

DIN EN 61158-100-26



ICS 35.100.70; 35.160

Entwurf

Einsprüche bis 2016-08-24
Vorgesehen als
teilweiser Ersatz für
DIN EN 61784-2:2015-02

**Industrielle Kommunikationsnetze –
Feldbus Spezifikationen und Profile –
Typ 26-Elemente und CPF 21 (FL-net)
(IEC 65C/832/CD:2015);
Text in Englisch, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications and Profiles –
Type 26 elements and CPF 21 (FL-net)
(IEC 65C/832/CD:2015);
Text in English, only on CD-ROM

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2016-06-24 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter www.entwuerfe.din.de bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@vde.com möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/stellungnahme oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 319 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Nationales Vorwort

Die englische Originalfassung des internationalen Dokuments IEC 65C/832/CD:2015 „Industrial communication networks – Fieldbus specifications and Profiles – Type 26 elements and CPF 21 (FL-net)“, (CD, en: Committee Draft) ist unverändert in diesen Norm-Entwurf übernommen worden.

Das internationale Dokument wurde vom SC 65C „Industrial networks“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet und den nationalen Komitees zur Stellungnahme vorgelegt.

Die IEC und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Da der Abstimmungszeitraum für einen FDIS bzw. Schluss-Entwurf prEN nur 2 Monate beträgt, und dann keine sachlichen Stellungnahmen mehr abgegeben werden können, sondern nur noch eine „JA/NEIN“-Entscheidung möglich ist, wobei eine „NEIN“-Entscheidung fundiert begründet werden muss, wird bereits der CD als DIN-Norm-Entwurf veröffentlicht, um die Stellungnahmen aus der Öffentlichkeit frühzeitig berücksichtigen zu können.

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

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.

Änderungen

Gegenüber DIN EN 61784-2:2015-02 wurden folgende Änderungen vorgenommen:

- a) a) Dieser Sammelentwurf enthält neben den einzelnen Entwürfe zu IEC 61158-3-26, IEC 61158-4-26, IEC 61158-5-26, IEC 61158-6-26, welche die neue Kommunikationsprofilfamilie 20 (FL-net) Typ 26-Elemente beschreibt, ebenfalls eine entsprechende Erweiterung (CPF 21) für die kommende Überarbeitung der DIN EN 61784-2 Ed.3.

Nationaler Anhang NA (informativ)

Die deutschen Übersetzungen haben Empfehlungscharakter und können nur die ungefähre Begrifflichkeit in Deutsch wiedergeben. Unter Experten wird empfohlen, ausschließlich die englischen Fachbegriffe zu verwenden, um die Gefahr von Missverständnissen zu reduzieren.

1 Anwendungsbereich^{N1)}

1.1 Überblick

Die Feldbus-Anwendungsschicht (FAL; en: field bus application layer) stellt Benutzerprogramme bereit, mit einem Mittel für den Zugriff auf die Feldbus-Kommunikationsumgebung. Diesbezüglich kann die FAL als ein „Fenster zwischen zugehörigen Anwendungsprogrammen“ angesehen werden.

Diese Norm stellt allgemeine Elemente für grundlegende zeitkritische und nicht-zeitkritische Nachrichtenkommunikation zwischen Anwendungsprogrammen in einer automatisierten Umgebung und für den Typ 26-Feldbus spezifische Materialien bereit. Der Begriff „zeitkritisch“ wird verwendet, um das Vorhandensein eines Zeitfensters darzustellen, innerhalb dessen der Abschluss einer oder mehrerer festgelegter Aktionen mit einem gewissen festgelegten Grad an Sicherheit gefordert ist. Eine Nichterfüllung des Abschlusses festgelegter Aktionen innerhalb des Zeitfensters birgt die Gefahr des Scheiterns der Anwendung die diese Aktion angefordert hat, mit der daraus entstehenden Gefahr für Betriebsmittel, Anlage und möglicherweise menschliches Leben.

Diese Norm legt in einer abstrakten Weise den extern sichtbaren Dienst fest, welcher durch die verschiedenen Typen der Feldbus-Anwendungsschicht bereitgestellt wird, hinsichtlich:

- a) eines abstrakten Modells für die Festlegung von Anwendungsressourcen (Objekten), die in der Lage sind, mittels der Verwendung des FAL-Dienstes durch Benutzer beeinflusst zu werden;
- b) der Aktionen und Ereignisse von Primitiven dieses Dienstes;
- c) der Parameter, welche den Aktionen und Ereignisse von Primitiven zugeordnet sind, und der Form, welche sie annehmen, und
- d) die Wechselbeziehung zwischen diesen Aktionen und Ereignissen und ihre gültigen Abfolgen.

Der Zweck der vorliegenden Norm ist die Festlegung der Dienste, welche bereitgestellt werden für:

- a) die FAL-Benutzer an der Grenze zwischen dem Benutzer und der Anwendungsschicht des Feldbusreferenzmodells, und
- b) das Systemmanagement an der Grenze zwischen der Anwendungsschicht und dem Systemmanagement des Feldbusreferenzmodells.

Diese Norm legt die Struktur und die Dienste der IEC-Feldbus-Anwendungsschicht in Übereinstimmung mit dem OSI-Basis-Referenzmodell (ISO/IEC 7498) und der Struktur der OSI-Anwendungsschicht (ISO/IEC 9545) fest.

FAL-Dienste und -Protokolle werden durch FAL-Anwendungsinstanzen (AE; application entity) bereitgestellt, welche in den Anwendungsprozessen enthalten sind. Die FAL-AE wird gebildet aus einer Gruppe von objektorientierten Anwendungsdienstelementen (ASE; en: Application Service Elements) und einer Schichtverwaltungsinstanz (LME; en: Layer Management Entity), welche die AE verwaltet. Die ASE stellen Kommunikationsdienste bereit, welche auf der Grundlage einer Gruppe verbundener Klassen von Anwendungsprozessobjekten (APO; en: application process object) betrieben werden. Eines der FAL-ASE ist

^{N1)} Nationale Fußnote: Die Nummerierung des IEC-Schriftstücks 65C/832/CD:2015-10 wurde beibehalten.

ein Verwaltungs-ASE, welches eine allgemeine Gruppe von Diensten für die Verwaltung der Instanzen von FAL-Klassen bereitstellt.

Obwohl diese Dienste aus der Sicht von Anwendungen festlegen, wie Anfragen und Antworten herausgegeben und geliefert werden, umfassen sie keine Festlegung dazu, was die anfragenden und antwortenden Anwendungen mit diesen zu machen haben. Das heißt, die Verhaltensaspekte der Anwendungen sind nicht festgelegt; nur eine Definition dazu, welche Anfragen und Antworten sie senden/empfangen können, ist festgelegt. Dies bietet Benutzern eine größere Flexibilität hinsichtlich der Normung eines solchen Verhaltens von Objekten. Zusätzlich zu diesen Diensten sind auch einige unterstützende Dienste in dieser Norm festgelegt, um für die Steuerung bestimmter Aspekte ihres Betriebs einen Zugriff auf FAL bereitzustellen.

1.2 Festlegungen

Das grundlegende Ziel dieser Norm ist die Festlegung der Merkmale eines konzeptuellen Dienstes der Anwendungsschicht, welcher für zeitkritische Kommunikation geeignet ist, und somit die Ergänzung des OSI-Basis-Referenzmodells für die Anleitung zur Entwicklung von Protokollen der Anwendungsschicht für zeitkritische Kommunikationen.

Ein sekundäres Ziel ist die Bereitstellung von Migrationspfaden von bereits bestehenden industriellen Kommunikationsprotokollen. Das letztere Ziel führt zur Vielfalt von normierten Diensten, wie den verschiedenen Typen nach IEC 61158, und den entsprechenden Protokollen, die in den Teilen von IEC 61158-6 normiert sind.

Diese Festlegung darf als Grundlage für formelle Programmierungsschnittstellen der Anwendungsschicht verwendet werden. Sie ist jedoch keine formelle Programmierungsschnittstelle und jede derartige Schnittstelle muss auch Implementierungsfragen behandeln, welche durch diese Norm nicht abgedeckt sind.

- a) die Größen und Oktettfolgen von verschiedenen Multi-Oktett-Dienstparametern, und
- b) die Beziehung von paarweisen Primitiven für Anfrage und Bestätigung oder Meldung und Antwort.

1.3 Konformität

Diese Norm legt weder einzelnen Implementierungen oder Produkte fest, noch schränkt sie die Implementierungen von Anwendungsschichtinstanzen innerhalb eines industriellen Automatisierungssystems ein.

Es besteht keine Konformität von Betriebsmitteln mit dieser Norm für die Dienstfestlegungen der Anwendungsschicht. Stattdessen wird die Konformität durch Implementierung der entsprechenden Anwendungsschichtprotokolle erreicht, welche allen nach dieser Norm festgelegten Dienstypen der Anwendungsschicht entsprechen.

3.2 Besondere Begriffe für die Feldbus-Anwendungsschicht Typ 26

Für die Anwendung des vorliegenden Dokuments gelten die folgenden Begriffe.

3.2.1

Anwendung

(en: application)

Funktion oder Datenstruktur, für welche Daten verarbeitet oder erzeugt werden

3.2.2

Interoperabilität auf Anwendungsschicht

(en: application layer interoperability)

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

3.2.3

Anwendungsobjekte

(en: application objects)

mehrere Objektklassen, die einen Nachrichtenaustausch über das Netzwerk und innerhalb des Netzwerkgerätes zur Laufzeit verwalten und bereitstellen

3.2.4

Anwendungsprozess

(en: application process)

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

3.2.5

Anwendungsprozesskennung

(en: application process identifier)

unterscheidet mehrere in einem Gerät verwendete Anwendungsprozesse

3.2.6

Anwendungsprozessobjekt

(en: application process object)

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

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

3.2.7

Anwendungsprozessobjektklasse

(en: application process object class)

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

3.2.8

Anwendungsbeziehung

(en: application relationship)

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

3.2.9

Anwendungsdienstelement für Anwendungsbeziehungen

(en: application relationship application service element)

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

3.2.10

Endpunkt der Anwendungsbeziehung

(en: application relationship endpoint)

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

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

3.2.11**Attribut**

(en: attribute)

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

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

3.2.12**Verhalten**

(en: behaviour)

Anzeige dafür, wie ein Objekt auf bestimmte Ereignisse reagiert

3.2.13**Bitnummer**

(en: bit-no)

kennzeichnet die Nummer eines Bits in einer Bitfolge oder einem Oktett

3.2.14**Kanal**

(en: channel)

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

3.2.15**Klasse**

(en: class)

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

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

3.2.16**Klassenattribute**

(en: class attributes)

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

3.2.17**Klassencode**

(en: class code)

eindeutige Kennung, die jeder Objektklasse zugewiesen ist

3.2.18**klassenspezifischer Dienst**

(en: class specific service)

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

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

3.2.19**Client**

a) Objekt, das die Dienste eines anderen Objekts (Servers) nutzt, um eine Aufgabe auszuführen;

b) Initiator einer Nachricht, auf die ein Server reagiert

3.2.20**allgemeiner Speicher**

(en: common-memory)

virtueller Speicher, auf welchen mit einer logisch eindeutigen Adresse zugegriffen werden kann, und der für zyklische Übertragung verwendet wird, er besteht aus dem Speicherbereich-1 und -2 der Speichergröße 512 Wörter beziehungsweise 8 192 Wörter und wird mit den Typ 26-Knoten in einem Typ 26-Feldbusnetz gemeinsam verwendet

3.2.21**Verbindung**

(en: connection)

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

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

3.2.22**Konsument**

(en: consumer)

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

3.2.23**Steuerbefehle**

(en: control commands)

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

3.2.24**Übertragungspfad**

(en: conveyance path)

unidirektionaler Strom von APDUs in einer Anwendungsbeziehung

3.2.25**zyklisch**

(en: cyclic)

sich regelmäßig wiederholend

3.2.26**Rahmen zyklischer Daten**

(en: cyclic-data frame)

Rahmen, der für die zyklische Übertragung des Typ 26 verwendet wird

3.2.27**Übertragung zyklischer Daten**

(en: cyclic-data transmission)

Datenübertragung, mit welcher eine Typ 26-Knoten Daten an einen vorbestimmten Bereich für den Knoten des allgemeinen Speichers aussendet, wobei der Knoten bei Rundsendungen (en: Broadcasting) an alle anderen Typ 26-Knoten jedes Mal ein Token bezieht

3.2.28**Datenkonsistenz**

(en: data consistency)

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

3.2.29**zugeordnete AR**

(en: dedicated AR)

Anwendungsbeziehung, die direkt vom FAL-Benutzer verwendet wird

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

3.2.30

Gerät

(en: device)

physikalische Hardware, die mit dem Verbindungszweig verbunden ist

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

3.2.31

Geräteprofil

(en: device profile)

eine Zusammenstellung geräteabhängiger Informationen und Funktionalitäten, welche Konsistenz zwischen ähnlichen Geräten desselben Gerätetyps bereitstellen

3.2.32

dynamische AR

(en: dynamic AR)

AR, für die die Nutzung der Prozeduren für die Herstellung von ARs erforderlich ist, um sie in einen aktiven Zustand zu versetzen

3.2.33

Endknoten

(en: end node)

erzeugender oder konsumierender Knoten

3.2.34

Endpunkt

(en: endpoint)

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

3.2.35

Fehler

(en: error)

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

3.2.36

Ereignis

(en: event)

eine Instanz einer Änderung von Zuständen

3.2.37

Protokoll einer FA-Verbindung

(en: FA-link protocol)

Kommunikationsprotokoll für einen Typ 26-Feldbus; legt die Verwaltung von Token, die Übertragung von zyklischen Daten und die Übertragung von Nachrichtendaten fest, mit welchen jeder der Typ 26-Knoten eine Datenübertragung auf gleichberechtigter Basis durchführen kann

3.2.38

FAL-Subnetz

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

3.2.39

Rahmen

(en: frame)

umgangssprachliches Synonym für DLPDU

3.2.40

Gruppe

(en: group)

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

3.2.41

Schnittstelle

(en: interface)

- a) gemeinsame Grenze zwischen zwei Funktionseinheiten, die durch funktionelle Merkmale, Signalgrößen oder andere Merkmale als geeignet festgelegt ist
- b) Zusammenstellung von FAL-Klassenattributen und -diensten, die eine spezifische Sicht auf die FAL-Klasse darstellt

3.2.42

Schnittstellenzeiger

(en: interface pointer)

Schlüsselattribut, das eine Objektschnittstelleninstanz eindeutig adressiert

3.2.43

Aufruf

(en: invocation)

Vorgang der Nutzung eines Dienstes oder einer anderen Ressource eines Anwendungsprozesses

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

3.2.44

E/A-Daten

(en: I/O data)

Objekt, das zu Verarbeitungszwecken zyklisch übertragen wird

3.2.45

Index

Adresse eines Objekts innerhalb eines Anwendungsprozesses

3.2.46

Instanz

(en: instance)

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

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

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

3.2.47

Instanzattribute

(en: instance attributes)

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

3.2.48

instanciert

(en: instantiated)

Objekt, das in einem Gerät erzeugt wurde

3.2.49**Funktion der Netzlastmessung**

(en: network load measurement function)

Vorkehrung zur Überwachung und Messung der Kommunikationslast in einem Typ 26-Feldbus oder einem Typ 26-Knoten

3.2.50**logisches Gerät**

(en: logical device)

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

3.2.51**Verwaltungsinformation**

(en: management information)

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

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

3.2.52**Rahmen für Nachrichtendaten**

(en: message-data frame)

Rahmen, der für die Übertragung von Nachrichtendaten des Typ 26 verwendet wird

3.2.53**Übertragung von Nachrichtendaten**

Unregelmäßige Übertragung, mit welcher ein Typ 26-Knoten die Nachrichtendaten einer Anfrage, einer Antwort, eines Informationsberichts, einer Benachrichtigung zum Auftreten eines Ereignisses am Knoten usw. an andere Typ 26-Knoten aussendet

3.2.54**Methode**

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

3.2.55**Modul**

Hardware oder logische Komponente eines physischen Gerätes

3.2.56**Mehrpunktverbindung**

(en: multipoint connection)

Verbindung von einem Knoten zu vielen Knoten

Anmerkung 1 zum Begriff: Eine Mehrpunktverbindung ermöglicht den Empfang von Nachrichten eines einzelnen Erzeugers durch viele Abonnenten-Knoten.

3.2.57**Netzwerk**

Reihe von Knoten, die über ein bestimmtes Kommunikationsmedium miteinander verbunden sind, einschließlich aller Repeater, Netzwerkbrücken, Router und Gateways unterer Schichten

3.2.58**Objekt**

abstrakte Darstellung einer bestimmten Komponente innerhalb eines Gerätes, üblicherweise eine Zusammenstellung zugehöriger Daten in Form von Variablen und Methoden (Verfahren) für Operationen in Bezug auf diese Daten mit klar definierter Schnittstelle und klar definiertem Verhalten

3.2.59

Aufruf objektorientierter entfernter Methoden

(en: object remote procedure call)

Modell für den objektorientierten oder komponentenbasierten Aufruf von entfernten Methoden

3.2.60

objektspezifischer Dienst

(en: object specific service)

Dienst, der für die Klasse eindeutig ist, die ihn festlegt

3.2.61

Teilnahme-Anfrage-Rahmen

(en: participation-request-frame)

Rahmen, der für die Benachrichtigung eines Typ 26-Knotens über seine Teilnahme-Anfrage in einem Typ 26-Feldbusnetz verwendet wird

3.2.62

gleichrangig

(en: peer)

Rolle eines AR-Endpunktes, in welcher er sowohl Client als auch Server sein kann

3.2.63

physikalisches Gerät

(en: physical device)

Automatisierungsgerät oder anderes Netzwerkgerät

3.2.64

Punkt-zu-Punkt-Verbindung

(en: point-to-point connection)

Verbindung, die zwischen genau zwei Anwendungsobjekten besteht

3.2.65

vordefinierter AR-Endpunkt

(en: pre-defined AR endpoint)

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

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

3.2.66

vorher aktivierter AR-Endpunkt

AR-Endpunkt, der während der Konfiguration der AE, die seine Endpunkte steuern, in einen aktiven Zustand versetzt wird

3.2.67

Eigenschaft

(en: property)

allgemeine Bezeichnung für beschreibende Informationen zu einem Objekt

3.2.68

Protokolltyp

(en: protocol type)

Kennung des verwendeten Kommunikationsprotokolls

3.2.69

Publisher

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

Anmerkung 1 zum Begriff: Ein Publisher kennt die Identität oder Anzahl der Subscriber möglicherweise nicht und kann seine APDUs mithilfe einer zugeordneten AR veröffentlichen.

3.2.70**Pull-Publisher**

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

3.2.71**veröffentlichender Pull-Manager**

(en: pull publishing manager)

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

3.2.72**Push-Publisher**

Typ von Publisher, der ein Objekt in einer unbestätigten Dienstanforderungs-APDU veröffentlicht

3.2.73**veröffentlichender Push-Manager**

(en: push publishing manager)

Typ von veröffentlichendem Manager, der die Veröffentlichung eines bestimmten Objektes mithilfe eines unbestätigten Dienstes anfordert

3.2.74**Pull-Subscriber**

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

3.2.75**Push-Subscriber**

Typ von Subscriber, der empfangene unbestätigte Dienstanforderungs-APDUs als veröffentlichte Objektdaten erkennt

3.2.76**Ressource**

Verarbeitungs- oder Informationsfähigkeit eines Subsystems

3.2.77**Laufzeitobjektmodell**

(en: runtime object model)

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

3.2.78**Sequenznummer (SEQ)**

deren Wert zur Bestimmung verwendet wird, ob eine empfangende Nachricht bereits vorher empfangen wurde und somit redundant ist; wird zum Zeitpunkt der Inbetriebnahme des Netzwerkes auf 0 (Null) gesetzt und wird jedes Mal, wenn eine Übertragung von Nachrichtendaten abgeschlossen ist um 1 (Eins) erhöht

3.2.79**Server**

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

3.2.80**Dienst**

(en: service)

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

3.2.81

Subscriber

Teilnehmer

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

3.2.82

Token

Recht zum Senden von Daten über das Übertragungsmedium, und wird in einem Typ 26-Netz nacheinander von Knoten zu Knoten weitergegeben

3.2.83

Token-Frame

Frame, der für die Übergabe von Token zwischen Typ 26-Knoten verwendet wird

3.2.84

Transaktionscode (TCD)

(en: transaction code)

wird verwendet, um jeden Frame der Datenübertragung zu unterscheiden und zu kennzeichnen, und repräsentiert die Transaktion eines empfangenen Rahmens

3.2.85

Trigger-Frame

Rahmen, der im Zustand der Netzinbetriebnahme verwendet wird, um einen Typ 26-Knoten bezüglich seines Eintritts in eine neue Teilnehmeranfrage in einem Typ 26-Netzwerk zu benachrichtigen

3.2.86

verbindungsloser Dienst

(en: unconnected service)

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

3.2.87

Zustand gültiger Verbindung

(en: valid-linking state)

Zustand, in welchem ein Token nacheinander zwischen Knoten in einem Typ 26-Netzwerk verbreitet wird

3.2.88

Version einer Sequenznummer (V_SEQ)

deren Wert zu dem Zeitpunkt gesetzt wird, wenn ein Typ 26-Knoten die Teilnahme in einem Typ 26-Feldbus-Netzwerk vollzieht, wird beibehalten, bis der Knoten aus dem Typ 26-Feldbus-Netzwerk ausscheidet, und wird für andere Knoten verwendet, um zu bestimmen, ob eine empfangene Nachricht vom Knoten gültig und nicht überfällig ist

3.2.89

virtueller Adressraum

(en: virtual address space)

virtueller Speicherplatz mit einer pseudologischen Adresse, wird für die Nachrichtenübertragung verwendet

– Leerseite –

CONTENTS

1		
2	FOREWORD.....	5
3	INTRODUCTION.....	7
4	1 Scope.....	8
5	1.1 Overview	8
6	1.2 Specifications	9
7	1.3 Conformance	9
8	2 Normative references	9
9	3 Terms, definitions, symbols, abbreviations and conventions	10
10	3.1 Referenced terms and definitions.....	10
11	3.1.1 ISO/IEC 7498-1 terms.....	10
12	3.1.2 ISO/IEC 8822 terms.....	10
13	3.1.3 ISO/IEC 9545 terms.....	10
14	3.1.4 ISO/IEC 8824-1 terms.....	11
15	3.1.5 IEC 61158-1 terms.....	11
16	3.2 Fieldbus application layer type 26-specific definitions	11
17	3.3 Abbreviations and symbols	19
18	3.4 Conventions.....	20
19	3.4.1 Overview	20
20	3.4.2 General conventions.....	20
21	3.4.3 Conventions for class definitions	20
22	3.4.4 Conventions for service definitions	22
23	4 Concepts	23
24	5 Data type ASE	23
25	5.1 Overview	23
26	5.2 Formal definition of data type objects.....	24
27	5.2.1 Data type class definitions	24
28	5.2.2 Attributes.....	24
29	5.3 FAL defined data types	25
30	5.3.1 Fixed length types	25
31	5.3.2 String types	29
32	5.4 Data type ASE service specification	29
33	6 Communication model specification	30
34	6.1 General.....	30
35	6.2 Protocol stack for type 26 fieldbus	30
36	6.3 Overview of type 26 communication model	31
37	6.4 Cyclic data communication service with Common-memory	32
38	6.4.1 Overview	32
39	6.4.2 Common-memory: allocation to each node.....	32
40	6.4.3 Data sharing among nodes with the CM.....	33
41	6.4.4 CM data type	34
42	6.5 ASEs	34
43	6.5.1 Overview of type 26 ASEs	34
44	6.5.2 Type 26 specific conventions for FAL service common parameters.....	35
45	6.5.3 Cyclic-data ASE.....	36
46	6.5.4 Message data ASE	41
47	6.5.5 Communication load measurement ASE	68

48	6.5.6	Network management ASE	74
49	6.5.7	General purpose command server ASE	86
50	6.5.8	AR ASE	89
51	6.5.9	FAL ASE summary	99
52		Bibliography.....	102
53			
54		Figure 1 – Protocol stack for type 26 fieldbus	31
55		Figure 2 – Unconfirmed Push-Publisher/Subscriber type interaction	31
56		Figure 3 – Unconfirmed/Confirmed Client/Server type interaction	32
57		Figure 4 – Common memory allocation	33
58		Figure 5 - Data sharing with the CM.....	34
59		Figure 6 – Node #01 for reception only	34
60		Figure 7 – Node #01 without the CM	34
61		Figure 8 – The structure of ASEs for type 26 FAL	35
62		Figure 9 – Virtual-address-space for Byte block	44
63		Figure 10 – Virtual-address-space for Word block.....	46
64		Figure 11 – AR ASE internal architecture	89
65		Figure 12 - Structure of IP address	98
66			
67		Table 1 – Write service parameters.....	38
68		Table 2 – Send-CM service parameters	39
69		Table 3 – Read service parameters.....	39
70		Table 4 – Update memory service parameters	40
71		Table 5 – Get- buffer service parameters	41
72		Table 6 – Byte block read service parameters.....	45
73		Table 7 – Byte block write service parameters	46
74		Table 8 – Word block read service parameters	47
75		Table 9 – Word block write service parameters	48
76		Table 10 – Network parameter read service parameters	49
77		Table 11 – Network parameter write service parameters	50
78		Table 12 – Stop command service parameters.....	51
79		Table 13 – Operation command service parameters.....	52
80		Table 14 – profile read service parameters	52
81		Table 15 – Transparent message service parameters	55
82		Table 16 – Log data read service parameters	55
83		Table 17 – Log data items.....	56
84		Table 18 – Log data clear service parameters.....	60
85		Table 19 – Message return service parameters.....	61
86		Table 20 – Vendor specific message service parameters	62
87		Table 21 – Set remote node configuration parameter service parameters	63
88		Table 22 – Data elements and Node configuration parameters.....	63
89		Table 23 – Service parameters of Read remote participating node management information parameter service.....	64
90			
91		Table 24 – Participating node management information parameters.....	64

92	Table 25 – Read remote node management information parameter service parameters	66
93	Table 26 – Node management information parameters	67
94	Table 27 – Read remote node setting information parameter service parameters	68
95	Table 28 – Node setting information parameters	68
96	Table 29 – Start TK-holding-time measurement service parameters	70
97	Table 30 – Terminate TK-holding-time measurement service parameters	71
98	Table 31 – Token-holding-time measurement result	71
99	Table 32 – Start GP_Comm sender log service parameters	72
100	Table 33 – Terminate GP_Comm sender log service parameters	73
101	Table 34 – GP_Comm sender log measurement result	73
102	Table 35 – Service parameters for Set configuration parameter	77
103	Table 36 – Configuration parameters	77
104	Table 37 – Read node management information parameter service parameters	78
105	Table 38 – Node management information parameters	78
106	Table 39 – Service parameters for Read participating node mgt. information parameter	79
107	Table 40 – Participating node management information parameters	79
108	Table 41 – Service parameters for Read network management information parameter	80
109	Table 42 – Network management information parameters	80
110	Table 43 – Service parameters for Read message sequence number management	
111	information	81
112	Table 44 – Read message sequence number management information parameters	81
113	Table 45 – Read node status service parameters	82
114	Table 46 – Read node status parameters	82
115	Table 47 – Upper layer operating condition matrix	83
116	Table 48 – Reset node service parameters	83
117	Table 49 – Set network address service parameters	84
118	Table 50 – Register service parameters	84
119	Table 51 – Event service parameters	85
120	Table 52 – Activate/Deactivate measurement service parameters	85
121	Table 53 – Get log data service parameters	86
122	Table 54 – Send command service parameters	88
123	Table 55 – CT send service parameters	92
124	Table 56 – MT send service parameters	93
125	Table 57 – CS send service parameters	94
126	Table 58 – Notify state change service parameters	95
127	Table 59 – Control measurement service parameters	96
128	Table 60 – DLSAP assignments	97
129	Table 61 – DLS Primitives and parameters	98
130	Table 62 – Lower layer T-profile and the required standards	99
131	Table 63 – Summary of FAL ASEs	99

132

133

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –

Part 5-26: Application layer service definition – Type 26 elements

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61158-5-26 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement and control.

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/XX/FDIS	65C/XX/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

The National Committees are requested to note that for this publication the stability date is 20XX.

THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE DELETED AT THE PUBLICATION STAGE.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

Contents highlighted in YELLOW is fixed at a later date.

THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE DELETED AT THE PUBLICATION STAGE.

204

INTRODUCTION

205 This part of IEC 61158 is one of a series produced to facilitate the interconnection of
206 automation system components. It is related to other standards in the set as defined by the
207 “three-layer” fieldbus reference model described in IEC 61158-1:201X.

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-26: Application layer service definition – Type 26 elements

1 Scope

1.1 Overview

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

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 26 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the different Types of fieldbus Application Layer in terms of

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

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

- a) the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model, and
- b) Systems Management at the boundary between the Application Layer and Systems Management of the Fieldbus Reference Model.

This standard specifies the structure and services of the IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

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

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

1.2 Specifications

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

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of services standardized as the various Types of IEC 61158, and the corresponding protocols standardized in subparts of IEC 61158-6.

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

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

1.3 Conformance

This standard does not specify individual implementations or products, nor does it constrain the implementations of application layer entities within industrial automation systems.

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

2 Normative references

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

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

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

IEC 61158-1:201X, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61784-2:201X, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

IEC 61158-6-26:—1, *Industrial communication networks - Fieldbus specifications – Part 6-26: Application layer protocol specification – Type 26 elements*

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

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

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

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

¹ In preparation

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

ISO/IEC 10646, *Information technology – Universal Multiple-Octet Coded Character Set(UCS)*

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

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

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

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

IETF RFC 793, *Transmission Control Protocol*, available at < <http://www.ietf.org> >

IETF RFC 796, *Address mappings*, available at < <http://www.ietf.org> >

IETF RFC 826, *An Ethernet Address Resolution Protocol*, available at < <http://www.ietf.org> >

IETF RFC 894, *A Standard for the Transmission of IP Datagrams over Ethernet Networks*, available at < <http://www.ietf.org> >

IETF RFC 919, *Broadcasting Internet Datagrams*, available at < <http://www.ietf.org> >

IETF RFC 922, *Broadcasting Internet Datagrams in the presence of subnets*, available at < <http://www.ietf.org> >

IETF RFC 950, *Internet Standard Subnetting Procedure*, available at < <http://www.ietf.org> >

3 Terms, definitions, symbols, abbreviations and conventions

3.1 Referenced terms and definitions

3.1.1 ISO/IEC 7498-1 terms

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

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

3.1.2 ISO/IEC 8822 terms

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

- a) abstract syntax
- b) presentation context

3.1.3 ISO/IEC 9545 terms

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

- a) application-association
- b) application-context
- c) application context name

- 345 d) application-entity-invocation
- 346 e) application-entity-type
- 347 f) application-process-invocation
- 348 g) application-process-type
- 349 h) application-service-element
- 350 i) application control service element

351 **3.1.4 ISO/IEC 8824-1 terms**

352 For the purposes of this document, the following terms as defined in ISO/IEC 8824-1 apply:

- 353 a) object identifier
- 354 b) type
- 355 c) value
- 356 d) simple type
- 357 e) structured type
- 358 f) tag
- 359 g) Boolean type
- 360 h) true
- 361 i) false
- 362 j) integer type
- 363 k) bitstring type
- 364 l) octetstring type

365 **3.1.5 IEC 61158-1 terms**

366 For the purposes of this document, the following terms as defined in ISO/IEC 61158-1 apply:

- 367 a) application relationship
- 368 b) conveyance path
- 369 c) client
- 370 d) dedicated AR
- 371 e) dynamic AR
- 372 f) error class
- 373 g) error code
- 374 h) name
- 375 i) numeric identifier
- 376 j) peer
- 377 k) pre-defined AR endpoint
- 378 l) pre-established AR endpoint
- 379 m) publisher
- 380 n) subscriber
- 381 o) server

382 **3.2 Fieldbus application layer type 26-specific definitions**

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

384 **3.2.1**

385 **application**

386 function or data structure for which data is consumed or produced

3.2.2**application layer interoperability**

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

3.2.3**application objects**

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

3.2.4**application process**

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

3.2.5**application process identifier**

distinguishes multiple application processes used in a device

3.2.6**application process object**

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

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

3.2.7**application process object class**

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

3.2.8**application relationship**

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

3.2.9**application relationship application service element**

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

3.2.10**application relationship endpoint**

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

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

3.2.11**attribute**

description of an externally visible characteristic or feature of an object

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

3.2.12**behavior**

indication of how an object responds to particular events

- 440 **3.2.13**
441 **bit-no**
442 designates the number of a bit in a bitstring or an octet
- 443 **3.2.14**
444 **channel**
445 single physical or logical link of an input or output application object of a server to the process
- 446 **3.2.15**
447 **class**
448 a set of objects, all of which represent the same kind of system component
- 449 Note 1 to entry: A class is a generalization of an object; a template for defining variables and methods. All objects
450 in a class are identical in form and behavior, but usually contain different data in their attributes.
- 451 **3.2.16**
452 **class attributes**
453 attribute that is shared by all objects within the same class
- 454 **3.2.17**
455 **class code**
456 unique identifier assigned to each object class
- 457 **3.2.18**
458 **class specific service**
459 service defined by a particular object class to perform a required function which is not
460 performed by a common service
- 461 Note 1 to entry: A class specific object is unique to the object class which defines it.
- 462 **3.2.19**
463 **client**
464 a) object which uses the services of another (server) object to perform a task
465 b) initiator of a message to which a server reacts
- 466 **3.2.20**
467 **common-memory**
468 virtual memory accessible with logically unique address is used for the cyclic-transmission, is
469 composed of the memory-area-1 and -2 of the memory size 512 words and 8 192 words
470 respectively, and is shared with the type 26 nodes in a type 26 fieldbus network
- 471 **3.2.21**
472 **connection**
473 logical binding between application objects that may be within the same or different devices
- 474 Note 1 to entry: Connections can be either point-to-point or multipoint.
- 475 **3.2.22**
476 **consumer**
477 node or sink that is receiving data from a producer
- 478 **3.2.23**
479 **control commands**
480 action invocations transferred from client to server to clear outputs, freeze inputs and/or
481 synchronize outputs
- 482 **3.2.24**
483 **conveyance path**
484 unidirectional flow of APDUs across an application relationship
- 485 **3.2.25**
486 **cyclic**
487 repetitive in a regular manner

3.2.26**cyclic-data frame**

frame used for the type 26 cyclic-transmission

3.2.27**cyclic-data transmission**

data transmission with which a type 26 node sends out data on a pre-assigned area for the node on the common-memory, in broadcasting to all other type 26 nodes every time the node obtains a token

3.2.28**data consistency**

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

3.2.29**dedicated AR**

AR used directly by the FAL User

Note 1 to entry: On dedicated ARs, only the FAL Header and the user data are transferred.

3.2.30**device**

physical hardware connected to the link

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

3.2.31**device profile**

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

3.2.32**dynamic AR**

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

3.2.33**end node**

producing or consuming node

3.2.34**endpoint**

one of the communicating entities involved in a connection

3.2.35**error**

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

3.2.36**event**

an instance of a change of conditions

3.2.37**FA-link protocol**

communication protocol for type 26 fieldbus; defines the token management, the cyclic data transmission and the message data transmission, with which each type 26 node can perform data transmission on an equal basis

3.2.38**FAL subnet**

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

- 538 **3.2.39**
539 **frame**
540 denigrated synonym for DLPDU
- 541 **3.2.40**
542 **group**
543 a) <general> a general term for a collection of objects.
544 b) <addressing> when describing an address, an address that identifies more than one entity
- 545 **3.2.41**
546 **interface**
547 a) shared boundary between two functional units, defined by functional characteristics,
548 signal characteristics, or other characteristics as appropriate
549 b) collection of FAL class attributes and services that represents a specific view on the FAL
550 class
- 551 **3.2.42**
552 **interface pointer**
553 key attribute that unambiguously addresses an object interface instance
- 554 **3.2.43**
555 **invocation**
556 act of using a service or other resource of an application process
- 557 Note 1 to entry: Each invocation represents a separate thread of control that can be described by its context.
558 Once the service completes, or use of the resource is released, the invocation ceases to exist. For service
559 invocations, a service that has been initiated but not yet completed is referred to as an outstanding service
560 invocation.
- 561 **3.2.44**
562 **I/O data**
563 object designated to be transferred cyclically for the purpose of processing
- 564 **3.2.45**
565 **index**
566 address of an object within an application process
- 567 **3.2.46**
568 **instance**
569 the actual physical occurrence of an object within a class that identifies one of many objects
570 within the same object class
- 571 EXAMPLE California is an instance of the object class state.
572 Note 1 to entry: The terms object, instance, and object instance are used to refer to a specific instance.
- 573 **3.2.47**
574 **instance attributes**
575 attribute that is unique to an object instance and not shared by the object class
- 576 **3.2.48**
577 **instantiated**
578 object that has been created in a device
- 579 **3.2.49**
580 **network load measurement function**
581 provision for monitoring and measurement of the communication load over a type 26 fieldbus
582 with a type 26 node
- 583 **3.2.50**
584 **logical device**
585 a certain FAL class that abstracts a software component or a firmware component as an
586 autonomous self-contained facility of an automation device

3.2.51**management information**

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

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

3.2.52**message-data frame**

frame used for the type 26 message-data transmission

3.2.53**message-data transmission**

sporadic data transmission with which a type 26 node sends out the message data of request, response, information report, notification upon event happened at the node and so forth, to other type 26 nodes

3.2.54**method**

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

3.2.55**module**

hardware or logical component of a physical device

3.2.56**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.

3.2.57**network**

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

3.2.58**object**

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

3.2.59**object remote procedure call**

model for object oriented or component based remote method invocation

3.2.60**object specific service**

service unique to the object class which defines it

3.2.61**participation-request-frame**

frame used for a type 26 node notifying of its participation-request in a type 26 fieldbus network

3.2.62**peer**

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

3.2.63**physical device**

an automation or other network device

- 637 **3.2.64**
638 **point-to-point connection**
639 connection that exists between exactly two application objects
- 640 **3.2.65**
641 **pre-defined AR endpoint**
642 AR endpoint that is defined locally within a device without use of the create service
- 643 Note 1 to entry: Pre-defined ARs that are not pre-established are established before being used
- 644 **3.2.66**
645 **pre-established AR endpoint**
646 AR endpoint that is placed in an established state during configuration of the AEs that control
647 its endpoints
- 648 **3.2.67**
649 **property**
650 a general term for descriptive information about an object
- 651 **3.2.68**
652 **protocol type**
653 used for communication protocol identification
- 654 **3.2.69**
655 **publisher**
656 role of an AR endpoint that transmits APDUs onto the fieldbus for consumption by one or
657 more subscribers
- 658 Note 1 to entry: A publisher may not be aware of the identity or the number of subscribers and it may publish its
659 APDUs using a dedicated AR.
- 660 **3.2.70**
661 **pull publisher**
662 type of publisher that publishes an object in response to a request received from its pull
663 publishing manager
- 664 **3.2.71**
665 **pull publishing manager**
666 type of publishing manager that requests that a specified object be published in a
667 corresponding response APDU
- 668 **3.2.72**
669 **push publisher**
670 type of publisher that publishes an object in an unconfirmed service request APDU
- 671 **3.2.73**
672 **push publishing manager**
673 type of publishing manager that requests that a specified object be published using an
674 unconfirmed service
- 675 **3.2.74**
676 **pull subscriber**
677 type of subscriber that recognizes received confirmed service response APDUs as published
678 object data
- 679 **3.2.75**
680 **push subscriber**
681 type of subscriber that recognizes received unconfirmed service request APDUs as published
682 object data
- 683 **3.2.76**
684 **resource**
685 a processing or information capability of a subsystem

3.2.77**runtime object model**

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

3.2.78**sequence number (SEQ)**

of which value is used for other nodes to determine whether a receiving message-data from a node has been previously received and is redundant, is set to 0 (zero) at the time of the network start-up, and is increased by 1 (one) every time on a message-data transmission completed

3.2.79**server**

a) role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request

b) object which provides services to another (client) object

3.2.80**service**

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

3.2.81**subscriber**

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

3.2.82**token**

right to send out data on the transmission medium, and is passed successively from node to node for type 26 fieldbus network

3.2.83**token-frame**

frame used for hand-over the token between type 26 nodes

3.2.84**transaction code (TCD)**

used to discriminate and identify each data transmission frame, and represents the transaction of a receiving frame

3.2.85**trigger-frame**

frame used in the network-startup state for a type 26 node notifying of its entrance into a new participation-request over a type 26 fieldbus network

3.2.86**unconnected service**

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

3.2.87**valid-linking state**

state in which a token is consecutively circulated among nodes over type 26 fieldbus network

3.2.88**version of sequence number (V_SEQ)**

of which value is set at the time when a type 26 node completes to participate in a type 26 fieldbus network, is maintained until the node drops out of the type 26 fieldbus network, and is used for other nodes to determine whether a receiving message from the node is valid and not overdue

735 **3.2.89**
736 **virtual-address-space**
737 virtual memory space with pseudo logical address, used for the message-transmission

738 **3.3 Abbreviations and symbols**

AE	Application entity
AL	Application layer
ALME	Application layer management entity
ALP	Application layer protocol
ALS	Application layer service
APO	Application object
AP	Application process
APDU	Application protocol data unit
API	Application process identifier
AR	Application relationship
AREP	Application relationship endpoint
ASCII	American Standard Code for Information Interchange
ASE	Application service element
BCT	Broadcast transmission
C_AD1	Data-head-address on common-memory-area-1
C_AD2	Data-head-address on common-memory-area-2
CIM	Computer integrated manufacturing
CM	Common-memory
Cnf	Confirmation
C_SZ1	Data-size on common-memory-area-1
C_SZ2	Data-size on common-memory-area-2
DA	Destination-address
DL-	(as a prefix) data-link-
DLC	Data-link connection
DLCEP	Data-link connection endpoint
DLL	Data-link layer
DLS	Data-link service
DLSAP	Data-link service access point
DLSDU	DL-service-data-unit
DNA	Destination-node-number
DNS	Domain name service
FAL	Fieldbus application layer
HMI	Human-machine interface
H_TYPE	Header-type
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
IP	Internet protocol
ISO	International Organization for Standardization
LME	Layer management entity
LSB	Least Significant Bit
M_ADD	Message-offset-address in the virtual-address-space

MAJ_VER	FA-link protocol version: major-version
M_CTL	Message-control
MFT	Allowable-minimum-frame-interval-time
MIN_VER	FA-link protocol version: minor-version
MSB	Most Significant Bit
MSN	Manufacturer's model name of a type 26 node
M_SZ	Message-data-size in the virtual-address-space
M_RLT	Message result
NDN	Node name of type 26 node
OSI	Open Systems Interconnect
PDU	Protocol data unit
PhL	Ph-layer
PTP	Point-to-Point
P_TYPE	Protocol-type
RCT	Allowable-refresh-cycle-time
Req	Request
RMT	Refresh-cycle-measurement-time
Rsp	Response
RT	Real Time
SA	Source-address
SAP	Service access point
SDU	Service data unit
SEQ	Sequence-number
SNA	Source-node-number
TCD	Transaction code
TCP	Transmission Control Protocol
TW	Observation time period for the token circulation
UDP	User Datagram protocol
ULS	Upper layer status
VDN	Vendor code of the node
V_SEQ	Version of sequence number

739

740 **3.4 Conventions**741 **3.4.1 Overview**

742 The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate
 743 subclause. Each ASE specification is composed of two parts, its class specification, and its
 744 service specification.

745 The class specification defines the attributes of the class. The attributes are accessible from
 746 instances of the class using the ASE services specified in 6.5 of this standard. The service
 747 specification defines the services that are provided by the ASE.

748 **3.4.2 General conventions**

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

750 **3.4.3 Conventions for class definitions**

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

753 **FAL ASE:** ASE Name

754	CLASS:	Class Name
755	CLASS ID:	#
756	PARENT CLASS:	Parent Class Name
757	ATTRIBUTES:	
758	1 (o) Key Attribute:	numeric identifier
759	2 (o) Key Attribute:	name
760	3 (m) Attribute:	attribute name (values)
761	4 (m) Attribute:	attribute name (values)
762	4.1 (s) Attribute:	attribute name (values)
763	4.2 (s) Attribute:	attribute name (values)
764	4.3 (s) Attribute:	attribute name (values)
765	5. (c) Constraint:	constraint expression
766	5.1 (m) Attribute:	attribute name (values)
767	5.2 (o) Attribute:	attribute name (values)
768	6 (m) Attribute:	attribute name (values)
769	6.1 (s) Attribute:	attribute name (values)
770	6.2 (s) Attribute:	attribute name (values)
771	SERVICES:	
772	1 (o) OpsService:	service name
773	2. (c) Constraint:	constraint expression
774	2.1 (o) OpsService:	service name
775	3 (m) MgtService:	service name

776
777 a) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class
778 being specified.

779 b) The "CLASS:" entry is the name of the class being specified. All objects defined using this
780 template will be an instance of this class. The class may be specified by this standard, or
781 by a user of this standard.

782 c) The "CLASS ID:" entry is a number that identifies the class being specified. This number is
783 unique within the FAL ASE that will provide the services for this class. When qualified by
784 the identity of its FAL ASE, it unambiguously identifies the class within the scope of the
785 FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between
786 1 and 255 are reserved by this standard to identify standardized classes. They have been
787 assigned to maintain compatibility with existing national standards. CLASS IDs between
788 256 and 2048 are allocated for identifying user defined classes.

789 d) The "PARENT CLASS:" entry is the name of the parent class for the class being specified.
790 All attributes defined for the parent class and inherited by it are inherited for the class
791 being defined, and therefore do not have to be redefined in the template for this class.

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

795 e) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the
796 class.

797 1) Each of the attribute entries contains a line number in column 1, a mandatory (m) /
798 optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label
799 in column 3, a name or a conditional expression in column 4, and optionally a list of
800 enumerated values in column 5. In the column following the list of values, the default
801 value for the attribute may be specified.

802 2) Objects are normally identified by a numeric identifier or by an object name, or by both.
803 In the class templates, these key attributes are defined under the key attribute.

804 3) The line number defines the sequence and the level of nesting of the line. Each
805 nesting level is identified by period. Nesting is used to specify

806 i) fields of a structured attribute (4.1, 4.2, 4.3),

807 ii) attributes conditional on a constraint statement (5). Attributes may be mandatory
808 (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require
809 constraint statements as does the attribute defined in (5.2).

810 iii) the selection fields of a choice type attribute (6.1 and 6.2).

811 f) The "SERVICES" label indicates that the following entries are services defined for the
812 class.

1) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.

2) The label "OpsService" designates an operational service (1).

3) The label "MgtService" designates an management service (2).

4) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

3.4.4 Conventions for service definitions

3.4.4.1 General

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

3.4.4.2 Service parameters

Service primitives, used to represent service user/service provider interactions (see ISO/IEC 10731), convey parameters that indicate information available in the user/provider interaction. In any particular interface, not all parameters need be explicitly stated.

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

- a) parameter name,
- b) request primitive, (transmitted from the sender)
- c) indication primitive, (transmitted to the receiver)
- d) response primitive, (transmitted from the receiver) and
- e) confirm primitive (transmitted to the sender).

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

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

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

- a parameter-specific constraint:

"(=)" indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table, and

- an indication that some note applies to the entry:

"(n)" indicates that the following note "n" contains additional information pertaining to the parameter and its use.

3.4.4.3 Service procedures

The procedures are defined in terms of:

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

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

4 Concepts

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: the class definitions, the services and the protocol specification. The first two are contained in this standard. The protocol specification for each of the ASEs is defined in IEC 61158-6-26:—.

The fundamental concepts and templates used to describe the application layer service in this standard follow Clause 9 of IEC 61158-1:201X.

The FAL defined in this standard identifies a peer based controller communications network supporting a Common-memory model in 6.4 as well as client/server models.

5 Data type ASE

5.1 Overview

Fieldbus data types specify the machine independent syntax for application data conveyed by FAL services. The FAL supports the definition and transfer of both basic and constructed data types. Basic types are atomic types. Constructed types are types composed of basic types and other constructed types.

IEC 61158-1:201X, 10.1 provides an overview of the data type ASE and the relationship between data types, and is incorporated herein by reference.

Most basic types are defined from a set of ISO/IEC 8824 types (simple types):

- Basic types have a constant length;
- Two variations are defined, one for defining data types whose length is an integral number of octets, and one for defining data types whose length is bits;
- Some ISO/IEC 8824 types have been extended for fieldbus specific use (specific types). They are defined as simple class subtypes.

Constructed types are of strings, arrays and structures:

- A string shall be composed of an ordered set, variable in number, of homogeneously typed fixed-length elements;
- An array shall be composed of an ordered set of homogeneously typed elements. There are no restrictions on the data type of array elements placed, but each element shall be of the same type;
- A structure shall be made of an ordered set of heterogeneously typed elements called fields. Like arrays, this standard does not restrict the data type of fields. However, the fields within a structure do not have to be of the same type.

Data types are defined as instances of the data type class. Defining new types shall be accomplished by providing a new numeric id and supplying values for the attributes defined for the data type class.

User data is transferred between applications by the FAL protocol. All encoding and decoding are performed by the FAL user. The rules for encoding user data in FAL protocol data units are data type dependent. These rules are defined in IEC 61158-6-26:—.

5.2 Formal definition of data type objects

5.2.1 Data type class definitions

The data type class specifies the root of the data type class tree. Its parent class "top" indicates the top of the FAL class tree. The general form of the template to describe the class definitions is shown below:

FAL ASE:		DATA TYPE ASE
CLASS:		DATA TYPE
CLASS ID:		5 (FIXED LENGTH & STRING), 6 (STRUCTURE), 12 (ARRAY)
PARENT CLASS:		TOP
ATTRIBUTES:		
1	(o)	Key Attribute: Data Type Numeric Identifier
2	(o)	Key Attribute: Data Type Name
3	(m)	Attribute: Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
4	(c)	Constraint: Format = FIXED LENGTH STRING
4.1	(m)	Attribute: Octet Length
5	(c)	Constraint: Format = STRUCTURE
5.1	(m)	Attribute: Number of Fields
5.2	(m)	Attribute: List of Fields
5.2.1	(o)	Attribute: Field Name
5.2.2	(m)	Attribute: Field Data Type
6	(c)	Constraint: Format = ARRAY
6.1	(m)	Attribute: Number of Array Elements
6.2	(m)	Attribute: Array Element Data Type

5.2.2 Attributes

Data type numeric identifier

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

Data type name

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

Format

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

Octet length

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

Number of fields

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

List of fields

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

Field name

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

Field data type

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

Number of array elements

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

963 **Array element data type**

964 This conditional attribute specifies the data type for the elements of an array. All elements of
 965 the array shall have the same data type. It shall be present when the value of the format
 966 attribute is "ARRAY". This attribute may itself specify a constructed data type either by
 967 referencing a constructed data type definition by its numeric id, or by embedding a
 968 constructed data type definition here. When embedding a description, the embedded-data-
 969 type description shown below shall be used.

970 **Embedded data type description**

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

975 **ATTRIBUTES:**

976	1	(m)	Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
977	2	(c)	Constraint:	Format = FIXED LENGTH STRING
978	2.1	(m)	Attribute:	Data Type Numeric ID value
979	2.2	(m)	Attribute:	Octet Length
980	3	(c)	Constraint:	Format = STRUCTURE
981	3.1	(m)	Attribute:	Number of Fields
982	3.2	(m)	Attribute:	List of Fields
983	3.2.1	(m)	Attribute:	Embedded Data Type Description
984	4	(c)	Constraint:	Format = ARRAY
985	4.1	(m)	Attribute:	Number of Array Elements
986	4.2	(m)	Attribute:	Embedded Data Type Description

987

988 **5.3 FAL defined data types**

989 **5.3.1 Fixed length types**

990 **5.3.1.1 Boolean types**

991 **5.3.1.1.1 Boolean**

992 **FAL ASE:** DATA TYPE ASE

993 **CLASS:** DATA TYPE

994 **CLASS ID:** 5

995 **PARENT CLASS:** TOP

996 **ATTRIBUTES:**

997	1	Data Type Numeric Identifier	=	1
998	2	Data Type Name	=	Boolean
999	3	Format	=	FIXED LENGTH
1000	4.1	Octet Length	=	1

1001 This data type expresses a Boolean data type with the values TRUE (non zero) and FALSE
 1002 (zero).

1003 **5.3.1.1.2 BOOL**

1004 This IEC 61131-3 type is the same as Boolean.

1005 **5.3.1.2 BitString types**

1006 **5.3.1.2.1 BitString8**

1007 **FAL ASE:** DATA TYPE ASE

1008 **CLASS:** DATA TYPE

1009 **CLASS ID:** 5

1010 **PARENT CLASS:** TOP

1011 **ATTRIBUTES:**

1012	1	Data Type Numeric Identifier	=	22
1013	2	Data Type Name	=	BitString8
1014	3	Format	=	FIXED LENGTH
1015	5.1	Octet Length	=	1

1016 This type contains 1 element of type BitString.

1017 5.3.1.2.2 BitString16

1018	FAL ASE:		DATA TYPE ASE
1019	CLASS:		DATA TYPE
1020	CLASS ID:	5	
1021	PARENT CLASS:	TOP	
1022	ATTRIBUTES:		
1023	1	Data Type Numeric Identifier	= 23
1024	2	Data Type Name	= BitString16
1025	3	Format	= FIXED LENGTH
1026	5.1	Octet Length	= 2

1027 5.3.1.2.3 WORD

1028 This IEC 61131-3 type is the same as BitString16.

1029 5.3.1.2.4 BitString32

1030	FAL ASE:		DATA TYPE ASE
1031	CLASS:		DATA TYPE
1032	CLASS ID:	5	
1033	PARENT CLASS:	TOP	
1034	ATTRIBUTES:		
1035	1	Data Type Numeric Identifier	= 24
1036	2	Data Type Name	= BitString32
1037	3	Format	= FIXED LENGTH
1038	5.1	Octet Length	= 4

1039 5.3.1.2.5 DWORD

1040 This IEC 61131-3 type is the same as BitString32.

1041 5.3.1.3 Numeric types

1042 5.3.1.3.1 Floating Point types

1043 5.3.1.3.1.1 Float32

1044	FAL ASE:		DATA TYPE ASE
1045	CLASS:		DATA TYPE
1046	CLASS ID:	5	
1047	PARENT CLASS:	TOP	
1048	ATTRIBUTES:		
1049	1	Data Type Numeric Identifier	= 8
1050	2	Data Type Name	= Float32
1051	3	Format	= FIXED LENGTH
1052	4.1	Octet Length	= 4

1053 This type has a length of four octets. The format for float32 is that defined by IEC 60559 as
1054 single precision.

1055 5.3.1.3.1.2 REAL

1056 This IEC 61131-3 type is the same as Float32.

1057 5.3.1.3.1.3 Float64

1058	FAL ASE:		DATA TYPE ASE
1059	CLASS:		DATA TYPE
1060	CLASS ID:	5	
1061	PARENT CLASS:	TOP	
1062	ATTRIBUTES:		
1063	1	Data Type Numeric Identifier	= 15
1064	2	Data Type Name	= Float64
1065	3	Format	= FIXED LENGTH
1066	4.1	Octet Length	= 8

1067 This type has a length of eight octets. The format for float64 is that defined by IEC 60559 as
1068 double precision.

1069 **5.3.1.3.2 Integer types**1070 **5.3.1.3.2.1 Integer8**

1071	FAL ASE:		DATA TYPE ASE
1072	CLASS:		DATA TYPE
1073	CLASS ID:	5	
1074	PARENT CLASS:	TOP	
1075	ATTRIBUTES:		
1076	1	Data Type Numeric Identifier	= 2
1077	2	Data Type Name	= Integer8
1078	3	Format	= FIXED LENGTH
1079	4.1	Octet Length	= 1

1080 This integer type is a two's complement binary number with a length of one octet.

1081 **5.3.1.3.2.2 SINT**

1082 This IEC 61131-3 type is the same as Integer8.

1083 **5.3.1.3.2.3 char**

1084 This data type is the same as Integer8.

1085 **5.3.1.3.2.4 Integer16**

1086	FAL ASE:		DATA TYPE ASE
1087	CLASS:		DATA TYPE
1088	CLASS ID:	5	
1089	PARENT CLASS:	TOP	
1090	ATTRIBUTES:		
1091	1	Data Type Numeric Identifier	= 3
1092	2	Data Type Name	= Integer16
1093	3	Format	= FIXED LENGTH
1094	4.1	Octet Length	= 2

1095 This integer type is a two's complement binary number with a length of two octets.

1096 **5.3.1.3.2.5 INT**

1097 This IEC 61131-3 type is the same as Integer16.

1098 **5.3.1.3.2.6 Integer32**

1099	FAL ASE:		DATA TYPE ASE
1100	CLASS:		DATA TYPE
1101	CLASS ID:	5	
1102	PARENT CLASS:	TOP	
1103	ATTRIBUTES:		
1104	1	Data Type Numeric Identifier	= 4
1105	2	Data Type Name	= Integer32
1106	3	Format	= FIXED LENGTH
1107	4.1	Octet Length	= 4

1108 This integer type is a two's complement binary number with a length of four octets.

1109 **5.3.1.3.2.7 DINT**

1110 This IEC 61131-3 type is the same as Integer32.

1111 **5.3.1.3.2.8 Integer64**

1112	FAL ASE:		DATA TYPE ASE
1113	CLASS:		DATA TYPE
1114	CLASS ID:	5	
1115	PARENT CLASS:	TOP	
1116	ATTRIBUTES:		
1117	1	Data Type Numeric Identifier	= 55
1118	2	Data Type Name	= Integer64
1119	3	Format	= FIXED LENGTH
1120	4.1	Octet Length	= 8

1121 This integer type is a two's complement binary number with a length of eight octets.

1122 **5.3.1.3.2.9 LINT**

1123 This IEC 61131-3 type is the same as Integer64.

1124 **5.3.1.3.3 Unsigned types**1125 **5.3.1.3.3.1 Unsigned8**1126 **FAL ASE:** DATA TYPE ASE1127 **CLASS:** DATA TYPE1128 **CLASS ID:** 51129 **PARENT CLASS:** TOP1130 **ATTRIBUTES:**

1131	1	Data Type Numeric Identifier	=	5
1132	2	Data Type Name	=	Unsigned8
1133	3	Format	=	FIXED LENGTH
1134	4.1	Octet Length	=	1

1135 This type is a binary number. The most significant bit of the most significant octet is always
 1136 used as the most significant bit of the binary number; no sign bit is included. This type has a
 1137 length of one octet.

1138 **5.3.1.3.3.2 USINT**

1139 This IEC 61131-3 type is the same as Unsigned8.

1140 **5.3.1.3.3.3 Unsigned16**1141 **FAL ASE:** DATA TYPE ASE1142 **CLASS:** DATA TYPE1143 **CLASS ID:** 51144 **PARENT CLASS:** TOP1145 **ATTRIBUTES:**

1146	1	Data Type Numeric Identifier	=	6
1147	2	Data Type Name	=	Unsigned16
1148	3	Format	=	FIXED LENGTH
1149	4.1	Octet Length	=	2

1150 This type is a binary number. The most significant bit of the most significant octet is always
 1151 used as the most significant bit of the binary number; no sign bit is included. This unsigned
 1152 type has a length of two octets.

1153 **5.3.1.3.3.4 UINT**

1154 This IEC 61131-3 type is the same as Unsigned16.

1155 **5.3.1.3.3.5 Unsigned32**1156 **FAL ASE:** DATA TYPE ASE1157 **CLASS:** DATA TYPE1158 **CLASS ID:** 51159 **PARENT CLASS:** TOP1160 **ATTRIBUTES:**

1161	1	Data Type Numeric Identifier	=	7
1162	2	Data Type Name	=	Unsigned32
1163	3	Format	=	FIXED LENGTH
1164	4.1	Octet Length	=	4

1165 This type is a binary number. The most significant bit of the most significant octet is always
 1166 used as the most significant bit of the binary number; no sign bit is included. This unsigned
 1167 type has a length of four octets.

1168 **5.3.1.3.3.6 UDINT**

1169 This IEC 61131-3 type is the same as Unsigned32.

1170 **5.3.1.3.3.7 Unsigned64**1171 **FAL ASE:** DATA TYPE ASE1172 **CLASS:** DATA TYPE1173 **CLASS ID:** 51174 **PARENT CLASS:** TOP1175 **ATTRIBUTES:**

1176	1	Data Type Numeric Identifier	=	56
1177	2	Data Type Name	=	Unsigned64
1178	3	Format	=	FIXED LENGTH
1179	4.1	Octet Length	=	8

1180 This type is a binary number. The most significant bit of the most significant octet is always
 1181 used as the most significant bit of the binary number; no sign bit is included. This unsigned
 1182 type has a length of eight octets.

1183 5.3.1.3.3.8 ULINT

1184 This IEC 61131-3 type is the same as Unsigned64.

1185 5.3.2 String types

1186 5.3.2.1 BitString

1187	FAL ASE:	DATA TYPE ASE
1188	CLASS:	DATA TYPE
1189	CLASS ID:	5
1190	PARENT CLASS:	TOP
1191	ATTRIBUTES:	
1192	1	Data Type Numeric Identifier = 14
1193	2	Data Type Name = BitString
1194	3	Format = STRING
1195	5.1	Octet Length = 1 to n

1196 This string type is defined as a series of BitString8 elements.

1197 5.3.2.2 OctetString

1198	FAL ASE:	DATA TYPE ASE
1199	CLASS:	DATA TYPE
1200	CLASS ID:	5
1201	PARENT CLASS:	TOP
1202	ATTRIBUTES:	
1203	1	Data Type Numeric Identifier = 10
1204	2	Data Type Name = OctetString
1205	3	Format = STRING
1206	4.1	Octet Length = 1 to n

1207 An OctetString is an ordered sequence of octets, numbered from 1 to n. Octet 1 of the
 1208 sequence is referred to as the first octet. IEC 61158-6-11 defines the order of transmission.

1209 5.3.2.3 VisibleString

1210	FAL ASE:	DATA TYPE ASE
1211	CLASS:	DATA TYPE
1212	CLASS ID:	5
1213	PARENT CLASS:	TOP
1214	ATTRIBUTES:	
1215	1	Data Type Numeric Identifier = 9
1216	2	Data Type Name = VisibleString
1217	3	Format = STRING
1218	4.1	Octet Length = 1 to n

1219 This type is defined as the ISO/IEC 646 string type.

1220 5.4 Data type ASE service specification

1221 There are no operational services defined for the type object.

6 Communication model specification

6.1 General

This standard specifies the application layer service of Type 26 fieldbus essential for the ISO/IEC 8802-3 based FA Link network (FL-net²), which will be incorporated as one of the communication networks for the Real-Time Ethernet (RTE) defined in IEC 61784-2:201X.

The Type 26 fieldbus meets the industrial automation market objective of providing predictable time deterministic and reliable time-critical data transfer and means, which allow co-existence with non-time-critical data transfer over the ISO/IEC 8802-3 series communications medium, for support of cooperation and synchronization between automation processes on field devices in a real-time application system. 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.

This standard specifies the Application layer service part of the protocol set of the communication profile and/or of one or more communication profiles related to the common family of CPF 21.

6.2 Protocol stack for type 26 fieldbus

The application layer exploits the services of the data-link layer immediately below, in terms of the “three-layer” Fieldbus Reference Model which is based in part on the OSI Basic Reference Model. The OSI reference model provides a layered approach to communications standards, whereby the layers can be developed and modified independently. IEC 61158 specifies functionality from top to bottom of a full OSI stack and, potentially, some functions for the users of the stack.

Figure 1 shows the protocol stack for a type 26 fieldbus, the OSI layers and the equivalent layers in the IEC 61158 basic fieldbus reference model.

Functions of the intermediate OSI layers, layers 3 through 6, are consolidated into the IEC 61158 data-link layer for a type 26 fieldbus as the lower layer of the FA Link protocol. The data-link layer is that of a general-purpose Ethernet with UDP (RFC 768), TCP (RFC 793), IP (RFC 791) and Ethernet layer (RFC 894, ISO/IEC 8802-3 or IEEE 802.3 MAC) as the main components. Type 26 fieldbus utilizes that of the services provided with a general-purpose Ethernet protocol suite.

The data-link layer for type 26 fieldbus shall specify a broadcast communication function as the mandatory fundamental function of the lower layer, and any other main function is not specified as of mandatory for the lower layer protocol.

The application layer for Type 26 fieldbus is the main component of the FA Link protocol stack, and is specified as the upper layer of the FA Link protocol. The application layer is mapped directly on the data-link layer.

Equivalent function of the OSI layer 5 and 6 are not used for a type 26 fieldbus. Features common to users of the fieldbus application layer are provided by the IEC 61158 application layer to simplify user operation. This standard defines the service elements and the services, and the protocol specification is defined in IEC 61158-6-26:—.

² FL-net is the trade name of JEMA/FL-net: The Japan Electrical Manufacturers' Association / the Factory Automation Link network. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance does not require use of the trade name. Use of the trade name requires permission of the trade name holder.

1262

OSI layer		IEC 61158 layer
7	Application	Application IEC 61158-5-26 IEC 61158-6-26
6	Presentation	
5	Session	
4	Transport	Data-link RFC 768, (RFC 793)
3	Network	RFC 791 RFC 919, RFC 922 (RFC 792, RFC 950)
2	Data-link	RFC 894/ ISO/IEC 8802-3 MAC (RFC 826)
1	Physical	Physical ISO/IEC 8802-3 Phy

1263

Figure 1 – Protocol stack for type 26 fieldbus

1264

1265 **6.3 Overview of type 26 communication model**

1266 Two types of communication service models are used in FAL type 26; i.e. the cyclic data- and
1267 the message data-communication service models as follows:

1268 a) Cyclic data communication service model:

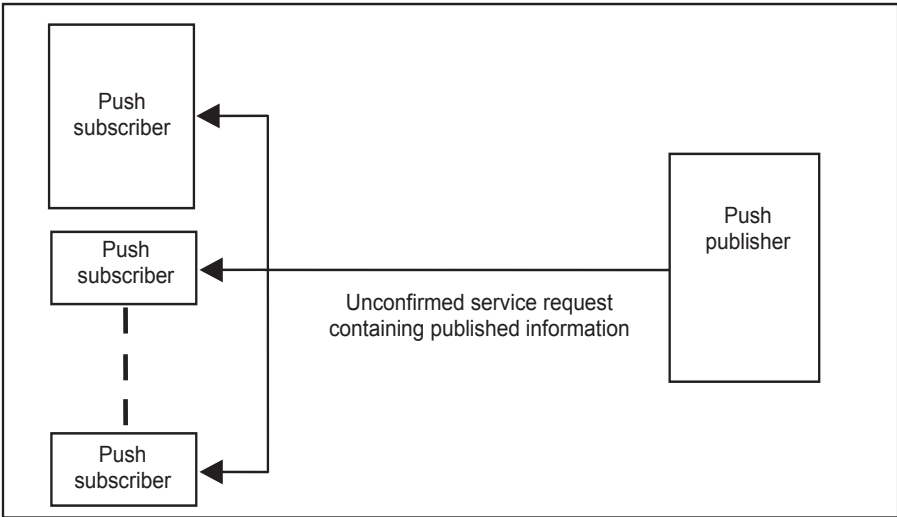
1269 Communication service model using the n : n-type distributed-shared-virtual memory with
1270 a common pseudo logical address among the type 26 nodes over a type 26 fieldbus
1271 network, is shown in Figure 2.

1272 This communication service is performed periodically with the Unconfirmed Push-
1273 Publisher/Subscriber type communications on a buffered AREP;

1274 b) Message data communication service model:

1275 Communication service model using the Unconfirmed/Confirmed Client/Server
1276 communications with Push type on queued AREPs is shown in Figure 3.

1277



1278

1279

Figure 2 – Unconfirmed Push-Publisher/Subscriber type interaction

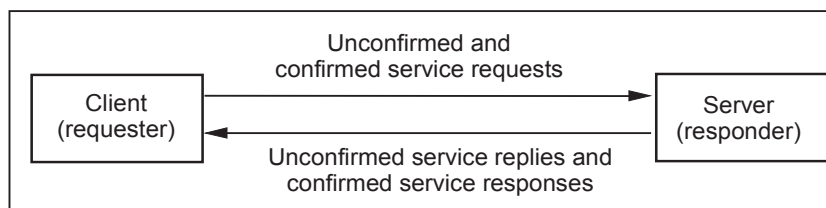


Figure 3 – Unconfirmed/Confirmed Client/Server type interaction

6.4 Cyclic data communication service with Common-memory

6.4.1 Overview

The FAL defined in this standard identifies a peer based controller communications network supporting a Common-memory model.

The type 26 fieldbus is applicable to a controller level network for efficiently executing the control and state data exchange between various kinds of field devices such as PLCs (Programmable Logic Controllers), RCs (Robot Controllers), NCs (Numerical Controllers), FA (Factory Automation) controllers, DCSs (Distributed Control systems), PCs (Personal Computers) for the control purpose and the human-machine interfaces and so forth, in industrial automation control systems in which a time-criticality for their data exchanges is required.

The communications between these field devices also requires simplicity in actual application programming as well as the time-critical transfer of the control and state data required among these field devices.

The Common-memory (CM) provides a function to share the control and state data seamlessly across an industrial control network system.

6.4.2 Common-memory: allocation to each node

The CM is a virtual common memory accessible with logically unique address across the type 26 nodes in a type 26 fieldbus network, and the APs running over each type 26 node can handle the CM as a global common memory shared among the type 26 nodes over the type 26 fieldbus network.

Cyclic-data transmission is used for refreshing and equalizing the contents of the CM among the type 26 nodes by means of a cyclic broadcasting performed at a constant rate after a type 26 node writes a data piece on a pre-allocated memory area of the CM for the type 26 node.

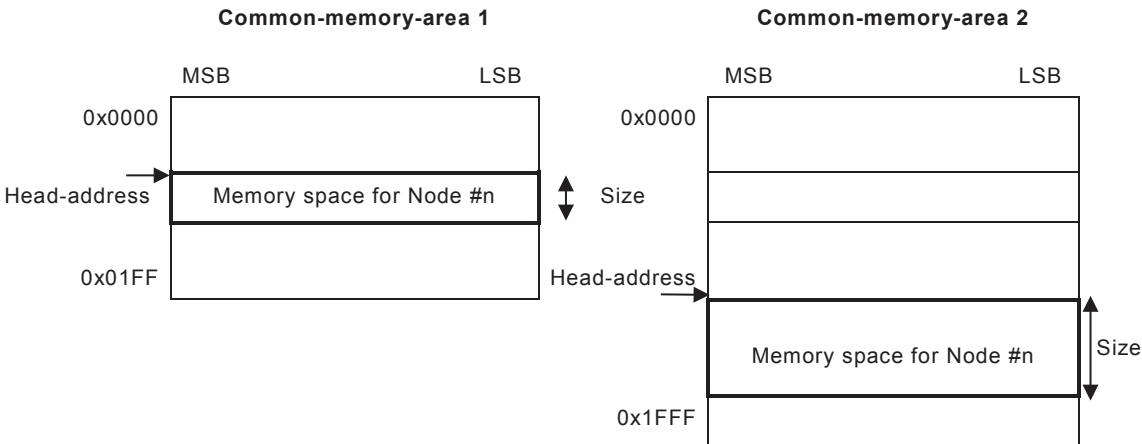
Figure 4 shows the CM which is composed of two memory areas; i.e. the memory-area-1 and -2 with the memory size of 512 words and 8 192 words respectively. The total memory size of the CM is up to 8 704 words.

The memory area allocated to each node as the output/write area on one of or both of the memory-area-1 and -2, is dependent on the implementation.

Any memory size in word boundary is allocable from 0 (zero) to the maximum space size of the memory-area-1 and/or -2 to each node, and the memory space allocation is designated with the memory size and the head-address. The memory space allocated to each node shall not overlap with each other. And the allocated memory space is identical to the cyclic-data transmission area by the node.

In case of total zero memory space allocated to a node, it means no cyclic-data transmission to be carried out by the node. The provision with the CM at each node is dependent on the implementation of each node, and it is applicable to a mixture system of the nodes with and without the CM.

1323



1324

Figure 4 – Common memory allocation

1325

1326 **6.4.3 Data sharing among nodes with the CM**

1327 The CM, by means of the cyclic-data transmission performed at a constant rate, provides the
1328 data distribution/refreshing function to all type 26 nodes in temporal and spatial coherency, so
1329 that each node can share the same data with each other and can make good use of the data;
1330 i.e. a memory space allocated to each node is a container for application data in general use
1331 and provides flexibility to apply in a variety of industrial application processes.

1332 In operational view of the CM, the memory area in the CM allocated to a certain node is used
1333 as the transmission area of the node by a cyclic-data transmission as a sender or a Push-
1334 publisher. On the other hand as receiving nodes of the cyclic-data transmission or Push-
1335 subscribers, the memory area of the sender node is used as a receiving area for the data
1336 from the sender node.

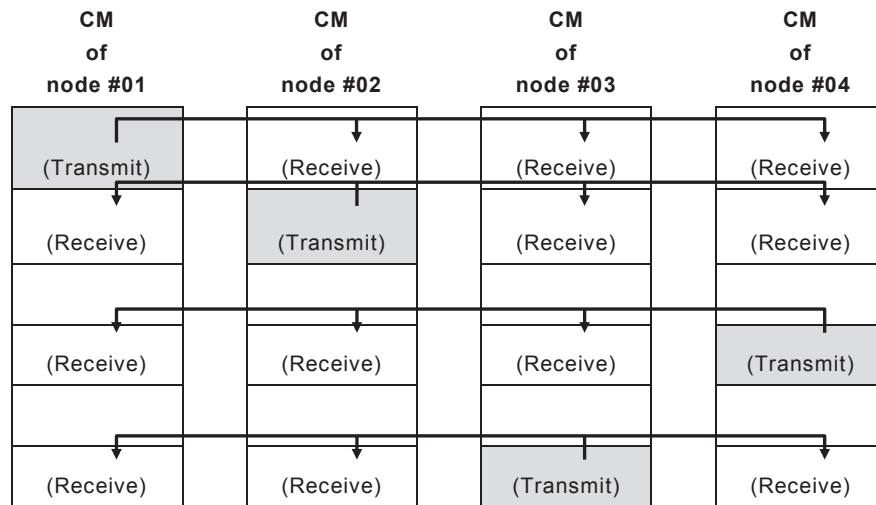
1337 Figure 5 shows the data sharing with the CM.

1338 The memory area or the transmission area by a cyclic-data transmission, allocated on the CM,
1339 shall not be overlapped with each other.

1340 It is available for a node to use the CM only as the receiving area for other nodes. Figure 6
1341 shows an example in which the node #01 is of reception only.

1342 In the case of a node without the CM, all the receiving data by a cyclic-data transmission shall
1343 be discarded. Figure 7 shows an example in which the node #01 is not the recipient or the
1344 Push-subscriber.

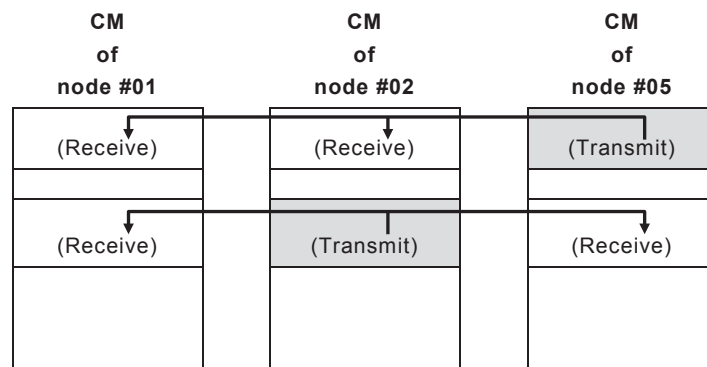
1345



1346

Figure 5 - Data sharing with the CM

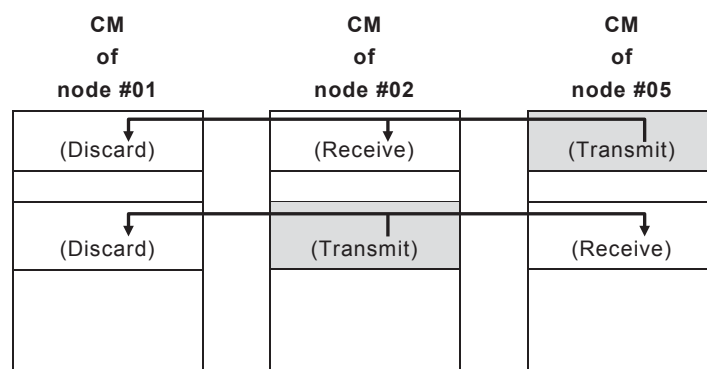
1347



1348

Figure 6 – Node #01 for reception only

1349



1350

Figure 7 – Node #01 without the CM

1351

6.4.4 CM data type

The data type applied to the CM is primarily based on the basic data type defined in Clause 5. Using these data types, Array or Structure is built and applied.

6.5 ASEs

6.5.1 Overview of type 26 ASEs

The FAL ASEs and their architectural relationships for type 26 FAL is shown in Figure 8.

Each ASE provides the services as follows:

- a) The cyclic-data ASE provides the cyclic data communication service with type 26 specific common-memory defined in 6.4 of IEC 61158-5-26:—, and the cyclic-data transmission is performed at a constant rate for updating the common-memory;
- b) Message data ASE provides the message-data communication service between nodes and the message-data transmission is performed on demand by the ALS-user in order to intercommunicate between the APs running on remote nodes;
- c) The load measurement ASE provides services to conduct the start and the termination of the communication load measurement on the target node, and provides the measurement result containing the token holding time related and the general purpose communication load related statistical information on that node available for analyzing the total communication load condition over the type 26 fieldbus network;
- d) The General purpose command server ASE provides the general purpose command server communication service, and provides the facilities for setting and reading the values of various network visible APOs of other nodes, and for requesting the start and the stop of the communication load measurement to other nodes;
- e) The Network management ASE provides services to access type 26 APO attributes; modify the APO instances; configure the nodes; inform the status report on the unexpected events, the errors and the status changes of the nodes; manage network visible APOs accessed through the FAL.

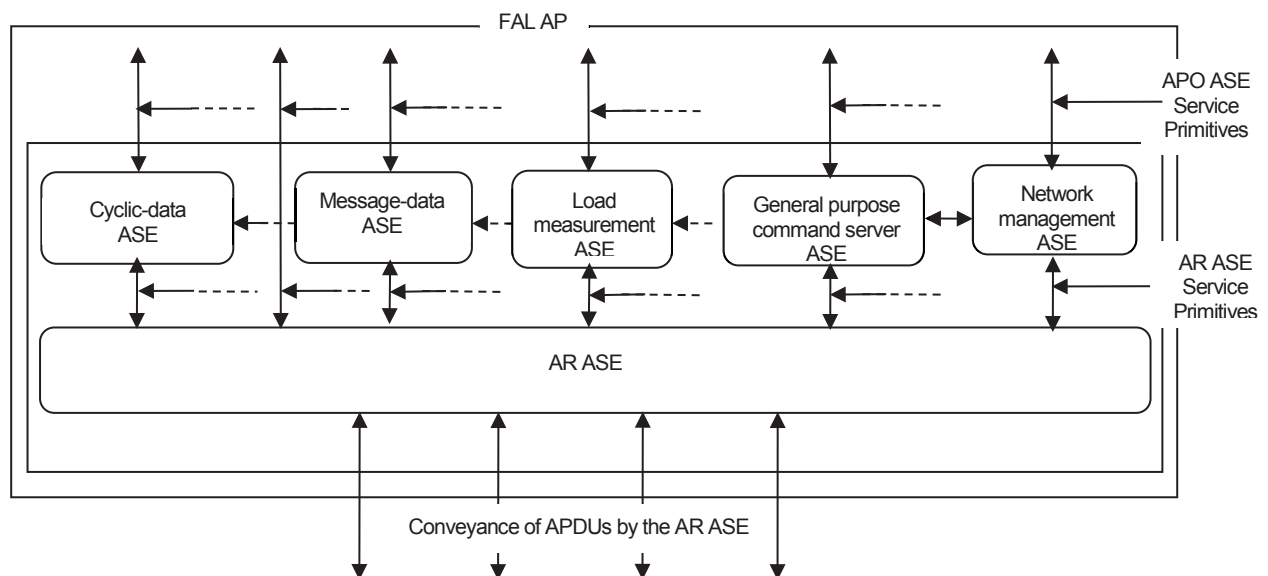


Figure 8 – The structure of ASEs for type 26 FAL

6.5.2 Type 26 specific conventions for FAL service common parameters

The following type 26 FAL service parameters are used commonly in the subclauses in which the corresponding services are defined. The definitions for these service parameters are provided herein, and are referred without the definitions throughout the remainder of this standard, otherwise extended definitions and/or additional information are provided to these parameters in each case.

Argument

This parameter conveys the service specific parameters of the service request.

DNA

This parameter specifies the destination-node-number.

1392 SNA

1393 This parameter indicates the source-node-number.

1394 TCD

1395 This attribute specifies the transaction identification to be used to convey the service and to
1396 identify the context of the application relationship as well as the AREP.

1397 AREP

1398 This attribute indicates information for local identification of the AREP to be used to convey
1399 the service and to identify the AR.

1400 Result (+)

1401 This selection type parameter conveys the service specific parameters of the service
1402 response, and indicates that the service request is successfully carried out.

1403 M_RLT

1404 This parameter contains the message result by the responder. The value is:

1405 – “Success – Successfully completed”.

1406 Result (–)

1407 This selection type parameter conveys the service specific parameters of the service
1408 response, and indicates that the service request is failed.

1409 Error info

1410 This parameter conveys the error information on Result (–) of the service response. The
1411 value contains:

1412 – “Failure – invalid requested parameter”;

1413 – “Failure –service unimplemented”;

1414 – “Failure –with other reason”.

1415 6.5.3 Cyclic-data ASE**1416 6.5.3.1 Overview**

1417 The cyclic-data ASE represents the cyclic data communication service with common-memory
1418 in 6.4, and the cyclic-data transmission is performed at a constant rate for updating the
1419 contents of the CM.

1420 6.5.3.2 Common memory model class specification**1421 6.5.3.2.1 Overview**

1422 The CM ASE provides the write and read data services to/from the CM for the ALS-user in
1423 order to intercommunicate between the APs running on remote nodes.

1424 6.5.3.2.2 Common memory class formal model

1425	FAL ASE:		CM ASE
1426	CLASS:		CM
1427	CLASS ID:		not used
1428	PARENT CLASS:		TOP
1429	ATTRIBUTES:		
1430	1	(o) Key Attribute:	Not used
1431	2	(o) Key Attribute:	TCD
1432	3	(m) Attribute:	State
1433	4	(m) Attribute:	AREP
1434	5	(m) Attribute:	Network address
1435	6	(m) Attribute:	Node number
1436	7	(m) Attribute:	CM (Area 1, Area 2)
1437	8	(m) Attribute:	CM_SZ (Size 1, Size 2)
1438	9	(m) Attribute:	Node status
1439	SERVICES:		
1440	1	(m) OpsService:	Write
1441	2	(m) OpsService:	Send-CM
1442	2	(m) OpsService:	Read
1443	3	(m) OpsService:	Update-memory

1444 4 (m) OpsService: Get-buffer

1445 **6.5.3.2.3 Attributes**

1446 **TCD**

1447 This attribute specifies the transaction identification to be used to convey the service and to
1448 identify the context of the application relationship as well as the AREP.

1449 **State**

1450 This attribute indicates the current state of the CM ASE (FSPM) that is defined in detail in
1451 IEC 61158-6-26:—.

1452 **AREP**

1453 This attribute indicates information for local identification of the AREP to be used to convey
1454 the service and to identify the AR.

1455 **Network address**

1456 This attribute identifies the network address of the node.

1457 **Node number**

1458 This attribute identifies the node number of the node.

1459 **CM (Area 1, Area 2)**

1460 This attribute identifies the common-memory-area-1 and -2.

1461 **CM_SZ (Size 1, Size 2)**

1462 This attribute indicates the data-sizes of the common-memory-area-1 and -2.

1463 **Node status**

1464 This attribute identifies the node status of this node.

1465 **6.5.3.2.4 Services**

1466 **Write**

1467 The Write request primitive is used by the ALS-user to update the content of the Common
1468 memory.

1469 **Send-CM**

1470 The Send-CM request primitive is used by the ALS-user to put the updated local Common
1471 memory data on the designated memory area to the send buffer and to send out the data out
1472 of the buffer as a publisher.

1473 **Read**

1474 The Read request primitive is used by the ALS-user to read the content of the Common
1475 memory.

1476 **Update memory**

1477 The Update Memory indication primitive is used to notify the ALS-user that the designated
1478 Common Memory area is updated.

1479 **Get-buffer**

1480 The Get-CM request primitive is used by the ALS-user to get the data out of the receive buffer
1481 in which the updated Common memory data on the designated memory area contains, and to
1482 update the local Common memory data on the designated memory area as a subscriber.

1483 **6.5.3.3 Service specification**

1484 **6.5.3.3.1 Supported services**

1485 The services provided by the CM ASE are as follows:

- 1486 • Write,
- 1487 • Send-CM,
- 1488 • Read,
- 1489 • Update memory,