindungszweig angeschlossene physische Hardware

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

3.6.43

Geräteprofil

(en: device profile)

Zusammenstellung geräteabhängiger Informationen und Funktionalitäten, um Konsistenz zwischen ähnlichen Geräten desselben Gerätetyps herzustellen

3.6.44

Diagnose-Informationen

(en: diagnosis information)

alle Daten, die serverseitig für Wartungszwecke verfügbar sind

3.6.45

Sammlung von Diagnose-Informationen

(en: diagnosis information collection)

Diagnose-Informationen zum System, die clientseitig zusammengetragen werden

3.6.46

dynamische AR

(en: dynamic AR)

Anwendungsbeziehung, die nur unter Verwendung der Prozeduren zum Aufbau von ARs in einen aktiven Zustand versetzt werden kann

3.6.47

Endknoten

(en: end node)

produzierender oder konsumierender Knoten

3.6.48

Endpunkt

(en: endpoint)

eine der Daten übermittelnden Entitäten, die an einer Verbindung beteiligt ist

3.6.49

Engineering

(en: engineering)

abstrakte Bezeichnung, welche die Client-Anwendung oder das Gerät kennzeichnet, die/das für die Konfiguration eines Automatisierungssystems durch Verbinden von Datenelementen verantwortlich ist

3.6.50

Fehler

(en: error)

Nichtübereinstimmung zwischen einem berechneten, beobachteten oder gemessenen Wert oder Zustand und dem festgelegten oder theoretisch richtigen Wert oder Zustand

3.6.51

Fehlerklasse

(en: error class)

allgemeine Gruppierung für zugehörige Fehlerdefinitionen und entsprechende Fehlercodes

3.6.52

Fehlercode

(en: error code)

Kennzeichnung eines bestimmten Fehlertyps innerhalb einer Fehlerklasse

3.6.53

Ereignis

(en: event)

Instanz einer Änderung von Zuständen

3.6.54

FAL-Subnetz

(en: FAL subnet)

Subnetzwerke, die aus einem oder mehreren Data Link-Segmenten bestehen und durch eine Untermenge der Netzwerkadresse identifiziert werden

Anmerkung 1 zum Begriff: FAL-Subnetze dürfen Bridges, jedoch keine Router umfassen.

3.6.55**FIFO-Variable**

(en: FIFO variable)

Variablenobjektklasse, die aus einem Satz von einheitlich typisierten Elementen besteht, wobei das erste geschriebene Element das erste Element ist, das gelesen werden kann

Anmerkung 1 zum Begriff: Auf dem Feldbus kann nur ein einziges vollständiges Element als Ergebnis eines einzigen Dienstaufrufs übertragen werden.

3.6.56**Rahmen**

(en: frame)

umgangssprachliches Synonym für DLPDU

3.6.57**Freeze**

(en: freeze)

Funktion an den DP-Slaves für den gleichzeitigen Datentransfer zwischen dem Eingabedatenobjekt und dem Prozess

3.6.58**Gruppe**

(en: group)

- a) <allgemein> eine allgemeine Bezeichnung für eine Zusammenstellung von Objekten; Spezifische Verwendungen:
- b) <Adressierung> bei der Beschreibung einer Adresse: eine Adresse, die mehr als eine Einheit identifiziert

3.6.59**Schnittstelle**

(en: interface)

- a) gemeinsam genutzte Grenze zwischen zwei funktionalen Einheiten, die durch funktionale Merkmale, Signalmerkmale oder entsprechend andere Merkmale festgelegt ist;
- b) Zusammenstellung von FAL-Klassenattributen und -diensten, die eine spezifische Sicht auf die FAL-Klasse darstellt

3.6.60**Schnittstellendefinitionssprache**

(en: interface definition language)

Syntax und Semantik zur formalen Beschreibung von Dienstparametern

Anmerkung 1 zum Begriff: Diese Beschreibung ist die Eingabe für das ORPC-Modell, insbesondere für das ORPC-Netzwerkprotokoll.

3.6.61**Schnittstellenzeiger**

(en: interface pointer)

Schlüsselattribut, das eine Objektschnittstelleninstanz eindeutig adressiert

3.6.62**Aufruf**

(en: invocation)

Vorgang der Nutzung eines Dienstes oder einer anderen Ressource eines Anwendungsprozesses

Anmerkung 1 zum Begriff: Jeder Aufruf bildet einen einzelnen Steuerungsthread, der durch seinen Kontext beschrieben werden kann. Sobald der Dienst ausgeführt oder die Nutzung der Ressource beendet ist, erlischt der Aufruf. Bei Dienstaufufen wird ein Dienst, der initiiert, aber noch nicht ausgeführt wurde, als ausstehender Dienstaufuf bezeichnet.

3.6.63**E/A-Daten**

(en: I/O data)

Objekt, das zu Verarbeitungszwecken zyklisch übertragen wird

3.6.64

kennungsbezogene Diagnose

(en: identifier related diagnosis)

Informationen bezogen auf Module für Wartungszwecke

3.6.65

Index

(en: index)

Adresse eines Objekts innerhalb eines Anwendungsprozesses

3.6.66

Instanz

(en: instance)

tatsächliches physisches Vorkommen eines Objekts innerhalb einer Klasse, das eines von mehreren Objekten innerhalb derselben Objektklasse identifiziert

BEISPIEL „Kalifornien“ ist eine Instanz der Objektklasse „US-Staat“.

Anmerkung 1 zum Begriff: Die Benennungen „Objekt“, „Instanz“ und „Objektinstanz“ beziehen sich auf eine spezifische Instanz.

3.6.67

Instanzattribut

(en: instance attributes)

Attribut, das für eine Objektinstanz eindeutig ist und sich nicht auf die Objektklasse bezieht

3.6.68

instanziiert

(en: instantiated)

Objekt, das in einem Gerät erzeugt wurde

3.6.69

logisches Gerät

(en: logical device)

eine bestimmte FAL-Klasse, die eine Softwarekomponente oder eine Firmwarekomponente als eigenständige Einrichtung eines Automatisierungsgerätes abstrahiert

3.6.70

Hersteller-ID

(en: manufacturer ID)

Kennzeichnung jedes Produktherstellers durch eine eindeutige Nummer

3.6.71

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.

3.6.72

Member

(en: member)

Teil eines Attributs, das als Element eines Feldes strukturiert ist

3.6.73

Nachrichtenrouter

(en: message router)

Objekt innerhalb eines Knotens, das Nachrichtenanforderungen an entsprechende Anwendungsobjekte verteilt

3.6.74**Methode**

(en: method)

<Objekt> Synonym für einen funktionsfähigen Dienst, der von dem Server-ASE bereitgestellt und von einem Client aufgerufen wird

3.6.75**Modul**

(en: module)

- a) <allgemein> Hardware oder logische Komponente eines physischen Gerätes;
- b) <Typ 3> adressierbare Einheit innerhalb des DP-Slaves

3.6.76**Mehrpunktverbindung**

(en: multipoint connection)

Verbindung von einem Knoten zu vielen Knoten

Anmerkung 1 zum Begriff: Durch Mehrpunktverbindungen können Nachrichten von einem einzelnen Produzenten von vielen Konsumentenknuten empfangen werden.

3.6.77**Netzwerk**

(en: network)

Menge von Knoten, die über ein bestimmtes Kommunikationsmedium miteinander verbunden sind, einschließlich zwischengeschalteter Repeater, Bridges, Router und Gateways unterer Schichten

3.6.78**Objekt**

(en: object)

abstrakte Darstellung einer bestimmten Komponente innerhalb eines Gerätes, normalerweise eine Zusammenstellung zugehöriger Daten (in Form von Variablen) und Methoden (Prozeduren) für Operationen mit diesen Daten, die eine klar definierte Schnittstelle und ein klar definiertes Verhalten aufweisen

3.6.79**ORPC**

(en: object remote procedure call)

Modell für den objektorientierten oder komponentenbasierten Aufruf von entfernten Methoden

3.6.80**objektspezifischer Dienst**

(en: object specific service)

Dienst, der für die Objektklasse, die ihn definiert, eindeutig ist

3.6.81**Absender**

(en: originator)

Client, der für den Aufbau eines Verbindungspfads zum Ziel verantwortlich ist

3.6.82**Peer**

(en: peer)

Rolle eines AR-Endpunktes, in der dieser in der Lage ist, sowohl als Client als auch als Server zu agieren

3.6.83**physisches Gerät**

(en: physical device)

Automatisierungsgerät oder anderes Netzwerkgerät

3.6.84

Punkt-zu-Punkt-Verbindung

(en: point-to-point connection)

Verbindung, die zwischen genau zwei Anwendungsobjekten besteht

3.6.85

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 vorher aktiviert sind, werden vor ihrer Nutzung aktiviert.

3.6.86

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

3.6.87

Prozessdaten

(en: process data)

Objekte, die bereits vorverarbeitet sind und zu Informationszwecken oder zum Zweck der weiteren Verarbeitung azyklisch übertragen werden

3.6.88

produzieren

(en: produce)

Vorgang des Sendens von Daten zum Empfang durch einen Konsumenten

3.6.89

Produzent

(en: producer)

Knoten, der für das Senden von Daten verantwortlich ist

3.6.90

Eigenschaft

(en: property)

allgemeine Bezeichnung für beschreibende Informationen zu einem Objekt

3.6.91

Provider

(en: provider)

Quelle einer Datenverbindung

3.6.92

Provider-ID

(en: providerID)

eindeutige Kennung innerhalb des Anwendungsbereiches des ACCO, die vom Provider zugewiesen wird, um die internen Daten einer konfigurierten Verbindungsquelle zu erkennen

3.6.93

Publisher

(en: publisher)

Rolle eines AR-Endpunktes, in der dieser APDUs auf den Feldbus für den Verbrauch durch einen oder mehrere Subscriber überträgt

Anmerkung 1 zum Begriff: Ein Publisher muss die Identität oder Anzahl der Subscriber nicht kennen und kann seine APDUs über eine zugeordnete AR veröffentlichen.

3.6.94**veröffentlichender Manager**

(en: publishing manager)

Rolle eines AR-Endpunktes, in der dieser eine oder mehrere bestätigte Dienstanforderungs-APDUs an einen Publisher ausgibt, um die Veröffentlichung eines angegebenen Objektes vom Publisher anzufordern

Anmerkung 1 zum Begriff: In dieser Norm sind zwei Typen von veröffentlichenden Managern festgelegt, die jeweils einzeln definiert sind: veröffentlichende Pull-Manager und veröffentlichende Push-Manager.

3.6.95**Pull-Publisher**

(en: pull publisher)

Typ eines Publishers, der ein Objekt als Antwort auf eine von dem entsprechenden veröffentlichenden Pull-Manager empfangende Anforderung veröffentlicht

3.6.96**veröffentlichender Pull-Manager**

(en: pull publishing manager)

Typ eines veröffentlichenden Managers, der die Veröffentlichung eines bestimmten Objekts in einer entsprechenden Antwort-APDU anfordert

3.6.97**Push-Publisher**

(en: push publisher)

Typ eines Publishers, der ein Objekt in einer unbestätigten Dienstanforderungs-APDU veröffentlicht

3.6.98**veröffentlichender Push-Manager**

(en: push publishing manager)

Typ eines veröffentlichenden Managers, der die Veröffentlichung eines bestimmten Objekts mithilfe eines unbestätigten Dienstes anfordert

3.6.99**Pull-Subscriber**

(en: pull subscriber)

Typ eines Subscribers, der empfangene bestätigte Dienstantwort-APDUs als veröffentlichte Objektdaten erkennt

3.6.100**Push-Subscriber**

(en: push subscriber)

Typ eines Subscribers, der empfangene unbestätigte Dienstanforderungs-APDUs als veröffentlichte Objektdaten erkennt

3.6.101**Qualitätscode**

(en: quality code)

zusätzliche Statusinformation eines Datenelementes

3.6.102**Ressource**

(en: resource)

Verarbeitungs- oder Informationsfähigkeit eines Subsystems

3.6.103**Routen-Endpunkt**

(en: route endpoint)

Objektcontainer, der Variablenobjekte einer Variablenklasse enthält

3.6.104

Laufzeitobjektmodell

(en: runtime object model)

Objekte, die in einem Gerät zusammen mit ihren Schnittstellen und Methoden vorhanden sind, auf welche zugegriffen werden kann

3.6.105

Server

(en: server)

- a) Rolle eines AREP, in der dieser eine bestätigte Dienstanwort-APDU an den Client, der die Anforderung initiiert hat, zurückgibt;
- b) Objekt, das einem anderen (Client-)Objekt Dienste bereitstellt

3.6.106

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

3.6.107

Slot

(en: slot)

Adresse eines Moduls innerhalb eines DP-Slaves

3.6.108

Subscriber

(en: subscriber)

Rolle eines AREP, in der dieser von einem Publisher produzierte APDUs empfängt

3.6.109

Sync

(en: sync)

Funktion an den DP-Slaves für den gleichzeitigen Datentransfer zwischen dem Ausgabedatenobjekt und dem Prozess

3.6.110

Ziel

(en: target)

Endknoten, mit dem eine Verbindung aufgebaut wird

3.6.111

verbindungsloser Dienst

(en: unconnected service)

Nachrichtendienst, der nicht auf den Aufbau einer Verbindung zwischen Geräten angewiesen ist, bevor Informationen ausgetauscht werden können

English Version

**Industrial communication networks - Fieldbus specifications -
Part 5-5: Application layer service definition - Type 5 elements
(IEC 61158-5-5:2014)**

Réseaux de communication industriels - Spécifications des
bus de terrain - Partie 5-5: Définition des services de la
couche application - Éléments de type 5
(CEI 61158-5-5:2014)

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

This European Standard was approved by CENELEC on 2014-09-22. 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

Foreword

The text of document 65C/763/FDIS, future edition 2 of IEC 61158-5-5, 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 61158-5-5:2014.

The following dates are fixed:

- latest date by which the document has to be (dop) 2015-06-22
implemented at national level by
publication of an identical national
standard or by endorsement
- latest date by which the national (dow) 2017-09-22
standards conflicting with the
document have to be withdrawn

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

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 61158-5-5:2014 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 61784-1	NOTE	Harmonized as EN 61784-1.
IEC 61784-2	NOTE	Harmonized as EN 61784-2.

CONTENTS

INTRODUCTION.....	8
1 Scope	9
1.1 General.....	9
1.2 Specifications	10
1.3 Conformance	10
2 Normative references	10
3 Terms and definitions	11
3.1 ISO/IEC 7498-1 terms.....	11
3.2 ISO/IEC 8822 terms.....	11
3.3 ISO/IEC 9545 terms.....	11
3.4 ISO/IEC 8824 terms.....	12
3.5 Fieldbus data-link layer terms	12
3.6 Fieldbus application layer specific terms and definitions	12
3.7 Abbreviations and symbols	22
3.8 Conventions	24
4 Concepts.....	27
5 Data type ASE.....	27
5.1 Overview	27
5.2 Formal definition of data type objects	27
5.3 FAL defined data types	29
5.4 Data type ASE service specification	66
6 Communication model specification	66
6.1 Concepts.....	66
6.2 ASEs.....	66
6.3 ARs.....	208
6.4 Summary of FAL classes	232
6.5 Permitted FAL services by AREP role	233
7 Type 5 communication model specification.....	234
7.1 Concepts.....	234
7.2 ASEs.....	257
7.3 FDA sessions	292
7.4 Summary of FAL Type 9 and Type 5 classes	302
7.5 Permitted FAL Type 9 and Type 5 services by AREP role	302
Bibliography	305
Annex ZA (normative) Normative references to international publications with their corresponding European publications.....	306
Figure 1 – The AR ASE conveys APDUs between APs	96
Figure 2 – 1-to-1 AR establishment	108
Figure 3 – 1-to-many AR establishment.....	108
Figure 4 – Event model overview.....	148
Figure 5 – Residence timeliness.....	222
Figure 6 – Synchronized timeliness	223
Figure 7 – Residence timeliness.....	229

Figure 8 – Synchronized timeliness	230
Figure 9 – VCR initiation	241
Figure 10 – Misordered message handling	247
Figure 11 – FF SM port message processing order	248
Figure 12 – FF FDA port message processing order	248
Figure 13 – FF TCP connection message processing order	249
Figure 14 – Session endpoint message processing order	249
Figure 15 – FDA LAN redundancy port message processing order	249
Figure 16 – Message processing by receiving entity	250
Table 1 – PERSISTDEF	34
Table 2 – VARTYPE	34
Table 3 – ITEMQUALITYDEF	36
Table 4 – STATEDEF	40
Table 5 – GROUPEXCEPTIONDEF	40
Table 6 – ACCESSRIGHTSDEF	40
Table 7 – HRESULT	41
Table 8 – UUID	47
Table 9 – Data type names for value	63
Table 10 – UUID	65
Table 11 – Create service parameters	67
Table 12 – Delete service parameters	68
Table 13 – Get attributes service parameters	69
Table 14 – Set attributes service parameters	72
Table 15 – Begin set attributes	73
Table 16 – End set attributes	74
Table 17 – Subscribe service parameters	83
Table 18 – Identify	86
Table 19 – Get status	87
Table 20 – Status notification	88
Table 21 – Initiate	89
Table 22 – Terminate	92
Table 23 – Conclude	94
Table 24 – Reject	94
Table 25 – Conveyance of service primitives by AREP role	97
Table 26 – Valid combinations of AREP roles involved in an AR	97
Table 27 – AR-Unconfirmed send	103
Table 28 – AR-Confirmed send	105
Table 29 – AR-Establish service	107
Table 30 – Valid combinations of AREP classes to be related	109
Table 31 – AR-Deestablish service	110
Table 32 – AR-Abort	111

Table 33 – AR-Compel service	112
Table 34 – AR-Get buffered message service	113
Table 35 – AR-Schedule communication service	114
Table 36 – AR-Cancel scheduled sequence service	115
Table 37 – AR-Status	116
Table 38 – AR-XON-OFF	117
Table 39 – AR-Remote read service	117
Table 40 – AR-Remote write service	119
Table 41 – Read service parameters	128
Table 42 – Read list service parameters	130
Table 43 – Write service parameters	133
Table 44 – Write list service parameters	135
Table 45 – Information report service	137
Table 46 – Information report list service	138
Table 47 – Exchange service parameters	140
Table 48 – Exchange list service parameters	144
Table 49 – Acknowledge event	156
Table 50 – Acknowledge event list service parameters	157
Table 51 – Enable event	159
Table 52 – Event notification service parameters	160
Table 53 – Enable event list	162
Table 54 – Notification recovery service parameters	163
Table 55 – Get event summary service parameters	164
Table 56 – Get event summary list service parameters	166
Table 57 – Query event summary list service parameters	169
Table 58 – Initiate load service parameters	176
Table 59 – Terminate load service parameters	178
Table 60 – Push segment service parameters	179
Table 61 – Pull segment service parameters	180
Table 62 – Discard service parameters	182
Table 63 – Pull upload sequencing of service primitives	183
Table 64 – Pull upload service parameter constraints	184
Table 65 – Pull upload state table	185
Table 66 – Pull download sequencing of service primitives	186
Table 67 – Pull download service parameter constraints	186
Table 68 – Pull download state table	187
Table 69 – Push download sequencing of service primitives	189
Table 70 – Push download service parameter constraints	189
Table 71 – Push download state table	190
Table 72 – Start service parameters	197
Table 73 – Stop service parameters	198
Table 74 – Resume service parameters	199

Table 75 – Reset service parameters	200
Table 76 – Kill service parameters	201
Table 77 – Action invoke service parameters.....	202
Table 78 – Action return service parameters	203
Table 79 – State transitions for a function invocation object.....	205
Table 80 – FAL class summary	232
Table 81 – Services by AREP role	233
Table 82 – Scope of Invoke Id.....	245
Table 83 – Types of misordering detectable by message numbers	246
Table 84 – Delivery of misordered message types on publisher/subscriber VCRs.....	246
Table 85 – Statistics gathered per VCR.....	246
Table 86 – Determination of misordering type at a subscriber VCR	247
Table 87 – Mapping of received messages to primitives	247
Table 88 – Mapping of received primitives to messages	248
Table 89 – Defined network addresses.....	251
Table 90 – Use of network addresses.....	252
Table 91 – Use of endpoint selectors in server VCRs	252
Table 92 – Use of endpoint selectors in publisher VCRs	253
Table 93 – Use of endpoint selectors in source VCRs.....	253
Table 94 – Network address and port numbers for device annunciation.....	255
Table 95 – Network address and port numbers for set/clear assignment info and clear address.....	255
Table 96 – Network address and port numbers for SM identify.....	255
Table 97 – Network address and port numbers for SM find tag	255
Table 98 – Network address and port numbers for clients and servers (part 1)	255
Table 99 – Network address and port numbers for clients and servers (part 2)	256
Table 100 – Network address and port numbers for publishers and subscribers	256
Table 101 – Network address and port numbers for report distribution	256
Table 102 – Network address and port numbers for LAN redundancy get and put information.....	256
Table 103 – Network address and port numbers for LAN redundancy diagnostics.....	256
Table 104 – VCR types.....	258
Table 105 – Use of VCR user id.....	259
Table 106 – Use of FDA address	259
Table 107 – Initiate.....	261
Table 108 – Connect option	262
Table 109 – Find tag query service parameters	267
Table 110 – SMK IDs.....	268
Table 111 – Find tag reply service parameters	269
Table 112 – Identify service parameters	271
Table 113 – Annunciate service parameters	274
Table 114 – Set assignment info service parameters	276
Table 115 – Clear assignment info service parameters	279

Table 116 – Clear address service parameters	281
Table 117 – Diagnostic message service	286
Table 118 – Get redundancy info service	287
Table 119 – Put redundancy info service	289
Table 120 – Get redundancy statistics service	291
Table 121 – Open session service	298
Table 122 – Idle session service	301
Table 123 – FAL class summary	302
Table 124 – Services by AREP role	303

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 service is provided by the application protocol making use of the services available from the data-link or other immediately lower layer. This standard defines the application service characteristics that fieldbus applications and/or system management may exploit.

Throughout the set of fieldbus standards, the term “service” refers to the abstract capability provided by one layer of the OSI Basic Reference Model to the layer immediately above. Thus, the application layer service defined in this standard is a conceptual architectural service, independent of administrative and implementation divisions.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-5: Application layer service definition – 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 service provided by the Type 5 fieldbus application layer in terms of

- a) an abstract model for defining application resources (objects) capable of being manipulated by users via the use of the FAL service,
- b) the primitive actions and events of the service;
- c) the parameters associated with each primitive action and event, and the form which they take; and
- d) the interrelationship between these actions and events, and their valid sequences.

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

- 1) the FAL user at the boundary between the user and the application layer of the fieldbus reference model, and
- 2) Systems Management at the boundary between the application layer and Systems Management of the fieldbus reference model.

This standard specifies the structure and services of the Type 2 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498) 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 characteristics of conceptual application layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of application layer protocols for time-critical communications.

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 services standardized as the various types of IEC 61158.

This specification may be used as the basis for formal application programming interfaces. Nevertheless, it is not a formal programming interface, and any such interface will need to address implementation issues not covered by this specification, including

- a) the sizes and octet ordering of various multi-octet service parameters, and
- b) the correlation of paired request and confirm, or indication and response, primitives.

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.

There is no conformance of equipment to this application layer service definition standard. Instead, conformance is achieved through implementation of conforming application layer protocols that fulfill the Type 5 application layer services as defined in this standard.

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 61131-3, *Programmable controllers – Part 3: Programming languages*

IEC 61158-1:2014, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-3-1, *Industrial communication networks – Fieldbus specifications – Part 3-1: Data-link layer service definition – Type 1 elements*

IEC 61158-4-1, *Industrial communication networks – Fieldbus specifications – Part 4-1: Data-link layer protocol specification – Type 1 elements*

IEC 61158-5:2014 (all parts), *Industrial communication networks – Fieldbus specifications – Part 5: Application layer service definition*

IEC 61158-6-5, *Industrial communication networks – Fieldbus specifications – Part 6-5: Application layer protocol specification – Type 5 elements*

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange*

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

ISO/IEC 8822, *Information technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 8824: 1990, *Information technology – Open Systems Interconnection – Specification of 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*

ANSI/IEEE 754-1985, *Binary Floating-Point Arithmetic*

3 Terms and definitions

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

3.1 ISO/IEC 7498-1 terms

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

3.2 ISO/IEC 8822 terms

- a) abstract syntax
- b) presentation context

3.3 ISO/IEC 9545 terms

- a) application-association
- b) application-context
- c) application context name
- d) application-entity-invocation
- e) application-entity-type

¹ Withdrawn.

- f) application-process-invocation
- g) application-process-type
- h) application-service-element
- i) application control service element

3.4 ISO/IEC 8824 terms

- a) object identifier
- b) type

3.5 Fieldbus data-link layer terms

For the purposes of this document, the following terms as defined in IEC 61158-3-1 and IEC 61158-4-1 apply.

- a) DL-Time
- b) DL-Scheduling-policy
- c) DLCEP
- d) DLC
- e) DL-connection-oriented mode
- f) DLPDU
- g) DLSDU
- h) DLSAP
- i) fixed tag
- j) generic tag
- k) link
- l) MAC ID
- m) network address
- n) node address
- o) node
- p) tag
- q) scheduled
- r) unscheduled

3.6 Fieldbus application layer specific terms and definitions

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

3.6.1

access protection

limitation of the usage of an application object to one client

3.6.2

active connection control object

instance of a certain FAL class that abstracts the interconnection facility (as Consumer and Provider) of an automation device

3.6.3

address assignment table

mapping of the client's internal I/O-Data object storage to the decentralised input and output data objects

3.6.4

allocate

take a resource from a common area and assign that resource for the exclusive use of a specific entity

3.6.5

application

function or data structure for which data is consumed or produced

3.6.6

application layer interoperability

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

3.6.7

application objects

multiple object classes that manage and provide a run time exchange of messages across the network and within the network device

3.6.8

application process

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

3.6.9

application process identifier

component that distinguishes multiple application processes used in a device

3.6.10

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.

3.6.11

application process object class

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

3.6.12

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.

3.6.13

application relationship application service element

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

3.6.14

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.

3.6.15

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.

3.6.16

behaviour

indication of how an object responds to particular events

3.6.17

bit-no

designates the number of a bit in a bitstring or an octet

3.6.18

channel

single physical or logical link of an input or output application object of a server to the process

3.6.19

channel related diagnosis

information concerning a specific element of an input or output application object, provided for maintenance purposes

EXAMPLE: validity of data

3.6.20

class

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

Note 1 to entry: A class is a generalisation of an 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.

3.6.21

class attributes

attribute that is shared by all objects within the same class

3.6.22

class code

unique identifier assigned to each object class

3.6.23

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.

3.6.24

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

3.6.25

configuration check

comparison of the expected I/O-Data object structuring of the client with the real I/O-Data object structuring to the server in the start-up phase

3.6.26

configuration data base

interconnection information maintained by the ACCO ASE

3.6.27

configuration fault

an unacceptable difference between the expected I/O-Data object structuring and the real I/O-Data object structuring, as detected by the server

3.6.28

configuration identifier

representation of a portion of I/O Data of a single input- and/or output-module of a server

3.6.29

connection

logical binding between application objects that may be within the same or different devices

Note 1 to entry: Connections may be either point-to-point or multipoint.

3.6.30

connection channel

description of a connection between a sink and a source of data items

3.6.31

connection ID

CID

identifier assigned to a transmission that is associated with a particular connection between producers and consumers, providing a name for a specific piece of application information

3.6.32

connection path

an octet stream that defines the application object to which a connection instance applies

3.6.33

connection point

buffer which is represented as a subinstance of an Assembly object

3.6.34

consume

act of receiving data from a producer

3.6.35

consumer

node or sink that is receiving data from a producer

3.6.36

consuming application

application that consumes data

3.6.37

control commands

action invocations transferred from client to server to clear outputs, freeze inputs and/or synchronise outputs

3.6.38

conveyance path

unidirectional flow of APDUs across an application relationship

3.6.39

cyclic

repetitive in a regular manner

3.6.40

data consistency

means for coherent transmission and access of the input- or output-data object between and within client and server

3.6.41

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.

3.6.42

device

physical hardware connected to the link

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

3.6.43

device profile

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

3.6.44

diagnosis information

all data available at the server for maintenance purposes

3.6.45

diagnosis information collection

system diagnosis information that is assembled at the client side

3.6.46

dynamic AR

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

3.6.47

end node

producing or consuming node

3.6.48

endpoint

one of the communicating entities involved in a connection

3.6.49

engineering

abstract term that characterizes the client application or device responsible for configuring an automation system via interconnecting data items

3.6.50

error

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

3.6.51

error class

general grouping for related error definitions and corresponding error codes

3.6.52

error code

identification of a specific type of error within an error class

3.6.53

event

an instance of a change of conditions

3.6.54

FAL subnet

subnetworks composed of one or more data link segments, identified by a subset of the network address

Note 1 to entry: FAL subnets are permitted to contain bridges but not routers.

3.6.55

FIFO variable

a Variable Object class, composed of a set of homogeneously typed elements, where the first written element is the first element that can be read

Note 1 to entry: On the fieldbus only one, complete element can be transferred as a result of one service invocation.

3.6.56

frame

denigrated synonym for DLPDU

3.6.57

freeze

function at the DP-slaves for simultaneous data transfer between the input data object and the process

3.6.58

group

a) <general> a general term for a collection of objects

Specific uses:

b) <addressing> when describing an address, an address that identifies more than one entity

3.6.59
interface

- a) shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics as appropriate
- b) collection of FAL class attributes and services that represents a specific view on the FAL class

3.6.60
interface definition language

syntax and semantics of describing service parameters in a formal way

Note 1 to entry: This description is the input for the ORPC model, especially for the ORPC wire protocol.

3.6.61
interface pointer

key attribute that unambiguously addresses an object interface instance

3.6.62
invocation

act of using a service or other resource of an application process

Note 1 to entry: Each invocation represents a separate thread of control that may be described by its context. Once the service completes, or use of the resource is released, the invocation ceases to exist. For service invocations, a service that has been initiated but not yet completed is referred to as an outstanding service invocation.

3.6.63
I/O data

object designated to be transferred cyclically for the purpose of processing

3.6.64
identifier related diagnosis

information dedicated to modules for maintenance purpose

3.6.65
index

address of an object within an application process

3.6.66
instance

the actual physical occurrence of an object within a class that identifies one of many objects within the same object class

EXAMPLE California is an instance of the object class US-state.

Note 1 to entry: The terms object, instance, and object instance are used to refer to a specific instance.

3.6.67
instance attributes

attribute that is unique to an object instance and not shared by the object class

3.6.68
instantiated

object that has been created in a device

3.6.69
logical device

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

3.6.70

manufacturer ID

identification of each product manufacturer by a unique number

3.6.71

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.

3.6.72

member

piece of an attribute that is structured as an element of an array

3.6.73

message router

object within a node that distributes messaging requests to appropriate application objects

3.6.74

method

<object> a synonym for an operational service which is provided by the server ASE and invoked by a client

3.6.75

module

a) <general> hardware or logical component of a physical device

b) <Type 3> addressable unit inside the DP-slave

3.6.76

multipoint connection

connection from one node to many

Note 1 to entry: Multipoint connections allow messages from a single producer to be received by many consumer nodes.

3.6.77

network

a set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways

3.6.78

object

abstract representation of a particular component within a device, usually a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behaviour

3.6.79

object remote procedure call

model for object oriented or component based remote method invocation

3.6.80

object specific service

service unique to the object class which defines it

3.6.81

originator

client responsible for establishing a connection path to the target

3.6.82

peer

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

3.6.83

physical device

automation or other network device

3.6.84

point-to-point connection

connection that exists between exactly two application objects

3.6.85

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.

3.6.86

pre-established AR endpoint

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

3.6.87

process data

object(s) which are already pre-processed and transferred acyclically for the purpose of information or further processing

3.6.88

produce

act of sending data to be received by a consumer

3.6.89

producer

node that is responsible for sending data

3.6.90

property

general term for descriptive information about an object

3.6.91

provider

source of a data connection

3.6.92

providerID

an unambiguous identifier within the scope of the ACCO assigned by the provider to recognize the internal data of a configured interconnection source

3.6.93

publisher

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

Note 1 to entry: A publisher may not be aware of the identity or the number of subscribers and it may publish its APDUs using a dedicated AR.

3.6.94

publishing manager

role of an AR endpoint in which it issues one or more confirmed service request APDUs to a publisher to request the publisher to publish a specified object

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

3.6.95

pull publisher

type of publisher that publishes an object in response to a request received from its pull publishing manager

3.6.96

pull publishing manager

type of publishing manager that requests that a specified object be published in a corresponding response APDU

3.6.97

push publisher

type of publisher that publishes an object in an unconfirmed service request APDU

3.6.98

push publishing manager

type of publishing manager that requests that a specified object be published using an unconfirmed service

3.6.99

pull subscriber

type of subscriber that recognizes received confirmed service response APDUs as published object data

3.6.100

push subscriber

type of subscriber that recognizes received unconfirmed service request APDUs as published object data

3.6.101

quality code

additional status information of a data item

3.6.102

resource

a processing or information capability of a subsystem

3.6.103

route endpoint

object container containing Variable Objects of a variable class

3.6.104

runtime object model

objects that exist in a device together with their interfaces and methods that are accessible

3.6.105

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

3.6.106

service

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

3.6.107

slot

address of a module within a DP-slave

3.6.108

subscriber

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

3.6.109

sync

function at the DP-slaves for simultaneous data transfer between the output data object and the process

3.6.110

target

end-node to which a connection is established

3.6.111

unconnected service

messaging service which does not rely on the set up of a connection between devices before allowing information exchanges

3.7 Abbreviations and symbols

ACCO	Active Connection Control Object
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
CID	Connection ID
CIM	Computer Integrated Manufacturing
CIP	Control and Information Protocol

CM_API	Actual Packet Interval
CM_RPI	Requested Packet Interval
Cnf	Confirmation
COR	Connection originator
CR	Communication Relationship
CREP	Communication Relationship End Point
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
DNS	Domain Name Service
DP	Decentralised Peripherals
FAL	Fieldbus Application Layer
FIFO	First In First Out
HMI	Human-Machine Interface
ID	Identifier
IDL	Interface Definition Language
IEC	International Electrotechnical Commission
Ind	Indication
IP	Internet Protocol
ISO	International Organization for Standardization
LDev	Logical Device
LME	Layer Management Entity
O2T	Originator to target (connection characteristics)
O⇒T	Originator to target (connection characteristics)
ORPC	Object Remote Procedure Call
OSI	Open Systems Interconnect
PDev	Physical Device
PDU	Protocol Data Unit
PL	Physical Layer
QoS	Quality of Service
QC	Quality Code
REP	Route Endpoint
Req	Request
Rsp	Response
RT	Runtime
SAP	Service Access Point
SCL	Security Level
SDU	Service Data Unit

SEM	State event matrix
SMIB	System Management Information Base
SMK	System Management Kernel
STD	State transition diagram, used to describe object behaviour
S-VFD	Simple Virtual Field Device
T2O	Target to originator (connection characteristics)
T⇒O	Target to originator (connection characteristics)
VAO	Variable Object

3.8 Conventions

3.8.1 Overview

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 two parts, its class specification, and its service specification.

The class specification defines the attributes of the class. The attributes are accessible from instances of the class using the Object Management ASE services specified in Clause 5 of this standard. The service specification defines the services that are provided by the ASE.

3.8.2 Conventions for class definitions

Class definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is shown below:

FAL ASE:		ASE Name
CLASS:	Class Name	
CLASS ID:		#
PARENT CLASS:		Parent Class Name
ATTRIBUTES:		
1	(o) Key Attribute:	numeric identifier
2	(o) Key Attribute:	name
3	(m) Attribute:	attribute name(values)
4	(m) Attribute:	attribute name(values)
4.1	(s) Attribute:	attribute name(values)
4.2	(s) Attribute:	attribute name(values)
4.3	(s) Attribute:	attribute name(values)
5.	(c) Constraint:	constraint expression
5.1	(m) Attribute:	attribute name(values)
5.2	(o) Attribute:	attribute name(values)
6	(m) Attribute:	attribute name(values)
6.1	(s) Attribute:	attribute name(values)
6.2	(s) Attribute:	attribute name(values)
SERVICES:		
1	(o) OpsService:	service name
2.	(c) Constraint:	constraint expression
2.1	(o) OpsService:	service name
3	(m) MgtService:	service name

- (1) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class being specified.

- (2) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this standard, or by a user of this standard.
- (3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 255 are reserved by this standard to identify standardized classes. They have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2048 are allocated for identifying user defined classes.
- (4) The "PARENT CLASS:" entry is the name of the parent class for the class being specified. All attributes defined for the parent class and inherited by it are inherited for the class being defined, and therefore do not have to be redefined in the template for this class.

NOTE The parent-class "TOP" indicates that the class being defined is an initial class definition. The parent class TOP is used as a starting point from which all other classes are defined. The use of TOP is reserved for classes defined by this standard.

- (5) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.
 - a) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.
 - b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.
 - c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify
 - i) fields of a structured attribute (4.1, 4.2, 4.3),
 - ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2).
 - iii) the selection fields of a choice type attribute (6.1 and 6.2).
- (6) The "SERVICES" label indicates that the following entries are services defined for the class.
 - a) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.
 - b) The label "OpsService" designates an operational service (1).
 - c) The label "MgtService" designates an management service (2).
 - d) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

3.8.3 Conventions for service definitions

3.8.3.1 General

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

The service model, service primitives, and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

3.8.3.2 Service parameters

Service primitives are used to represent service user/service provider interactions (ISO/IEC 10731). They convey parameters which indicate information available in the user/provider interaction.

NOTE 1 See the Note under 3.8.3.3 relative to the non-inclusion of service parameters that are appropriate to a protocol specification or programming interface specification or implementation specification, but not to an abstract service definition.

This standard uses a tabular format to describe the component parameters of the service primitives. The parameters that apply to each group of service primitives are set out in tables throughout the remainder of this standard. Each table consists of up to six columns: a column for the name of the service parameter, and a column each for those primitives and parameter-transfer directions used by the service. The possible six columns are

- 1) the parameter name;
- 2) the request primitive's input parameters;
- 3) the request primitive's output parameters;

NOTE 2 This is a seldom-used capability. Unless otherwise specified, request primitive parameters are input parameters.

- 4) the indication primitive's output parameters;
- 5) the response primitive's input parameters; and
- 6) the confirm primitive's output parameters.

NOTE 3 The request, indication, response and confirm primitives are also known as requestor.submit, acceptor.deliver, acceptor.submit, and requestor.deliver primitives, respectively (see ISO/IEC 10731).

One parameter (or component of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive specified in the column:

- M parameter is mandatory for the primitive
- U parameter is a User option, and may or may not be provided depending on dynamic usage of the service user. When not provided, a default value for the parameter is assumed.
- C parameter is conditional upon other parameters or upon the environment of the service user.
- (blank) parameter is never present.
- S parameter is a selected item.

Some entries are further qualified by items in brackets. These may be

- a) a parameter-specific constraint:
“(=)” indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.
- b) an indication that some note applies to the entry:
“(n)” indicates that the following note "n" contains additional information pertaining to the parameter and its use.

3.8.3.3 Service procedures

The procedures are defined in terms of

- the interactions between application entities through the exchange of fieldbus Application Protocol Data Units, and
- the interactions between an application layer service provider and an application layer service user in the same system through the invocation of application layer service primitives.

These procedures are applicable to instances of communication between systems which support time-constrained communications services within the fieldbus Application Layer.

NOTE The IEC 61158-5 series of standards define sets of abstract services. They are neither protocol specifications nor implementation specifications nor concrete programming interface specifications. Therefore there are restrictions on the extent to which service procedures can be mandated in the parts of IEC 61158-5. Protocol aspects that can vary among different protocol specifications or different implementations that instantiate the same abstract services are unsuitable for inclusion in these service definitions, except at the level of abstraction that is necessarily common to all such expressions.

For example, the means by which service providers pair request and reply PDUs is appropriate for specification in an IEC 61158-6 protocol specification standard but not in an IEC 61158-5 abstract service definition standard. Similarly, local implementation methods by which a service provider or service user pairs request and confirm(ation) primitives, or indication and response primitives, is appropriate for an implementation specification or for a programming interface specification, but not for an abstract service standard or for a protocol standard, except at a level of abstraction that is necessarily common to all embodiments of the specifying standard. In all cases, the abstract definition is not permitted to over-specify the more concrete instantiating realization.

Further information on the conceptual service procedures of an implementation of a protocol that realizes the services of one of the IEC 61158-5 abstract service definitions can be found in IEC 61158-1, 9.6.

4 Concepts

The common concepts and templates used to describe the application layer service in this standard are detailed in IEC 61158-1, Clause 9.

5 Data type ASE

5.1 Overview

An overview of the data type ASE and the relationships between data types is provided in IEC 61158-1, 10.1.

5.2 Formal definition of data type objects

5.2.1 Data type class

5.2.1.1 Template

The data type class specifies the root of the data type class tree. Its parent class "top" indicates the top of the FAL class tree.

FAL ASE:		DATA TYPE ASE
CLASS:	DATA TYPE	
CLASS ID:		5 (FIXED LENGTH & STRING), 6 (STRUCTURE), 12 (ARRAY)
PARENT CLASS:		TOP
ATTRIBUTES:		
1	(m) Key Attribute:	Data type Numeric Identifier
2	(o) Key Attribute:	Data type Name
3	(m) Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
4	(c) Constraint:	Format = FIXED LENGTH STRING
4.1	(m) Attribute:	Octet Length

5	(c)	Constraint:	Format = STRUCTURE
5.1	(m)	Attribute:	Number of Fields
5.2	(m)	Attribute:	List of Fields
5.2.1	(o)	Attribute:	Field Name
5.2.2	(m)	Attribute:	Field Data type
6	(c)	Constraint:	Format = ARRAY
6.1	(m)	Attribute:	Number of Array Elements
6.2	(m)	Attribute :	Array Element Data type

5.2.1.2 Attributes

Data type Numeric Identifier

This attribute identifies the numeric identifier of the related data type.

Data type Name

This optional attribute identifies the name of the related data type.

Format

This attribute identifies the data type as a fixed-length, string, array, or data structure.

Octet Length

This conditional attribute defines the representation of the dimensions of the associated type object. It is present when the value of the format attribute is "FIXED LENGTH" or "STRING". For FIXED LENGTH data types, it represents the length in octets. For STRING data types, it represents the length in octets for a single element of a string.

Number of Fields

This conditional attribute defines the number of fields in a structure. It is present when the value of the format attribute is "STRUCTURE".

List of Fields

This conditional attribute is an ordered list of fields contained in the structure. Each field is specified by its number and its type. Fields are numbered sequentially from 0 (zero) in the order in which they occur. Partial access to fields within a structure is supported by identifying the field by number. This attribute is present when the value of the format attribute is "STRUCTURE".

Field Name

This conditional, optional attribute specifies the name of the field. It may be present when the value of the format attribute is "STRUCTURE".

Field Data type

This conditional attribute specifies the data type of the field. It is present when the value of the format attribute is "STRUCTURE". This attribute may itself specify a constructed data type either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the Embedded Data type description shown below is used.

Number of Array Elements

This conditional attribute defines the number of elements for the array type. Array elements are indexed starting at "0" through "n-1" where the size of the array is "n" elements. This attribute is present when the value of the format attribute is "ARRAY".

Array Element Data type

This conditional attribute specifies the data type for the elements of an array. All elements of the array have the same data type. It is present when the value of the format attribute is "ARRAY". This attribute may itself specify a constructed data type either by referencing a constructed data type

definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the Embedded Data type description shown below is used.

Embedded Data type Description

This attribute is used to recursively define embedded data types within a structure or array. The template below defines its contents. The attributes shown in the template are defined above in the data type class, except for the Embedded Data type attribute, which is a recursive reference to this attribute. It is used to define nested elements.

ATTRIBUTES:

1	(m)	Attribute:	Format(FIXED LENGTH, STRING, STRUCTURE, ARRAY)
2	(c)	Constraint:	Format = FIXED LENGTH STRING
2.1	(m)	Attribute:	Data type Numeric ID value
2.2	(m)	Attribute:	Octet Length
3	(c)	Constraint:	Format = STRUCTURE
3.1	(m)	Attribute:	Number of Fields
3.2	(m)	Attribute:	List of Fields
3.2.1	(m)	Attribute:	Embedded Data type Description
4	(c)	Constraint:	Format = ARRAY
4.1	(m)	Attribute:	Number of Array Elements
4.2	(m)	Attribute:	Embedded Data type Description

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Boolean types

5.3.1.1.1 Boolean

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	1
2	Data type Name	=	Boolean
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

This data type expresses a Boolean data type with the values TRUE and FALSE.

5.3.1.1.2 BOOL

This IEC 61131-3 type is the same as Boolean.

5.3.1.1.3 VT_BOOLEAN

CLASS: Data type

ATTRIBUTES:

2	Data type Name =	VT_BOOLEAN
4	Format	= FIXED LENGTH
4.1	Octet Length	= 2

This data type expresses a Boolean data type with the values TRUE (-1) and FALSE (0) (see Integer16).

5.3.1.2 Bitstring types

5.3.1.2.1 BitString8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	22
2	Data type Name	=	Bitstring8
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	1

This type contains 1 element of type BitString.

5.3.1.2.2 OCTET

This IEC 61131-3 type is the same as Bitstring8.

5.3.1.2.3 BitString16

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	23
2	Data type Name	=	Bitstring16
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	2

5.3.1.2.4 WORD

This IEC 61131-3 type is the same as Bitstring16.

5.3.1.2.5 BitString32

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	24
2	Data type Name	=	Bitstring32
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	4

5.3.1.2.6 DWORD

This IEC 61131-3 type is the same as Bitstring32.

5.3.1.2.7 BitString64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	57
2	Data type Name	=	Bitstring64
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	8

5.3.1.2.8 LWORD

This IEC 61131-3 type is the same as Bitstring64.

5.3.1.3 Currency types

5.3.1.3.1 currency

CLASS: Data type

ATTRIBUTES:

2	Data type Name =	currency
3	Format	= FIXED LENGTH
4.1	Octet Length	= 8

This data type defines a signed 64-bit integer in units of 1/10,000 (or 1/100 of a cent). A currency number stored as an 8-octet, two's complement integer, scaled by 10,000 to give a fixed-point number with 15 digits to the left of the decimal point and 4 digits to the right. This representation provides a range of $\pm 922337203685477,5807$. This data type is useful for calculations involving money, or for any fixed-point calculation where accuracy is particularly important.

5.3.1.4 Date types

5.3.1.4.1 BinaryDate

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier =	11
2	Data type Name	= BinaryDate
3	Format	= FIXED LENGTH
4.1	Octet Length	= 7

This data type is composed of six elements of unsigned values and expresses calendar date and time. The first element is an Unsigned16 data type and gives the fraction of a minute in milliseconds. The second element is an Unsigned8 data type and gives the fraction of an hour in minutes. The third element is an Unsigned8 data type and gives the fraction of a day in hours. The fourth element is an Unsigned8 data type. Its upper three (3) bits give the day of the week and its lower five (5) bits give the day of the month. The fifth element is an Unsigned8 data type and gives the month. The last element is Unsigned8 data type and gives the year.

5.3.1.4.2 BinaryDate2000

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier =	51
2	Data type Name	= BinaryDate2000
3	Format	= FIXED LENGTH
4.1	Octet Length	= 8

This data type is composed of six elements of unsigned values and expresses calendar date and time. The first element is an Unsigned16 data type and gives the fraction of a minute in milliseconds. The second element is an Unsigned8 data type and gives the fraction of an hour in minutes. The third element is an Unsigned8 data type and gives the fraction of a day in hours. The fourth element is an Unsigned8 data type. Its upper three (3) bits give the day of the week and its lower five (5) bits give the day of the month. The fifth element is an Unsigned8 data type and gives the month. The last element is Unsigned16 data type and gives the year.

5.3.1.4.3 Date

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier =	50
2	Data type Name	= Date
3	Format	= FIXED LENGTH
4.1	Octet Length	= 7

This data type is composed of six elements of unsigned values and expresses calendar date and time. The first element is an Unsigned16 data type and gives the fraction of a minute in milliseconds. The second element is an Unsigned8 data type and gives the fraction of an hour in minutes. The third element is an Unsigned8 data type and gives the fraction of a day in hours with the most significant bit indicating Standard Time or Daylight Saving Time. The fourth element is an Unsigned8 data type. Its upper three (3) bits give the day of the week and its lower five (5) bits give the day of the month. The fifth element is an Unsigned8 data type and gives the month. The last element is Unsigned8 data type and gives the year. The values 0 ... 50 correspond to the years 2000 to 2050, the values 51 ... 99 correspond to the years 1951 to 1999.

5.3.1.4.4 DATE

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
2	Data type Name	=	DATE
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This IEC 61131-3 type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of two octets. It expresses the date as a number of days, starting from 1972.01.01 (January 1st, 1972), the start of the Coordinated Universal Time (UTC) era, until 2151.06.06 (June 6th, 2151), i.e. a total range of 65536 days.

5.3.1.4.5 date

This data type is the same as Float64.

The data type date has a resolution in the range of one nanosecond. It is valid for dates between 1 January 0100 and 31 December 9999. The value 0,0 has been defined for 30 December 1899, 00:00. The integer part of the value represents the days after 30 December 1899 (for dates before this day, the corresponding value is negative); the fractional part defines the time at that day.

5.3.1.4.6 TimeOfDay

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	12
2	Data type Name	=	TimeOfDay
4	Format	=	FIXED LENGTH
4.1	Octet Length	=	6

This data type is composed of two elements of unsigned values and expresses the time of day and the date. The first element is an Unsigned32 data type and gives the time after the midnight in milliseconds. The second element is an Unsigned16 data type and gives the date counting the days from January 1, 1984.

5.3.1.4.7 TimeOfDay with date indication

This data type is the same as the TimeOfDay data type defined above.

5.3.1.4.8 TimeOfDay without date indication

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	52
2	Data type Name	=	TimeOfDay without date indication

4	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This data type is composed of one element of an unsigned value and expresses the time of day. The element is an Unsigned32 data type and gives the time after the midnight in milliseconds.

5.3.1.4.9 TIME_OF_DAY

This IEC 61131-3 type is the same as TimeofDay without date indication.

5.3.1.4.10 TimeDifference

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 13
2	Data type Name = TimeDifference
3	Format = FIXED LENGTH
4.1	Octet Length = 4 or 6

This data type is composed of two elements of unsigned values that express the difference in time. The first element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds. The optional second element is an Unsigned16 data type that provides the difference in days.

5.3.1.4.11 TimeDifference with date indication

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 53
2	Data type Name = TimeDifference with date indication
3	Format = FIXED LENGTH
4.1	Octet Length = 6

This data type is composed of two elements of unsigned values that express the difference in time. The first element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds. The second element is an Unsigned16 data type that provides the difference in days.

5.3.1.4.12 TimeDifference without date indication

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 54
2	Data type Name = TimeDifference without date indication
3	Format = FIXED LENGTH
4.1	Octet Length = 4

This data type is composed of one element of an unsigned value that express the difference in time. The element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds.

5.3.1.4.13 TimeValue

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 21
2	Data type Name = Time Value
3	Format = FIXED LENGTH
4.1	Octet Length = 8

This simple type expresses the time or time difference in a two's complement binary number with a length of eight octets. The unit of time is 1/32 millisecond.

5.3.1.4.14 UniversalTime

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 16
- 2 Data type Name = UniversalTime
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 12

This simple type is composed of twelve elements of type VisibleString. (YYMMDDHHMMSS). It is the same as that defined in ISO/IEC 8824, except that the local time differential is not supported.

5.3.1.4.15 FieldbusTime

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 17
- 2 Data type Name = FieldbusTime
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 7

This data type is defined in this standard as DL-Time.

5.3.1.5 Enumerated types

5.3.1.5.1 PERSISTDEF

CLASS: Data type

ATTRIBUTES:

- 2 Data type Name = PERSISTDEF
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

The allowed values are shown in Table 1.

Table 1 – PERSISTDEF

Value
CBAVolatile
CBAPendingPersistent
CBAPersistent

5.3.1.5.2 VARTYPE

CLASS: Data type

ATTRIBUTES:

- 2 Data type Name = VARTYPE
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

The VARTYPE specifies the data type that governs the interpretation of the data. The allowed values are shown in Table 2.

Table 2 – VARTYPE

Value	Data type
VT_EMPTY	no value
VT_NULL	null value (no valid data)
VT_BOOL	VT_BOOLEAN