

**DIN EN IEC 61158-6-2**

ICS 35.100.70; 35.160

Ersatz für  
DIN EN 61158-6-2:2015-09  
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –  
Feldbusse –  
Teil 6-2: Protokollspezifikation des Application Layer  
(Anwendungsschicht) – Typ 2-Elemente  
(IEC 61158-6-2:2019);  
Englische Fassung EN IEC 61158-6-2:2019, nur auf CD-ROM**

Industrial communication networks –  
Fieldbus specifications –  
Part 6-2: Application layer protocol specification – Type 2 elements  
(IEC 61158-6-2:2019);  
English version EN IEC 61158-6-2:2019, only on CD-ROM

Réseaux de communication industriels –  
Spécifications des bus de terrain –  
Partie 6-2: Spécification du protocole de la couche application – Éléments de type 2  
(IEC 61158-6-2:2019);  
Version anglaise EN IEC 61158-6-2:2019, seulement en CD-ROM

Gesamtumfang 284 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

## Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2019-07-25 angenommene Europäische Norm als DIN-Norm ist 2020-08-01.

Für DIN EN 61158-6-2:2015-09 besteht eine Übergangsfrist bis 2022-07-25.

## Nationales Vorwort

*Vorausgegangener Norm-Entwurf: E DIN EN 61158-600:2017-12.*

Dieser Entwurf betrifft auch die Normen DIN EN IEC 61158-6-3:2020-08, DIN EN IEC 61158-6-4:2020-08, DIN EN IEC 61158-6-10:2020-08, DIN EN IEC 61158-6-12:2020-08, DIN EN IEC 61158-6-19:2020-08, DIN EN IEC 61158-6-21:2020-08, DIN EN IEC 61158-6-23:2020-08, DIN EN IEC 61158-6-25:2020-08 und DIN EN IEC 61158-6-26:2020-08.

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Industrielle Kommunikation“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE ([www.dke.de](http://www.dke.de)) zuständig.

Die enthaltene IEC-Publikation wurde vom SC 65C „Industrial networks“ erarbeitet und unterliegt dem Copyright der IEC .

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

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

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

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

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

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

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN IEC 61158-6-2 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt und Abschnitt 3 (außer Unterabschnitte 3.4 und 3.5), der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für die meisten 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.

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 IEC 61158 legen eine Anzahl von Feldbus-Protokolltypen fest. Die vorliegende Norm legt die Dienste fest, die durch den Data-Link Layer der einzelnen Typen zur Verfügung gestellt werden.

## Änderungen

Gegenüber DIN EN 61158-6-2:2015-09 wurden folgende Änderungen vorgenommen:

- a) Klarstellungen der Antwort-PDU-Formate im Falle eines Ausfalls in 4.1.5;
- b) Klarstellungen zur Definition und Behandlung von Verbindungsanfrage-Prioritäten in 4.1.6.5 und 4.1.6.6;
- c) Klärung des verbleibenden Verbindungspaths in 4.1.6.12;
- d) Erweiterungen der allgemeinen Syntax in 4.1.8.1;
- e) Erweiterungen und Klarstellungen von Identitätsobjekt-PDUs in 4.1.8.2;
- f) Aktualisierungen von Assembly-Objekt-PDUs in 4.1.8.4;
- g) Erweiterungen und Klarstellungen von Time-Sync-Objekt-PDUs in 4.1.8.6;
- h) Aktualisierungen von Parameterobjekt-PDUs in 4.1.8.7;
- i) Erweiterungen der Connection Manager-Objekt-PDUs in 4.1.8.8;
- j) Erläuterungen zu Connection-Objekt-PDUs in 4.1.8.9;
- k) Erweiterungen und Klarstellungen von Nachrichten- und Verbindungspfaden in 4.1.9;
- l) Aktualisierungen von Objekt- und Dienstklassencodes in 4.1.10 und Fehlercodes in 4.1.11;
- m) Aktualisierungen der Datentypen in 4.2.4, 5.1.3 und 5.2.3;
- n) Aktualisierungen der abstrakten Kapselungssyntax in 4.3;
- o) Aktualisierungen der DLL-Mapping-Protokollmaschine 2 in Abschnitt 11;
- p) verschiedene redaktionelle Korrekturen.

## 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 61158-6: 2005-05  
 DIN EN 61158-6-2: 2008-09, 2013-03, 2015-09

## Nationaler Anhang NA (informativ)

### Auszugsweise Übersetzung

Reihenfolge und Inhalt der folgenden Abschnitte sind identisch mit denen in den jeweiligen Abschnitten der Englischen Fassung.

#### EINLEITUNG

Dieser Teil von IEC 61158 gehört zu einer Normenreihe, die zur Erleichterung der Verbindung von Komponenten von Automatisierungssystemen erstellt wurde. Er ist mit anderen Normen der Reihe verbunden, wie durch das in IEC 61158-1 beschriebene „dreischichtige“ Feldbus-Referenzmodell definiert.

Das Anwendungsprotokoll stellt den Anwendungsdienst bereit, indem es die Dienste nutzt, die über die Datenverbindung oder eine andere unmittelbar darunter liegende Schicht verfügbar sind. Das vorrangige Ziel dieses Dokuments ist es, eine Reihe von Regeln für die Kommunikation zu erstellen, die sich auf die Verfahren beziehen, die von den Peer Application Entities (AE) zum Zeitpunkt der Kommunikation durchzuführen sind. Diese Regeln für die Kommunikation sollen eine stabile Grundlage für die Weiterentwicklung bilden, damit sie einer Vielzahl von Zwecken dienen können:

- als Leitfaden für Implementierer und Konstrukteure;
- zur Verwendung bei der Prüfung und Beschaffung von Ausrüstung;
- als Teil einer Vereinbarung über die Zulassung von Systemen in der offenen Systemumgebung;
- als eine Erweiterung des Verständnisses von zeitkritischer Kommunikation innerhalb der OSI.

Dieses Dokument befasst sich insbesondere mit der Kommunikation und dem Zusammenwirken von Sensoren, Effektoren und anderen Automatisierungsgeräten. Durch die Verwendung dieses Dokuments zusammen mit anderen Normen, die innerhalb der OSI- oder Feldbus-Referenzmodelle vorhanden sind, können andernfalls inkompatible Systeme in beliebiger Kombination zusammenarbeiten.

## 1 Anwendungsbereich

### 1.1 Allgemeines

Der Feldbus Application Layer (FAL, en: Fieldbus Application Layer) stellt Anwenderprogramme mit Mitteln zum Zugriff auf die Feldbuskommunikationsumgebung bereit. Insofern kann der FAL als ein „Fenster zwischen entsprechenden Anwendungsprogrammen“ angesehen werden.

Dieser Teil der IEC 61158 legt allgemeine Elemente für grundlegende zeitkritische und nicht zeitkritische Messaging-Kommunikationen zwischen Anwendungsprogrammen in einem Automatisierungsumfeld und für Typ-2-Feldbusse spezifisches Material fest. Der Begriff „zeitkritisch“ wird zum Angeben des Vorhandenseins eines Zeitfensters, in dem eine oder mehrere festgelegte Aktionen mit einem gewissen definierten Sicherheitsgrad abzuschließen sind, verwendet. Das Nichtabschließen festgelegter Aktionen in dem Zeitfenster riskiert den Ausfall der Anwendungen, die die Aktionen anfordern, zusammen mit einem begleitenden Risiko für Betriebsmittel, die Anlage und möglicherweise Menschenleben.

Diese Internationale Norm legt Interaktionen zwischen entfernten Anwendungen fest und definiert das nach außen erkennbare Verhalten, das durch den Typ-2-Feldbus Application Layer bereitgestellt wird, hinsichtlich

- a) der formalen abstrakten Syntax, die die zwischen kommunizierenden Anwendungsinstanzen übermittelten Application Layer-Protokolldateneinheiten definiert;
- b) der Übertragungssyntax, die die Codierungsregeln definiert, die auf die Application Layer Protokolldateneinheiten angewendet werden;

- c) des Anwendungskontextzustandsautomaten, der das zwischen kommunizierenden Anwendungsinstanzen erkennbare Anwendungsdienstverhalten definiert;
- d) der Anwendungsbeziehungszustandsautomaten, die das zwischen kommunizierenden Anwendungsinstanzen erkennbare Kommunikationsverhalten definieren.

Der Zweck dieses Dokuments ist es, das Protokoll zu definieren, das bereitgestellt wird, um

- a) die Drahtdarstellung der in IEC 61158-5-2 definierten Dienstprimitive zu definieren und
- b) das nach außen erkennbare Verhalten zu definieren, das im Zusammenhang mit der entsprechenden Übertragung steht.

Dieses Dokument legt das Protokoll des Typ-2-Feldbus Application Layer in Übereinstimmung mit dem OSI-Basisreferenzmodell (ISO/IEC 7498-1) und der OSI Application Layer-Struktur (ISO/IEC 9545) fest.

## 1.2 Spezifikationen

Das wesentliche Ziel dieses Dokuments ist es, die Syntax und das Verhalten des Application Layer-Protokolls, das die in IEC 61158-5-2 definierten Application Layer-Dienste übermittelt, festzulegen.

Ein Nebenziel ist es, Migrationspfade von zuvor bestehenden industriellen Kommunikationsprotokollen bereitzustellen.

## 1.3 Konformität

Dieses Dokument legt keine individuellen Implementierungen oder Produkte fest, noch beschränkt es die Implementierungen von Application Layer-Instanzen in industriellen Automatisierungssystemen. Konformität wird durch Implementierung dieser Application Layer-Protokollspezifikation erreicht.

## 3 Begriffe, Symbole, Abkürzungen und Vereinbarungen

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe, Symbole, Abkürzungen und Vereinbarungen.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- IEC Electropedia: verfügbar unter <http://www.electropedia.org/>
- ISO Online Browsing Platform: verfügbar unter <http://www.iso.org/obp>

### 3.1 Begriffe von anderen ISO/IEC-Normen

#### 3.1.1 Begriffe von ISO/IEC 7498-1

- a) abstrakte Syntax (en: abstract syntax)
- b) Anwendungsinstanz (en: application entity)
- c) Anwendungsprozess (en: application process)
- d) Anwendungsprotokolldateneinheit (en: application protocol data unit)
- e) Anwendungsdienstelement (en: application service element)
- f) Aufruf einer Anwendungsinstanz (en: application entity invocation)
- g) Aufruf eines Anwendungsprozesses (en: application process invocation)
- h) Anwendungstransaktion (en: application transaction)
- i) Darstellungskontext (en: presentation context)

- j) reales offenes System (en: real open system)
- k) Übertragungssyntax (en: transfer syntax)

### 3.1.2 Begriffe von ISO/IEC 9545

- a) Anwendungsassoziation (en: application-association)
- b) Anwendungskontext (en: application-context)
- c) Anwendungskontextname (en: application context name)
- d) Anwendungsinstanzaufruf (en: application-entity-invocation)
- e) Anwendungsinstanztyp (en: application-entity-type)
- f) Anwendungsprozessaufruf (en: application-process-invocation)
- g) Anwendungsprozessstyp (en: application-process-type)
- h) Anwendungsdienstelement (en: application-service-element)
- i) Anwendungssteuerdienstelement (en: application control service element)

### 3.1.3 Begriffe von ISO/IEC 8824-1

- a) Objektbezeichner (en: object identifier)
- b) Typ (en: type)
- c) Wert (en: value)
- d) einfacher Typ (en: simple type)
- e) strukturierter Typ (en: structured type)
- f) Komponenten-Typ (en: component type)
- g) Tag (en: tag)
- h) boolescher Typ (en: Boolean type)
- i) wahr (en: true)
- j) falsch (en: false)
- k) ganzzahliger Typ (en: integer type)
- l) Bit-String-Typ (en: bitstring type)
- m) Oktet-String-Typ (en: octetstring type)
- n) Null-Typ (en: null type)
- o) Sequenz-Typ (en: sequence type)
- p) Sequenz vom Typ (en: sequence of type)
- q) Auswahl-Typ (en: choice type)
- r) getaggter Typ (en: tagged type)
- s) beliebiger Typ (en: any type)
- t) Modul (en: module)
- u) Produktion (en: production)

**3.1.4 Begriffe von ISO/IEC 8825-1**

- a) Codierung (eines Datenwerts) (en: encoding (of a data value))
- b) Datenwert (en: data value)
- c) Bezeichner-Oktette (en: identifier octets) (in diesem Dokument wird die Singularform verwendet)
- d) Längen-Oktett(e) (en: length octet(s)) (in diesem Dokument werden sowohl Singular- als auch Pluralformen verwendet)
- e) Inhalts-Oktette (en: contents octets)

**3.2 Begriffe von IEC 61158-5-2**

- a) Anwendungsbeziehung (en: application relationship)
- b) Client (en: client)
- c) Peer (en: peer)
- d) Server (en: server)

**3.3 Zusätzliche Begriffe****3.3.1****zuweisen**

(en: allocate)

Ressource aus einem allgemeinen Bereich nehmen und diese Ressource für die ausschließliche Nutzung durch eine spezifische Instanz zuordnen

**3.3.2****Anwendung**

(en: application)

Funktion oder Datenstruktur, für die Daten konsumiert oder produziert werden

**3.3.3****Anwendungsobjekte**

(en: application objects)

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

**3.3.4****Attribut**

(en: attribute)

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

Anmerkung 1 zum Begriff: 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.

**3.3.5****Verhalten**

(en: behavior)

Angabe, wie ein Objekt auf bestimmte Ereignisse reagiert

### 3.3.6

#### **Best-Master-Clock-Algorithmus**

(en: Best Master Clock Algorithm)

#### **BMC**

(en: BMCA)

Algorithmus, der von jedem Knoten durchgeführt wird, um die Uhr zu bestimmen, die die Master-Uhr in einem Subnetz und die Grandmaster-Uhr für die Domäne wird

Anmerkung 1 zum Begriff: Der Algorithmus vergleicht vorrangig „Priority1“, die Uhrenqualität, „Priority2“ und die Quellidentität, um den besten Master unter den verfügbaren Kandidaten zu ermitteln.

### 3.3.7

#### **Grenzuhr**

(en: boundary clock)

Uhr, die über mehrere PTP-(Precision Time Protocol)-Ports in einer Domäne verfügt und den in der Domäne verwendeten Zeitmaßstab aufrechterhält

Anmerkung 1 zum Begriff: Sie darf als die Zeitquelle dienen, d. h. eine Master-Uhr sein, und mit einer anderen Uhr synchronisiert werden, d. h. eine Slave-Uhr sein.

[QUELLE: IEC 61588:2009, 3.1.3, modifiziert - zweiter Satz in eine Anmerkung geändert]

### 3.3.8

#### **aufgerufen**

(en: called)

Dienstbenutzer oder Dienstanbieter, der ein Anzeigeprimitiv oder eine Anforderungs-APDU empfängt

### 3.3.9

#### **aufrufend**

(en: calling)

Dienstbenutzer oder Dienstanbieter, der ein Anforderungsprimitiv oder eine Anforderungs-APDU initiiert

### 3.3.10

#### **Klasse**

(en: class)

Menge von Objekten, die allesamt die gleiche Art Systemkomponente repräsentieren

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

### 3.3.11

#### **Klassenattribut**

(en: class attribute)

Attribut, das allen Objekten innerhalb derselben Klasse gemeinsam ist

### 3.3.12

#### **Klassencode**

(en: class code)

eindeutige Kennung, die jeder Objektklasse zugewiesen ist

### 3.3.13

#### **klassenspezifischer Dienst**

(en: class specific service)

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

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

**3.3.14****Client**

(en: client)

- a) Objekt, das die Dienste eines anderen (Server-)Objekts zur Ausführung einer Aufgabe nutzt
- b) Initiator einer Nachricht, auf die ein Server reagiert

**3.3.15****Uhr**

(en: clock)

am PTP (Precision Time Protocol) teilnehmender Knoten, der eine Messung der seit einem festgelegten Bezugszeitpunkt abgelaufenen Zeit bereitstellen kann

Anmerkung 1 zum Begriff: In IEC 61588:2009 sind drei Typen von Uhren festgelegt: Grenzuhren, transparente Uhren und gewöhnliche Uhren.

[QUELLE: IEC 61588:2009, 3.1.4, modifiziert - andere Anmerkung]

**3.3.16****Kommunikationsobjekte**

(en: communication objects)

Komponenten, die zur Laufzeit für einen Nachrichtenaustausch im Netzwerk sorgen und diesen verwalten

BEISPIELE Verbindungsmanager-Objekt, UCMM-(Unconnected Message Manager)-Objekt und Nachrichtenrouter-Objekt.

**3.3.17****Verbindung**

(en: connection)

logische Verknüpfung zwischen Anwendungsobjekten, die sich in demselben Gerät oder in unterschiedlichen Geräten befinden dürfen

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

**3.3.18****Verbindungs-ID**

(en: connection ID)

**CID**

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

**3.3.19****Verbindungspfad**

(en: connection path)

Oktett-Strom, der das Anwendungsobjekt definiert, auf das sich eine Verbindungsinstanz bezieht

**3.3.20****Verbindungspunkt**

(en: connection point)

Puffer, der als Unterinstanz eines Assembly-Objekts dargestellt ist

**3.3.21****konsumieren**

(en: consume)

Vorgang des Empfangs von Daten von einem Produzenten

### 3.3.22

#### **Konsument**

(en: consumer)

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

### 3.3.23

#### **konsumierende Anwendung**

(en: consuming application)

Anwendung, die Daten konsumiert

### 3.3.24

#### **zyklisch**

(en: cyclic)

sich regelmäßig wiederholend

### 3.3.25

#### **Gerät**

(en: device)

physische Hardware, die mit dem Verbindungszweig verbunden ist

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

### 3.3.26

#### **Geräteprofil**

(en: device profile)

Zusammenstellung geräteabhängiger Informationen und Funktionalitäten zur Bereitstellung von Konsistenz zwischen ähnlichen Geräten desselben Gerätetyps

### 3.3.27

#### **Domäne**

(en: domain)

logische Gruppierung von Uhren, die sich mithilfe des Protokolls miteinander synchronisieren, die aber nicht zwangsläufig mit Uhren in einer anderen Domäne synchronisiert werden

[QUELLE: IEC 61588:2009, 3.1.7]

### 3.3.28

#### **Endknoten**

(en: end node)

erzeugender oder konsumierender Knoten

### 3.3.29

#### **Endpunkt**

(en: end point)

eine der Kommunikationsinstanzen, die zu einer Verbindung gehören

### 3.3.30

#### **Bezugszeitpunkt**

(en: epoch)

Anfang einer Zeitskala

[QUELLE: IEC 61588:2009, 3.1.9]

### 3.3.31

#### **Fehler**

(en: error)

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

**3.3.32****Rahmen**

(en: frame)

umgangssprachliches Synonym für DLPDU

**3.3.33****Grandmaster-Uhr**

(en: grandmaster clock)

innerhalb einer Domäne: Uhr, die als endgültige Zeitquelle für die Synchronisation der Uhren unter Verwendung des PTP-Protokolls dient

[QUELLE: IEC 61588:2009, 3.1.13]

**3.3.34****Instanz**

(en: instance)

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

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

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

**3.3.35****Instanzattribut**

(en: instance attribute)

Attribut, dessen Wert für eine Objektinstanz eindeutig ist und dessen Definition alle Instanzen eines Objekts gemeinsam haben

**3.3.36****instanziiert**

(en: instantiated)

Objekt, das in einem Gerät erzeugt wurde

**3.3.37****Interoperabilität**

(en: interoperability)

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

**3.3.38****Keeper**

(en: Keeper)

Objekt, das für die Verteilung von Konfigurationsdaten des Verbindungszweigs an alle Knoten auf dem Verbindungszweig verantwortlich ist

**3.3.39****Little-Endian**

(en: little endian)

Modell der Speicherorganisation, bei dem das am wenigsten signifikante Oktett an der kleinsten Adresse gespeichert wird, oder für die Übertragung, bei dem das am wenigsten signifikante Oktett zuerst übertragen wird

Anmerkung 1 zum Begriff: Native Datentypen vom Typ 2 werden in Little-Endian-Reihenfolge gesendet.

### 3.3.40

#### **L-Paket**

(en: Lpacket)

#### **Link-Paket**

(en: Link packet)

Teil von Anwendungsinformationen, der eine Größe, ein Steuerungsoktett, ein Tag und Verbindungsdaten enthält

Anmerkung 1 zum Begriff: Der Data-Link Layer von Peers verwenden L-Pakete, um Dienstdateneinheiten von höhere Schichten im OSI-Stapel zu senden und zu empfangen.

### 3.3.41

#### **Verwaltungsinformationen**

(en: management information)

über das Netzwerk zugängliche Informationen, die die Verwaltung des Betriebs des Feldbussystems, einschließlich des Application Layer, unterstützen

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

### 3.3.42

#### **Master-Uhr**

(en: master clock)

im Zusammenhang mit einem einzelnen PTP-Kommunikationspfad: Uhr, die die Zeitquelle ist, mit der alle anderen Uhren auf diesem Pfad synchronisiert werden

[QUELLE: IEC 61588:2009, 3.1.17]

### 3.3.43

#### **Member**

(en: member)

Teil eines Attributs, das als Element eines Feldes strukturiert ist

### 3.3.44

#### **Nachrichtenrouter**

(en: Message Router)

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

### 3.3.45

#### **Mehrpunktverbindung**

(en: multipoint connection)

Verbindung von einem Knoten zu vielen Knoten

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

### 3.3.46

#### **Netzwerk**

(en: network)

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

### 3.3.47

#### **Objekt**

(en: object)

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

**3.3.48****objektspezifischer Dienst**

(en: object specific service)

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

**3.3.49****gewöhnliche Uhr**

(en: ordinary clock)

Uhr, die über einen einzelnen PTP-Port in einer Domäne verfügt und die in der Domäne verwendete Zeitskala aufrechterhält

Anmerkung 1 zum Begriff: Sie darf als eine Zeitquelle dienen, d. h. eine Master-Uhr sein, oder mit einer anderen Uhr synchronisiert werden, d. h. eine Slave-Uhr sein.

[QUELLE: IEC 61588:2009, 3.1.22, modifiziert - zweiter Satz in eine Anmerkung geändert]

**3.3.50****Absender**

(en: originator)

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

**3.3.51****übergeordnete Uhr**

(en: parent clock)

Master-Uhr, mit der eine Uhr synchronisiert wird

[QUELLE: IEC 61588:2009, 3.1.23]

**3.3.52****Punkt-zu-Punkt-Verbindung**

(en: point-to-point connection)

Verbindung, die zwischen genau zwei Anwendungsobjekten besteht

**3.3.53****Precision Time Protocol**

(en: Precision Time Protocol)

**PTP**

in IEC 61588:2009 festgelegtes Protokoll

Anmerkung 1 zum Begriff: Wird „PTP“ adjektivisch gebraucht, weist dies darauf hin, dass das modifizierte Nomen in IEC 61588:2009 festgelegt ist oder im Zusammenhang damit gedeutet wird.

[QUELLE: IEC 61588:2009, 3.1.28, modifiziert - zweiter Satz in eine Anmerkung geändert]

**3.3.54****produzieren**

(en: produce)

Vorgang des Sendens von Daten zum Empfang durch einen Konsumenten

**3.3.55****Produzent**

(en: producer)

Knoten, der für das Senden von Daten verantwortlich ist

**3.3.56****PTP-Nachricht**

(en: PTP message)

einer der in IEC 61588:2009 festgelegten Nachrichtentypen

[QUELLE: IEC 61588:2009, 3.1.33]

### 3.3.57

#### **PTP-Port**

(en: PTP port)

logischer Zugangspunkt einer Uhr für die PTP-Kommunikation mit dem Kommunikationsnetzwerk

[QUELLE: IEC 61588:2009, 3.1.35]

### 3.3.58

#### **empfangend**

(en: receiving)

Dienstbenutzer, der ein bestätigtes oder ein unbestätigtes Dienstprimitiv empfängt, oder Dienstanbieter, der eine bestätigte APDU oder eine unbestätigte APDU empfängt

### 3.3.59

#### **Ressource**

(en: resource)

Verarbeitungs- oder Informationsfähigkeit eines Subsystems

### 3.3.60

#### **sendend**

(en: sending)

Dienstbenutzer, der ein bestätigtes oder ein unbestätigtes Dienstprimitiv sendet, oder Dienstanbieter, der eine bestätigte APDU oder eine unbestätigte APDU sendet

### 3.3.61

#### **Server**

(en: server)

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

### 3.3.62

#### **Dienst**

(en: service)

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

### 3.3.63

#### **synchronisierte Uhren**

(en: synchronized clocks)

(mit einer gewissen Messunsicherheit) zwei Uhren, die denselben Bezugszeitpunkt haben und bei denen Messungen des Zeitpunkts eines einzelnen Ereignisses zu einem beliebigen Zeitpunkt nicht mehr als die angegebene Messunsicherheit voneinander abweichen

[QUELLE: IEC 61588:2009, 3.1.40, modifiziert - umformuliert]

### 3.3.64

#### **Systemzeit**

(en: system time)

absoluter Zeitwert nach der Definition der Zeitsynchronisation für die Kommunikationsprofilfamilie 2 (CPF2) im Kontext eines verteilten Zeitsystems, in dem alle Geräte über eine lokale Uhr verfügen, die mit einer gemeinsamen Master-Uhr synchronisiert wird

Anmerkung 1 zum Begriff: Im Rahmen der Kommunikationsprofilfamilie 2 (CPF2) entspricht die Systemzeit einem ganzzahligen 64-Bit-Wert in Nanosekunden, wobei sich der Wert „0“ auf das Datum „1970-01-01“ bezieht.

### 3.3.65

#### **Ziel**

(en: target)

Endknoten, mit dem eine Verbindung aufgebaut wird

**3.3.66****temporärer Knoten**

(en: temporary node)

transienter Knoten (en: transient node)

**3.3.67****Transaktions-ID**

(en: transaction id)

Feld innerhalb eines UCMM-Headers, das eine Antwort auf die zugehörige Anforderung abstimmt

**3.3.68****transparente Uhr**

(en: transparent clock)

Gerät, das die Zeit misst, die eine PTP-Ereignisnachricht durch das Gerät benötigt, und das diese Informationen den Uhren, die diese PTP-Ereignisnachricht empfangen, bereitstellt

[QUELLE: IEC 61588:2009, 3.1.46, modifiziert - gekürzt]

**3.3.69****Manager für unverbundene Nachrichten**

(en: unconnected message manager)

**UCMM**

Komponente innerhalb eines Knotens, die verbindungslose explizite Nachrichten überträgt und empfängt und sie direkt an das Nachrichtenrouter-Objekt sendet

**3.3.70****verbindungsloser Dienst**

(en: unconnected service)

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

**3.3.71****Anbieter-ID**

(en: vendor ID)

Kennzeichnung jedes Produktherstellers/Anbieters durch eine eindeutige Nummer

Anmerkung 1 zum Begriff: Anbieter-ID werden von der ODVA, Inc. vergeben.

– Leerseite –

English Version

Industrial communication networks - Fieldbus specifications -  
Part 6-2: Application layer protocol specification - Type 2 elements  
(IEC 61158-6-2:2019)

Réseaux de communication industriels - Spécifications des bus de  
terrain - Partie 6-2: Spécification du protocole de la couche  
application - Éléments de type 2  
(IEC 61158-6-2:2019)

Industrielle Kommunikationsnetze - Feldbusse -  
Teil 6-2: Protokollspezifikation des Application Layer  
(Anwendungsschicht) - Typ 2-Elemente  
(IEC 61158-6-2:2019)

This European Standard was approved by CENELEC on 2019-07-25. 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, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.



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

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

## **European foreword**

The text of document 65C/948/FDIS, future edition 4 of IEC 61158-6-2, 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 IEC 61158-6-2:2019.

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) 2020-04-25
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2022-07-25

This document supersedes EN 61158-6-2:2014 and all of its amendments and corrigenda (if any).

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

## **Endorsement notice**

The text of the International Standard IEC 61158-6-2:2019 was approved by CENELEC as a European Standard without any modification.

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

|                  |      |                                                   |
|------------------|------|---------------------------------------------------|
| IEC 61131-3      | NOTE | Harmonized as EN 61131-3.                         |
| IEC 61784-1:2019 | NOTE | Harmonized as EN IEC 61784-1:2019 (not modified). |
| IEC 61784-2:2019 | NOTE | Harmonized as EN IEC 61784-2:2019 (not modified). |

## Annex ZA (normative)

### Normative references to international publications with their corresponding European publications

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

NOTE 1 Where an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: [www.cenelec.eu](http://www.cenelec.eu).

| Publication              | Year | Title                                                                                                                                                              | EN/HD            | Year |
|--------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|------|
| IEC 61158-1              | 2019 | Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series                                 | EN IEC 61158-1   | 2019 |
| IEC 61158-3-2            | 2014 | Industrial communication networks - Fieldbus specifications - Part 3-2: Data-link layer service definition - Type 2 elements                                       | EN 61158-3-2     | 2014 |
| IEC 61158-3-2:2014/Amd 1 | 2019 |                                                                                                                                                                    | + A1             | 2019 |
| IEC 61158-4-2            | 2019 | Industrial communication networks - Fieldbus specifications - Part 4-2: Data-link layer protocol specification - Type 2 elements                                   | EN IEC 61158-4-2 | 2019 |
| IEC 61158-5-2            | 2019 | Industrial communication networks - Fieldbus specifications - Part 5-2: Application layer service definition - Type 2 elements                                     | EN IEC 61158-5-2 | 2019 |
| IEC 61588                | 2009 | Precision clock synchronization protocol for-networked measurement and control systems                                                                             |                  | -    |
| IEC 61784-3-2            | -    | Industrial communication networks - Profiles -- Part 3-2: Functional safety fieldbuses - Additional specifications for CPF 2                                       |                  | -    |
| IEC 61800-7-202          | -    | Adjustable speed electrical power drive systems - Part 7-202: Generic interface and use of profiles for power drive systems - Profile type 2 specification         | EN 61800-7-202   | -    |
| IEC 62026-3              | 2014 | Low-voltage switchgear and controlgear -- Controller-device interfaces (CDIs) - Part 3: DeviceNet                                                                  |                  | -    |
| ISO 639-2                | -    | Codes for the representation of names of-languages - Part-2: Alpha-3 code                                                                                          |                  | -    |
| ISO 11898                | 1993 | Road vehicles - Interchange of digital-information - Controller area network (CAN) for high-speed communication                                                    |                  | -    |
| ISO/IEC 7498-1           | -    | Information technology - Open Systems-Interconnection - Basic reference model: The basic model                                                                     |                  | -    |
| ISO/IEC 8824-1           | -    | Information technology - Abstract Syntax-Notation One (ASN.1): Specification of basic notation                                                                     |                  | -    |
| ISO/IEC 8825-1           | -    | Information technology - ASN.1 encoding rules:- Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) |                  | -    |

|                     |      |                                                                                                                                |   |
|---------------------|------|--------------------------------------------------------------------------------------------------------------------------------|---|
| ISO/IEC 9545        | -    | Information technology - Open Systems-Interconnection - Application layer structure                                            | - |
| ISO/IEC 10646       | -    | Information technology_- Universal Coded-Character Set (UCS)                                                                   | - |
| ISO/IEC 10731       | -    | Information technology - Open Systems-Interconnection - Basic Reference Model - Conventions for the definition of OSI services | - |
| IEEE Std 802.1D     | 2004 | IEEE standard for local and metropolitan area networks – Media Access Control (MAC) bridges                                    |   |
| IEEE Std 802.1Q     | 2005 | IEEE standard for local and metropolitan area networks – Virtual bridged local area networks                                   |   |
| IEEE Std 802.3      | 2015 | IEEE Standard for Ethernet                                                                                                     | - |
| IETF RFC 1035       | -    | Domain Names - Implementation and-Specification                                                                                | - |
| IETF RFC 1112       | -    | Host Extensions for IP Multicasting                                                                                            | - |
| IETF RFC 1117       | -    | Internet numbers                                                                                                               | - |
| IETF RFC 1122       | -    | Requirements for Internet Hosts --Communication Layers                                                                         | - |
| IETF RFC 1759       | -    | Printer MIB                                                                                                                    | - |
| IETF RFC 2236       | -    | Internet Group Management Protocol, Version 2-                                                                                 | - |
| IETF RFC 2474       | -    | Definition of the Differentiated Services Field-(DS Field) in the IPv4 and IPv6 Headers                                        | - |
| IETF RFC 2475       | -    | An Architecture for Differentiated Services                                                                                    | - |
| IETF RFC 2597       | -    | Assured Forwarding PHB Group                                                                                                   | - |
| IETF RFC 2873       | -    | TCP Processing of the IPv4 Precedence Field                                                                                    | - |
| IETF RFC 3140       | -    | Per Hop Behavior Identification Codes                                                                                          | - |
| IETF RFC 3246       | -    | An Expedited Forwarding PHB (Per-Hop-Behavior)                                                                                 | - |
| IETF RFC 3376       | -    | Internet Group Management Protocol, Version 3-                                                                                 | - |
| IETF RFC 4594       | -    | Configuration Guidelines for DiffServ Service-Classes                                                                          | - |
| IETF RFC 791        | -    | Internet protocol darpa internet program-protocol specification                                                                | - |
| ISO/IEC/IEEE 8802-3 | -    | Standard for Ethernet                                                                                                          | - |

## CONTENTS

|                                                                                                                                 |     |
|---------------------------------------------------------------------------------------------------------------------------------|-----|
| European foreword .....                                                                                                         | 2   |
| Annex ZA (normative) Normative references to international publications<br>with their corresponding European publications ..... | 3   |
| INTRODUCTION .....                                                                                                              | 17  |
| 1 Scope .....                                                                                                                   | 18  |
| 1.1 General .....                                                                                                               | 18  |
| 1.2 Specifications .....                                                                                                        | 18  |
| 1.3 Conformance .....                                                                                                           | 19  |
| 2 Normative references .....                                                                                                    | 19  |
| 3 Terms, definitions, symbols, abbreviated terms and conventions .....                                                          | 21  |
| 3.1 Terms and definitions from other ISO/IEC standards .....                                                                    | 21  |
| 3.1.1 Terms and definitions from ISO/IEC 7498-1 .....                                                                           | 21  |
| 3.1.2 Terms and definitions from ISO/IEC 9545 .....                                                                             | 21  |
| 3.1.3 Terms and definitions from ISO/IEC 8824-1 .....                                                                           | 21  |
| 3.1.4 Terms and definitions from ISO/IEC 8825-1 .....                                                                           | 22  |
| 3.2 Terms and definitions from IEC 61158-5-2 .....                                                                              | 22  |
| 3.3 Additional terms and definitions .....                                                                                      | 22  |
| 3.4 Abbreviated terms and symbols .....                                                                                         | 29  |
| 3.5 Conventions.....                                                                                                            | 30  |
| 3.5.1 General concept.....                                                                                                      | 30  |
| 3.5.2 Attribute specification .....                                                                                             | 30  |
| 3.5.3 Common services.....                                                                                                      | 31  |
| 3.5.4 State machine conventions .....                                                                                           | 34  |
| 4 Abstract syntax.....                                                                                                          | 36  |
| 4.1 FAL PDU abstract syntax .....                                                                                               | 36  |
| 4.1.1 General.....                                                                                                              | 36  |
| 4.1.2 PDU structure.....                                                                                                        | 37  |
| 4.1.3 UCMM_PDUs.....                                                                                                            | 39  |
| 4.1.4 Transport_Headers .....                                                                                                   | 40  |
| 4.1.5 CM_PDUs.....                                                                                                              | 43  |
| 4.1.6 CM PDU components.....                                                                                                    | 57  |
| 4.1.7 MR headers .....                                                                                                          | 66  |
| 4.1.8 OM_Service_PDU .....                                                                                                      | 67  |
| 4.1.9 Message and connection paths .....                                                                                        | 121 |
| 4.1.10 Class, attribute and service codes .....                                                                                 | 135 |
| 4.1.11 Error codes .....                                                                                                        | 141 |
| 4.2 Data abstract syntax specification .....                                                                                    | 155 |
| 4.2.1 Transport format specification .....                                                                                      | 155 |
| 4.2.2 Abstract syntax notation .....                                                                                            | 155 |
| 4.2.3 Control network data specification .....                                                                                  | 155 |
| 4.2.4 Data type specification / dictionaries .....                                                                              | 157 |
| 4.3 Encapsulation abstract syntax.....                                                                                          | 160 |
| 4.3.1 Encapsulation protocol .....                                                                                              | 160 |
| 4.3.2 Command descriptions.....                                                                                                 | 163 |
| 4.3.3 Common packet format.....                                                                                                 | 174 |

|        |                                                         |     |
|--------|---------------------------------------------------------|-----|
| 5      | Transfer syntax .....                                   | 178 |
| 5.1    | Compact encoding .....                                  | 178 |
| 5.1.1  | Encoding rules .....                                    | 178 |
| 5.1.2  | Encoding constraints .....                              | 179 |
| 5.1.3  | Examples .....                                          | 179 |
| 5.2    | Data type reporting .....                               | 186 |
| 5.2.1  | Object data representation .....                        | 186 |
| 5.2.2  | Elementary data type reporting .....                    | 186 |
| 5.2.3  | Constructed data type reporting .....                   | 187 |
| 6      | Structure of FAL protocol state machines .....          | 193 |
| 7      | AP-Context state machine .....                          | 193 |
| 7.1    | Overview .....                                          | 193 |
| 7.2    | Connection object state machine.....                    | 193 |
| 7.2.1  | I/O Connection instance behavior .....                  | 193 |
| 7.2.2  | Bridged Connection instance behavior.....               | 198 |
| 7.2.3  | Explicit Messaging Connection instance behavior .....   | 199 |
| 8      | FAL service protocol machine (FSPM).....                | 202 |
| 8.1    | General .....                                           | 202 |
| 8.2    | Primitive definitions .....                             | 202 |
| 8.3    | Parameters of primitives .....                          | 207 |
| 8.4    | FSPM state machines.....                                | 207 |
| 9      | Application relationship protocol machines (ARPMs)..... | 208 |
| 9.1    | General .....                                           | 208 |
| 9.2    | Connection-less ARPM (UCMM) .....                       | 208 |
| 9.2.1  | General .....                                           | 208 |
| 9.2.2  | Primitive definitions.....                              | 208 |
| 9.2.3  | Parameters of primitives .....                          | 209 |
| 9.2.4  | UCMM state machines .....                               | 210 |
| 9.2.5  | Examples of UCMM sequences .....                        | 215 |
| 9.2.6  | Management UCMM.....                                    | 217 |
| 9.3    | Connection-oriented ARPMs (transports).....             | 218 |
| 9.3.1  | Transport PDU buffer .....                              | 218 |
| 9.3.2  | Transport classes .....                                 | 218 |
| 9.3.3  | Common primitive definitions .....                      | 219 |
| 9.3.4  | Parameters of common primitives.....                    | 220 |
| 9.3.5  | Transport state machines – class 0.....                 | 220 |
| 9.3.6  | Transport state machines – class 1.....                 | 223 |
| 9.3.7  | Transport state machines – class 2.....                 | 229 |
| 9.3.8  | Transport state machines – class 3.....                 | 237 |
| 9.3.9  | Transport state machines – classes 4, 5, 6.....         | 246 |
| 9.3.10 | Transport state machines – class 4.....                 | 246 |
| 9.3.11 | Transport state machines – class 5.....                 | 246 |
| 9.3.12 | Transport state machines – class 6.....                 | 246 |
| 10     | DLL mapping protocol machine 1 (DMPM 1) .....           | 246 |
| 10.1   | General .....                                           | 246 |
| 10.2   | Link producer .....                                     | 246 |
| 10.3   | Link consumer.....                                      | 247 |

|        |                                                            |     |
|--------|------------------------------------------------------------|-----|
| 10.4   | Primitive definitions.....                                 | 247 |
| 10.4.1 | Primitives exchanged between DMPM and ARPM.....            | 247 |
| 10.4.2 | Parameters of ARPM/DMPM primitives.....                    | 247 |
| 10.4.3 | Primitives exchanged between data-link layer and DMPM..... | 248 |
| 10.4.4 | Parameters of DMPM/Data-link Layer primitives.....         | 248 |
| 10.4.5 | Network connection ID.....                                 | 249 |
| 10.5   | DMPM state machine.....                                    | 249 |
| 10.5.1 | DMPM states.....                                           | 249 |
| 10.5.2 | Functions used by DMPM.....                                | 251 |
| 10.6   | Data-link Layer service selection.....                     | 251 |
| 11     | DLL mapping protocol machine 2 (DMPM 2).....               | 251 |
| 11.1   | General.....                                               | 251 |
| 11.2   | Mapping of UCMM PDUs.....                                  | 251 |
| 11.2.1 | General.....                                               | 251 |
| 11.2.2 | Common requirements for Connection Manager PDU's.....      | 253 |
| 11.2.3 | Forward_open PDU for class 2 and class 3 connections.....  | 254 |
| 11.2.4 | Forward_open for class 0 and class 1 connections.....      | 255 |
| 11.2.5 | Forward_close.....                                         | 259 |
| 11.3   | Mapping of transport class 0 and class 1 PDUs.....         | 259 |
| 11.3.1 | Class 0 and class 1 PDUs.....                              | 259 |
| 11.3.2 | No dependency on TCP connections.....                      | 260 |
| 11.3.3 | Class 0 and class 1 packet ordering.....                   | 260 |
| 11.3.4 | Screening incoming connected data.....                     | 261 |
| 11.4   | Mapping of transport class 2 and class 3 PDU's.....        | 261 |
| 11.5   | Mapping of transport classes 4 to 6.....                   | 262 |
| 11.6   | IGMP Usage.....                                            | 262 |
| 11.6.1 | Background (informative).....                              | 262 |
| 11.6.2 | IGMP Membership Report messages.....                       | 262 |
| 11.6.3 | IGMP Leave Group messages.....                             | 263 |
| 11.7   | Quality of Service (QoS) for CP 2/2 messages.....          | 263 |
| 11.7.1 | Overview.....                                              | 263 |
| 11.7.2 | DSCP format.....                                           | 264 |
| 11.7.3 | IEEE 802.1D/IEEE 802.Q format.....                         | 264 |
| 11.7.4 | Mapping CPF 2 traffic to DSCP and IEEE 802.1D.....         | 265 |
| 11.7.5 | CP 2/2 usage of DSCP.....                                  | 265 |
| 11.7.6 | CP 2/2 usage of IEEE 802.1D/IEEE 802.1Q.....               | 265 |
| 11.7.7 | User considerations with IEEE 802.1D/IEEE 802.1Q.....      | 266 |
| 11.8   | Management of an encapsulation session.....                | 266 |
| 11.8.1 | Phases of an encapsulation session.....                    | 266 |
| 11.8.2 | Establishing a session.....                                | 266 |
| 11.8.3 | Terminating a session.....                                 | 266 |
| 11.8.4 | Maintaining a session.....                                 | 267 |
| 11.8.5 | TCP connection management.....                             | 267 |
| 12     | DLL mapping protocol machine 3 (DMPM 3).....               | 267 |
|        | Bibliography.....                                          | 268 |

|                                                                                                   |     |
|---------------------------------------------------------------------------------------------------|-----|
| Figure 1 – Attribute table format and terms .....                                                 | 30  |
| Figure 2 – Service request/response parameter .....                                               | 31  |
| Figure 3 – Example of an STD .....                                                                | 35  |
| Figure 4 – Network connection parameters.....                                                     | 57  |
| Figure 5 – Priority/Tick_time bit definition.....                                                 | 60  |
| Figure 6 – Member ID/EX description (WORD).....                                                   | 75  |
| Figure 7 – Transport Class Trigger attribute.....                                                 | 110 |
| Figure 8 – CP2/3_initial_comm_characteristics attribute format .....                              | 113 |
| Figure 9 – Segment type .....                                                                     | 122 |
| Figure 10 – Port segment .....                                                                    | 123 |
| Figure 11 – Logical segment encoding.....                                                         | 125 |
| Figure 12 – Extended network segment.....                                                         | 131 |
| Figure 13 – Symbolic segment encoding .....                                                       | 131 |
| Figure 14 – Encapsulation message .....                                                           | 160 |
| Figure 15 – FixedLengthBitString compact encoding bit placement rules .....                       | 183 |
| Figure 16 – Example compact encoding of a SWORD FixedLengthBitString.....                         | 183 |
| Figure 17 – Example compact encoding of a WORD FixedLengthBitString .....                         | 183 |
| Figure 18 – Example compact encoding of a DWORD FixedLengthBitString .....                        | 183 |
| Figure 19 – Example compact encoding of a LWORD FixedLengthBitString .....                        | 183 |
| Figure 20 – Example 1 of formal encoding of a structure type specification.....                   | 188 |
| Figure 21 – Example 2 of formal encoding of a structure type specification.....                   | 188 |
| Figure 22 – Example 3 of formal encoding of a handle structure type specification .....           | 189 |
| Figure 23 – Example 4 of formal encoding of a handle structure type specification .....           | 189 |
| Figure 24 – Example 5 of abbreviated encoding of a structure type specification.....              | 190 |
| Figure 25 – Example 1 of formal encoding of an array type specification.....                      | 191 |
| Figure 26 – Example 2 of formal encoding of an array type specification.....                      | 192 |
| Figure 27 – Example 1 of abbreviated encoding of an array type specification.....                 | 192 |
| Figure 28 – Example 2 of abbreviated encoding of an array type specification.....                 | 193 |
| Figure 29 – I/O Connection object state transition diagram .....                                  | 194 |
| Figure 30 – Bridged Connection object state transition diagram.....                               | 198 |
| Figure 31 – Explicit Messaging Connection object state transition diagram .....                   | 200 |
| Figure 32 – State transition diagram of UCMM client9.....                                         | 210 |
| Figure 33 – State transition diagram of high–end UCMM server .....                                | 212 |
| Figure 34 – State transition diagram of low–end UCMM server.....                                  | 214 |
| Figure 35 – Sequence diagram for a UCMM with one outstanding message.....                         | 216 |
| Figure 36 – Sequence diagram for a UCMM with multiple outstanding messages.....                   | 217 |
| Figure 37 – TPDU buffer.....                                                                      | 218 |
| Figure 38 – Data flow diagram using a client transport class 0 and server transport class 0 ..... | 221 |
| Figure 39 – Sequence diagram of data transfer using transport class 0.....                        | 221 |
| Figure 40 – Class 0 client STD .....                                                              | 222 |
| Figure 41 – Class 0 server STD .....                                                              | 223 |
| Figure 42 – Data flow diagram using client transport class 1 and server transport class 1 .....   | 224 |

|                                                                                                                                       |     |
|---------------------------------------------------------------------------------------------------------------------------------------|-----|
| Figure 43 – Sequence diagram of data transfer using client transport class 1 and server transport class 1.....                        | 224 |
| Figure 44 – Class 1 client STD.....                                                                                                   | 226 |
| Figure 45 – Class 1 server STD .....                                                                                                  | 228 |
| Figure 46 – Data flow diagram using client transport class 2 and server transport class 2.....                                        | 230 |
| Figure 47 – Diagram of data transfer using client transport class 2 and server transport class 2 without returned data.....           | 231 |
| Figure 48 – Sequence diagram of data transfer using client transport class 2 and server transport class 2 with returned data.....     | 232 |
| Figure 49 – Class 2 client STD.....                                                                                                   | 233 |
| Figure 50 – Class 2 server STD .....                                                                                                  | 235 |
| Figure 51 – Data flow diagram using client transport class 3 and server transport class 3.....                                        | 237 |
| Figure 52 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 without returned data ..... | 238 |
| Figure 53 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 with returned data.....     | 239 |
| Figure 54 – Class 3 client STD.....                                                                                                   | 241 |
| Figure 55 – Class 3 server STD .....                                                                                                  | 244 |
| Figure 56 – Data flow diagram for a link producer and consumer .....                                                                  | 246 |
| Figure 57 – State transition diagram for a link producer .....                                                                        | 250 |
| Figure 58 – State transition diagram for a link consumer.....                                                                         | 251 |
| Figure 59 – DS field in the IP header .....                                                                                           | 264 |
| Figure 60 – IEEE 802.1Q tagged frame .....                                                                                            | 264 |
| Table 1 – Get_Attributes_All response service rules .....                                                                             | 32  |
| Table 2 – Example class level object/service specific response data of Get_Attributes_All .....                                       | 32  |
| Table 3 – Example Get_Attributes_All data array method .....                                                                          | 32  |
| Table 4 – Set_Attributes_All request service rules .....                                                                              | 33  |
| Table 5 – Example Set_Attributes_All attribute ordering method .....                                                                  | 33  |
| Table 6 – Example Set_Attributes_All data array method .....                                                                          | 33  |
| Table 7 – State event matrix format .....                                                                                             | 35  |
| Table 8 – Example state event matrix.....                                                                                             | 36  |
| Table 9 – UCMM_PDU header format .....                                                                                                | 39  |
| Table 10 – UCMM command codes.....                                                                                                    | 39  |
| Table 11 – Transport class 0 header.....                                                                                              | 40  |
| Table 12 – Transport class 1 header.....                                                                                              | 41  |
| Table 13 – Transport class 2 header.....                                                                                              | 41  |
| Table 14 – Transport class 3 header.....                                                                                              | 41  |
| Table 15 – Real-time data header – exclusive owner .....                                                                              | 41  |
| Table 16 – Real-time data header– redundant owner.....                                                                                | 42  |
| Table 17 – Forward_Open request format .....                                                                                          | 46  |
| Table 18 – Forward_Open_Good response format .....                                                                                    | 46  |
| Table 19 – Forward_Open_Bad response format.....                                                                                      | 47  |
| Table 20 – Large_Forward_Open request format.....                                                                                     | 48  |

|                                                                        |    |
|------------------------------------------------------------------------|----|
| Table 21 – Large_Forward_Open_Good response format .....               | 48 |
| Table 22 – Large_Forward_Open_Bad response format.....                 | 49 |
| Table 23 – Forward_Close request format .....                          | 50 |
| Table 24 – Forward_Close_Good response format .....                    | 50 |
| Table 25 – Forward_Close_Bad response format.....                      | 51 |
| Table 26 – Unconnected_Send request format.....                        | 51 |
| Table 27 – Unconnected_Send_Good response format .....                 | 52 |
| Table 28 – Unconnected_Send_Bad response format.....                   | 53 |
| Table 29 – Unconnected_Send request format (modified).....             | 54 |
| Table 30 – Unconnected_Send_Good response format (modified) .....      | 54 |
| Table 31 – Unconnected_Send_Bad response format (modified) .....       | 55 |
| Table 32 – Get_Connection_Data request format.....                     | 55 |
| Table 33 – Get_Connection_Data response format.....                    | 55 |
| Table 34 – Search_Connection_Data request format .....                 | 56 |
| Table 35 – Get_Connection_Owner request format.....                    | 56 |
| Table 36 – Get_Connection_Owner response format.....                   | 57 |
| Table 37 – Time-out multiplier .....                                   | 60 |
| Table 38 – Tick time units.....                                        | 61 |
| Table 39 – Encoded application path ordering.....                      | 65 |
| Table 40 – Transport class, trigger and Is_Server format .....         | 66 |
| Table 41 – MR_Request_Header format.....                               | 66 |
| Table 42 – MR_Response_Header format.....                              | 67 |
| Table 43 – Structure of Get_Attributes_All_ResponsePDU body .....      | 67 |
| Table 44 – Structure of Set_Attributes_All_RequestPDU body .....       | 68 |
| Table 45 – Structure of Get_Attribute_List_RequestPDU body .....       | 68 |
| Table 46 – Structure of Get_Attribute_List_ResponsePDU body .....      | 68 |
| Table 47 – Structure of Set_Attribute_List_RequestPDU body.....        | 68 |
| Table 48 – Structure of Set_Attribute_List_ResponsePDU body.....       | 69 |
| Table 49 – Structure of Reset_RequestPDU body.....                     | 69 |
| Table 50 – Structure of Reset_ResponsePDU body .....                   | 69 |
| Table 51 – Structure of Start_RequestPDU body.....                     | 69 |
| Table 52 – Structure of Start_ResponsePDU body.....                    | 69 |
| Table 53 – Structure of Stop_RequestPDU body.....                      | 70 |
| Table 54 – Structure of Stop_ResponsePDU body .....                    | 70 |
| Table 55 – Structure of Create_RequestPDU body .....                   | 70 |
| Table 56 – Structure of Create_ResponsePDU body.....                   | 70 |
| Table 57 – Structure of Delete_RequestPDU body.....                    | 70 |
| Table 58 – Structure of Delete_ResponsePDU body .....                  | 71 |
| Table 59 – Structure of Get_Attribute_Single_ResponsePDU body .....    | 71 |
| Table 60 – Structure of Set_Attribute_Single_RequestPDU body .....     | 71 |
| Table 61 – Structure of Set_Attribute_Single_ResponsePDU body .....    | 71 |
| Table 62 – Structure of Find_Next_Object_Instance_RequestPDU body..... | 72 |

|                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------|----|
| Table 63 – Structure of Find_Next_Object_Instance_ResponsePDU body .....                                    | 72 |
| Table 64 – Structure of Apply_Attributes_RequestPDU body .....                                              | 72 |
| Table 65 – Structure of Apply_Attributes_ResponsePDU body .....                                             | 72 |
| Table 66 – Structure of Save_RequestPDU body .....                                                          | 72 |
| Table 67 – Structure of Save_ResponsePDU body .....                                                         | 73 |
| Table 68 – Structure of Restore_RequestPDU body .....                                                       | 73 |
| Table 69 – Structure of Restore_ResponsePDU body .....                                                      | 73 |
| Table 70 – Structure of Get_Member_ResponsePDU body .....                                                   | 73 |
| Table 71 – Structure of Set_Member_RequestPDU body .....                                                    | 73 |
| Table 72 – Structure of Set_Member_ResponsePDU body .....                                                   | 74 |
| Table 73 – Structure of Insert_Member_RequestPDU body .....                                                 | 74 |
| Table 74 – Structure of Insert_Member_ResponsePDU body .....                                                | 74 |
| Table 75 – Structure of Remove_Member_ResponsePDU body .....                                                | 74 |
| Table 76 – Common structure of _Member_RequestPDU body (basic format) .....                                 | 75 |
| Table 77 – Common structure of _Member_ResponsePDU body (basic format) .....                                | 75 |
| Table 78 – Common structure of _Member_RequestPDU body (extended format) .....                              | 76 |
| Table 79 – Common structure of _Member_ResponsePDU body (extended format) .....                             | 76 |
| Table 80 – Extended Protocol ID .....                                                                       | 76 |
| Table 81 – Structure of _Member_RequestPDU body (Multiple Sequential Members) .....                         | 77 |
| Table 82 – Structure of _Member_ResponsePDU body (Multiple Sequential Members) .....                        | 77 |
| Table 83 – Structure of _Member_RequestPDU body (International String Selection) .....                      | 78 |
| Table 84 – Structure of _Member_ResponsePDU body (International String Selection) .....                     | 78 |
| Table 85 – Structure of Group_Sync_RequestPDU body .....                                                    | 78 |
| Table 86 – Structure of Group_Sync_ResponsePDU body .....                                                   | 78 |
| Table 87 – Structure of Multiple_Service_Packet_RequestPDU body .....                                       | 79 |
| Table 88 – Structure of Multiple_Service_Packet_ResponsePDU body .....                                      | 79 |
| Table 89 – Identity object class attributes .....                                                           | 80 |
| Table 90 – Identity object instance attributes .....                                                        | 80 |
| Table 91 – Identity object bit definitions for status instance attribute .....                              | 81 |
| Table 92 – Default values for extended device status field (bits 4 to 7) of status instance attribute ..... | 82 |
| Table 93 – Identity object bit definitions for protection mode instance attribute .....                     | 82 |
| Table 94 – Class level object/service specific response data of Get_Attributes_All .....                    | 82 |
| Table 95 – Instance level object/service specific response data of Get_Attributes_All .....                 | 83 |
| Table 96 – Object-specific request parameter for Reset .....                                                | 83 |
| Table 97 – Reset service parameter values .....                                                             | 84 |
| Table 98 – Communication link attributes that shall be preserved .....                                      | 84 |
| Table 99 – Structure of Flash_LEDs_RequestPDU body .....                                                    | 84 |
| Table 100 – Message Router object class attributes .....                                                    | 85 |
| Table 101 – Message Router object instance attributes .....                                                 | 85 |
| Table 102 – Class level object/service specific response data of Get_Attributes_All .....                   | 85 |
| Table 103 – Instance level object/service specific response data of Get_Attributes_All .....                | 86 |
| Table 104 – Structure of Symbolic_Translation_RequestPDU body .....                                         | 86 |

|                                                                                                                      |     |
|----------------------------------------------------------------------------------------------------------------------|-----|
| Table 105 – Structure of Symbolic_Translation_ResponsePDU body .....                                                 | 86  |
| Table 106 – Object specific status for Symbolic_Translation service .....                                            | 86  |
| Table 107 – Assembly object class attributes .....                                                                   | 87  |
| Table 108 – Assembly object instance attributes .....                                                                | 87  |
| Table 109 – Assembly Instance ID ranges .....                                                                        | 88  |
| Table 110 – Acknowledge Handler object class attributes .....                                                        | 88  |
| Table 111 – Acknowledge Handler object instance attributes .....                                                     | 89  |
| Table 112 – Structure of Add_AckData_Path_RequestPDU body .....                                                      | 89  |
| Table 113 – Structure of Remove_AckData_Path_RequestPDU body .....                                                   | 89  |
| Table 114 – Time Sync object class attributes .....                                                                  | 90  |
| Table 115 – Time Sync object instance attributes .....                                                               | 90  |
| Table 116 – ClockIdentity encoding for different network implementations .....                                       | 94  |
| Table 117 – ClockClass values .....                                                                                  | 94  |
| Table 118 – TimeAccuracy values .....                                                                                | 94  |
| Table 119 – TimePropertyFlags bit values .....                                                                       | 95  |
| Table 120 – TimeSource values .....                                                                                  | 95  |
| Table 121 – Types of Clock .....                                                                                     | 96  |
| Table 122 – Network protocol to PortPhysicalAddressInfo mapping .....                                                | 96  |
| Table 123 – Parameter object class attributes .....                                                                  | 97  |
| Table 124 – Parameter Class Descriptor bit values .....                                                              | 97  |
| Table 125 – Parameter object instance attributes .....                                                               | 98  |
| Table 126 – Semantics of Descriptor Instance attribute .....                                                         | 99  |
| Table 127 – Descriptor Scaling bits usage .....                                                                      | 99  |
| Table 128 – Minimum and Maximum Value semantics .....                                                                | 100 |
| Table 129 – Scaling Formula attributes .....                                                                         | 101 |
| Table 130 – Scaling links .....                                                                                      | 101 |
| Table 131 – Class level object/service specific response data of Get_Attributes_All .....                            | 102 |
| Table 132 – Instance level object/service specific response data of Get_Attributes_All (Parameter object stub) ..... | 102 |
| Table 133 – Instance level object/service specific response data of Get_Attributes_All (full Parameter object) ..... | 103 |
| Table 134 – Structure of Get_Enum_String_RequestPDU body .....                                                       | 103 |
| Table 135 – Structure of Get_Enum_String_ResponsePDU body .....                                                      | 104 |
| Table 136 – Enumerated strings Type versus Parameter data type .....                                                 | 104 |
| Table 137 – Connection Manager object class attributes .....                                                         | 104 |
| Table 138 – Connection Manager object instance attributes .....                                                      | 105 |
| Table 139 – Class level object/service specific response data of Get_Attributes_All .....                            | 106 |
| Table 140 – Instance level object/service specific response data of Get_Attributes_All .....                         | 106 |
| Table 141 – Instance level object/service specific request data of Set_Attributes_All .....                          | 107 |
| Table 142 – Connection object class attributes .....                                                                 | 107 |
| Table 143 – Connection object instance attributes .....                                                              | 107 |
| Table 144 – Values assigned to the state attribute .....                                                             | 108 |
| Table 145 – Values assigned to the instance_type attribute .....                                                     | 109 |

|                                                                                 |     |
|---------------------------------------------------------------------------------|-----|
| Table 146 – Possible values within Direction Bit.....                           | 110 |
| Table 147 – Possible values within Production Trigger Bits .....                | 110 |
| Table 148 – Possible values within Transport Class Bits.....                    | 111 |
| Table 149 – TransportClass_Trigger attribute values summary .....               | 112 |
| Table 150 – Transport Class 0 client behavior summary.....                      | 112 |
| Table 151 – Transport Class 1, 2 and 3 client behavior summary.....             | 112 |
| Table 152 – Values defined for the CP2/3_produced_connection_id attribute.....  | 113 |
| Table 153 – Values defined for the CP2/3_consumed_connection_id attribute ..... | 113 |
| Table 154 – Values for the Initial Production Characteristics nibble.....       | 114 |
| Table 155 – Values for the Initial Consumption Characteristics nibble .....     | 115 |
| Table 156 – Values for the watchdog_timeout_action .....                        | 118 |
| Table 157 – Structure of Connection_Bind_RequestPDU body .....                  | 119 |
| Table 158 – Object specific status for Connection_Bind service .....            | 120 |
| Table 159 – Structure of Producing_Application_Lookup_RequestPDU body.....      | 120 |
| Table 160 – Structure of Producing_Application_Lookup_ResponsePDU body .....    | 120 |
| Table 161 – Producing_Application_Lookup Service status codes .....             | 120 |
| Table 162 – Possible port segment examples .....                                | 123 |
| Table 163 – TCP/IP link address examples .....                                  | 124 |
| Table 164 – Extended Logical Type.....                                          | 125 |
| Table 165 – Electronic key segment format.....                                  | 127 |
| Table 166 – Logical segments examples .....                                     | 128 |
| Table 167 – Network segments.....                                               | 128 |
| Table 168 – Extended subtype definitions .....                                  | 131 |
| Table 169 – Symbolic segment examples.....                                      | 132 |
| Table 170 – Data segment .....                                                  | 132 |
| Table 171 – ANSI_Extended_Symbol segment.....                                   | 133 |
| Table 172 – Addressing categories .....                                         | 135 |
| Table 173 – Class code ID ranges .....                                          | 135 |
| Table 174 – Attribute ID ranges .....                                           | 136 |
| Table 175 – Service code ranges.....                                            | 136 |
| Table 176 – Class codes.....                                                    | 137 |
| Table 177 – Reserved class attributes for all object class definitions .....    | 138 |
| Table 178 – Common services list .....                                          | 138 |
| Table 179 – Identity object specific services list.....                         | 139 |
| Table 180 – Message Router object specific services list.....                   | 139 |
| Table 181 – Acknowledge Handler object specific services list.....              | 139 |
| Table 182 – Parameter object specific services list .....                       | 139 |
| Table 183 – Services specific to Connection Manager .....                       | 140 |
| Table 184 – Services specific to Connection object.....                         | 140 |
| Table 185 – Device type numbering.....                                          | 141 |
| Table 186 – Connection Manager service request error codes .....                | 142 |
| Table 187 – General status codes .....                                          | 151 |

|                                                                                    |     |
|------------------------------------------------------------------------------------|-----|
| Table 188 – Extended status code for a general status of "Key Failure in path..... | 153 |
| Table 189 – Identity object status codes .....                                     | 153 |
| Table 190 – Encapsulation header .....                                             | 161 |
| Table 191 – Encapsulation command codes.....                                       | 161 |
| Table 192 – Encapsulation status codes .....                                       | 162 |
| Table 193 – Nop request encapsulation header.....                                  | 163 |
| Table 194 – RegisterSession request encapsulation header .....                     | 164 |
| Table 195 – RegisterSession request data portion .....                             | 164 |
| Table 196 – RegisterSession reply encapsulation header .....                       | 164 |
| Table 197 – RegisterSession reply data portion (successful) .....                  | 165 |
| Table 198 – UnRegisterSession request encapsulation header .....                   | 166 |
| Table 199 – ListServices request encapsulation header.....                         | 166 |
| Table 200 – ListServices reply encapsulation header .....                          | 167 |
| Table 201 – ListServices reply data portion (successful) .....                     | 167 |
| Table 202 – Communications capability flags .....                                  | 168 |
| Table 203 – ListIdentity request encapsulation header .....                        | 168 |
| Table 204 – ListIdentity reply encapsulation header .....                          | 169 |
| Table 205 – ListIdentity reply data portion (successful) .....                     | 169 |
| Table 206 – CPF 2 identity item .....                                              | 170 |
| Table 207 – ListInterfaces request encapsulation header.....                       | 171 |
| Table 208 – ListInterfaces reply encapsulation header .....                        | 171 |
| Table 209 – SendRRData request encapsulation header.....                           | 172 |
| Table 210 – SendRRData request data portion .....                                  | 172 |
| Table 211 – SendRRData reply encapsulation header .....                            | 173 |
| Table 212 – SendUnitData request encapsulation header.....                         | 173 |
| Table 213 – SendUnitData request data portion .....                                | 174 |
| Table 214 – Common packet format .....                                             | 174 |
| Table 215 – CPF item format.....                                                   | 174 |
| Table 216 – Item Type ID numbers .....                                             | 174 |
| Table 217 – Null address item .....                                                | 175 |
| Table 218 – Connected address item.....                                            | 176 |
| Table 219 – Sequenced address item.....                                            | 176 |
| Table 220 – Unconnected data item.....                                             | 176 |
| Table 221 – Connected data item.....                                               | 176 |
| Table 222 – Sockaddr info items .....                                              | 177 |
| Table 223 – Usage of CPF items.....                                                | 178 |
| Table 224 – BOOLEAN encoding.....                                                  | 179 |
| Table 225 – Example compact encoding of a BOOL value.....                          | 179 |
| Table 226 – Encoding of SignedInteger values .....                                 | 179 |
| Table 227 – Example compact encoding of a SignedInteger value .....                | 179 |
| Table 228 – UnsignedInteger values .....                                           | 180 |
| Table 229 – Example compact encoding of an UnsignedInteger.....                    | 180 |

|                                                                                       |     |
|---------------------------------------------------------------------------------------|-----|
| Table 230 – FixedLengthReal values .....                                              | 180 |
| Table 231 – Example compact encoding of a REAL value .....                            | 180 |
| Table 232 – Example compact encoding of a LREAL value .....                           | 180 |
| Table 233 – FixedLengthReal values .....                                              | 181 |
| Table 234 – STRING value .....                                                        | 181 |
| Table 235 – STRING2 value .....                                                       | 181 |
| Table 236 – STRINGN value .....                                                       | 181 |
| Table 237 – SHORT_STRING value .....                                                  | 182 |
| Table 238 – Example compact encoding of a STRING value .....                          | 182 |
| Table 239 – Example compact encoding of STRING2 value .....                           | 182 |
| Table 240 – SHORT_STRING type .....                                                   | 182 |
| Table 241 – Example compact encoding of a single dimensional ARRAY .....              | 184 |
| Table 242 – Example compact encoding of a multi-dimensional ARRAY .....               | 184 |
| Table 243 – Example compact encoding of a STRUCTURE .....                             | 185 |
| Table 244 – Identification codes and descriptions of elementary data types .....      | 186 |
| Table 245 – Identification codes and descriptions of constructed data types .....     | 187 |
| Table 246 – Formal structure encoding definition .....                                | 188 |
| Table 247 – Formal structure with handles encoding definition .....                   | 189 |
| Table 248 – Abbreviated structure encoding definition .....                           | 190 |
| Table 249 – Formal array encoding definition .....                                    | 190 |
| Table 250 – Abbreviated array encoding definition .....                               | 192 |
| Table 251 – I/O Connection state event matrix .....                                   | 194 |
| Table 252 – Bridged Connection state event matrix .....                               | 198 |
| Table 253 – Explicit Messaging Connection state event matrix .....                    | 200 |
| Table 254 – Primitives issued by FAL user to FSPM .....                               | 203 |
| Table 255 – Primitives issued by FAL user to FSPM .....                               | 203 |
| Table 256 – Primitives issued by FSPM to FAL user .....                               | 206 |
| Table 257 – Parameters used with primitives exchanged between FAL user and FSPM ..... | 207 |
| Table 258 – Primitives issued by FSPM to ARPM .....                                   | 209 |
| Table 259 – Primitives issued by ARPM to FSPM .....                                   | 209 |
| Table 260 – Parameters used with primitives exchanged between FSPM and ARPM .....     | 209 |
| Table 261 – UCMM client states .....                                                  | 210 |
| Table 262 – State event matrix of UCMM client .....                                   | 211 |
| Table 263 – High-end UCMM server states .....                                         | 212 |
| Table 264 – State event matrix of high-end UCMM server .....                          | 213 |
| Table 265 – Low-end UCMM server states .....                                          | 214 |
| Table 266 – State event matrix of low-end UCMM server .....                           | 215 |
| Table 267 – Notification .....                                                        | 218 |
| Table 268 – Transport classes .....                                                   | 219 |
| Table 269 – Primitives issued by FSPM to ARPM .....                                   | 219 |
| Table 270 – Primitives issued by ARPM to FSPM .....                                   | 220 |
| Table 271 – Parameters used with primitives exchanged between FSPM and ARPM .....     | 220 |

|                                                                                       |     |
|---------------------------------------------------------------------------------------|-----|
| Table 272 – Class 0 transport client states .....                                     | 221 |
| Table 273 – Class 0 client SEM.....                                                   | 222 |
| Table 274 – Class 0 transport server states .....                                     | 223 |
| Table 275 – Class 0 server SEM .....                                                  | 223 |
| Table 276 – Class 1 transport client states .....                                     | 226 |
| Table 277 – Class 1 client SEM.....                                                   | 226 |
| Table 278 – Class 1 transport server states .....                                     | 227 |
| Table 279 – Class 1 server SEM .....                                                  | 228 |
| Table 280 – Class 2 transport client states .....                                     | 233 |
| Table 281 – Class 2 client SEM.....                                                   | 234 |
| Table 282 – Class 2 transport server states .....                                     | 235 |
| Table 283 – Class 2 server SEM .....                                                  | 236 |
| Table 284 – Class 3 transport client states .....                                     | 240 |
| Table 285 – Class 3 client SEM.....                                                   | 241 |
| Table 286 – Class 3 transport server states .....                                     | 243 |
| Table 287 – Class 3 server SEM .....                                                  | 245 |
| Table 288 – Primitives issued by ARPM to DMPM .....                                   | 247 |
| Table 289 – Primitives issued by DMPM to ARPM .....                                   | 247 |
| Table 290 – Parameters used with primitives exchanged between ARPM and DMPM.....      | 247 |
| Table 291 – Primitives exchanged between data-link layer and DMPM .....               | 248 |
| Table 292 – Parameters used with primitives exchanged between DMPM and Data-link..... | 248 |
| Table 293 – Selection of connection ID.....                                           | 249 |
| Table 294 – Link producer states .....                                                | 250 |
| Table 295 – State event matrix of link producer .....                                 | 250 |
| Table 296 – Link consumer states.....                                                 | 250 |
| Table 297 – State event matrix of link consumer .....                                 | 251 |
| Table 298 – UCMM request.....                                                         | 252 |
| Table 299 – UCMM reply.....                                                           | 253 |
| Table 300 – Network Connection ID selection .....                                     | 254 |
| Table 301 – Sockaddr Info usage.....                                                  | 256 |
| Table 302 – Example multicast assignments .....                                       | 259 |
| Table 303 – UDP data format for class 0 and class 1 .....                             | 259 |
| Table 304 – Transport class 2 and class 3 connected data .....                        | 261 |
| Table 305 – Default DSCP and IEEE 802.1D mapping.....                                 | 265 |

## INTRODUCTION

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

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

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

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

## INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

### Part 6-2: Application layer protocol specification – Type 2 elements

## 1 Scope

### 1.1 General

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

This part of IEC 61158 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 2 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 International Standard specifies interactions between remote applications and defines the externally visible behavior provided by the Type 2 fieldbus application layer in terms of

- a) the formal abstract syntax defining the application layer protocol data units conveyed between communicating application entities;
- b) the transfer syntax defining encoding rules that are applied to the application layer protocol data units;
- c) the application context state machine defining the application service behavior visible between communicating application entities;
- d) the application relationship state machines defining the communication behavior visible between communicating application entities.

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

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

This document specifies the protocol of the Type 2 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI application layer structure (ISO/IEC 9545).

### 1.2 Specifications

The principal objective of this document is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-2.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols.

### **1.3 Conformance**

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

## **2 Normative references**

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

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

IEC 61158-1:2019, Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series

IEC 61158-3-2:2014, Industrial communication networks – Fieldbus specifications – Part 3-2: Data-link layer service definition – Type 2 elements  
IEC 61158-3-2:2014/AMD1:2019

IEC 61158-4-2:2019, Industrial communication networks – Fieldbus specifications – Part 4-2: Data-link layer protocol specification – Type 2 elements

IEC 61158-5-2:2019, Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition – Type 2 elements

IEC 61588:2009, Precision clock synchronization protocol for networked measurement and control systems

IEC 61784-3-2, Industrial communication networks – Profiles – Part 3-2: Functional safety fieldbuses – Additional specifications for CPF 2

IEC 61800-7-202, Adjustable speed electrical power drive systems – Part 7-202: Generic interface and use of profiles for power drive systems – Profile type 2 specification

IEC 62026-3:2014, Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) – Part 3: DeviceNet

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

ISO/IEC/IEEE 8802-3, Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Standard for Ethernet

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

ISO/IEC 8825-1, Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

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

ISO/IEC 10646, Information technology – Universal Coded Character Set (UCS)

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

ISO 639-2, Codes for the representation of names of languages – Part 2: Alpha-3 code

ISO 11898:1993<sup>1</sup>, Road vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication

IEEE Std 802.1D-2004, IEEE standard for local and metropolitan area networks – Media Access Control (MAC) bridges, available at <<http://www.ieee.org>> [viewed 2018-09-04]

IEEE Std 802.1Q-2005<sup>2</sup>, IEEE standard for local and metropolitan area networks – Virtual bridged local area networks, available at <<http://www.ieee.org>> [viewed 2018-09-04]

IEEE Std 802.3-2015, IEEE Standard for Ethernet, available at <<http://www.ieee.org>> [viewed 2018-09-04]

IETF RFC 791, Internet Protocol, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1035, Domain Names – Implementation and Specification, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1112, Host Extensions for IP Multicasting, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1117, Internet Numbers, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1122, Requirements for Internet Hosts – Communication Layers, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1759, Printer MIB, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2236, Internet Group Management Protocol, *Version 2*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2475, An Architecture for Differentiated Services, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2597, Assured Forwarding PHB Group, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2873, TCP Processing of the IPv4 Precedence Field, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 3140, Per Hop Behavior Identification Codes, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 3246, An Expedited Forwarding PHB (Per-Hop Behavior), available at <<http://www.ietf.org>> [viewed 2018-09-04]

---

<sup>1</sup> A newer edition of this document has been published, but only the cited edition applies.

<sup>2</sup> A newer edition of this document has been published, but only the cited edition applies.

IETF RFC 3376, Internet Group Management Protocol, Version 3, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 4594, Configuration Guidelines for DiffServ Service Classes, available at <<http://www.ietf.org>> [viewed 2018-09-04]

### 3 Terms, definitions, symbols, abbreviated terms and conventions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

#### 3.1 Terms and definitions from other ISO/IEC standards

##### 3.1.1 Terms and definitions from ISO/IEC 7498-1

- abstract syntax
- application entity
- application process
- application protocol data unit
- application service element
- application entity invocation
- application process invocation
- application transaction
- presentation context
- real open system
- transfer syntax

##### 3.1.2 Terms and definitions from ISO/IEC 9545

- application-association
- application-context
- application context name
- application-entity-invocation
- application-entity-type
- application-process-invocation
- application-process-type
- application-service-element
- application control service element

##### 3.1.3 Terms and definitions from ISO/IEC 8824-1

- object identifier

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

#### **3.1.4 Terms and definitions from ISO/IEC 8825-1**

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

#### **3.2 Terms and definitions from IEC 61158-5-2**

- a) application relationship
- b) client
- c) peer
- d) server

#### **3.3 Additional terms and definitions**

##### **3.3.1**

##### **allocate**

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

### 3.3.2

#### **application**

function or data structure for which data is consumed or produced

### 3.3.3

#### **application objects**

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

### 3.3.4

#### **attribute**

description of an externally visible characteristic or feature of an object

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

### 3.3.5

#### **behavior**

indication of how an object responds to particular events

### 3.3.6

#### **Best Master Clock Algorithm BMCA**

algorithm performed by each node to determine the clock that will become the master clock on a subnet and the grandmaster clock for the domain

Note 1 to entry: The algorithm primarily compares priority1, clock quality, priority2, and source identity to determine the best master among available candidates.

### 3.3.7

#### **boundary clock**

clock that has multiple Precision Time Protocol (PTP) ports in a domain and maintains the timescale used in the domain

Note 1 to entry: It may serve as the source of time, i.e., be a master clock, and may synchronize to another clock, i.e., be a slave clock.

[SOURCE: IEC 61588:2009, 3.1.3, modified – second sentence changed to a Note]

### 3.3.8

#### **called**

service user or a service provider that receives an indication primitive or a request APDU

### 3.3.9

#### **calling**

service user or a service provider that initiates a request primitive or a request APDU

### 3.3.10

#### **class**

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

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

### 3.3.11

#### **class attribute**

attribute that is shared by all objects within the same class

### 3.3.12

#### **class code**

unique identifier assigned to each object class

### 3.3.13

#### **class specific service**

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

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

### 3.3.14

#### **client**

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

### 3.3.15

#### **clock**

node participating in the Precision Time Protocol (PTP) that is capable of providing a measurement of the passage of time since a defined epoch

Note 1 to entry: There are three types of clocks in IEC 61588:2009, boundary, transparent and ordinary clocks.

[SOURCE: IEC 61588:2009, 3.1.4, modified – different Note]

### 3.3.16

#### **communication objects**

components that manage and provide a run time exchange of messages across the network

EXAMPLES Connection Manager object, Unconnected Message Manager (UCMM) object, and Message Router object.

### 3.3.17

#### **connection**

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

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

### 3.3.18

#### **connection ID**

##### **CID**

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

### 3.3.19

#### **connection path**

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

### 3.3.20

#### **connection point**

buffer which is represented as a subinstance of an Assembly object

### 3.3.21

#### **consume**

act of receiving data from a producer

### 3.3.22

#### **consumer**

node or sink that is receiving data from a producer

### 3.3.23

#### **consuming application**

application that consumes data

### 3.3.24

#### **cyclic**

repetitive in a regular manner

### 3.3.25

#### **device**

physical hardware connected to the link

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

### 3.3.26

#### **device profile**

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

### 3.3.27

#### **domain**

logical grouping of clocks that synchronize to each other using the protocol, but that are not necessarily synchronized to clocks in another domain

[SOURCE: IEC 61588:2009, 3.1.7]

### 3.3.28

#### **end node**

producing or consuming node

### 3.3.29

#### **end point**

one of the communicating entities involved in a connection

### 3.3.30

#### **epoch**

origin of a time scale

[SOURCE: IEC 61588:2009, 3.1.9]

### 3.3.31

#### **error**

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

### 3.3.32

#### **frame**

denigrated synonym for DLPDU

### 3.3.33

#### **grandmaster clock**

within a domain, clock that is the ultimate source of time for clock synchronization using the PTP protocol

[SOURCE: IEC 61588:2009, 3.1.13]

### 3.3.34

#### **instance**

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

EXAMPLE California is an instance of the object class state.

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

### 3.3.35

#### **instance attribute**

attribute whose value is unique to an object instance and whose definition is shared by all instances of an object

### 3.3.36

#### **instantiated**

object that has been created in a device

### 3.3.37

#### **interoperability**

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

### 3.3.38

#### **Keeper**

object responsible for distributing link configuration data to all nodes on the link

### 3.3.39

#### **little endian**

model of memory organization which stores the least significant octet at the lowest address, or for transfer, which transfers the lowest order octet first

Note 1 to entry: Native Type 2 data types are sent in little endian order.

### 3.3.40

#### **Lpacket**

Link packet

piece of application information that contains a size, control octet, tag, and link data

Note 1 to entry: Peer data-link layers use Lpackets to send and receive service data units from higher layers in the OSI stack.

### 3.3.41

#### **management information**

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

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

### 3.3.42

#### **master clock**

in the context of a single Precision Time Protocol (PTP) communication path, clock that is the source of time to which all other clocks on that path synchronize

[SOURCE: IEC 61588:2009, 3.1.17]

### 3.3.43

#### **member**

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

#### 3.3.44

##### **Message Router**

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

#### 3.3.45

##### **multipoint connection**

connection from one node to many

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

#### 3.3.46

##### **network**

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

#### 3.3.47

##### **object**

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

#### 3.3.48

##### **object specific service**

service unique to the object class which defines it

#### 3.3.49

##### **ordinary clock**

clock that has a single Precision Time Protocol (PTP) port in a domain and maintains the timescale used in the domain

Note 1 to entry: It may serve as a source of time, i.e., be a master clock, or may synchronize to another clock, i.e., be a slave clock.

[SOURCE: IEC 61588:2009, 3.1.22, modified – second sentence changed to a Note]

#### 3.3.50

##### **originator**

client responsible for establishing a connection path to the target

#### 3.3.51

##### **parent clock**

master clock to which a clock is synchronized

[SOURCE: IEC 61588:2009, 3.1.23]

#### 3.3.52

##### **point-to-point connection**

connection that exists between exactly two application objects

#### 3.3.53

##### **Precision Time Protocol**

##### **PTP**

protocol defined by IEC 61588:2009

Note 1 to entry: As an adjective, it indicates that the modified noun is specified in or interpreted in the context of IEC 61588:2009.

[SOURCE: IEC 61588:2009, 3.1.28, modified – second sentence changed to a Note]

**3.3.54**

**produce**

act of sending data to be received by a consumer

**3.3.55**

**producer**

node that is responsible for sending data

**3.3.56**

**PTP message**

one of the message types defined in IEC 61588:2009

[SOURCE: IEC 61588:2009, 3.1.33]

**3.3.57**

**PTP port**

logical access point of a clock for PTP communications to the communications network

[SOURCE: IEC 61588:2009, 3.1.35]

**3.3.58**

**receiving**

service user that receives a confirmed primitive or an unconfirmed primitive, or a service provider that receives a confirmed APDU or an unconfirmed APDU

**3.3.59**

**resource**

processing or information capability of a subsystem

**3.3.60**

**sending**

service user that sends a confirmed primitive or an unconfirmed primitive, or a service provider that sends a confirmed APDU or an unconfirmed APDU

**3.3.61**

**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

**3.3.62**

**service**

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

**3.3.63**

**synchronized clocks**

(to a specified uncertainty) two clocks which have the same epoch and for which measurements of the time of a single event at an arbitrary time differ by no more than the specified uncertainty

[SOURCE: IEC 61588:2009, 3.1.40, modified – reworded]

**3.3.64**

**system time**

absolute time value as defined by CPF2 time synchronization in the context of a distributed time system where all devices have a local clock that is synchronized with a common master clock

Note 1 to entry: In the context of CPF2, System Time is a 64-bit integer value in units of nanoseconds with a value of 0 corresponding to the date 1970-01-01.

### 3.3.65

#### **target**

end-node to which a connection is established

### 3.3.66

#### **temporary node**

transient node

### 3.3.67

#### **transaction id**

field within a UCMM header that matches a response with the associated request

### 3.3.68

#### **transparent clock**

device that measures the time taken for a Precision Time Protocol (PTP) event message to transit the device and provides this information to clocks receiving this PTP event message

[SOURCE: IEC 61588:2009, 3.1.46, modified – truncated]

### 3.3.69

#### **unconnected message manager**

##### **UCMM**

component within a node that transmits and receives unconnected explicit messages and sends them directly to the Message Router object

### 3.3.70

#### **unconnected service**

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

### 3.3.71

#### **vendor ID**

identification of each product manufacturer/vendor by a unique number

Note 1 to entry: Vendor IDs are assigned by ODVA, Inc.

## 3.4 Abbreviated terms and symbols

|              |                                                                       |
|--------------|-----------------------------------------------------------------------|
| <b>ASCII</b> | American Standard Code for Information Interchange                    |
| <b>CID</b>   | connection ID                                                         |
| <b>DLL</b>   | data-link layer                                                       |
| <b>DSCP</b>  | DiffServ Codepoint                                                    |
| <b>IGMP</b>  | Internet Group Management Protocol (see IETF RFC 1112, IETF RFC 2236) |
| <b>IP</b>    | Internet Protocol (see IETF RFC 791)                                  |
| <b>IPv4</b>  | Internet Protocol version 4 (see IETF RFC 791)                        |
| <b>IPv6</b>  | Internet Protocol version 6 (see IETF RFC 791)                        |
| <b>OSI</b>   | open systems interconnection (see ISO/IEC 7498-1)                     |
| <b>PDU</b>   | protocol data unit                                                    |
| <b>PHB</b>   | per-hop behavior                                                      |
| <b>PTP</b>   | Precision Time Protocol (see IEC 61588:2009)                          |
| <b>QoS</b>   | quality of service                                                    |
| <b>Rcv</b>   | receive                                                               |

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <b>Rx</b>         | receive                                                    |
| <b>SDU</b>        | service data unit                                          |
| <b>SEM</b>        | state event matrix                                         |
| <b>STD</b>        | state transition diagram, used to describe object behavior |
| <b>TCP</b>        | Terminal Control Protocol (see IETF RFC 793)               |
| <b>ToS</b>        | type of service                                            |
| <b>TPDU</b>       | transport protocol data unit                               |
| <b>Tx</b>         | transmit                                                   |
| <b>UDP</b>        | User Datagram Protocol (see IETF RFC 768)                  |
| <b>Xmit</b>       | transmit                                                   |
| <b>CM_API</b>     | actual packet interval                                     |
| <b>O2T or O⇒T</b> | originator to target (connection parameters)               |
| <b>CM_RPI</b>     | requested packet interval                                  |
| <b>TUI</b>        | table unique identifier                                    |
| <b>T2O or T⇒O</b> | target to originator (connection parameters)               |

### 3.5 Conventions

#### 3.5.1 General concept

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of three parts: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5-2. The protocol specification for each of the ASEs is defined in this document.

The class definitions define the attributes of the classes supported by each ASE. The attributes are accessible from instances of the class using the Management ASE services specified in IEC 61158-5-2. The service specification defines the services that are provided by the ASE.

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

Bold font is used in this document to highlight parameter names or important requirement elements from surrounding text. Italic font is also used in Tables and Notes for that purpose for better visibility.

#### 3.5.2 Attribute specification

Attributes are defined in an Attribute Table using the format and terms defined in Figure 1.

| Attribute ID | Name | Data type | Semantics of values |
|--------------|------|-----------|---------------------|
|--------------|------|-----------|---------------------|

**Figure 1 – Attribute table format and terms**

The **Attribute ID** shall be a unique integer identification value assigned to an attribute. The valid ranges for Attribute IDs shall be as specified in 0. The Attribute ID shall identify the particular attribute being accessed.

**Name** shall refer to the name assigned to the attribute. An attribute may contain sub-elements, which may also have names, as is the case with the **STRUCT** of data type. The attribute name shall be the name which appears first, or at the top row in the name column (the row that contains the Attribute ID).

Every attribute shall be assigned a **Data type** which shall be either an elementary or a derived data type. The data types specified for the defined attributes shall be used in all implementations.

NOTE The elementary data types are defined in IEC 61158-5-2.

**Semantics of values** shall specify the meaning of the value(s) of the attribute. If this information needs more room than can fit in the table it will immediately follow the Class Attribute table. Included in the Class Attribute table will be an appropriate reference to this information.

If a Class Attribute is optional, then a default value or a special case processing method shall be defined such that the Client (Requester) can process the error message that occurs when accessing those objects that choose not to implement the class attribute.

### 3.5.3 Common services

#### 3.5.3.1 Service\_PDU definitions

Each service has unique parameters for request and response. The service request and response parameters shall be defined using service Request/Response parameter tables as defined in Figure 2.

| Name | Type | Semantics of values |
|------|------|---------------------|
|------|------|---------------------|

Figure 2 – Service request/response parameter

**Name** shall refer to the name given to the service request/response parameter.

**Type** shall specify the data type of the service request/response parameter.

**Semantics of values** shall specify the meaning of the values of the service request/response parameter, e.g. “the value is counts of microseconds.”

| Rule number | Rule                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1           | <p>If the definition of the Get_Attributes_All response includes optional attributes, then default values shall be specified in the response description. Optional attributes at the end of the list that are not implemented may be omitted in the response data. Optional attributes in the middle of the list that are not implemented shall be included in the response data, and set to specified default values.</p> <p>If any of the following optional attributes are included in an object specification, but not supported in the implementation of the object, then the following shall be adhered to:</p> <ul style="list-style-type: none"> <li>– if the class attribute “Optional attribute list” is not supported, the default value of zero shall be inserted into the response array and no optional attribute numbers shall follow;</li> <li>– if the class attribute “optional service list” is not supported, the default value of zero shall be inserted into the response array and no optional service numbers shall follow.</li> </ul> |
| 2           | If new attributes are added to an existing object, those attributes shall be added to the end of the response attribute list or data array to ensure compatibility with different object revisions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3           | Whichever method is used to specify the response, it shall be done in such a way as to be unambiguous, including rules to deal with variable length fields and padding.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 4           | The Get_Attributes_All response for objects specified in this document shall include only the open attributes; it shall not include any vendor specific attributes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

#### 3.5.3.2 Get\_Attributes\_All response

##### 3.5.3.2.1 General definition

When the Get\_Attributes\_All common service is included in the list of supported System/Object Management services for an object class, then the **Get\_Attributes\_All** response shall be detailed for this class: the sequence or order of the data returned in the Service\_ResponsePDU shall be specified. There are three ways in which the Get\_Attributes\_All Service\_ResponsePDU may be specified:

- list the ordering of the attributes in the response message;
- specify the actual data array of the response message;
- combine the above two methods such that the actual data array also contains the attribute numbers.

Whichever method is used, the rules specified in Table 1 shall be adhered to when specifying the Get\_Attributes\_All Service\_ResponsePDU of an object class for both the Class Attributes and the Instance Attributes.

**Table 1 – Get\_Attributes\_All response service rules**

Table 2 is an example of the attribute ordering method of specifying the service data portion of a Get\_Attributes\_All response for class level attributes of an object which supports **optional gettable** class attributes 1, 2, 3 and 4.

**Table 2 – Example class level object/service specific response data of Get\_Attributes\_All**

| Class attribute ID | Attribute name and default value                       |
|--------------------|--------------------------------------------------------|
| 1                  | Revision, default = 0x0001                             |
| 2                  | Max Instance, default = 0x0000                         |
| 3                  | Number of Instances, default = 0x0000                  |
| 4                  | Attribute list, number of attributes, default = 0x0000 |

Table 3 is an example of the data array method of specifying the service data portion of a Get\_Attributes\_All response for class level attributes of an object which supports **optional gettable** class attributes 1, 2, 3 and 4.

**Table 3 – Example Get\_Attributes\_All data array method**

| Octet  | Bit 7                                                                  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|------------------------------------------------------------------------|-------|-------|-------|-------|-------|-------|-------|
| 0      | Revision (low octet) Default = 1                                       |       |       |       |       |       |       |       |
| 1      | Revision (high octet) Default = 0                                      |       |       |       |       |       |       |       |
| 2      | Max Instance (low octet) Default = 0                                   |       |       |       |       |       |       |       |
| 3      | Max Instance (high octet) Default = 0                                  |       |       |       |       |       |       |       |
| 4      | Number of Instances (low octet) Default = 0                            |       |       |       |       |       |       |       |
| 5      | Number of Instances (high octet) Default = 0                           |       |       |       |       |       |       |       |
| 6      | Optional Attribute List: number of attributes (low octet) Default = 0  |       |       |       |       |       |       |       |
| 7      | Optional Attribute List: number of attributes (high octet) Default = 0 |       |       |       |       |       |       |       |
| 8      | Optional Attribute List: optional attribute #1 (low octet)             |       |       |       |       |       |       |       |
| 9      | Optional Attribute List: optional attribute #1 (high octet)            |       |       |       |       |       |       |       |
| 2n + 6 | Optional Attribute List: optional attribute #n (low octet)             |       |       |       |       |       |       |       |
| 2n + 7 | Optional Attribute List: optional attribute #n (high octet)            |       |       |       |       |       |       |       |

### 3.5.3.2.2 Revisions

The defined Get\_Attributes\_All response for an object may increase in size with each revision of the object; however, to insure interoperability, the format of the first part of the response shall remain parseable by a client of the older revision of the object. Clients (Requesters) need not make use of this compatibility requirement.

NOTE The Revision class attribute is not the revision of an implementation (which is reflected in the Identity object, Minor/Major revision status bits), but the revision of the class definition.

### 3.5.3.3 Set\_Attributes\_All request

#### 3.5.3.3.1 General definition

When the Set\_Attributes\_All common service is included in the list of supported common services, then the format of the **Set\_Attributes\_All request** shall be included in the object specification. The sequence or order of the data supplied in the Service Data portion of the request shall be specified. There are three ways in which the Set\_Attributes\_All request may be specified:

- list the order of the attributes in the request message;
- specify the actual data array of the request message;
- combine the above two methods such that the actual data array also contains the attribute numbers.

Whichever method is used, the rules specified in Table 4 shall be adhered to when specifying an object's Set\_Attributes\_All request in the object specification for both the Class Attributes and the Instance Attributes.

**Table 4 – Set\_Attributes\_All request service rules**

| Rule number | Rule                                                                                                                                                                                                                            |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1           | An object shall support the Set_Attributes_All service only if all settable attributes shown in the Set_Attributes_All request are implemented as settable.                                                                     |
| 2           | Default values shall be specified for all optional attributes that are not implemented. If an implementation does not support an optional attribute, it shall accept the specified default value for the unsupported attribute. |
| 3           | If new settable attributes are added to an existing object, those attributes shall be added to the end of the request attribute list or data array.                                                                             |
| 4           | Whichever method is used to specify the response, it shall be done in such a way as to be unambiguous, including rules to deal with variable length fields and padding.                                                         |
| 5           | The Set_Attributes_All request response for objects specified in this document shall include only the open attributes. It shall not include any vendor specific attributes.                                                     |

Table 5 is an example of the attribute ordering method of specifying the service data portion of a Set\_Attributes\_All request for instance level attributes of an object which supports required settable instance attributes 7, 8, 9, 10, 11 and 12.

**Table 5 – Example Set\_Attributes\_All attribute ordering method**

| Class Attribute ID | Attribute name  |
|--------------------|-----------------|
| 7                  | Output Range    |
| 8                  | Value Data Type |
| 9                  | Fault State     |
| 10                 | Idle State      |
| 11                 | Fault Value     |
| 12                 | Idle Value      |

Table 6 is an example of the data array method of specifying the service data portion of a Set\_Attributes\_All request for instance level attributes of an object which supports required settable instance attributes 7, 8, 9, 10, 11 and 12.

**Table 6 – Example Set\_Attributes\_All data array method**

| Octet | Bit 7        | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|--------------|-------|-------|-------|-------|-------|-------|-------|
| 0     | Output Range |       |       |       |       |       |       |       |

| Octet | Bit 7                    | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|--------------------------|-------|-------|-------|-------|-------|-------|-------|
| 1     | Value Data Type          |       |       |       |       |       |       |       |
| 2     | Fault State              |       |       |       |       |       |       |       |
| 3     | Idle State               |       |       |       |       |       |       |       |
| 4     | Fault Value (low octet)  |       |       |       |       |       |       |       |
| 5     | Fault Value (high octet) |       |       |       |       |       |       |       |
| 6     | Idle Value (low octet)   |       |       |       |       |       |       |       |
| 7     | Idle Value (high octet)  |       |       |       |       |       |       |       |

### 3.5.3.3.2 Revisions

A server processing a Set\_Attributes\_All request may need to process more data than is expected by the server if the client has implemented a newer revision of this object while the server an older revision. As a result, the server object may have to process a Set\_Attributes\_All request that contains more data because the additional attributes were not recognized. If the server receives more data in a Set\_Attributes\_All request than it expects the server shall respond with a general status code equal to 0x15 (too much data).

NOTE The Revision class attribute is not the revision of an implementation (which is reflected in the Identity object, Minor/Major revision status bits), but the revision of the class definition.

### 3.5.4 State machine conventions

#### 3.5.4.1 General

State changes may be triggered by events (internal or external) or service invocations. Reaction to service invocations may depend on the value(s) of the attribute(s) accessed by the service.

Behaviour of the state machine shall be defined for combinations of:

- the **event** the machine receives;
- the **state** the machine is in when it receives notification of a state-changing event.

To define behaviour in these terms, a State Transition Diagram (STD) and a State Event Matrix (SEM) are used when applicable in the state machine specification.

#### 3.5.4.2 State Transition Diagram (STD)

An **event** is an external stimulus that may cause a state transition. An STD graphically illustrates the states of an object and includes events, service calls and changes of attributes that cause it to transition to another state. Figure 3 shows an example of an STD.