

DIN EN 61158-5-10

ICS 35.100.70; 35.160

Ersatz für
DIN EN 61158-5-10:2008-09
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 5-10: Dienstfestlegungen des Application Layer
(Anwendungsschicht) –
Typ 10-Elemente
(IEC 61158-5-10:2010);
Englische Fassung EN 61158-5-10:2012, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 5-10: Application layer service definition –
Type 10 elements
(IEC 61158-5-10:2010);
English version EN 61158-5-10:2012, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 5-10: Définition des services des couches d'application –
Éléments de type 10
(CEI 61158-5-10:2010);
Version anglaise EN 61158-5-10:2012, seulement en CD-ROM

Gesamtumfang 760 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2012-03-28 angenommene Europäische Norm als DIN-Norm ist 2013-01-01.

Für DIN EN 61158-5-10:2008-09 gilt eine Übergangsfrist bis zum 2015-03-28.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN IEC 61158-500:2009-06.

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

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

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

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

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

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

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

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

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 61158-5-10 veröffentlicht. Zu deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer 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 der Anwendungsschicht für Typ 10 fest.

Änderungen

Gegenüber DIN EN 61158-5-10:2008-09 wurden folgende Änderungen vorgenommen:

- a) Berichtigungen;
- b) Verbesserungen;
- c) Verbesserung der Synchronisation;
- d) Verbesserung der Hochlaufzeit nach Spannungswiederkehr.

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 61158-5-10: 2008-09

Nationaler Anhang NA (informativ)

3.2 Zusätzliche Begriffe zur verteilten Automatisierung

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.2.1

ACCO

(en: **active connection control object (ACCO)**)

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

3.2.2

Konfigurationsdatenbank

(en: **configuration data base**)

Verbindungsinformationen, die von dem ACCO-ASE verwaltet werden

3.2.3

Verbindung

(en: **connection**)

logischer Verbindungszweig zwischen Senke und Quelle von Attributen und Diensten an unterschiedlichen benutzerdefinierten Schnittstellen von benutzerdefinierten RT-Auto-Objekten

3.2.4

Verbindungskanal

(en: **connection channel**)

Beschreibung einer Verbindung zwischen einer Senke und einer Quelle von Datenelementen

3.2.5

Konsument

(en: **consumer**)

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

3.2.6

Konsumenten-ID

(en: **consumerID**)

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

3.2.7

Daten-Marshalling

(en: **data marshaling**)

Codierung von Parametern der FAL-Dienstelemente in Bezug auf deren Schnittstellendefinition

ANMERKUNG Daten-Marshalling ist Teil des abstrakten ORPC-Modells.

3.2.8

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

3.2.9

Ereignis

(en: **event**)

Instanz einer Änderung von Zuständen

3.2.10

Schnittstelle

(en: **interface**)

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

3.2.11

Schnittstellendefinitionssprache

(en: **interface definition language**)

Syntax und Semantik zur formalen Beschreibung von Dienstparametern

ANMERKUNG Diese Beschreibung ist die Eingabe für das ORPC-Modell, insbesondere für das ORPC-Netzwerkprotokoll.

3.2.12

Schnittstellenzeiger

(en: **interface pointer**)

Schlüsselattribut, das eine Objektschnittstelleninstanz eindeutig adressiert

3.2.13

logisches Gerät

(en: **logical device**)

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

3.2.14

Methode

(en: **method**)

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

3.2.15

ORPC

(en: **object remote procedure call**)

Modell für den objektorientierten oder komponentenbasierten Aufruf von entfernten Methoden

3.2.16**physikalisches Gerät****(en: physical device)**

bestimmte FAL-Klasse, die die Hardware-Einrichtungen eines Automatisierungsgerätes abstrahiert

3.2.17**Eigenschaft****(en: property)**

Synonym für ASE-Attribute, die mithilfe von funktionsfähigen ASE-Diensten gelesen oder geschrieben werden können

ANMERKUNG Diese Dienste werden im Allgemeinen mit „get_<Attributname>“ oder „set_<Attributname>“ bezeichnet und entsprechen den IDL-Schlüsselwörtern „propget“ und „propput“.

3.2.18**Provider****(en: provider)**

Quelle einer Datenverbindung

3.2.19**Provider-ID****(en: providerID)**

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

3.2.20**Qualitätscode****(en: quality code)**

zusätzliche Statusinformation eines Datenelementes

3.2.21**mit Qualitätscode****(en: quality code aware)**

Attribut der RT-Auto-Klasse, welches angibt, dass ein RT-Auto-Objekt einen Statuscode für seine Datenelemente verwendet

3.2.22**ohne Qualitätscode****(en: quality code unaware)**

Gegenteil von „mit Qualitätscode“

3.2.23**RT-Auto****(en: RT-Auto)**

FAL-Klasse, die die Automatisierungsfunktion als prozessbezogene Komponente eines Automatisierungsgerätes abstrahiert

3.2.24**Laufzeitobjektmodell****(en: runtime object model)**

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

3.3 Zusätzliche Begriffe zur dezentralisierten Peripherie

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.3.1

Alarm

(en: alarm)

Aktivierung eines Ereignisses, das einen kritischen Zustand kennzeichnet

3.3.2

Alarmbestätigung

(en: alarm ack)

Bestätigung eines Ereignisses, das einen kritischen Zustand kennzeichnet

3.3.3

Alarmdatenobjekt

(en: alarm data object)

Objekte, die kritische Zustände kennzeichnen und vom Typ des Gerätes/Slots/Subslots/Alarms referenziert werden

3.3.4

zuweisen

(en: allocate)

eine Ressource aus einem allgemeinen Bereich für die ausschließliche Nutzung durch eine spezifische Einheit zuweisen

3.3.5

Anwendung

(en: application)

Funktion oder Datenstruktur, für die Daten verarbeitet oder erzeugt werden

3.3.6

Interoperabilität auf Anwendungsschicht

(en: application layer interoperability)

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

3.3.7

Anwendungsprozess

(en: application process)

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

3.3.8

Anwendungsprozesskennung

(en: application process identifier)

unterscheidet mehrere in einem Gerät verwendete Anwendungsprozesse

ANMERKUNG Die Anwendungsprozesskennung wird von PROFIBUS International (PI) zugewiesen.

3.3.9

Anwendungsprozessobjekt

(en: application process object)

Komponente eines Anwendungsprozesses, die über eine FAL-Anwendungsbeziehung identifizierbar und zugänglich ist

ANMERKUNG Definitionen von Anwendungsprozessobjekten bestehen aus einem Satz von Werten für die Attribute ihrer Klasse (siehe Definition von „Klasse von Anwendungsprozessobjekten“). Auf die Definitionen von Anwendungsprozessobjekten kann mithilfe der Dienste des Object-Management-ASE der FAL von einem entfernten Standort aus zugegriffen werden. Object-Management-Dienste der FAL können zum Laden und Aktualisieren von Objektdefinitionen, zum Lesen von Objektdefinitionen und zur dynamischen Erstellung und Löschung von Anwendungsprozessobjekten und deren Definitionen genutzt werden.

3.3.10**Klasse von Anwendungsprozessobjekten****(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

3.3.11**Anwendungsbeziehung****(en: application relationship)**

kooperative Verbindung zwischen zwei oder mehr Aufrufen von Anwendungseinheiten zum Zweck des Informationsaustausches und der Koordination des gemeinsamen Betriebs. Diese Beziehung wird entweder durch den Austausch von Protokolldateneinheiten der Anwendung oder als Ergebnis von Vorkonfigurationsaktivitäten aktiviert

3.3.12**Anwendungsdienstelement für Anwendungsbeziehungen****(en: application relationship application service element)**

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

3.3.13**Endpunkt der Anwendungsbeziehung****(en: application relationship endpoint)**

Kontext und Verhalten einer Anwendungsbeziehung entsprechend der Sicht und Verwaltung durch einen der in die Anwendungsbeziehung eingeschlossenen Anwendungsprozesse

ANMERKUNG Jeder Anwendungsprozess, der in die Anwendungsbeziehung eingeschlossen ist, verwaltet seinen eigenen Endpunkt der Anwendungsbeziehung.

3.3.14**Attribut****(en: attribute)**

Beschreibung eines von außen sichtbaren Merkmals oder einer von außen sichtbaren Eigenschaft eines Objekts

ANMERKUNG Die Attribute eines Objekts enthalten Informationen zu veränderlichen Teilen eines Objekts. Üblicherweise stellen sie Statusinformationen bereit oder steuern den Betrieb eines Objekts. Attribute können auch das Verhalten eines Objekts beeinflussen. Attribute werden in Klassenattribute und Instanzattribute unterteilt.

3.3.15**backup****(en: backup)**

Status der E/A-AR, der angibt, dass sich diese in Bereitschaft befindet

3.3.16**Verhalten****(en: behavior)**

Angabe, wie ein Objekt auf bestimmte Ereignisse reagiert

3.3.17**Kanal****(en: channel)**

Darstellung eines einzelnen physikalischen oder logischen Verbindungszweiges eines Eingabe- oder Ausgabe-Anwendungsprozessobjektes eines Servers zum Prozess, um die Adressierung von Diagnose-Informationen zu unterstützen

ANMERKUNG Der Kanal stellt typischerweise einen einzelnen Stecker oder eine einzelne Klemme als reale Schnittstelle eines Moduls oder Untermoduls dar. Diese Referenz wird verwendet, um Ausfallpunkte innerhalb von Diagnose-PDUs zu identifizieren.

3.3.18

kanalbezogene Diagnose

(en: **channel related diagnosis**)

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

BEISPIEL Offene Schleife.

3.3.19

Klasse

(en: **class**)

Satz von Objekten, die allesamt den gleichen Typ von Systemkomponente repräsentieren

ANMERKUNG Eine Klasse ist eine Verallgemeinerung eines Objektes; 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.

3.3.20

Klassenattribute

(en: **class attributes**)

Attribut, das von allen Objekten innerhalb derselben Klasse gemeinsam genutzt wird

3.3.21

Klassencode

(en: **class code**)

eindeutige Kennung, die jeder Objektklasse zugewiesen ist

3.3.22

klassenspezifischer Dienst

(en: **class specific service**)

Dienst, der durch eine bestimmte Objektklasse definiert ist, eine erforderliche Funktion, die nicht von einem allgemeinen Dienst ausgeführt wird, auszuführen

ANMERKUNG Ein klassenspezifisches Objekt ist für die Objektklasse, die es definiert, eindeutig.

3.3.23

clear

(en: **clear**)

Status des E/A-Controllers, der angibt, dass der Steuerungsalgorithmus derzeit nicht ausgeführt wird

3.3.24

Client

(en: **client**)

- a) Objekt, das die Dienste eines anderen Objekts (Servers) nutzt, um eine Aufgabe auszuführen
- b) Initiator einer PDU, auf die ein Server reagiert

3.3.25

gemeinsames Profil

(en: **common profile**)

Zusammenstellung geräteunabhängiger Informationen und Funktionalitäten, um Konsistenz zwischen allen Geräten herzustellen

3.3.26

Kommunikationsdatenobjekt

(en: **communication data object**)

Objekte, die Parameter von Kommunikationsbeziehungen sind und vom Gerät/Slot/Subslot/Index referenziert werden

3.3.27**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 Startphase

3.3.28**Konfigurationsfehler****(en: configuration fault)**

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

3.3.29**Konfigurationskennung****(en: configuration identifier)**

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

3.3.30**konsumieren****(en: consume)**

Vorgang des Empfangs von Daten von einem Provider

3.3.31**Konsument****(en: consumer)**

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

3.3.32**Kontextmanagement****(en: context management)**

über das Netzwerk zugängliche Informationen (Kommunikationsobjekte), die die Verwaltung des Betriebs des Feldbussystems, einschließlich der Anwendungsschicht, unterstützen

ANMERKUNG Zur Verwaltung gehören Funktionen wie die Steuerung, die Überwachung und die Diagnose.

3.3.33**Übertragungspfad****(en: conveyance path)**

unidirektionaler Strom von APDUs über eine Anwendungsbeziehung

3.3.34**zyklisch****(en: cyclic)**

sich regelmäßig wiederholend

3.3.35**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

3.3.36**Gerät****(en: device)**

physikalische Hardware, die mit dem Verbindungszweig verbunden ist

ANMERKUNG Ein Gerät kann mehr als einen Knoten umfassen.

3.3.37

Geräte-ID

(en: device ID)

vom Anbieter zugewiesene Kennung des Gerätetyps

3.3.38

Geräteprofil

(en: device profile)

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

3.3.39

Diagnosedatenobjekt

(en: diagnosis data object)

Objekte, die Diagnose-Informationen enthalten und vom Gerät/Slot/Subslot/Index referenziert werden

3.3.40

Diagnose-Informationen

(en: diagnosis information)

alle Daten, die serverseitig für Wartungszwecke verfügbar sind

3.3.41

dynamische Neukonfiguration

(en: dynamic reconfiguration)

Änderung von E/A-Datenobjekten ohne Unterbrechung einer bestehenden Anwendungsbeziehung und fortlaufende Aktualisierung der nicht geänderten E/A-Datenobjekte

3.3.42

Endpunkt

(en: endpoint)

eine der Kommunikationseinheiten, die zu einer Verbindung gehören

3.3.43

Engineering

(en: engineering)

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

3.3.44

Fehler

(en: error)

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

3.3.45

Fehlerklasse

(en: error class)

allgemeine Gruppierung für zugehörige Fehlerdefinitionen und entsprechende Fehlercodes

3.3.46

Fehlercode

(en: error code)

Kennzeichnung eines bestimmten Fehlertyps innerhalb einer Fehlerklasse

3.3.47

Ereignis

(en: event)

Instanz einer Änderung von Zuständen

3.3.48**erweiterte kanalbezogene Diagnose****(en: extended channel related diagnosis)**

Informationen in Bezug auf ein spezifisches Element eines spezifischen Anwendungsprozessobjektes, die zu Wartungszwecken bereitgestellt werden

BEISPIEL Ausfall des Verbindungszweiges.

3.3.49**schnelle Inbetriebnahme****(en: fast startup)**

beschleunigte Prozedur zur Einrichtung eines E/A-Gerätes nach einem Aus- und Einschalten

3.3.50**Rahmen****(en: frame)**

Einheit des Datentransfers auf einem Verbindungszweig

3.3.51**Identifikationsdatenobjekt****(en: identification data object)**

Objekte, die Informationen zum Hersteller und Typ des Gerätes, Moduls und Submoduls enthalten und vom Gerät/Slot/Subslot/Index referenziert werden

3.3.52**impliziter AR-Endpunkt****(en: implicit AR endpoint)**

AR-Endpunkt, der lokal innerhalb eines Gerätes ohne Nutzung des Erstellungsdienstes definiert ist

3.3.53**Index****(en: index)**

Adresse eines Datensatzobjektes innerhalb eines Anwendungsprozesses

3.3.54**Instanz****(en: instance)**

tatsächliches physikalisches Vorkommen eines Objektes innerhalb einer Klasse, das ein oder mehrere Objekte innerhalb derselben Objektklasse identifiziert

3.3.55**Instanzattribut****(en: instance attributes)**

Attribut, das für eine Objektinstanz eindeutig ist und von der Objektklasse nicht gemeinsam genutzt wird

3.3.56**instanziiert****(en: instantiated)**

Objekt, das in einem Gerät erzeugt wurde

3.3.57**Aufruf****(en: invocation)**

Vorgang der Nutzung eines Dienstes oder einer anderen Ressource eines Anwendungsprozesses

ANMERKUNG Jeder Aufruf stellt einen separaten Kontrollthread dar, 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.

3.3.58

E/A-Controller

(en: **IO controller**)

Steuergerät, das für mehrere E/A-Geräte (Feldgeräte) als Client agiert

ANMERKUNG Hierbei handelt es sich üblicherweise um eine speicherprogrammierbare Steuerung oder ein verteiltes Steuerungssystem.

3.3.59

E/A-Datenobjekt

(en: **IO data object**)

Objekt, das zu Verarbeitungszwecken zyklisch übertragen wird und das vom Gerät/Slot/Subslot referenziert wird

3.3.60

E/A-Gerät

(en: **IO device**)

Feldgerät, das für den E/A-Betrieb als Server agiert

3.3.61

E/A-Parameterserver

(en: **IO parameter server**)

Server für Anwendungsparameter von E/A-Geräten (Client)

ANMERKUNG Hierbei handelt es sich üblicherweise um ein Gerät zur Sicherung von Parameterdaten und zur Protokollierung von Online-Änderungen von Geräteparametern.

3.3.62

E/A-Subsystem

(en: **IO subsystem**)

Subsystem, das einen E/A-Controller und alle zugehörigen E/A-Geräte umfasst

3.3.63

E/A-Supervisor

(en: **IO supervisor**)

Engineering-Gerät, das die Inbetriebnahme und Diagnose eines E/A-Systems verwaltet

3.3.64

E/A-System

(en: **IO system**)

System, das aus allen seinen E/A-Subsystemen besteht

ANMERKUNG Eine SPS mit mehr als einem E/A-Controller (Netzwerkschnittstelle) steuert beispielsweise ein E/A-System, das aus einem E/A-Subsystem für jeden E/A-Controller besteht.

3.3.65

isochroner Modus

(en: **Isochronous mode**)

E/A-System, das eng getaktet mit einem Jitter von weniger als 1 µs ausgeführt wird

3.3.66

Member

(en: **member**)

Teil eines Attributes, das als Element eines Feldes strukturiert ist

3.3.67

Methode

(en: **method**)

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

3.3.68**Modul****(en: module)**

Hardware oder logische Komponente eines physikalischen Gerätes

3.3.69**Netzwerk****(en: network)**

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

3.3.70**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 in Bezug auf diese Daten mit einer klar definierten Schnittstelle und klar definiertem Verhalten

3.3.71**objektspezifischer Dienst****(en: object specific service)**

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

3.3.72**operate****(en: operate)**

Status des E/A-Controllers, der angibt, dass der Steuerungsalgorithmus derzeit ausgeführt wird

3.3.73**Peer****(en: peer)**

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

3.3.74**physikalisches Gerät****(en: physical device)**

Automatisierungsgerät oder anderes Netzwerkgerät

3.3.75**Punkt-zu-Punkt-Verbindung****(en: point-to-point connection)**

Verbindung, die zwischen genau zwei Anwendungsprozessobjekten besteht

3.3.76**primary****(en: primary)**

Status der E/A-AR, der angibt, dass sich diese im Betriebszustand befindet

ANMERKUNG Neben der primären E/A-AR kann auch eine E/A-AR zur Sicherung vorhanden sein. Im Beispiel für die Redundanz und dynamische Neukonfiguration von E/A-Daten verwendet.

3.3.77**Provider****(en: provider)**

Knoten oder Quelle, der/die Daten an einen oder viele Konsumenten sendet

3.3.78**PTCP-Domäne****(en: PTCP domain)**

bestimmte Anzahl von PTCP-Unterdomeänen in einem IP-Subnetz

3.3.79

PTCP-Unterlandäne

(en: **PTCP subdomain**)

bestimmte Menge von DTEs mit synchronisierten Uhren

3.3.80

Datensatzobjekt

(en: **record data object**)

Objekte, die bereits vorverarbeitet sind und zu Informationszwecken oder zum Zweck der weiteren Verarbeitung azyklisch übertragen werden und die vom Gerät/Slot/Subslot/Index referenziert werden

3.3.81

Ressource

(en: **resource**)

Verarbeitungs- oder Informationsfähigkeit

3.3.82

run

(en: **run**)

Status des E/A-Controllers, der angibt, dass der Steuerungsalgorithmus derzeit in Betrieb ist

3.3.83

Server

(en: **server**)

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

3.3.84

Dienst

(en: **service**)

Operation oder Funktion, die ein Objekt und/oder eine Objektklasse auf Aufforderung eines anderen Objekts und/oder einer anderen Objektklasse ausführt

3.3.85

Slot

(en: **slot**)

Adresse einer strukturellen Einheit innerhalb eines E/A-Gerätes

ANMERKUNG Innerhalb eines modularen Gerätes adressiert ein Slot typischerweise ein physikalisches Modul. Innerhalb kompakter Geräte adressiert ein Slot typischerweise eine logische Funktion oder ein virtuelles Modul.

3.3.86

Stopp

(en: **stop**)

Status des E/A-Controllers, der angibt, dass der Steuerungsalgorithmus derzeit nicht ausgeführt wird

3.3.87

Untermodule

(en: **submodule**)

Hardware oder logische Komponente eines Moduls

3.3.88

Subslot

(en: **subslot**)

Adresse einer strukturellen Einheit innerhalb eines Slots

ANMERKUNG Ein Subslot kann eine physikalische Schnittstelle für Untermodule innerhalb eines Moduls adressieren. Im Allgemeinen stellt ein Subslot eine zweite Ebene für die Strukturierung von Daten innerhalb eines Gerätes dar.

3.3.89**Anbieter-ID****(en: vendor ID)**

zentrale Verwaltungsnummer, die als Herstellerkennung verwendet wird

ANMERKUNG Die Anbieter-ID wird von PROFIBUS International (PI) zugewiesen.

3.4 Zusätzliche Begriffe zur Medienredundanz

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.4.1**Ausfall****(en: failure)**

Beendigung der Fähigkeit eines Elementes, eine geforderte Funktion auszuführen

ANMERKUNG 1 Nach einem Ausfall befindet sich das Element in einem Fehlzustand.

ANMERKUNG 2 Bei einem Ausfall handelt es sich im Unterschied zu einem Fehlzustand, der einen Zustand beschreibt, um ein Ereignis.

ANMERKUNG 3 Diese Definition gilt nicht für Elemente, die nur aus Software bestehen.

3.4.2**Fehlzustand****(en: fault)**

Zustand eines Elementes, der durch sein Unvermögen gekennzeichnet ist, eine geforderte Funktion auszuführen, außer dem Unvermögen während vorbeugender Wartungs- oder anderer geplanter Maßnahmen oder aufgrund fehlender externer Ressourcen

ANMERKUNG Ein Fehlzustand ist oft das Ergebnis eines Ausfalls des Elementes selbst, kann aber auch ohne vorausgehenden Ausfall auftreten.

3.4.3**Wiederherstellung****(en: recovery)**

Ereignis, bei dem das Element die Fähigkeit zur Ausführung einer geforderten Funktion nach einem Fehlzustand wiedererlangt

3.4.4**Wiederherstellungszeit****(en: recovery time)**

für die Wiederherstellung erforderliche Zeitdauer

3.4.5**Redundanz****(en: redundancy)**

Vorhandensein von zwei oder mehr Mitteln zur Ausführung einer geforderten Funktion in einem Element

ANMERKUNG In diesem Kontext bezieht sich „Redundanz“ auf das Vorhandensein von mehr als einem Pfad (bestehend aus Verbindungszweigen und Switches) zwischen Endknoten.

3.4.6**Ring****(en: ring)**

Netzwerk, bei dem jeder Knoten in einer Reihe mit zwei anderen Knoten verbunden ist

ANMERKUNG 1 Die Knoten sind miteinander in der logischen Form eines Kreises verbunden.

ANMERKUNG 2 Rahmen werden der Reihe nach zwischen aktiven Knoten übertragen, wobei jeder Knoten in der Lage ist, den Rahmen vor der Weiterleitung zu untersuchen und zu verändern.

3.4.7

Ring-Port

(en: ring port)

Port eines Switches, an den ein Ring-Verbindungsweig angeschlossen ist

3.4.8

Ring-Verbindungsweig

(en: ring link)

Verbindungsweig, der zwei Switches eines Rings miteinander verbindet

3.4.9

Switch-Knoten

(en: switch node)

MAC-Bridge, wie in IEEE 802.1D definiert

ANMERKUNG Wird in diesem Kontext als „Switch“ bezeichnet.

English version

**Industrial communication networks -
Fieldbus specifications -
Part 5-10: Application layer service definition -
Type 10 elements
(IEC 61158-5-10:2010)**

Réseaux de communication industriels -
Spécifications des bus de terrain -
Partie 5-10: Définition des services des
couches d'application -
Eléments de type 10
(CEI 61158-5-10:2010)

Industrielle Kommunikationsnetze -
Feldbusse -
Teil 5-10: Dienstfestlegungen des
Application Layer (Anwendungsschicht) -
Typ 10-Elemente
(IEC 61158-5-10:2010)

This European Standard was approved by CENELEC on 2012-03-28. 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, 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.

CENELEC

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

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 65C/606/FDIS, future edition 2 of IEC 61158-5-10, 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 61158-5-10:2012.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2012-12-28
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2015-03-28

This document supersedes EN 61158-5-10:2008.

EN 61158-5-10:2012 includes the following significant technical change with respect to EN 61158-5-10:2008:

- corrections;
- improvements;
- optimization of the synchronization;
- optimization of the startup time from power down.

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.

Endorsement notice

The text of the International Standard IEC 61158-5-10:2010 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following note has to be added for the standard indicated:

IEC/TR 61158-1:2010 NOTE Harmonized as CLC/TR 61158-1:2010 (not modified).

CONTENTS

INTRODUCTION	15
1 Scope	16
1.1 Overview.....	16
1.2 Specifications.....	17
1.3 Conformance	17
2 Normative references.....	17
3 Terms, definitions, abbreviations, symbols and conventions	19
3.1 Referenced terms and definitions	19
3.2 Additional terms and definitions for distributed automation.....	20
3.3 Additional terms and definitions for decentralized periphery	22
3.4 Additional terms and definitions for media redundancy	29
3.5 Abbreviations and symbols.....	30
3.6 Additional abbreviations and symbols for distributed automation.....	31
3.7 Additional abbreviations and symbols for decentralized periphery.....	32
3.8 Additional abbreviations and symbols for media redundancy	33
3.9 Conventions	33
4 Concepts	36
5 Data type ASE	36
5.1 General.....	36
5.2 Formal definition of data type objects	38
5.3 FAL defined data types.....	40
5.4 Data type ASE service specification	82
6 Communication model for common services.....	82
6.1 Concepts.....	82
6.2 ASE data types	82
6.3 ASEs.....	83
7 Communication model for distributed automation	180
7.1 Concepts.....	180
7.2 ASE data types	187
7.3 ASEs.....	191
7.4 ARs.....	398
7.5 Summary of FAL classes.....	401
7.6 Summary of FAL services	402
8 Communication model for decentralized periphery	404
8.1 Concepts.....	404
8.2 ASE data types	420
8.3 ASEs.....	420
8.4 Behavior of an IO device	683
8.5 Behavior of an IO controller.....	726
8.6 Application characteristics	729
Annex A (informative) Device instances	732
Annex B (informative) Components of an Ethernet interface.....	734
Annex C (informative) Scheme of MAC address assignment	738
Annex D (informative) Collection of objects	739
Annex E (informative) Measurement of the fast startup time	740

Bibliography.....	741
Annex ZA (normative) Normative references to international publications with their corresponding European publications.....	742
Figure 1 – Data type class hierarchy example.....	36
Figure 2 – NetworkTime date relation.....	61
Figure 3 – PTCP applications.....	95
Figure 4 – Clock drift measurement.....	106
Figure 5 – Multiple synchronization.....	106
Figure 6 – MRP stack.....	109
Figure 7 – Ring topology with one manager and clients.....	120
Figure 8 – MRM in an open ring.....	120
Figure 9 – More than one MRM in the ring.....	121
Figure 10 – Media redundancy diagnosis dependencies.....	122
Figure 11 – Locating the destination for redundant RT frames.....	161
Figure 12 – Example of periods at a local port.....	168
Figure 13 – FAL ASEs communication architecture.....	183
Figure 14 – Runtime object model.....	184
Figure 15 – Relationship between engineering and runtime.....	185
Figure 16 – Navigation in the runtime object model.....	186
Figure 17 – Operating state block diagram.....	224
Figure 18 – Device status model for the common diagnosis.....	224
Figure 19 – ACCO ASE structure.....	254
Figure 20 – Productive operation of data connections.....	255
Figure 21 – Quality code transfer – standard behavior.....	266
Figure 22 – Startup of a connection.....	267
Figure 23 – Quality code with communication fault.....	267
Figure 24 – Quality code when a connection is cleared.....	268
Figure 25 – Quality code when a connection is deactivated.....	268
Figure 26 – Quality code during the transfer of "incorrect" connection data.....	269
Figure 27 – Quality code for provider in "CBARReady" state.....	269
Figure 28 – Quality code when clearing an object from the provider.....	270
Figure 29 – Quality code when a connection is forced.....	270
Figure 30 – Quality code at QoS violation.....	271
Figure 31 – Push mode.....	277
Figure 32 – Pull mode overview.....	278
Figure 33 – Detailed sequence chart of the pull mode.....	279
Figure 34 – QoS and ORPC communication channel.....	280
Figure 35 – QoS Violation within Pull Mode.....	281
Figure 36 – Monitoring the providers heartbeat.....	282
Figure 37 – State machine RemoteACCO.....	284
Figure 38 – State machine RemoteACCOPROVIDER.....	284
Figure 39 – State machine RemoteACCOPROVIDER _{ORPC}	285
Figure 40 – State machine AR _{ORPC} – PROVIDER.....	285
Figure 41 – State machine GetConnectionData – PROVIDER.....	286
Figure 42 – State machine ProviderConnection.....	286
Figure 43 – State machine ProvConnActivation.....	287
Figure 44 – State machine WorkerORPC.....	287
Figure 45 – Communication stack of distributed automation devices.....	288

Figure 46 – Application relations between devices.....	289
Figure 47 – Communication relations	289
Figure 48 – RT communication channel	291
Figure 49 – Interaction between provider and consumer.....	293
Figure 50 – State machine AR _{SRT} – Consumer	295
Figure 51 – State machine AR _{SRT} – Provider	295
Figure 52 – State machine AccoDataCR – Consumer	296
Figure 53 – State machine AccoDataCR – Provider	297
Figure 54 – RT frame layout.....	298
Figure 55 – Establishing an AccoDataCR.....	298
Figure 56 – Flowchart of the copy cycle for local connections	299
Figure 57 – State machine connect attempt.....	303
Figure 58 – Productive operation of data connections (ORPC channel)	309
Figure 59 – Productive operation of data connections (RT channel)	310
Figure 60 – Productive operation of data connections (Local channel).....	311
Figure 61 – Data flow for cyclic RT	311
Figure 62 – Failure of the provider in productive operation (ORPC push mode)	314
Figure 63 – Failure of the provider in productive operation (ORPC pull mode)	314
Figure 64 – Scenario 1: Provider failure in productive operation (RT).....	316
Figure 65 – Scenario 2: Recovery from provider failure in productive operation (RT)	317
Figure 66 – Failure of the consumer (push mode).....	317
Figure 67 – Failure of the consumer (pull mode).....	318
Figure 68 – Failure of the consumer	319
Figure 69 – Failure of the provider when setting up connections	321
Figure 70 – Information levels	322
Figure 71 – ACCO ASE status model for the common diagnosis	322
Figure 72 – ACCO ASE status model for the detailed diagnosis	323
Figure 73 – Structure of the transmitted connection data	363
Figure 74 – Example of communication between controlling devices and field devices	406
Figure 75 – Example of communication between an engineering station and several controlling and field devices.....	406
Figure 76 – Example of communication between field devices and a server station	407
Figure 77 – Example of communication between field devices.....	407
Figure 78 – Structural units of one arbitrary API of an IO device (general)	409
Figure 79 – Example 1 structural units for interfaces and ports within API 0.....	410
Figure 80 – Example 2 structural units for interfaces and ports within API 0.....	411
Figure 81 – Overview of application processes.....	413
Figure 82 – IO device with APs, slots and subslots.....	414
Figure 83 – Application Process with application process objects (APOs).....	416
Figure 84 – Access to a remote APO	417
Figure 85 – Access to a remote APO for provider/consumer association	418
Figure 86 – Example of one AR with two AREPs.....	419
Figure 87 – Relation of a record data object to one real object.....	421
Figure 88 – Relation of a record data object to two real objects	422
Figure 89 – Overview IO ASE service interactions.....	432
Figure 90 – Example of a resource model at the alarm source.....	503
Figure 91 – General isochronous application model (example).....	566
Figure 92 – ASE relations in an IO device operating in isochronous mode.....	572
Figure 93 – State machine relations in an IO device operating in isochronous mode.....	572

Figure 94 – SyncCtl state diagram	576
Figure 95 – Output state diagram	578
Figure 96 – Input state diagram	582
Figure 97 – Assignment of communication relationship to application relationship	661
Figure 98 – Implicit application relationship	664
Figure 99 – Example IO application relationship (one-to-one)	666
Figure 100 – Example IO application relationship one-to-many	667
Figure 101 – Overview ASE state machines for IO device	683
Figure 102 – State diagram application startup IO device	685
Figure 103 – State diagram neighborhood check	692
Figure 104 – State diagram PD parameter check IO device	701
Figure 105 – State diagram for a submodule	711
Figure 106 – State diagram IO controller during startup	727
Figure 107 – Example of network topology including slower wireless segments	730
Figure 108 – Example of media redundancy including wireless segments	731
Figure A.1 – Instance model	732
Figure B.1 – Scheme of an Ethernet interface	734
Figure B.2 – Scheme of an Ethernet interface with bridging ability	735
Figure B.3 – Scheme of an Ethernet interface with optical ports	736
Figure B.4 – Scheme of an Ethernet interface with bridging ability using radio communication	737
Figure B.5 – Scheme of an Ethernet interface with radio communication	737
Figure C.1 – Scheme of MAC address assignment	738
Figure D.1 – Example for an intersection of IO device, slot, and AR	739
Figure E.1 – Measurement of the fast startup time	740
Table 1 – V2 octets	40
Table 2 – L2 octets	41
Table 3 – PERSISTDEF	43
Table 4 – VARTYPE	43
Table 5 – ITEMQUALITYDEF	44
Table 6 – STATEDEF	47
Table 7 – GROUPEXCEPTIONDEF	47
Table 8 – ACCESSRIGHTSDEF	48
Table 9 – HRESULT	48
Table 10 – E2 octets	51
Table 11 – E2 value range	51
Table 12 – Unipolar2.16 octets	51
Table 13 – Unipolar2.16 value range	51
Table 14 – N2 value range	53
Table 15 – N4 value range	53
Table 16 – X2 value range	54
Table 17 – X4 value range	55
Table 18 – C4 value range	55
Table 19 – T2 value range	57
Table 20 – T2 value range	57
Table 21 – D2 value range	58
Table 22 – R2 value range	58
Table 23 – UUID for decentralized peripherals	59
Table 24 – UUID for distributed automation	60

Table 25 – NetworkTime values	61
Table 26 – NetworkTime octets	61
Table 27 – UNICODEString values	63
Table 28 – UTF-8 character encoding scheme	63
Table 29 – OctetString2+Unsigned8 octets	74
Table 30 – Float32+Unsigned8 octets	75
Table 31 – Unsigned8+Unsigned8 octets	75
Table 32 – Data Types for Value in a VARIANT	78
Table 33 – Unsigned16_S octets	79
Table 34 – Unsigned16_S meaning	79
Table 35 – Integer16_S octets	79
Table 36 – Integer16_S meaning	80
Table 37 – Unsigned8_S octets	80
Table 38 – Unsigned8_S meaning	80
Table 39 – OctetString_S octets	81
Table 40 – OctetString_S status bits	81
Table 41 – F message trailer with 4 octets	81
Table 42 – F message trailer with 5 octets	82
Table 43 – Get	87
Table 44 – Set	89
Table 45 – Identify	92
Table 46 – Hello	93
Table 47 – Start bridge	100
Table 48 – Start slave	101
Table 49 – Start master	102
Table 50 – Stop bridge	103
Table 51 – Stop slave	103
Table 52 – Stop master	104
Table 53 – Sync state change	105
Table 54 – Start MRM	114
Table 55 – Stop MRM	115
Table 56 – Redundancy state change	116
Table 57 – Start MRC	117
Table 58 – Stop MRC	118
Table 59 – Neighborhood changed	118
Table 60 – MRP network/connection parameters	123
Table 61 – MRM parameters	123
Table 62 – MRC parameters	123
Table 63 – Set Prov Data	125
Table 64 – Set Prov Status	125
Table 65 – PPM Activate	126
Table 66 – Close	127
Table 67 – Start	128
Table 68 – Error	128
Table 69 – Get Cons Data	128
Table 70 – Get cons status	129
Table 71 – Set RedRole	130
Table 72 – CPM activate	131
Table 73 – APMS Activate	134

Table 74 – APMR Activate	135
Table 75 – APMS A Data	137
Table 76 – APMR A Data	137
Table 77 – APMR Ack	138
Table 78 – APMS Error	138
Table 79 – APMS Error ERRCLS/ERRCODE	139
Table 80 – APMR Error	139
Table 81 – APMR Error ERRCLS/ERRCODE	139
Table 82 – APMS_Close	140
Table 83 – APMR_Close	140
Table 84 – Connect	141
Table 85 – Release	142
Table 86 – Read	143
Table 87 – Write	144
Table 88 – Control	145
Table 89 – System capabilities	150
Table 90 – Auto negotiation support and status	151
Table 91 – MDI Power Support	151
Table 92 – Link aggregation status	152
Table 93 – Remote systems data change	155
Table 94 – Allowed values of ReductionRatio	158
Table 95 – Frame IDs for RT_CLASS_3	158
Table 96 – Sync Frame	159
Table 97 – FrameSendOffset	159
Table 98 – Tx Port Entry	159
Table 99 – Port state change	162
Table 100 – Set port state	163
Table 101 – Flush filtering data base	163
Table 102 – IFW IRT Schedule Add	164
Table 103 – IFW IRT Schedule Remove	164
Table 104 – IFW Schedule	165
Table 105 – MAU type change	169
Table 106 – Set MAU type	170
Table 107 – IP Multicast address	172
Table 108 – Set ARP Cache	172
Table 109 – Enterprise number	175
Table 110 – Vendor OUI	175
Table 111 – IRT Schedule Add	176
Table 112 – IRT Schedule Remove	177
Table 113 – Schedule	177
Table 114 – N Data	178
Table 115 – A Data	179
Table 116 – C Data	180
Table 117 – Connectable data types	188
Table 118 – Supported data types according to the Base Object Version	189
Table 119 – Usage of character sets	191
Table 120 – QueryInterface (Unknown interface)	193
Table 121 – AddRef (Unknown interface)	194
Table 122 – Release (Unknown interface)	194

Table 123 – GetTypeInfoCount (Dispatch interface)	195
Table 124 – GetTypeInfo (Dispatch interface)	196
Table 125 – GetIdsOfNames (Dispatch interface)	197
Table 126 – Invoke (Dispatch interface)	198
Table 127 – CRC table for the PDev stamp calculation (hexadecimal values).....	203
Table 128 – get_Producer (Physical device interface)	204
Table 129 – get_Product (Physical device interface)	205
Table 130 – get_SerialNo (Physical device interface)	206
Table 131 – get_ProductionDate (Physical device interface).....	207
Table 132 – Revision (Physical device interface).....	208
Table 133 – get_LogicalDevice (Physical device interface).....	209
Table 134 – Type (Physical device interface)	210
Table 135 – PROFINetRevision (Physical device interface)	211
Table 136 – get_PDevStamp (Physical device interface).....	212
Table 137 – get_Count (Browse interface).....	213
Table 138 – BrowseItems (Browse interface)	214
Table 139 – get_Count2 (Browse interface).....	215
Table 140 – BrowseItems2 (Browse interface).....	217
Table 141 – Save (Persist interface).....	218
Table 142 – Save2 (Persist interface).....	219
Table 143 – get_Name (Logical Device interface).....	225
Table 144 – get_Producer (Logical Device interface).....	226
Table 145 – get_Product (Logical Device interface).....	227
Table 146 – get_SerialNo (Logical Device interface)	228
Table 147 – get_ProductionDate (Logical Device interface).....	229
Table 148 – Revision (Logical Device interface)	230
Table 149 – get_ACCO (Logical Device interface).....	231
Table 150 – get_RTAuto (Logical Device interface).....	232
Table 151 – PROFINetRevision (Logical Device interface).....	233
Table 152 – ComponentInfo (Logical Device interface).....	234
Table 153 – get_State (State interface)	235
Table 154 – Activate (State interface).....	236
Table 155 – Deactivate (State interface).....	237
Table 156 – Reset (State interface).....	238
Table 157 – AdviseState (State interface)	239
Table 158 – UnadviseState (State interface)	240
Table 159 – get_Time (Time interface).....	241
Table 160 – put_Time (Time interface).....	242
Table 161 – get_Count (Browse interface).....	243
Table 162 – BrowseItems (Browse interface)	244
Table 163 – get_Count2 (Browse interface).....	245
Table 164 – BrowseItems2 (Browse interface).....	246
Table 165 – GroupError (Group error interface).....	248
Table 166 – AdviseGroupError (Group Error interface).....	249
Table 167 – UnadviseGroupError (Group Error interface).....	250
Table 168 – PingFactor values.....	252
Table 169 – QoS subtypes in the ORPC communication channel	261
Table 170 – QoS subtypes in the RT communication channel	261
Table 171 – QoS Types and Values.....	262

Table 172 – Epsilon value for connectable data types	263
Table 173 – Quality Codes.....	265
Table 174 – Quality code priority table	272
Table 175 – Maximum ORPC substitute value apply time.....	282
Table 176 – Maximum GetConnectionData hold time	283
Table 177 – Usage of RT Variants	290
Table 178 – Mapping QoS Value to RT cycle time.....	291
Table 179 – Maximum RT Substitute Value Apply Time	292
Table 180 – Time Intervals and Timeouts.....	312
Table 181 – Error codes for the ACCO ASE detailed diagnosis	324
Table 182 – AddConnections (ACCO Management interface)	326
Table 183 – RemoveConnections (ACCO Management interface)	327
Table 184 – ClearConnections (ACCO Management interface).....	328
Table 185 – SetActivationState (ACCO Management interface)	329
Table 186 – GetInfo (ACCO Management interface)	331
Table 187 – GetIDs (ACCO Management interface).....	332
Table 188 – GetConnections (ACCO Management interface).....	333
Table 189 – ReviseQoS (ACCO Management interface)	334
Table 190 – get_PingFactor (ACCO Management interface)	335
Table 191 – put_PingFactor (ACCO Management interface)	336
Table 192 – get_CDBCcookie (ACCO Management interface).....	337
Table 193 – GetConsIDs (ACCO Management interface).....	338
Table 194 – GetConsConnections (ACCO Management interface).....	339
Table 195 – DiagConsConnections (ACCO Management interface).....	340
Table 196 – GetProvIDs (ACCO Management interface).....	341
Table 197 – GetProvConnections (ACCO Management interface).....	343
Table 198 – GetDiagnosis (ACCO Management interface).....	344
Table 199 – Request.....	346
Table 200 – Connect (ACCO Server interface).....	353
Table 201 – Disconnect (ACCO Server interface).....	355
Table 202 – DisconnectMe (ACCO Server interface).....	356
Table 203 – SetActivation (ACCO Server interface)	357
Table 204 – Ping (ACCO Server interface).....	358
Table 205 – Connect2 (ACCO Server interface).....	359
Table 206 – GetConnectionData (ACCO Server interface)	361
Table 207 – OnDataChanged (ACCO Callback interface).....	363
Table 208 – Version.....	364
Table 209 – Flags	364
Table 210 – Gnip (ACCO Callback interface)	366
Table 211 – ReadItems (ACCO Sync interface)	367
Table 212 – WriteItems (ACCO Sync interface).....	368
Table 213 – WriteItemsQCD (ACCO Sync interface).....	370
Table 214 – GroupError (Group Error interface)	371
Table 215 – AdviseGroupError (Group Error interface).....	372
Table 216 – UnadviseGroupError (Group Error interface).....	373
Table 217 – ConnectCR (ACCO Server SRT interface).....	374
Table 218 – DisconnectCR (ACCO Server SRT interface).....	376
Table 219 – Connect (ACCO Server SRT interface).....	377
Table 220 – Disconnect (ACCO Server SRT interface).....	379

Table 221 – DisconnectMe (ACCO Server SRT interface).....	380
Table 222 – SetActivation (ACCO Server SRT interface).....	381
Table 223 – Hresult values for access to properties of Custom RT-Auto objects.....	386
Table 224 – Hresult values for access to properties of the System RT-Auto object.....	387
Table 225 – Common hresult values on access to properties of RT-Auto objects.....	387
Table 226 – Quality code for access to properties of Custom RT-Auto objects.....	387
Table 227 – Quality code for access to properties of the System RT-Auto object.....	388
Table 228 – get_Name (RT-Auto interface).....	388
Table 229 – Revision (RT-Auto interface).....	389
Table 230 – ComponentInfo (RT-Auto interface).....	390
Table 231 – get_Count (Browse interface).....	391
Table 232 – BrowseItems (Browse interface).....	392
Table 233 – get_Count2 (Browse interface).....	393
Table 234 – BrowseItems2 (Browse interface).....	395
Table 235 – get_StateCollection (System properties interface).....	396
Table 236 – get_StampCollection (System Properties interface).....	397
Table 237 – CoCreateInstance.....	399
Table 238 – CoDisconnectObject.....	400
Table 239 – Call.....	400
Table 240 – Distributed automation FAL class summary.....	402
Table 241 – Assignment of the services to client and server.....	402
Table 242 – Requirements and features.....	405
Table 243 – Persistence behavior for record data objects.....	424
Table 244 – Read.....	425
Table 245 – Read Query.....	427
Table 246 – Write.....	429
Table 247 – Set input.....	439
Table 248 – Set Input IOCS.....	440
Table 249 – Get Input.....	441
Table 250 – Get Input IOCS.....	442
Table 251 – New Input.....	442
Table 252 – Set input APDU data status.....	443
Table 253 – New Input APDU Data Status.....	444
Table 254 – Read Input Data.....	445
Table 255 – Set Output.....	448
Table 256 – Set Output IOCS.....	449
Table 257 – Get Output.....	450
Table 258 – Get Output IOCS.....	451
Table 259 – New Output.....	451
Table 260 – Set Output APDU Data Status.....	452
Table 261 – New Output APDU Data Status.....	453
Table 262 – Read Output Data.....	455
Table 263 – Read Output Substitute Data.....	457
Table 264 – Write Output Substitute Data.....	459
Table 265 – Read Logbook.....	463
Table 266 – Logbook Event.....	464
Table 267 – Dependencies within channel properties.....	468
Table 268 – Ext Channel Error type.....	470
Table 269 – Ext Channel Add Value for Accumulative Info.....	471

Table 270 – Dependencies within Channel Properties for manufacturer specific diagnosis	472
Table 271 – Read Device Diagnosis	473
Table 272 – Diagnosis Item.....	476
Table 273 – Diagnosis Event	481
Table 274 – State table Diagnosis entry	483
Table 275 – Functions used in state tables.....	484
Table 276 – State table maintenance required entry.....	485
Table 277 – State table maintenance demanded entry	486
Table 278 – State table qualified entry	487
Table 279 – Alarm type.....	491
Table 280 – Channel Diagnosis	492
Table 281 – Manufacturer Specific Diagnosis.....	493
Table 282 – Submodule Diagnosis State.....	493
Table 283 – AR Diagnosis State	493
Table 284 – User Structure Identifier	494
Table 285 – Semantics of Specifier	495
Table 286 – Alarm Notification	499
Table 287 – Alarm Ack.....	502
Table 288 – Module State	508
Table 289 – Usage with respect to CR type.....	510
Table 290 – Detail.....	511
Table 291 – ARInfo.....	512
Table 292 – Ident Info	512
Table 293 – Connect.....	518
Table 294 – Connect Device Access	525
Table 295 – Release.....	527
Table 296 – Abort	527
Table 297 – End Of Parameter.....	528
Table 298 – Application Ready.....	529
Table 299 – Ready For Companion.....	531
Table 300 – Read Expected Identification	532
Table 301 – Read Real Identification.....	535
Table 302 – Read Identification Difference	538
Table 303 – Read AP Data	540
Table 304 – Read I&M0 Filter Data	541
Table 305 – Read I&M0 Data.....	545
Table 306 – Write I&M1 Data	547
Table 307 – Read I&M1 Data.....	549
Table 308 – Write I&M2 Data	551
Table 309 – Read I&M2 Data.....	553
Table 310 – Write I&M3 Data	555
Table 311 – Read I&M3 Data.....	557
Table 312 – Write I&M4 Data	559
Table 313 – Read I&M4 Data.....	561
Table 314 – Write IsoM Data.....	567
Table 315 – Read IsoM Data	569
Table 316 – SYNCH Event.....	571
Table 317 – Primitives issued by the AL to the SyncCtl state machine.....	573
Table 318 – Primitive issued by the SyncCtl state machine to the user.....	574

Table 319 – Primitives issued by the Input state machine to the user.....	574
Table 320 – Primitive issued by the Output state machine to the user.....	574
Table 321 – Primitives issued by the SyncCtl to the output state machine	574
Table 322 – Primitives issued by the output to the SyncCtl state machine	574
Table 323 – Primitives issued by the SyncCtl to the input state machine.....	574
Table 324 – Primitives issued by the output state machine to the AL.....	575
Table 325 – Primitives issued by the AL to the output state machine	575
Table 326 – Primitives issued by the input state machine to the AL	575
Table 327 – Primitives issued by the AL to the input state machine	575
Table 328 – SyncCtl state table.....	577
Table 329 – Output state table	579
Table 330 – Input state table.....	583
Table 331 – Subslot number for interface submodules	588
Table 332 – Subslot number for port submodules.....	589
Table 333 – Subslot Number for Interface Submodules.....	591
Table 334 – Subslot Number for Sync Interface Submodules	591
Table 335 – Sync Properties Role.....	592
Table 336 – Sync Class	593
Table 337 – Fiber Optic Types	593
Table 338 – Fiber Optic Cable Types	594
Table 339 – Write Expected Port Data	598
Table 340 – Write Adjusted Port Data	601
Table 341 – Read real port data.....	603
Table 342 – Read Expected Port Data	606
Table 343 – Read Adjusted Port Data	608
Table 344 – Write IR Data.....	611
Table 345 – Read IR Data.....	614
Table 346 – Write Sync Data.....	617
Table 347 – Read Real Sync Data	620
Table 348 – Read Expected Sync Data.....	623
Table 349 – Read PDev Data.....	626
Table 350 – Sync State Info	631
Table 351 – Write Fiber Optic Data	633
Table 352 – Read Real Fiber Optic Data.....	636
Table 353 – Write MRP Interface Data	638
Table 354 – Read MRP Interface Data.....	641
Table 355 – Write MRP Port Data	643
Table 356 – Read MRP Port Data	645
Table 357 – Write FSU Data	647
Table 358 – Read FSU Data	649
Table 359 – Write Network Component Data.....	651
Table 360 – Read Network Component Data	653
Table 361 – Read Real Interface Data	655
Table 362 – Set Time.....	659
Table 363 – Device Access.....	670
Table 364 – Companion AR.....	670
Table 365 – Media Redundancy.....	674
Table 366 – Frame ID	675
Table 367 – Read AR Data	680

Table 368 – State table application startup IO device	686
Table 369 – State table functions for startup IO device.....	692
Table 370 – State table neighborhood check.....	693
Table 371 – State table functions for neighborhood check.....	700
Table 372 – State table PD parameter check IO device.....	701
Table 373 – State table functions PD parameter check IO device.....	706
Table 374 – State table fiber optic maintenance required	707
Table 375 – State table fiber optic maintenance demanded.....	708
Table 376 – State table fiber optic diagnosis	709
Table 377 – State table for a submodule	711
Table 378 – State table for plug	721
Table 379 – State table for pull.....	723
Table 380 – State table of PTCP behavior.....	724
Table 381 – Functions used by PTCP behavior.....	725
Table 382 – State table IO controller during startup.....	728
Table 383 – Functions used by state table IO controller	729

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/TR 61158–1.

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-10: Application layer service definition – Type 10 elements

1 Scope

1.1 Overview

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 10 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the type 10 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 type 10 IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498) and the OSI Application Layer Structure (ISO/IEC 9545).

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

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

1.2 Specifications

The principal objective of this standard is to specify the characteristics of conceptual application layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of application layer protocols for time-critical communications.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of services standardized as the various Types of IEC 61158, 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 conformance of equipment to this application layer service definition standard mainly achieved through implementation of the modeled behavior of an application layer user (e.g. see user state machines) accompanied by implementation of conforming application layer protocols that fulfill the application layer services as defined in this standard.

2 Normative references

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

IEC 60559, *Binary floating-point arithmetic for microprocessor systems*

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

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

IEC 61158-5-3:2010¹, *Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definitions – Type 3 elements*

IEC 61158-6-3:2010¹, *Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements*

IEC 61158-6-10:2010¹, *Industrial communication networks – Fieldbus specifications – Part 6-10: Application layer protocol specification – Type 10 elements*

IEC 61375-1, *Electric railway equipment – Train bus – Part 1: Train communication network*

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

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

¹ To be published.

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

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

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 8859-1, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

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

ISO/IEC 10646-1, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

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

IEEE 802-2001, *IEEE Standard for Local and metropolitan area networks: Overview and architecture*, available at <<http://www.ieee.org>>

IEEE 802.1AB-2005, *IEEE Standard for Local and metropolitan area networks: Station and media access control connectivity discovery*, available at <<http://www.ieee.org>>

IEEE 802.1D-2004, *IEEE Standard for Local and Metropolitan Area Networks – Media Access Control (MAC) Bridges*, available at <<http://www.ieee.org>>

IEEE 802.1Q-2005, *IEEE Standard for Local and metropolitan area networks – Virtual bridged local area networks*, available at <<http://www.ieee.org>>

IEEE 802.3-2005, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer*, available at <<http://www.ieee.org>>

IETF RFC 768, *User Datagram Protocol*, available at <<http://www.ietf.org>>

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

IETF RFC 792, *Internet Control Message Protocol*, available at <<http://www.ietf.org>>

IETF RFC 826, *An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*, available at <<http://www.ietf.org>>

IETF RFC 1034, *Domain names – concepts and facilities*, available at <<http://www.ietf.org>>

IETF RFC 1112, *Host Extensions for IP Multicasting*, available at <<http://www.ietf.org>>

IETF RFC 1305, *Network Time Protocol (Version 3) – Specification, Implementation and Analysis*, available at <<http://www.ietf.org>>

IETF RFC 1952, *GZIP file format specification version 4.3*, available at <<http://www.ietf.org>>

IETF RFC 2131, *Dynamic Host Configuration Protocol*, available at <<http://www.ietf.org>>

IETF RFC 2365, *Administratively Scoped IP Multicast*, available at <<http://www.ietf.org>>

IETF RFC 2674, *Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions*, available at <<http://www.ietf.org>>

IETF RFC 2737, *Entity MIB (Version 2)*, available at <<http://www.ietf.org>>

IETF RFC 2863, *The Interfaces Group MIB*, available at <<http://www.ietf.org>>

IETF RFC 3330, *Special-Use IPv4 Addresses*, available at <<http://www.ietf.org>>

IETF RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*, available at <<http://www.ietf.org>>

IETF RFC 3490, *Internationalizing Domain Names in Applications (IDNA)*, available at <<http://www.ietf.org>>

IETF RFC 3621, *Power Ethernet MIB*, available at <<http://www.ietf.org>>

IETF RFC 3636, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)*, available at <<http://www.ietf.org>>

The Open Group – Publication C706, *Technical standard DCE1.1: Remote Procedure Call*, available at <<http://www.opengroup.org/onlinepubs/9629399/toc.htm>>

3 Terms, definitions, abbreviations, symbols and conventions

3.1 Referenced terms and definitions

3.1.1 ISO/IEC 7498-1 terms

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

- 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 8824-1 apply:

- a) object identifier
- b) type

3.2 Additional terms and definitions for distributed automation

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

3.2.1

active connection control object (ACCO)

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

3.2.2

configuration data base

interconnection information maintained by the ACCO ASE

3.2.3

connection

the logical link between sink and source of attributes and services at different custom interfaces of Custom RT-Auto objects

3.2.4

connection channel

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

3.2.5

consumer

node or sink that is receiving data from a producer

3.2.6

consumerID

unambiguous identifier within the scope of the ACCO assigned by the consumer to recognize the internal data of a configured interconnection sink.

3.2.7

data marshaling

the encoding of parameters of the FAL service primitives with respect to their interface definition

NOTE This is part of the abstract ORPC model.

3.2.8 engineering

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

3.2.9 event

instance of a change of conditions

3.2.10 interface

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

3.2.11 interface definition language

syntax and semantic of describing service parameters in a formal way

NOTE This description is the input for the ORPC model, especially for the ORPC wire protocol.

3.2.12 interface pointer

key attribute that unambiguously addresses an object interface instance

3.2.13 logical device

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

3.2.14 method

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

3.2.15 object remote procedure call

model for object oriented or component based remote method invocation

3.2.16 physical device

certain FAL class that abstracts the hardware facilities of an automation device

3.2.17 property

synonym for ASE attributes which are readable or writeable via operational ASE services

NOTE These services are generally named "get_<Attribute Name>" or "set_<Attribute Name" and correspond with the IDL keywords "propget" and "propput."

3.2.18 provider

source of a data connection.

3.2.19 providerID

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

3.2.20

quality code

additional status information of a data item

3.2.21

quality code aware

attribute of the RT-Auto class that indicates that an RT-Auto object uses a status code for its data items

3.2.22

quality code unaware

opposite of quality code aware

3.2.23

RT-Auto

FAL class that abstracts the automation function as a process-related component of an automation device

3.2.24

runtime object model

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

3.3 Additional terms and definitions for decentralized periphery

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

3.3.1

alarm

activation of an event that shows a critical state

3.3.2

alarm ack

acknowledgment of an event that shows a critical state

3.3.3

alarm data object

object(s) which represent critical states referenced by device/slot/subslot/alarm type

3.3.4

allocate

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

3.3.5

application

function or data structure for which data is consumed or produced

3.3.6

application layer interoperability

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

3.3.7

application process

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

3.3.8

application process identifier

distinguishes multiple application processes used in a device

NOTE Application process identifier is assigned by PROFIBUS International (PI).

3.3.9

application process object

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

NOTE 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 process objects and their corresponding definitions.

3.3.10

application process object class

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

3.3.11

application relationship

cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation. This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities

3.3.12

application relationship application service element

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

3.3.13

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 Each application process involved in the application relationship maintains its own application relationship endpoint.

3.3.14

attribute

description of an externally visible characteristic or feature of an object

NOTE 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.

3.3.15

backup

status of the IO AR, which indicates that it, is in the standby state

3.3.16

behavior

indication of how an object responds to particular events

3.3.17

channel

representation of a single physical or logical link of an input or output application process object of a server to the process in order to support addressing of diagnosis information

NOTE The channel typically represents a single connector or clamp as a real interface of a module or sub-module. This reference is used to identify points of failure within diagnosis PDUs.

3.3.18

channel related diagnosis

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

EXAMPLE open loop

3.3.19

class

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

NOTE 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.

3.3.20

class attributes

attribute that is shared by all objects within the same class

3.3.21

class code

unique identifier assigned to each object class

3.3.22

class specific service

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

NOTE A class specific object is unique to the object class which defines it.

3.3.23

clear

status of the IO controller, which indicates that the control algorithm is currently not running

3.3.24

client

- a) object which uses the services of another (server) object to perform a task
- b) initiator of a PDU to which a server reacts

3.3.25

common profile

collection of device independent information and functionality providing consistency between all devices

3.3.26

communication data object

object(s) which are parameter of communication relationships and referenced by device/ slot/ subslot/ index

3.3.27

configuration check

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

3.3.28

configuration fault

unacceptable difference between the expected IO-Data object structuring and the real IO-Data object structuring, as detected by the server

3.3.29

configuration identifier

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

3.3.30

consume

act of receiving data from a provider

3.3.31

consumer

node or sink receiving data from a provider

3.3.32

context management

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

NOTE Managing includes functions such as controlling, monitoring, and diagnosing.

3.3.33

conveyance path

unidirectional flow of APDUs across an application relationship

3.3.34

cyclic

repetitive in a regular manner

3.3.35

data consistency

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

3.3.36

device

physical hardware connected to the link

NOTE A device may contain more than one node.

3.3.37

device ID

vendor assigned device type identification

3.3.38

device profile

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

3.3.39

diagnosis data object

object(s) which contain diagnosis information referenced by device/slot/subslot/index

3.3.40

diagnosis information

all data available at the server for maintenance purposes

3.3.41

dynamic reconfiguration

change of IO data objects without interruption of an established application relationship and continuous updating of non-changed IO data objects

3.3.42

endpoint

one of the communicating entities involved in a connection

3.3.43

engineering

abstract term that characterizes the client application or device responsible for configuring an automation system

3.3.44

error

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

3.3.45

error class

general grouping for related error definitions and corresponding error codes

3.3.46

error code

identification of a specific type of error within an error class

3.3.47

event

instance of a change of conditions

3.3.48

extended channel related diagnosis

information concerning a specific element of a specific application process object, provided for maintenance purposes

EXAMPLE Link Fail

3.3.49

fast startup

accelerated procedure to setup an IO device after a power cycle

3.3.50

frame

unit of data transfer on a link

3.3.51

identification data object

object(s) that contain information about device, module and sub-module manufacturer and type referenced by device/slot/subslot/index

3.3.52

implicit AR endpoint

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

3.3.53

index

address of a record data object within an application process

3.3.54

instance

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

3.3.55

instance attributes

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

3.3.56

instantiated

object that has been created in a device

3.3.57

invocation

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

NOTE 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.

3.3.58

IO controller

controlling device, which acts as client for several IO devices (field devices)

NOTE This is usually a programmable controller or a distributed control system.

3.3.59

IO data object

object designated to be transferred cyclically for the purpose of processing and referenced by device/slot/subslot

3.3.60

IO device

field device which acts as server for IO operation

3.3.61

IO parameter server

server for application parameter of IO devices (client)

NOTE This is usually a device to backup parameter data and to log online changes of device parameter.

3.3.62

IO subsystem

subsystem composed of one IO controller and all its associated IO devices

3.3.63

IO supervisor

engineering device which manages commissioning and diagnosis of an IO system

3.3.64

IO system

system composed of all its IO subsystems

NOTE As an example a PLC with more than one IO controller (network interface) controls one IO system composed of an IO subsystem for each IO controller.

3.3.65

Isochronous mode

IO system operating tightly synchronized with a jitter of less than 1 μ s

3.3.66

member

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

3.3.67

method

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

3.3.68
module

hardware or logical component of a physical device

3.3.69
network

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

3.3.70
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 a clearly defined interface and behavior

3.3.71
object specific service

service unique to the object class which defines it

3.3.72
operate

status of the IO controller that indicates that the control algorithm is currently running

3.3.73
peer

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

3.3.74
physical device

automation or other network device

3.3.75
point-to-point connection

connection that exists between exactly two application process objects

3.3.76
primary

status of the IO AR that indicates that it is in the operating state

NOTE Besides a primary IO AR a backup IO AR may exist. In example used for redundancy and dynamic reconfiguration of IO data.

3.3.77
provider

node or source sending data to one or many consumer

3.3.78
PTCP domain

certain number of PTCP subdomains in one IP subnet

3.3.79
PTCP subdomain

certain amount of DTEs with synchronized clocks

3.3.80
record data object

object(s) which are already pre-processed and transferred acyclically for the purpose of information or further processing and referenced by device/slot/subslot/index

3.3.81

resource

processing or information capability

3.3.82

run

status of the IO controller which indicates that the control algorithm is currently operating

3.3.83

server

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

3.3.84

service

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

3.3.85

slot

address of a structural unit within an IO device

NOTE Within a modular device, a slot typically addresses a physical module. Within compact devices, a slot typically addresses a logical function or virtual module.

3.3.86

stop

status of the IO controller which indicates that the control algorithm is currently not running

3.3.87

submodule

hardware or logical component of a module

3.3.88

subslot

address of a structural unit within a slot

NOTE A subslot may address a physical interface for submodules within a module. Generally, a subslot is a second level to structure data within a device.

3.3.89

vendor ID

central administrative number used as manufacturer identification

NOTE The vendor ID is assigned by PROFIBUS International (PI).

3.4 Additional terms and definitions for media redundancy

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

3.4.1

failure

termination of the ability of an item to perform a required function

NOTE 1 After failure the item has a fault.

NOTE 2 "Failure" is an event, as distinguished from "fault", which is a state.

NOTE 3 This concept as defined does not apply to items considering of software only.

3.4.2

fault

state of an item characterized by its inability to perform a required function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources

NOTE A fault is often the result of a failure of the item itself, but may exist without prior failure.

3.4.3

recovery

event when the item regains the ability to perform a required function, after a fault

3.4.4

recovery time

time period required for recovery

3.4.5

redundancy

existence in an item of two or more means of performing a required function

NOTE In this context, the existence of more than one path (consisting of links and switches) between end nodes.

3.4.6

ring

network where each node is connected in series to two other nodes

NOTE 1 Nodes are connected to one another in the logical shape of a circle.

NOTE 2 Frames are passed sequentially between active nodes, each able to examine or modify the frame before forwarding it.

3.4.7

ring port

port of a switch to which a ring link is attached

3.4.8

ring link

link that connects two switches of a ring

3.4.9

switch node

MAC bridge as defined in IEEE 802.1D

NOTE It will be called "switch" in this context.

3.5 Abbreviations and symbols

ACCO	Active Connection Control Object
AE	Application Entity
AL	Application Layer
ALME	Application Layer Management Entity
ALP	Application Layer Protocol
APO	Application Process Object
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Identifier
AR	Application Relationship
AREP	Application Relationship End Point
ASCII	American Standard code for Information Interchange
ASE	Application Service Element
CID	Connection ID
CIM	Computer Integrated Manufacturing
CIP	Control and Information Protocol
CM_API	Actual Packet Interval

CM_RPI	Requested Packet Interval
Cnf	Confirmation
COR	Connection originator
CR	Communication Relationship
CREP	Communication Relationship End Point
DL-	(as a prefix) Data Link-
DLC	Data Link Connection
DLL	Data Link Layer
DLM	Data Link-management
DLSAP	Data Link Service Access Point
DLSDU	DL-service-data-unit
DNS	Domain Name Service
DP	Decentralised Peripherals
FAL	Fieldbus Application Layer
FIFO	First In First Out
HMI	Human-machine Interface
ID	Identifier
IDL	Interface Definition Language
IEC	International Electrotechnical Commission
Ind	Indication
IP	Internet Protocol
ISO	International Organization for Standardization
LDev	Logical device
LME	Layer Management Entity
O2T	Originator to target (connection characteristics)
OPT	Originator to target (connection characteristics)
ORPC	Object Remote Procedure Call
OSI	Open Systems Interconnect
PDev	Physical device
PDU	Protocol Data Unit
PL	Physical Layer
QoS	Quality of Service
QC	Quality code
REP	Route Endpoint
Req	Request
Rsp	Response
RT	Runtime
SAP	Service Access Point
SCL	Security Level
SDU	Service Data Unit
SEM	State event matrix
SMIB	System Management Information Base
SMK	System Management Kernel
STD	State transition diagram, used to describe object behaviour
S-VFD	Simple Virtual Field Device
T2O	Target to originator (connection characteristics)
TPO	Target to originator (connection characteristics)
VAO	Variable Object

3.6 Additional abbreviations and symbols for distributed automation

ACCO	Active Connection Control Object
IDL	Interface Definition Language
IP	Internet Protocol
LDev	Logical device
ORPC	Object Remote Procedure Call
PDev	Physical device
QoS	Quality of Service
QC	Quality code
RT	Runtime
TCP	Transmission Control Protocol

3.7 Additional abbreviations and symbols for decentralized periphery

AE	Application Entity
AL	Application Layer
ALME	Application Layer Management Entity
ALP	Application Layer Protocol
ALPMI	Alarm Protocol machine Initiator
ALPMR	Alarm Protocol machine Responder
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Identifier
APO	Application Process Object
AR	Application Relationship
AREP	Application Relationship End Point
ARP	Address Resolution Protocol
ASCII	American Standard code for Information Interchange
ASE	Application Service Element
BMC	Best Master Clock
CM	Context Management
Cnf	Confirmation
CR	Communication Relationship
CREP	Communication Relationship End Point
DCE	OSF Distributed Computing Environment
DCP	Discovery and basic Configuration Protocol
DCPMCR	DCP Multicast Receiver
DCPMCS	DCP Multicast Sender
DCPUCR	DCP Unicast Receiver
DCPUCS	DCP Unicast Sender
DHCP	Dynamic Host Configuration Protocol
DIM	Device Interface Module
DL-	(as a prefix) Data Link-
DLC	Data Link Connection
DLL	Data Link Layer
DLPDU	Data Link-Protocol Data Unit
DLSDU	DL-service-data-unit
DNS	Domain Name Service
DTE	Data Terminal Equipment
FAL	Fieldbus Application Layer
FIFO	First In First Out
GSDML	General Station Description Markup Language
I&M	Identification and Maintenance Profile
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
IOCS	Input Output Object Consumer Status
IOPS	Input Output Object Provider Status
IP	Internet Protocol
IR	Isochronous Relay
IRT	Isochronous Real Time Protocol
ISO	International Organization for Standardization
IsoM	Isochronous Mode
LED	Light Emitting Diode
LLDP	Link Layer Discovery Protocol
LME	Layer Management Entity
lsb	Least Significant Bit
LT	Length/type
MAC	Medium Access Control
msb	Most Significant Bit

NCA	Network Computing Architecture
OSI	Open Systems Interconnect
PDU	Protocol Data Unit
PI	PROFIBUS International see www.profibus.com
PL	Physical Layer
PTCP	Precision Transparent Clock Protocol
QoS	Quality of Service
Req	Request
RPC	Remote Procedure Call
Rsp	Response
RT	Real Time Protocol
RTA	Real Time Protocol Acyclic
RTC	Real Time Protocol Cyclic
RTE	Real Time Ethernet
SDU	Service Data Unit
TLV	Type Length Value (coding rule)
UDP	User Datagram Protocol
UUID	Universal Unique Identifier
VLAN	Virtual Local Area Network

3.8 Additional abbreviations and symbols for media redundancy

MRC	Media Redundancy Client
MRM	Media Redundancy Manager
MRP	Media Redundancy Protocol

3.9 Conventions

3.9.1 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.

3.9.2 General conventions

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

3.9.3 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.