

DIN EN 61158-5-3

ICS 35.100.70; 35.160

Ersatz für
DIN EN 61158-5-3:2013-01
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 5-3: Dienstfestlegungen des Application Layer
(Anwendungsschicht) – Typ 3-Elemente
(IEC 61158-5-3:2014);
Englische Fassung EN 61158-5-3:2014, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 5-3: Application layer service definition – Type 3 elements
(IEC 61158-5-3:2014);
English version EN 61158-5-3:2014, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 5-3: Définition des services de la couche application – Éléments de type 3
(CEI 61158-5-3:2014);
Version anglaise EN 61158-5-3:2014, seulement on CD-ROM

Gesamtumfang 453 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 61158-5-3:2013-01 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 Abschnietsnummer, 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 61158-5-3 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 3 fest.

Änderungen

Gegenüber DIN EN 61158-5-3:2013-01 wurden folgende Änderungen vorgenommen:

- a) redaktionelle Überarbeitung von Tabelle A.15 und A.16;
- b) Liste der Patente auf den neuesten Stand gebracht.

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 61158-5-3: 2008-09, 2013-01

Nationaler Anhang NA (informativ)

1 Anwendungsbereich^{N1)}

1.1 Allgemeines

Die Norm ist eine von vielen aus einer Reihe, welche erzeugt wurden, um die Verbindung von Geräten eines Automatisierungssystems zu vereinfachen. Sie ist verwandt mit anderen Normen in der Zusammenstellung, wie vom Drei-Schicht(en: three-layer)-Feldbusreferenzmodell beschrieben und in IEC 61158 festgelegt.

Dieser Teil enthält spezifischen Inhalt zum Feldbus Typ 3.

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

Diese Norm stellt für Typ 3-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 aufgeförderte 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 3 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 2-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 Applikationsprozessobjekt(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.

^{N1)} Die Nummerierung aus IEC 61158-5-3 wurde beibehalten.

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 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 3 der Application Layer-Dienste, wie in dieser Norm beschrieben, erfüllen.

3.2 Spezielle Begriffe für den Feldbus-Application Layer Typ 3

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.2.1

Zugriffsschutz

(en: access protection)

Beschränkung der Nutzung eines Anwendungsobjekts auf einen Client

3.2.2

Adresszuweisungstabelle

(en: address assignment table)

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

3.2.3

zuweisen

(en: allocate)

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

3.2.4

Anwendung

(en: application)

Funktion oder Datenstruktur, für die Daten konsumiert oder produziert werden

3.2.5

Interoperabilität auf Application Layer

(en: application layer interoperability)

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

3.2.6

Anwendungsobjekte

(en: application objects)

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

3.2.7

Anwendungsprozesskennung

(en: application process identifier)

unterscheidet mehrere in einem Gerät verwendete Anwendungsprozesse

3.2.8

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 ihrer Klasse zusammen (siehe Definition von „Klasse von Anwendungsprozessobjekten“). Auf die Definitionen von Anwendungsprozessobjekten kann mithilfe der Dienste des Objektverwaltungs-ASE der FAL von einem entfernten Standort aus zugegriffen werden. Objektverwaltungsdienste des 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.

3.2.9

Anwendungsprozessobjektklasse

(en: application process object class)

Klasse von Anwendungsprozessobjekten, die in Bezug auf den Satz der über das Netzwerk zugänglichen Attribute und Dienste der Anwendungsprozessobjekte definiert ist

3.2.10

Anwendungsbeziehung

(en: application relationship)

kooperative Verbindung zwischen zwei oder mehr Aufrufen von Anwendungsentitäten zum Zweck des Informationsaustausches und der Koordination des gemeinsamen Betriebs.

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

3.2.11

Anwendungsdienstelement für Anwendungsbeziehungen

(en: application relationship application service element)

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

3.2.12

Endpunkt der Anwendungsbeziehung

(en: application relationship endpoint)

Kontext und Verhalten einer Anwendungsbeziehung entsprechend der Sicht und Verwaltung durch einen der Anwendungsprozesse innerhalb der Anwendungsbeziehung

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

3.2.13

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.

3.2.14

Verhalten

(en: behavior)

Anzeige dafür, wie das Objekt auf bestimmte Ereignisse reagiert

3.2.15

Kanal

(en: channel)

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

3.2.16

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.

3.2.17

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.

3.2.18

Klassenattribute

(en: class attributes)

Attribut, welches allen Objekten innerhalb derselben Klasse gemeinsam ist

3.2.19

Klassencode

(en: class code)

eindeutige Kennung, die jeder Objektklasse zugewiesen ist

3.2.20

klassenspezifischer Dienst

(en: class specific service)

durch eine bestimmte Objektklasse definierter Dienst, um eine erforderliche Funktion auszuführen, welche nicht als allgemeiner Dienst ausgeführt wird

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

3.2.21

Client

(en: client)

<Objektsicht> Objekt, das die Dienste eines anderen (Server-)Objekts zur Ausführung einer Aufgabe nutzt

3.2.22

Client

(en: client)

<Kommunikationssicht> Initiator einer Nachricht, auf die ein Server reagiert

3.2.23

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

3.2.24

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

3.2.25

Konfigurationskennung

(en: configuration identifier)

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

3.2.26

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.

Anmerkung 2 zum Begriff: Der logische Verbindungsweig zwischen der Senke und der Quelle von Attributen und Diensten an unterschiedlichen benutzerdefinierten Schnittstellen von RT-Auto-ASEs wird als „Verschaltung“ bezeichnet. Es wird zwischen Datenverschaltung und Ereignisverschaltung unterschieden. Der logische Verbindungsweig und der Datenfluss zwischen der Senke und der Quelle von Automatisierungsdatenelementen werden als Datenverschaltung bezeichnet. Der logische Verbindungsweig und der Datenfluss zwischen der Senke (Methode) und der Quelle (Ereignis) von Betriebsdiensten werden als Ereignisverschaltung bezeichnet.

3.2.27

konsumieren

(en: consume)

Vorgang des Empfangs von Daten von einem Produzenten

3.2.28

Konsument

(en: consumer)

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

3.2.29

konsumierende Anwendung

(en: consuming application)

Anwendung, die Daten konsumiert

3.2.30

zyklisch

(en: cyclic)

sich regelmäßig wiederholend

3.2.31

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

3.2.32

standardmäßige DL-Adresse

(en: default DL-address)

Wert 126 als Anfangswert für DL-Adressen, der vor dem Betrieb mit einem DP-Master (Klasse 1) geändert werden muss (z. B. durch Zuweisung einer DL-Adresse über den Feldbus)

3.2.33

Gerät

(en: device)

an den Verbindungszweig angeschlossene physische Hardware

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

3.2.34

Diagnose-Informationen

(en: diagnosis information)

alle Daten, die serverseitig für Wartungszwecke verfügbar sind

3.2.35

Sammlung von Diagnose-Informationen

(en: diagnosis information collection)

Diagnose-Informationen zum System, die clientseitig zusammengetragen werden

3.2.36

DP-Master (Klasse 1)

(en: DP-master (class 1))

Steuergerät, das mehrere DP-Slaves (Feldgeräte) steuert

Anmerkung 1 zum Begriff: Hierbei handelt es sich üblicherweise um eine speicherprogrammierbare Steuerung oder ein verteiltes Steuerungssystem.

3.2.37

DP-Master (Klasse 2)

(en: DP-master (class 2))

Steuergerät, das Konfigurationsdaten (Parametersätze) und Diagnose-Daten eines DP-Masters (Klasse 1) verwaltet und das zusätzlich alle Kommunikationsfunktionen eines DP-Masters (Klasse 1) ausführen kann

3.2.38

DP-Slave

(en: DP-slave)

Feldgerät, das einem DP-Master (Klasse 1) als Provider für zyklischen E/A-Datenaustausch zugewiesen werden kann; zusätzlich könnten azyklische Funktionen und Alarmer bereitgestellt werden

3.2.39

Endpunkt

(en: endpoint)

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

3.2.40

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

3.2.41

Fehler

(en: error)

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

3.2.42

Fehlerklasse

(en: error class)

allgemeine Gruppierung für zugehörige Fehlerdefinitionen und entsprechende Fehlercodes

3.2.43

Fehlercode

(en: error code)

Kennzeichnung eines bestimmten Fehlertyps innerhalb einer Fehlerklasse

3.2.44

Ereignis

(en: event)

Instanz einer Änderung von Zuständen

3.2.45

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.

3.2.46

Freeze

(en: freeze)

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

3.2.47

Gruppe

(en: group)

<allgemein> eine allgemeine Bezeichnung für eine Zusammenstellung von Objekten. Spezifische Verwendungen

3.2.48

Gruppe

(en: group)

<Adressierung> bei der Beschreibung einer Adresse eine Adresse, die mehr als eine Entität identifiziert

3.2.49

Gruppe

(en: group)

<Typ 3> Satz von DP-Slaves, die eine Freeze- oder Sync-Funktion ausführen

3.2.50

Schnittstelle

(en: interface)

<allgemein> gemeinsam genutzte Grenze zwischen zwei funktionalen Einheiten, die durch funktionale Merkmale, Signalmerkmale oder entsprechend andere Merkmale festgelegt ist

3.2.51

Schnittstelle

(en: interface)

<FAL> Zusammenstellung von FAL-Klassenattributen und -diensten, die eine spezifische Sicht auf die FAL-Klasse darstellt

3.2.52

Schnittstellenzeiger

(en: interface pointer)

Schlüsselattribut, das eine Objektschnittstelleninstanz eindeutig adressiert

3.2.53

Aufruf

(en: invocation)

Vorgang der Nutzung eines Dienstes oder einer anderen Ressource eines Anwendungsprozesses

Anmerkung 1 zum Begriff: Jeder Aufruf bildet einen einzelnen Steuerungsstrang der durch seinen Kontext beschrieben werden kann. Sobald der Dienst ausgeführt oder die Nutzung der Ressource beendet ist, erlischt der Aufruf. Bei Dienstaufrufen wird ein Dienst, der initiiert, aber noch nicht ausgeführt wurde, als ausstehender Dienstaufruf bezeichnet. Bei Dienstaufrufen kann außerdem eine Aufruf-ID verwendet werden, um den Dienstaufruf eindeutig zu identifizieren und ihn von anderen ausstehenden Dienstaufrufen zu unterscheiden.

3.2.54

E/A-Daten

(en: I/O data)

Objekt, das zu Verarbeitungszwecken zyklisch übertragen wird

3.2.55

Identnummer

(en: ident number)

Gerätetyp eines DP-Masters (Klasse 1) oder eines DP-Slaves

3.2.56

kennungsbezogene Diagnose

(en: identifier related diagnosis)

Informationen bezogen auf Module für Wartungszwecke

3.2.57

Index

(en: index)

Adresse eines Objekts innerhalb eines Anwendungsprozesses

3.2.58

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 „Staat“.

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

3.2.59

instanziiert

(en: instantiated)

Objekt, das in einem Gerät erzeugt wurde

3.2.60

Hersteller-ID

(en: manufacturer ID)

Kennzeichnung jedes Produktherstellers durch eine eindeutige Nummer

3.2.61

Master-Parametersatz

(en: master parameter set)

Konfigurations- und Parametrisierungsdaten aller DP-Slaves, die dem entsprechenden DP-Master zugewiesen sind, sowie die Bus-Parameter

3.2.62

Member

(en: member)

Teil eines Attributs, das als Element eines Feldes strukturiert ist

3.2.63

Nachrichtenrouter

(en: message router)

Objekt innerhalb eines Knotens, das Nachrichten anforderungen an entsprechende Anwendungsobjekte verteilt

3.2.64

Methode

(en: method)

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

3.2.65

Modul

(en: module)

<allgemein> Hardware oder logische Komponente eines physischen Gerätes

3.2.66

Modul

(en: module)

<Typ 3> adressierbare Einheit innerhalb des DP-Slaves

3.2.67

Netzwerk

(en: network)

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

3.2.68

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

3.2.69

physisches Gerät

(en: physical device)

<allgemein> Automatisierungsgerät oder anderes Netzwerkgerät

3.2.70

Prozessdaten

(en: process data)

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

3.2.71

produzieren

(en: produce)

Vorgang des Sendens von Daten zum Empfang durch einen Konsumenten

3.2.72

Produzent

(en: producer)

Knoten, der für das Senden von Daten verantwortlich ist

3.2.73

Provider

(en: provider)

Quelle einer Datenverbindung

3.2.74

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.

3.2.75

tatsächliche Konfiguration

(en: real configuration)

Eingabe- und Ausgabedatenstruktur des DP-Slaves, einschließlich der Festlegung von Datenkonsistenz

3.2.76

Ressource

(en: resource)

Verarbeitungs- oder Informationsfähigkeit eines Subsystems

3.2.77

Server

(en: server)

<Kommunikationssicht> Rolle eines AREP, in der dieser eine bestätigte Dienstantwort-APDU an den Client, der die Anforderung initiiert hat, zurückgibt

3.2.78

Server

(en: server)

<Objektsicht> Objekt, das einem anderen (Client-)Objekt Dienste bereitstellt

3.2.79

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

3.2.80

Slot

(en: slot)

Adresse eines Moduls innerhalb eines DP-Slaves

3.2.81

Subscriber

(en: subscriber)

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

3.2.82

Sync

(en: sync)

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

3.2.83

Ziel

(en: target)

Endknoten, mit dem eine Verbindung aufgebaut wird

3.2.84

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 5-3: Application layer service definition - Type 3 elements
(IEC 61158-5-3:2014)**

Réseaux de communication industriels - Spécifications des
bus de terrain - Partie 5-3: Définition des services de la
couche application - Éléments de type 3
(CEI 61158-5-3:2014)

Industrielle Kommunikationsnetze - Feldbusse -
Teil 5-3: Dienstfestlegungen des Application Layer
(Anwendungsschicht) - Typ 3-Elemente
(IEC 61158-5-3:2014)

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 3 of IEC 61158-5-3, 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 61158-5-3:2014.

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 61158-5-3:2012.

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 61158-5-3:2014 was approved by CENELEC as a European Standard without any modification.

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

IEC 61158-6-10:2014	NOTE	Harmonized as EN 61158-6-10 ¹⁾ (not modified).
IEC 61784-1:2014	NOTE	Harmonized as EN 61784-1:2014 (not modified).
IEC 61784-2:2014	NOTE	Harmonized as EN 61784-2 ¹⁾ (not modified).

¹⁾ To be published.

CONTENTS

INTRODUCTION.....	12
1 Scope	13
1.1 General.....	13
1.2 Specifications	14
1.3 Conformance.....	14
2 Normative references	14
3 Terms, definitions, abbreviations, symbols and conventions	15
3.1 Referenced terms and definitions	15
3.2 Fieldbus Application Layer Type 3 – specific terms and definitions.....	17
3.3 Abbreviations and symbols	24
3.4 Conventions	25
4 Concepts.....	30
5 Data type ASE.....	30
6 Communication model specification	31
6.1 DP concepts	31
6.2 ASEs.....	50
6.3 Summary of FAL classes	429
6.4 Permitted FAL services by AREP role	430
6.5 Conformance classes	434
6.6 Application characteristics	435
Bibliography	437
Annex ZA (normative) Normative references to international publications with their corresponding European publications	438
Figure 1 – Example of DP communication with a single controlling device.....	33
Figure 2 – Example of DP communication with several controlling devices	33
Figure 3 – Example of DP communication between field devices.....	34
Figure 4 – DP-slave model (modular DP-slave)	36
Figure 5 – DP-slave model (compact DP-slave).....	37
Figure 6 – Overview of application processes	38
Figure 7 – DP-slave model (modular DP-slave)	39
Figure 8 – Application Service Elements (ASEs).....	41
Figure 9 – Application Process with application Objects (APOs).....	42
Figure 10 – Access to a remote APO.....	43
Figure 11 – Access to a remote APO for publisher/subscriber association	44
Figure 12 – Example of one AR with two AREPs.....	45
Figure 13 – Relation of a simple process data object to the real object.....	51
Figure 14 – Relation of a combined process data object to the real objects	53
Figure 15 – Sequence of an isochronous DP cycle with one DP-master (class 1).....	84
Figure 16 – Additional time relationships in a DP system operating in isochronous mode.....	85
Figure 17 – DP system with optimized isochronous DP cycle	87
Figure 18 – Buffered synchronized isochronous mode at the DP-master (class 1).....	88

Figure 19 – Enhanced synchronized isochronous mode at the DP-master (class 1)	89
Figure 20 – Input, output and PLL state machine interaction	90
Figure 21 – PLL state diagram	95
Figure 22 – OUTPUT state diagram	99
Figure 23 – INPUT state diagram	103
Figure 24 – Treatment of an alarm in the DP system	136
Figure 25 – Load Region state diagram for erasable memory.....	236
Figure 26 – Load region state diagram for non-erasable memory	237
Figure 27 – Function invocation state diagram.....	269
Figure 28 – System architecture.....	283
Figure 29 – Assignment of communication relationship to application relationship	290
Figure 30 – MS0 application relationship	296
Figure 31 – Output buffer model of a DP-slave without sync functionality	297
Figure 32 – Output buffer model of a DP-slave with sync functionality.....	297
Figure 33 – Input buffer model of a DP-slave without freeze functionality	298
Figure 34 – Input buffer model of a DP-slave with freeze functionality	298
Figure 35 – MS1 application relationship	299
Figure 36 – MS2 application relationship	299
Figure 37 – Example of inter-network communication	300
Figure 38 – Example without inter-network addressing	301
Figure 39 – First example with inter-network addressing	301
Figure 40 – Second example with inter-network addressing	302
Figure 41 – MS3 application relationship	304
Figure 42 – MM1 application relationship	304
Figure 43 – MM2 application relationship	305
Figure 44 – Cycle time of the DP system.....	436
Table 1 – Requirements and features of fieldbus DP	32
Table 2 – Status values of the service primitives	49
Table 3 – Access Rights MS1	52
Table 4 – Access Rights MS2	52
Table 5 – Access Rights MS1	55
Table 6 – Access Rights MS2	55
Table 7 – SCL matching rules	56
Table 8 – Read.....	56
Table 9 – Write.....	58
Table 10 – Data transport	59
Table 11 – Format (simple input data description)	63
Table 12 – Consistency (simple input data description).....	63
Table 13 – Format (simple output data).....	65
Table 14 – Consistency (simple output data)	65
Table 15 – Format (extended input data).....	66

Table 16 – Consistency (extended input data)	67
Table 17 – Format (extended output data)	68
Table 18 – Consistency (extended output data)	68
Table 19 – Set Input	69
Table 20 – Read Input	70
Table 21 – Get Input	71
Table 22 – New Input	72
Table 23 – Set Output	73
Table 24 – Final	74
Table 25 – Read Output	75
Table 26 – Get Output	76
Table 27 – Clear Flag	76
Table 28 – New Flag	76
Table 29 – New Output	77
Table 30 – Clear Flag	77
Table 31 – Global Control	78
Table 32 – Clear Command	78
Table 33 – Sync Command	78
Table 34 – Freeze Command	79
Table 35 – New publisher data	79
Table 36 – Get publisher data	80
Table 37 – New Flag	80
Table 38 – SYNCH	81
Table 39 – SYNCH Delayed	81
Table 40 – DX Finished	82
Table 41 – SYNCH Event	82
Table 42 – Status	82
Table 43 – Primitives issued by the AL to the PLL state machine	91
Table 44 – Primitives issued by the user to the PLL state machine	91
Table 45 – Allowed values of Status	92
Table 46 – Primitives issued by the user to the input state machine	92
Table 47 – Primitives issued by the user to the output state machine	92
Table 48 – Primitives issued by the PLL to the output state machine	92
Table 49 – Primitives issued by the output to the PLL state machine	92
Table 50 – Primitives issued by the PLL to the input state machine	93
Table 51 – Primitives issued by the output to the input state machine	93
Table 52 – Primitives issued by the output state machine to the AL	93
Table 53 – Primitives issued by the AL to the output state machine	93
Table 54 – Primitives issued by the input state machine to the AL	94
Table 55 – Primitives issued by the AL to the input state machine	94
Table 56 – PLL state table	96
Table 57 – OUTPUT state table	100

Table 58 – INPUT state table	104
Table 59 – Identifier status	106
Table 60 – Channel type	107
Table 61 – IO type	108
Table 62 – Status type	108
Table 63 – Status specifier	109
Table 64 – Status specifier	110
Table 65 – Module status	110
Table 66 – Status specifier	111
Table 67 – Link status	112
Table 68 – Link error	112
Table 69 – Set Slave Diag	113
Table 70 – Ext Diag Flag	114
Table 71 – Get Slave Diag	116
Table 72 – Read Slave Diag	126
Table 73 – New Slave Diag	135
Table 74 – Alarm type	137
Table 75 – Add Ack	138
Table 76 – Alarm specifier	138
Table 77 – Alarm notification	139
Table 78 – Alarm Ack	140
Table 79 – Prm data type	145
Table 80 – Supported feature	155
Table 81 – Supported profile feature	155
Table 82 – Role	156
Table 83 – Check user Prm	158
Table 84 – Prm structure	159
Table 85 – MS1 Command	162
Table 86 – Check user Prm result	163
Table 87 – Status values	164
Table 88 – Check Ext user Prm	165
Table 89 – Check Ext user Prm result	168
Table 90 – Status values	169
Table 91 – Check Cfg	169
Table 92 – Check Cfg result	170
Table 93 – Status values	171
Table 94 – Set Cfg	171
Table 95 – Get Cfg	172
Table 96 – Set Slave Add	173
Table 97 – Initiate	174
Table 98 – Abort	177
Table 99 – Instance	177

Table 100 – MS0 init DP-slave	178
Table 101 – MS1 init DP-slave	178
Table 102 – MS2 init DP-slave	179
Table 103 – DP-slave started	179
Table 104 – Alarm limit	180
Table 105 – DP-slave stopped	180
Table 106 – Reset DP-slave.....	181
Table 107 – DP-slave fault.....	181
Table 108 – Application ready DP-slave	181
Table 109 – Start subscriber	182
Table 110 – Stop subscriber	182
Table 111 – Publisher active	183
Table 112 – Status.....	184
Table 113 – Init DP-master CI1	184
Table 114 – DP-master CI1 started	185
Table 115 – Alarm limit	186
Table 116 – DP-master CI1 stopped.....	186
Table 117 – Reset DP-master CI1	186
Table 118 – DP-master CI1 fault	187
Table 119 – DP-master CI1 reject.....	187
Table 120 – Set mode DP-master CI1	188
Table 121 – DP-master CI1 mode changed.....	189
Table 122 – Load bus Par DP-master CI1.....	190
Table 123 – Mark DP-master CI1	191
Table 124 – Abort DP-master CI1	191
Table 125 – Read value DP-master CI1	192
Table 126 – Delete SC DP-master CI1	192
Table 127 – DP-master CI1 event.....	193
Table 128 – Init DP-master CI2	194
Table 129 – Reset DP-master CI2	195
Table 130 – DP-master CI2 fault	195
Table 131 – DP-master CI2 reject.....	195
Table 132 – DP-master CI2 closed	196
Table 133 – DP-master CI2 event.....	196
Table 134 – USIF state	198
Table 135 – Data rate	202
Table 136 – USIF state	202
Table 137 – Isochronous mode	203
Table 138 – Slave type	206
Table 139 – Alarm mode	207
Table 140 – Get Master Diag	210
Table 141 – MDiag identifier	210

Table 142 – Start Seq.....	211
Table 143 – Area code (start seq)	212
Table 144 – Download	213
Table 145 – Upload	214
Table 146 – End Seq	215
Table 147 – Act Para Brct	216
Table 148 – Area code (Act Para Brct)	216
Table 149 – Act param.....	217
Table 150 – Area code (Act param).....	218
Table 151 – Activate	218
Table 152 – Access rights MS1	220
Table 153 – Access rights MS2	221
Table 154 – Load region state.....	221
Table 155 – Initiate load	223
Table 156 – Default values for the parameter Intersegment Request Timeout	224
Table 157 – Push segment	225
Table 158 – Pull segment	227
Table 159 – Terminate load	229
Table 160 – Primitives issued by the user to the Load Region state machine	231
Table 161 – Primitives issued by the Load Region state machine to the user	232
Table 162 – Primitives issued by the Function Invocation to the Load Region state machine.....	232
Table 163 – Primitives issued by the Load Region to the Function Invocation state machine.....	233
Table 164 – Load Region state definitions.....	233
Table 165 – Load Region function table	234
Table 166 – Load Region state table for erasable memory.....	237
Table 167 – Load Region state table for non-erasable memory	250
Table 168 – Access rights MS1	254
Table 169 – Access rights MS2	254
Table 170 – Function Invocation state	255
Table 171 – Load Region object in use.....	255
Table 172 – Access rights MS1	257
Table 173 – Access rights MS2	257
Table 174 – Load Region object in use.....	257
Table 175 – Start.....	258
Table 176 – Stop	259
Table 177 – Resume.....	260
Table 178 – Reset	261
Table 179 – Get FI state	262
Table 180 – Call	263
Table 181 – Primitives issued by the user to the Function Invocation state machine.....	265
Table 182 – Primitives issued by the Function Invocation state machine to the user.....	266
Table 183 – Primitives issued by the Load Region to the Function Invocation state machine.....	266

Table 184 – Primitives issued by the Function Invocation to the Load Region state machine	267
Table 185 – Function Invocation state definitions	267
Table 186 – Function definitions	268
Table 187 – Function Invocation state table	269
Table 188 – CS status	285
Table 189 – Summertime	285
Table 190 – Synchronization active	286
Table 191 – Announcement hour	286
Table 192 – Summertime	287
Table 193 – Accuracy	287
Table 194 – Set time	288
Table 195 – Sync interval violation	289
Table 196 – Parameter of Initiate service without inter-network addressing	301
Table 197 – Parameter of Initiate service with inter-network addressing (first example)	302
Table 198 – Parameter of Initiate service with inter-network addressing (second example)	303
Table 199 – AR type	309
Table 200 – Sync supported	310
Table 201 – Freeze supported	311
Table 202 – Group identifier	313
Table 203 – DPV1 enabled	313
Table 204 – Fail safe	314
Table 205 – WD Base	314
Table 206 – No Add change	316
Table 207 – Alarm mode supported	320
Table 208 – Isochronous mode supp	324
Table 209 – Isochronous mode	324
Table 210 – Alarm mode	325
Table 211 – Time device type	326
Table 212 – S_SAP_index	329
Table 213 – D_addr	330
Table 214 – Service_activate	331
Table 215 – Role_in_service	332
Table 216 – Indication_mode	332
Table 217 – Max_DLSDU_length_req_low	333
Table 218 – Max_DLSDU_length_req_high	333
Table 219 – Max_DLSDU_length_ind_low	334
Table 220 – Max_DLSDU_length_ind_high	335
Table 221 – S_SAP_index	340
Table 222 – D_SAP_index	340
Table 223 – D_addr	341
Table 224 – Service_activate	341
Table 225 – Role_in_service	342

Table 226 – Indication_mode	342
Table 227 – Max_DLSDU_length_req_low.....	343
Table 228 – Max_DLSDU_length_req_high	343
Table 229 – Max_DLSDU_length_ind_low	344
Table 230 – Max_DLSDU_length_ind_high.....	344
Table 231 – Sync.....	345
Table 232 – Freeze	346
Table 233 – DPV1 enabled	347
Table 234 – Fail safe	347
Table 235 – Enable publisher.....	348
Table 236 – WD Base	348
Table 237 – Alarm mode.....	349
Table 238 – Fail safe	359
Table 239 – S_SAP_index	366
Table 240 – D_SAP_index	367
Table 241 – D_addr	367
Table 242 – Service_activate	367
Table 243 – Role_in_service.....	368
Table 244 – Max_DLSDU_length_req_low.....	368
Table 245 – Max_DLSDU_length_req_high	368
Table 246 – Max_DLSDU_length_ind_low	369
Table 247 – Max_DLSDU_length_ind_high.....	369
Table 248 – DLL init DP-slave.....	370
Table 249 – Load ARL DP-slave	371
Table 250 – Get ARL DP-slave	377
Table 251 – Set ARL isochronous mode.....	383
Table 252 – Load ARL DP-master CI1	384
Table 253 – Get ARL DP-master CI1	387
Table 254 – ARL Slave update DP-master CI1	389
Table 255 – Load ARL DP-master CI2.....	391
Table 256 – Get ARL DP-master CI2.....	392
Table 257 – Load CRL DP-slave	393
Table 258 – Load CRL DXB link entries	395
Table 259 – Get CRL DP-slave	396
Table 260 – Load CRL DP-master CI1.....	398
Table 261 – Get CRL DP-master CI1.....	410
Table 262 – CRL Slave activate	422
Table 263 – CRL Slave new Prm.....	423
Table 264 – CRL Slave new Prm data	424
Table 265 – Load CRL DP-master CI2.....	426
Table 266 – Get CRL DP-master CI2.....	428
Table 267 – Fieldbus AL class summary	429

Table 268 – Assignment of the services to DP-masters and DP-slaves.....	431
Table 269 – Support of AR types in the different DP-device types	432
Table 270 – Support of services at the different AREPs respectively CREPs	433
Table 271 – Conformance classes DP-master (class 1)	435
Table 272 – Conformance classes DP-master (class 2)	435

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 61158-1:2013.

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 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 5-3: Application layer service definition – Type 3 elements

1 Scope

1.1 General

This standard 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.

This sub-part contains material specific to Type 3 fieldbus.

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 3 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 different Types of 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

- a) the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model; and
- b) 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 IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

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

One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

1.2 Specifications

The principal objective of this standard is to specify the 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, and the corresponding protocols standardized in subparts of IEC 61158-6.

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 do they 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 any given Type of application layer services as defined in this part of IEC 61158.

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 61131-1, *Programmable controllers – Part 1: General information*

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

IEC 61158-3-3:2014, *Industrial communication networks – Fieldbus specifications – Part 3-3: Data-link layer service definition – Type 3 elements*

IEC 61158-4-3:2014, *Industrial communication networks – Fieldbus specifications – Part 4-3: Data-link layer protocol specification – Type 3 elements*

IEC 61158-5-10:2014, *Industrial communication networks – Fieldbus specifications – Part 5-10: Application layer service definition – Type 10 elements*

IEC 61158-6-3:2014, *Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements*

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

ISO/IEC 7498-3, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing*

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

ISO/IEC 8824-1, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

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*

3 Terms, definitions, abbreviations, symbols and conventions

For the purposes of this document, the following terms, definitions, abbreviations, symbols and conventions as defined in ISO/IEC 7498-1 apply:

3.1 Referenced terms and definitions

3.1.1 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.

3.1.2 ISO/IEC 8822 terms

For the purposes of this document, the following terms as defined in ISO/IEC 8822 apply:

- a) abstract syntax
- b) presentation context.

3.1.3 ISO/IEC 9545 terms

For the purposes of this document, the following terms as defined in ISO/IEC 9545 apply:

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

3.1.4 ISO/IEC 8824-1 terms

For the purposes of this document, the following terms as defined in ISO/IEC 8824 apply:

- a) object identifier
- b) type

3.1.5 Fieldbus Data Link Layer terms

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

3.2 Fieldbus Application Layer Type 3 – specific terms and definitions

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

3.2.1

access protection

limitation of the usage of an application object to one client

3.2.2

address assignment table

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

3.2.3

allocate

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

3.2.4

application

function or data structure for which data is consumed or produced

3.2.5

application layer interoperability

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

3.2.6

application objects

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

3.2.7

application process identifier

distinguishes multiple application processes used in a device

3.2.8

application process object

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

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

3.2.9

application process object class

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

3.2.10

application relationship

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

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

3.2.11

application relationship application service element

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

3.2.12

application relationship endpoint

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

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

3.2.13

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 behavior of an object. Attributes are divided into class attributes and instance attributes.

3.2.14

behavior

indication of how an object responds to particular events

3.2.15

channel

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

3.2.16

channel related diagnosis

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

EXAMPLE: validity of data

3.2.17

class

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

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

3.2.18

class attributes

attribute that is shared by all objects within the same class

3.2.19

class code

unique identifier assigned to each object class

3.2.20

class specific service

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

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

3.2.21

client

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

3.2.22

client

<communication view> initiator of a message to which a server reacts

3.2.23

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

3.2.24

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

3.2.25

configuration identifier

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

3.2.26

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.

Note 2 to entry: The logical link between sink and source of attributes and services at different custom interfaces of RT-Auto ASES is referred to as interconnection. There is a distinction between data and event interconnections. The logical link and the data flow between sink and source of automation data items is referred to as data interconnection. The logical link and the data flow between sink (method) and source (event) of operational services is referred to as event interconnection.

3.2.27

consume

act of receiving data from a producer

3.2.28

consumer

node or sink that is receiving data from a producer

3.2.29

consuming application

application that consumes data

3.2.30

cyclic

repetitive in a regular manner

3.2.31

data consistency

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

3.2.32

default DL-address

value 126 as an initial value for DL-address, which has to be changed (e.g. by assignment of an DL-address via the fieldbus) before operation with a DP-master (class 1)

3.2.33

device

physical hardware connected to the link

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

3.2.34

diagnosis information

all data available at the server for maintenance purposes

3.2.35

diagnosis information collection

system diagnosis information that is assembled at the client side

3.2.36

DP-master (class 1)

a controlling device which controls several DP-slaves (field devices)

Note 1 to entry: This is usually a programmable controller or a distributed control system.

3.2.37

DP-master (class 2)

controlling device which manages configuration data (parameter sets) and diagnosis data of a DP-master (Class 1), and that additionally can perform all communication capabilities of a DP-master (Class 1)

3.2.38

DP-slave

field device that can be assigned to one DP-master (Class 1) as a provider for cyclic I/O data exchange; in addition acyclic functions and alarms could be provided

3.2.39

endpoint

one of the communicating entities involved in a connection

3.2.40

engineering

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

3.2.41

error

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

3.2.42

error class

general grouping for related error definitions and corresponding error codes

3.2.43

error code

identification of a specific type of error within an error class

3.2.44

event

an instance of a change of conditions

3.2.45

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.

3.2.46

freeze

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

3.2.47

group

<general> a general term for a collection of objects. Specific uses:

3.2.48

group

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

3.2.49

group

<Type 3> set of DP-slaves which perform a Freeze or Sync function

3.2.50

interface

<general> shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics as appropriate

3.2.51

interface

<FAL> collection of FAL class attributes and services that represents a specific view on the FAL class

3.2.52

interface pointer

key attribute that unambiguously addresses an object interface instance

3.2.53

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. Also for service invocations, an Invoke ID may be used to unambiguously identify the service invocation and differentiate it from other outstanding service invocations.

3.2.54

I/O data

object designated to be transferred cyclically for the purpose of processing

3.2.55

ident number

DP-master (Class 1) or DP-slave device type

3.2.56

identifier related diagnosis

information dedicated to modules for maintenance purpose

3.2.57

index

address of an object within an application process

3.2.58

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 state.

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

3.2.59

instantiated

object that has been created in a device

3.2.60

manufacturer ID

identification of each product manufacturer by a unique number

3.2.61

master parameter set

the configuration and parameterization data of all DP-slaves that are assigned to the corresponding DP-master and the bus parameters

3.2.62

member

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

3.2.63

message router

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

3.2.64

method

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

3.2.65

module

<general> hardware or logical component of a physical device

3.2.66

module

<Type 3> addressable unit inside the DP-slave

3.2.67

network

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

3.2.68

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 behavior

3.2.69

physical device

<general> an automation or other network device

3.2.70

process data

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

3.2.71

produce

act of sending data to be received by a consumer

3.2.72

producer

node that is responsible for sending data

3.2.73

provider

source of a data connection

3.2.74

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.

3.2.75

real configuration

input and output data structure of the DP-slave, including definition of data consistency

3.2.76

resource

a processing or information capability of a subsystem

3.2.77

server

< communication view> role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request

3.2.78

server

< object view> object which provides services to another (client) object

3.2.79

service

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

3.2.80

slot

address of a module within a DP-slave

3.2.81

subscriber

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

3.2.82

sync

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

3.2.83

target

end-node to which a connection is established

3.2.84

unconnected service

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

3.3 Abbreviations and symbols

AE	Application Entity
AL	Application Layer
ALME	Application Layer Management Entity
ALP	Application Layer Protocol
APO	Application Object
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Identifier
AR	Application Relationship
AREP	Application Relationship End Point
ASE	Application Service Element
Cnf	Confirmation
CR	Communication Relationship
CREP	Communication Relationship End Point
DL-	(as a prefix) data-link-
DLL	Data-link layer
DLM	Data-link-management
DLSAP	Data-link Service Access Point
DLSDU	DL-service-data-unit
DP	Decentralised Peripherals
FAL	Fieldbus Application Layer
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
ISO	International Organization for Standardization
LME	Layer Management Entity

OSI	Open Systems Interconnect
PDev	Physical Device
PDU	Protocol Data Unit
PL	Physical Layer
QoS	Quality of Service
Req	Request
Rsp	Response
SAP	Service Access Point

3.4 Conventions

3.4.1 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.

3.4.2 General conventions

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

3.4.3 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 2 048 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 a 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.

3.4.4 Conventions for service definitions

3.4.4.1 General

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

3.4.4.2 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. In any particular interface, not all parameters need be explicitly stated.

The service specifications of this standard use a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the

- a) Parameter name,
- b) request primitive,
- c) indication primitive,
- d) response primitive, and
- e) confirm primitive.

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.

3.4.4.3 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.

3.4.5 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.

3.4.6 General conventions

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

3.4.7 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 2 048 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 (see 5.1) or optional (see 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 (see 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 a 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.

3.4.8 Conventions for service definitions

3.4.8.1 General

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

3.4.8.2 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. In any particular interface, not all parameters need be explicitly stated.

The service specifications of this standard use a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the

- 1) Parameter name,
- 2) request primitive,
- 3) indication primitive,
- 4) response primitive, and
- 5) confirm primitive.

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.

3.4.8.3 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.

4 Concepts

IEC 61158-1, Clause 9 describes the concepts of the application layer service descriptions and the templates used in this standard.

5 Data type ASE

Fieldbus data types specify the machine independent syntax for application data conveyed by FAL services. The fieldbus application layer supports the definition and transfer of both basic and constructed data types.

IEC 61158-5-10, Clause 5 provides a collection of all data types required by FAL services of Type 3 and Type 10 as well as data types allowed to be used by application profiles.

Encoding rules for the FAL data types used by Type 3 are provided in IEC 61158-6-3.

6 Communication model specification

6.1 DP concepts

6.1.1 User requirements of the fieldbus DP system

The typical automation system consists of one or several programmable controllers connected by I/O-systems to the machine/process. The I/O-system is mostly implemented as a set of single point connections with a well-defined electrical interface such as 24 V digital signals or 4 mA to 20 mA analogue signals. The overall structure of the programmable controller is defined in IEC 61131-1 and widely used.

To switch from I/O-systems plugged in a programmable controller to decentralized I/O-systems a serial link is necessary not only for the exchange of process data, but also to transmit diagnosis, configuration data and parameters.

Furthermore many applications in factory automation require very short reaction times.

Therefore the user interface in the programmable controller for the exchange of process data should be designed in a way to allow direct access to these variables without specific function calls.

In addition a reliable data transfer is required for industrial applications.

The integration of simple field devices requires a simple protocol architecture to allow low cost implementations.

To support the integration of smart field devices acyclic communication services and alarms are needed.

In order to achieve an automation system consisting of devices from different manufacturers strict interoperability of all components is necessary.

In accordance with the IEC 61131-1 structure the decentralized structure should be invisible to an existing control program.

For efficient communication between DP-slaves, as required in motion control applications, the fieldbus provides a Publisher/Subscriber communication mechanism.

The fieldbus provides a mechanism for the synchronization of the application of the DP-master (Class 1) and the DP-slaves with the communication cycle for highly synchronized real time applications.

For time stamping and other actions with precise timing requirements a mechanism to synchronise clocks is offered.

Field devices requires a mechanism to upload (Pull) and/or to download (Push) unstructured LR Data and a mechanism to invoke predefined functions within the DP-slave.

Fieldbus DP provides mechanism to build redundant DP-slaves, which may operate in control systems with several master implementations.

6.1.2 Features of fieldbus DP

Combining the structure defined in IEC 61131-1 with fieldbus DP the following advantages are achieved.

- Changes in the plant cause only small changes in wiring.
- Critical signals can be transferred over short distances to the remote field device (sensor/actuator) and to/from the programmable controller over long distances via fieldbus DP with high accuracy.
- Parameterization and diagnosis can be done without additional wiring.
- Commissioning of subparts without additional installations.

Table 1 gives an overview of the requirements and features of a fieldbus DP system.

Table 1 – Requirements and features of fieldbus DP

Requirement	Feature of fieldbus DP
Short reaction time	Exchange of more than 1 000 Inputs and Outputs with 32 field devices in less than 10 ms
Automation system consists of one or several programmable controller	Mono-master or Multi-master Operation
Simple field devices	Simple protocol, low cost communication interface
Reliable data transfer	Hamming Distance 4 guaranteed; Error recovery mechanism; Use of proven transmission technology
Excellent diagnosis	Various diagnosis in Masters as well as in Slaves
User interface with direct access to the input and output data	Buffer interface for the input and output data
Smart field devices	Acyclic communication which provides a flexible and enhanced addressing scheme of data within field devices. The possibility to transfer an alarm from the field device to the programmable controller with an explicit acknowledgement
Interoperability	Precise and complete definitions including the definition of the system behavior
Efficient communication between DP-slaves for motion control applications	The Publisher/Subscriber communication mechanism allows direct communication between DP-slaves without interactions from the User of the DP-master (Class 1)
Synchronization of the application of the DP-master (Class 1) and the DP-slaves	The Isochronous Mode provides a mechanism for the synchronization of the application of the DP-master (Class 1) and the DP-slaves with the communication cycle. The cycle jitter is less than 1 µs for data rates above 1,5 Mbit/s
Mechanism to load unstructured data blocks	To load unstructured data blocks fieldbus DP provides the Load Region mechanism
Invocation of predefined functions in the DP-slave	With the Function Invocation model predefined functions within the DP-slave may be invoked
Synchronization of clocks	With the Time Synchronization an uniform Network Time (UTC) with an accuracy of less than 1 ms can be achieved

6.1.3 DP communication model

6.1.3.1 Associations

The fieldbus communication model supports the communication of field devices with a single controlling device (e.g. programmable controller or distributed control system) via associations (see

Figure 1). Furthermore, the communication between several controlling devices and their assigned field devices is also supported (see Figure 2). In addition to these basic communication models, fieldbus supports also the cyclic communication between field devices (see Figure 3).

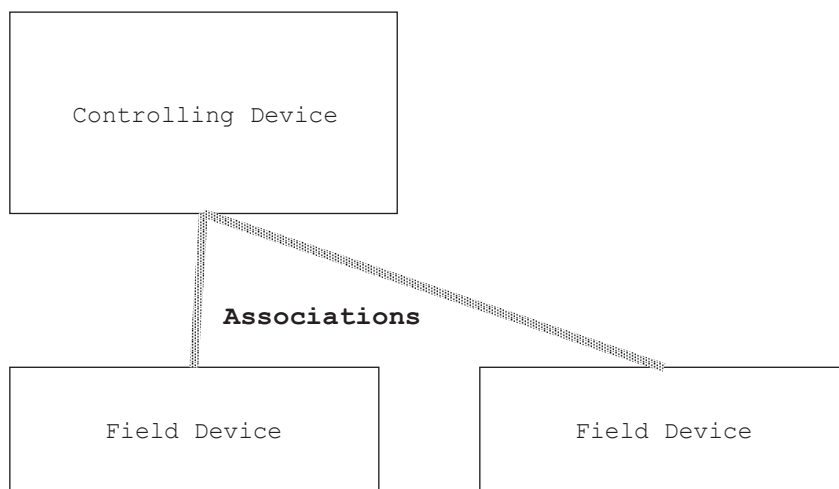


Figure 1 – Example of DP communication with a single controlling device

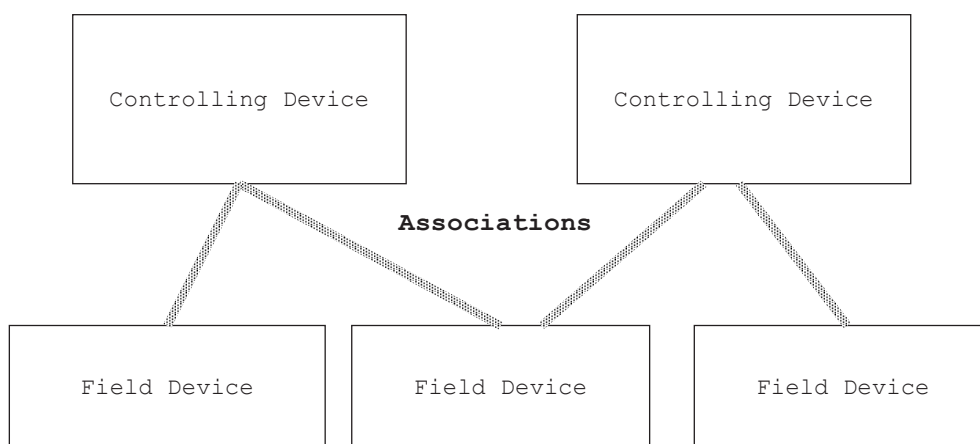


Figure 2 – Example of DP communication with several controlling devices

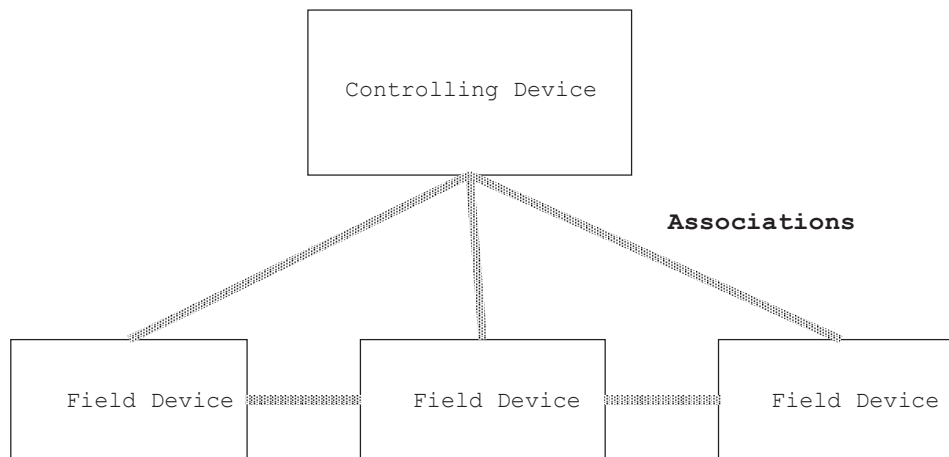


Figure 3 – Example of DP communication between field devices

6.1.3.2 Device types

6.1.3.2.1 DP-master (class 1)

The DP-master (Class 1) is a controlling device which is associated with one or more DP-slaves (field devices). The DP-master (Class 1) performs one or more of the following basic functionalities:

- cyclic exchange of I/O data with related DP-slaves,
- diagnosis with the DP-slaves,
- configuration of DP-slaves,
- treatment of configuration and diagnosis requests of a DP-master (Class 2).

The extended functionalities of a DP-master (Class 1) are:

- acyclic access to process data of the DP-slaves,
- treatment of alarms from DP-slaves,
- support of the Isochronous Mode for synchronization purposes with DP-slaves,
- support of the DXB mechanism for cyclic data exchange between DP-slaves,
- support of the up- and/or download of LR Data with DP-slaves,
- invocation of predefined functions within DP-slaves,
- support for clock synchronisation of the DP-slave's and other DP-master's clocks.

6.1.3.2.2 DP-master (class 2)

The DP-master (Class 2) is a controlling device which manages provision of configuration data (parameter sets) and collection of Diagnosis data for/from a DP-master (Class 1). Additionally the DP-master (Class 2) can perform all basic and some extended communication capabilities of a DP-master (Class 1) to a DP-slave.

6.1.3.2.3 DP-slave

6.1.3.2.3.1 General

The DP-slave is a field device and performs the following activities depending on the functionality.

The basic functionalities are:

- cyclic exchange of I/O data with the assigned DP-master,
- diagnosis with the assigned DP-master,
- treatment of configuration requests of a DP-master.

The extended functionalities of a DP-slave are:

- provision of acyclic access to process data for DP-masters,
- provision of alarms to the assigned DP-master,
- support of the Isochronous Mode for synchronization purposes with the DP-master (Class 1),
- cyclic data exchange between DP-slaves using the Publisher/Subscriber communication,
- support of the up- and/or download of LR Data,
- support of predefined functions which may be invoked by the DP-master (Class 1) or the DP-master (Class 2),
- provision to synchronise the local clock with a DP-master's clock,
- support of DP-slave Redundancy.

DP-slaves are composed of one or more modules. These modules usually reflect the hardware components of the field device.

6.1.3.2.3.2 Module

The DP Application Layer uses modules to reflect the structuring of components (e.g. hardware units, logical units) inside a DP-slave. A Module may have one or more channels which represent the real structure of the Input and/or Output Data. These channels may be a further subdivision of the Input and/or Output Data object.

Each module will be addressed by a slot number (1 to 254). Numbering is without gaps, ascending, beginning with 1. If a slot is not occupied with a module, an empty slot will be registered under the corresponding slot number in the configuration. The slot number "zero" (Slot = 0) refers to the DP-slave itself. Modules contain data which is index addressable. The concrete use of slot number and index to address data within the device is manufacturer specific.

Furthermore to each module a Configuration Identifier has to be assigned. Numbering is without gaps, ascending, beginning with 0. If a slot is not occupied with a module, a Cfg Identifier with the Input and Output Data length equal zero has to be assigned in the configuration (empty slot).

A module provides one or several of the following data elements:

- Diagnosis,
- I/O Data,
- Alarm,
- Process Data,
- LR Data,
- Context,
- Function Invocation Data.

In modular DP-slaves modules determine the actual configuration of the field device (see Figure 4). The number of modules is fixed according to the device specification (maximum number of modules is 244).

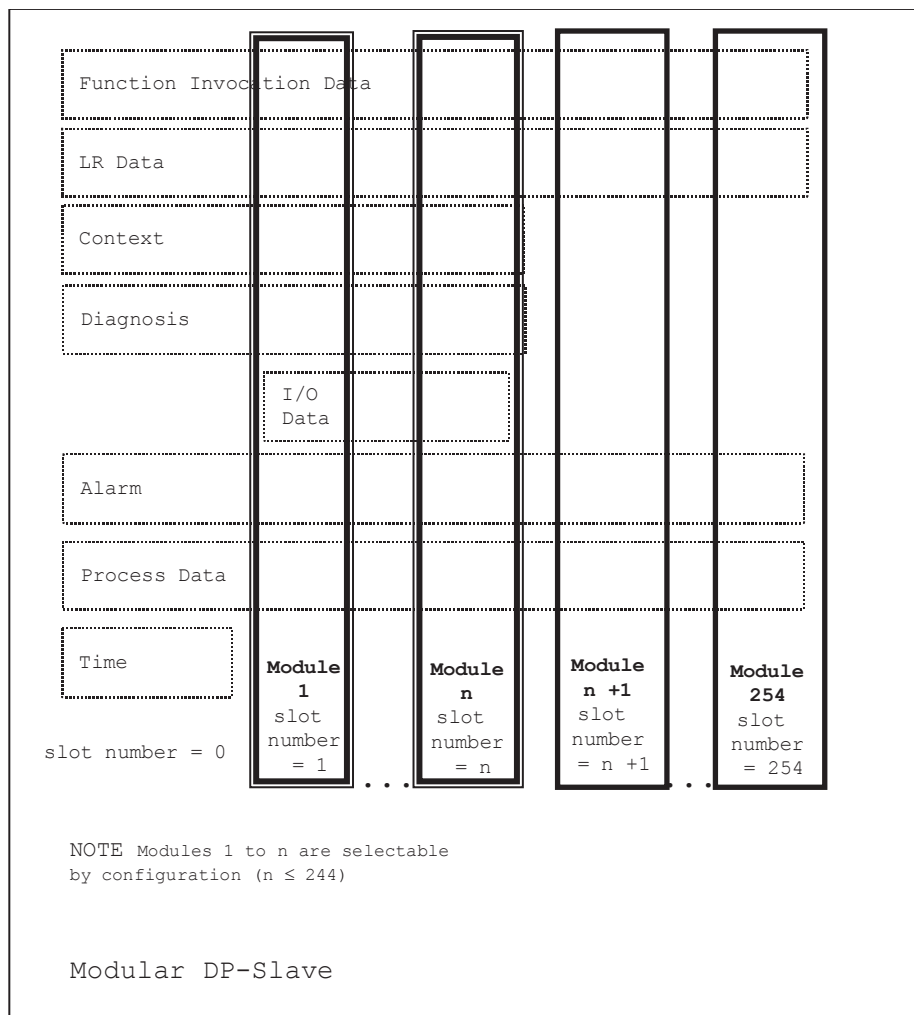


Figure 4 – DP-slave model (modular DP-slave)

In compact DP-slaves with respect to the device specification the number of modules is fixed and therefore also the configuration (for example only one module, see Figure 5).