

DIN EN 61158-5-2

ICS 35.100.70; 35.160

Ersatz für
DIN EN 61158-5-2:2008-09
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 5-2: Dienstfestlegungen des Application Layer
(Anwendungsschicht) –
Typ 2-Elemente
(IEC 61158-5-2:2010);
Englische Fassung EN 61158-5-2:2012, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 5-2: Application layer service definition –
Type 2 elements
(IEC 61158-5-2:2010);
English version EN 61158-5-2:2012, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 5-2: Définition des services des couches d'application –
Éléments de type 2
(CEI 61158-5-2:2010);
Version anglaise EN 61158-5-2:2012, seulement en CD-ROM

Gesamtumfang 203 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Anwendungsbeginn

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

Für DIN EN 61158-5-2:2008-09 gilt eine Übergangsfrist bis zum 2015-03-28.

Nationales Vorwort

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

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

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

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

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

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

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

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

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

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 61158-5-2 veröffentlicht. Zu deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Einführung

Ein Feldbus ist ein digitaler serieller Datenbus für die Kommunikation zwischen Geräten der industriellen Automatisierungstechnik wie beispielsweise Messeinrichtungen, Stellgliedern, Reglern und speicherprogrammierbaren Steuerungen. Die Normen der Reihe DIN EN 61158 legen eine Anzahl von Feldbus-Protokolltypen fest. Die vorliegende Norm legt die Dienste der Anwendungsschicht für Typ 2 fest.

Änderungen

Gegenüber DIN EN 61158-5-2:2008-09 wurden folgende Änderungen vorgenommen:

- a) Abschnitt 2 und die Literaturhinweise auf den neuesten Stand gebracht;
- b) in 6.1.3 neue Objekte berücksichtigt;
- c) in 6.2.1.2.6 das Time Sync ASE-Objekt auf den neuesten Stand gebracht (damit es zur Neuausgabe der IEC 61558 passt);
- d) neuen Unterabschnitt 6.2.1.2.7 eingeführt, der neue Parameter für das ASE-Objekte berücksichtigt;
- e) in 6.2.1.3 Dienste für Time Sync- und Parameter-ASE hinzugefügt bzw. auf den neuesten Stand gebracht;
- f) in 6.2.2.3 Connection Manager ASE-Dienste geringfügig geändert;
- g) in 6.4 den Parameter ASE zur Objekt-Tabelle hinzugefügt;
- h) in 6.5 den Inhalt der Dienstetabelle für die Time Sync- und Parameter ASEs auf den neuesten Stand gebracht.

Frühere Ausgaben

DIN V 19245-1: 1988-01
 DIN 19245-1: 1991-04
 DIN 19245-2: 1991-04
 DIN EN 50170/1: 1997-07
 DIN EN 50170/2: 1997-07
 DIN EN 50170/3: 1997-07
 DIN EN 50254: 1999-07
 DIN EN 50170/A2: 2000-08
 DIN EN 50170/A1: 2002-11
 DIN EN 50170/A3: 2002-11
 DIN EN 61158-5: 2005-05
 DIN EN 61158-5-2: 2008-09

Nationaler Anhang NA (informativ)

3.6 Spezielle Begriffe für die Feldbus-Anwendungsschicht Typ 2

3.6.1

zuweisen

(en: allocate)

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

3.6.2

Anwendung

(en: application)

Funktion oder Datenstruktur, für die Daten verarbeitet oder erzeugt werden

3.6.3

Anwendungsobjekte

(en: application objects)

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

3.6.4

Anwendungsprozess

(en: application process)

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

3.6.5

Anwendungsprozessobjekt

(en: application process object)

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

3.6.6

Klasse von Anwendungsprozessobjekten

(en: application process object class)

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

3.6.7

Anwendungsbeziehung

(en: application relationship)

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

3.6.8

Anwendungsdienstelement für Anwendungsbeziehungen

(en: application relationship application service element)

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

3.6.9

Endpunkt der Anwendungsbeziehung

(en: application relationship endpoint)

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

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

3.6.10

Attribut

(en: attribute)

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

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

3.6.11

Verhalten

(en: behavior)

Angabe, wie ein Objekt auf bestimmte Ereignisse reagiert

3.6.12**Grenzuhr****(en: boundary clock)**

Uhr mit mehr als einem einzigen PTP-Port, wobei jeder PTP-Port Zugang zu einem separaten PTP-Kommunikationspfad bereitstellt

ANMERKUNG Grenzuhren werden eingesetzt, um Schwankungen, die durch Router und ähnliche Netzwerkelemente erzeugt werden, zu beseitigen.

3.6.13**Klasse****(en: class)**

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

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

3.6.14**Klassenattribute****(en: class attributes)**

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

3.6.15**Klassencode****(en: class code)**

eindeutige Kennung, die jeder Objektklasse zugewiesen ist

3.6.16**klassenspezifischer Dienst****(en: class specific service)**

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

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

3.6.17**Client****(en: client)**

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

3.6.18**Uhr****(en: clock)**

Gerät, das eine Messung der seit einem festgelegten Bezugszeitpunkt vergangenen Zeit bereitstellt

ANMERKUNG In IEC 61588:2009 sind zwei Typen von Uhren festgelegt: Grenzuhren und gewöhnliche Uhren.

3.6.19**Kommunikationsobjekte****(en: communication objects)**

Komponenten, die einen Austausch von Nachrichten über das Netzwerk zur Laufzeit bereitstellen und verwalten

BEISPIELE Verbindungsmanager-Objekt, UCMM-Objekt und Nachrichtenrouter-Objekt.

3.6.20

Verbindung

(en: connection)

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

ANMERKUNG 1 Verbindungen können entweder Punkt-zu-Punkt-Verbindungen oder Mehrpunktverbindungen sein.

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

3.6.21

Verbindungs-ID

(en: connection ID (CID))

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

3.6.22

Verbindungspfad

(en: connection path)

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

3.6.23

Verbindungspunkt

(en: connection point)

Puffer, der als Unterinstanz eines Assembly-Objekts dargestellt ist

3.6.24

konsumieren

(en: consume)

Vorgang des Empfangs von Daten von einem Produzenten

3.6.25

Konsument

(en: consumer)

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

3.6.26

konsumierende Anwendung

(en: consuming application)

Anwendung, die Daten konsumiert

3.6.27

zyklisch

(en: cyclic)

sich regelmäßig wiederholend

3.6.28

Gerät

(en: device)

physikalische Hardware, die mit dem Verbindungszweig verbunden ist

ANMERKUNG Ein Gerät kann mehr als einen Knoten umfassen.

3.6.29**Geräteprofil****(en: device profile)**

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

3.6.30**Endknoten****(en: end node)**

produzierender oder konsumierender Knoten

3.6.31**Endpunkt****(en: endpoint)**

eine der Kommunikationseinheiten, die zu einer Verbindung gehören

3.6.32**Bezugszeitpunkt****(en: epoch)**

Referenzzeitpunkt, der den Anfang einer Zeitskala festlegt [IEC 61588:2009]

3.6.33**Fehler****(en: error)**

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

3.6.34**Ereignis****(en: event)**

Instanz einer Änderung von Zuständen

3.6.35**Rahmen****(en: frame)**

umgangssprachliches Synonym für DLPDU

3.6.36**Grandmaster-Uhr****(en: grandmaster clock)**

Uhr, die als primäre Zeitquelle dient, mit der alle anderen Uhren (innerhalb einer Zusammenstellung von Uhren nach IEC 61588) letztlich synchronisiert werden

3.6.37**Gruppe****(en: group)**

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

3.6.38**Schnittstelle****(en: interface)**

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

3.6.39

Aufruf

(en: invocation)

Vorgang der Nutzung eines Dienstes oder einer anderen Ressource eines Anwendungsprozesses

ANMERKUNG Jeder Aufruf stellt einen separaten Kontrollthread dar, der durch seinen Kontext beschrieben werden kann. Sobald der Dienst ausgeführt oder die Nutzung der Ressource beendet ist, erlischt der Aufruf. Bei Dienstaufrufen wird ein Dienst, der initiiert, aber noch nicht ausgeführt wurde, als ausstehender Dienstaufruf bezeichnet. Bei Dienstaufrufen kann außerdem eine Aufruf-ID verwendet werden, um den Dienstaufruf eindeutig zu identifizieren und ihn von anderen ausstehenden Dienstaufrufen zu unterscheiden.

3.6.40

Instanz

(en: instance)

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

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

ANMERKUNG Die Benennungen „Objekt“, „Instanz“ und „Objektinstanz“ beziehen sich auf eine spezifische Instanz.

3.6.41

Instanzattribut

(en: instance attributes)

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

3.6.42

instanziiert

(en: instantiated)

Objekt, das in einem Gerät erzeugt wurde

3.6.43

logisches Gerät

(en: logical device)

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

3.6.44

LPaket (oder Link-Paket)

(en: Lpacket (or Link packet))

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

ANMERKUNG Die Sicherungsschichten von Peers verwenden LPakete, um Dienstdateneinheiten an höhere Schichten im OSI-Stapel zu senden und um Dienstdateneinheiten von diesen zu empfangen.

3.6.45

Verwaltungsinformationen

(en: management information)

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

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

3.6.46

Master-Uhr

(en: master clock)

einzelne Uhr, die innerhalb der einzelnen Regionen als primäre Zeitquelle dient und die wiederum mit anderen Master-Uhren und letztlich mit der Grandmaster-Uhr synchronisiert wird

ANMERKUNG Ein System von Uhren nach IEC 61588 kann in Regionen aufgeteilt werden, die mithilfe von Grenzuhrn voneinander abgegrenzt werden.

3.6.47

Member

(en: member)

Teil eines Attributes, das als Element eines Feldes strukturiert ist

3.6.48

Nachrichtenrouter

(en: Message Router)

Objekt innerhalb eines Knotens, das Nachrichtenanforderungen an entsprechende Anwendungsobjekte verteilt

3.6.49

Methode

(en: method)

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

3.6.50

Modul

(en: module)

Hardware oder logische Komponente eines physikalischen Gerätes

3.6.51

Mehrpunktverbindung

(en: multipoint connection)

Verbindung von einem Knoten zu vielen Knoten

ANMERKUNG Durch Mehrpunktverbindungen können Nachrichten von einem einzelnen Produzenten von vielen Konsumentenknuten empfangen werden.

3.6.52

Netzwerk

(en: network)

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

3.6.53

Objekt

(en: object)

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

3.6.54

objektspezifischer Dienst

(en: object specific service)

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

3.6.55

gewöhnliche Uhr

(en: ordinary clock)

Uhr nach IEC 61588 mit einem einzigen PTP-Port

3.6.56

Absender

(en: originator)

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

3.6.57

Peer

(en: peer)

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

3.6.58

physikalisches Gerät

(en: physical device)

Automatisierungsgerät oder anderes Netzwerkgerät

3.6.59

Punkt-zu-Punkt-Verbindung

(en: point-to-point connection)

Verbindung, die zwischen genau zwei Anwendungsobjekten besteht

3.6.60

Precision Time Protocol

(PTP)

(en: Precision Time Protocol (PTP))

Name für das Protokoll zur Zeitsynchronisation

3.6.61

produzieren

(en: produce)

Vorgang des Sendens von Daten zum Empfang durch einen Konsumenten

3.6.62

Produzent

(en: producer)

Knoten, der für das Senden von Daten verantwortlich ist

3.6.63

Eigenschaft

(en: property)

allgemeine Bezeichnung für beschreibende Informationen zu einem Objekt

3.6.64

PTP-Nachricht

(en: PTP message)

Nachricht zur Zeitsynchronisation nach IEC 61588

ANMERKUNG In IEC 61588:2009 werden fünf vorgesehene Nachrichtentypen definiert: Sync, Delay_Req, Follow-up, Delay_Resp, und Management.

3.6.65

PTP-Port

(en: PTP port)

logischer Zugangspunkt für Kommunikationen nach IEC 61588 zu der Uhr, die den Port enthält

3.6.66

Ressource

(en: resource)

Verarbeitungs- oder Informationsfähigkeit eines Subsystems

3.6.67

Seriennummer

(en: serial number)

eindeutiger ganzzahliger 32-Bit-Wert, der jedem Gerät mit Typ-2-Kommunikationsfähigkeiten durch den jeweiligen Hersteller/Anbieter zugewiesen wird

ANMERKUNG Anbieter-ID und Seriennummer bilden zusammen eine eindeutige Kennung für jedes Gerät.

3.6.68

Server

(en: server)

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

3.6.69

Dienst

(en: service)

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

3.6.70

synchronisierte Uhren

(en: synchronized clocks)

(mit einer gewissen Messunsicherheit) zwei Uhren, die denselben Bezugszeitpunkt haben und bei denen Messungen eines beliebigen Zeitintervalls mit beiden Uhren nicht mehr als die angegebene Messunsicherheit voneinander abweichen

ANMERKUNG Die von zwei synchronisierten Uhren generierten Zeitstempel für das gleiche Ereignis dürfen nicht mehr als die angegebene Messunsicherheit voneinander abweichen.

3.6.71

Systemzeit

(en: System Time)

absoluter Zeitwert, wie für die Zeitsynchronisation nach Kommunikationsprofilfamilie 2 festgelegt, im Kontext eines verteilten Systems, in dem alle Geräte über lokale Uhren verfügen, die mit einer gemeinsamen Master-Uhr synchronisiert werden

ANMERKUNG Im Zusammenhang mit der Kommunikationsprofilfamilie 2 entspricht die Systemzeit einem ganzzahligen 64-Bit-Wert in Nanosekunden mit einem Wert von 0 entsprechend dem Datum 01.01.1972.

3.6.72

Ziel

(en: target)

Endknoten, mit dem eine Verbindung aufgebaut wird

3.6.73

Unconnected Message Manager (UCMM)

(en: Unconnected Message Manager (UCMM))

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

3.6.74

verbindungsloser Dienst

(en: unconnected service)

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

3.6.75

Anbieter-ID

(en: vendor ID)

Kennzeichnung jedes Produktherstellers/Anbieters durch eine eindeutige Nummer

ANMERKUNG Anbieter-IDs werden von der ODVA, Inc. vergeben.

– Leerseite –

English version

**Industrial communication networks -
Fieldbus specifications -
Part 5-2: Application layer service definition -
Type 2 elements
(IEC 61158-5-2:2010)**

Réseaux de communication industriels -
Spécifications des bus de terrain -
Partie 5-2: Définition des services des
couches d'application -
Eléments de type 2
(CEI 61158-5-2:2010)

Industrielle Kommunikationsnetze -
Feldbusse -
Teil 5-2: Dienstfestlegungen des
Application Layer (Anwendungsschicht) -
Typ 2-Elemente
(IEC 61158-5-2:2010)

This European Standard was approved by CENELEC on 2012-03-28. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

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

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 65C/606/FDIS, future edition 2 of IEC 61158-5-2, prepared by SC 65C, "Industrial networks", of IEC/TC 65, "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61158-5-2:2012.

The following dates are fixed:

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

This document supersedes EN 61158-5-2:2008.

EN 61158-5-2:2012 includes the following significant technical changes with respect to EN 61158-5-2:2008:

- Clause 2 and Bibliography: update of normative and bibliographic references;
- subclause 6.1.3: update lists with new objects
- subclause 6.2.1.2.6: update of the Time Sync ASE/object (to match new EN 61558)
- new subclause 6.2.1.2.7: new Parameter ASE/object
- subclause 6.2.1.3: update/add services for Time Sync and Parameter ASEs
- subclause 6.2.2.3: minor updates to the Connection Manager ASE services
- subclause 6.4: add Parameter ASE to the object table
- subclause 6.5: update contents of service table for Time Sync and Parameter ASEs

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

Endorsement notice

The text of the International Standard IEC 61158-5-2:2010 was approved by CENELEC as a European Standard without any modification.

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

IEC 61131-1	NOTE Harmonized as EN 61131-1.
IEC 61158-2:2010	NOTE Harmonized as EN 61158-2:2010 (not modified).
IEC 62026-3	NOTE Harmonized as EN 62026-3.

CONTENTS

INTRODUCTION	6
1 Scope	7
1.1 Overview	7
1.2 Specifications	8
1.3 Conformance	8
2 Normative references	8
3 Terms, definitions, symbols, abbreviations and conventions	9
3.1 ISO/IEC 7498-1 terms	9
3.2 ISO/IEC 8822 terms	9
3.3 ISO/IEC 9545 terms	9
3.4 ISO/IEC 8824 terms	10
3.5 Type 2 fieldbus data-link layer terms	10
3.6 Type 2 fieldbus application-layer specific definitions	10
3.7 Type 2 abbreviations and symbols	17
3.8 Conventions	18
4 Common concepts	20
5 Data type ASE	20
5.1 General	20
5.2 Formal definition of data type objects	20
5.3 FAL defined data types	21
5.4 Data type ASE service specification	30
6 Communication model specification	30
6.1 Concepts	30
6.2 ASEs	38
6.3 ARs	151
6.4 Summary of FAL classes	187
6.5 Permitted FAL services by AR type	188
Bibliography	189
Annex ZA (normative) Normative references to international publications with their corresponding European publications	190
Figure 1 – Overview of ASEs and object classes	32
Figure 2 – Addressing format using MAC, class, instance and attribute IDs	33
Figure 3 – Identity object state transition diagram	49
Figure 4 – Static Assembly state transition diagram	54
Figure 5 – Dynamic Assembly state transition diagram	55
Figure 6 – Typical timing relationships for acknowledged data production	64
Figure 7 – Example of a COS system with two acking devices	64
Figure 8 – Message flow in COS connection – one Connection object, one consumer	64
Figure 9 – Message flow in COS connection – multiple consumers	65
Figure 10 – CPF2 time synchronization offset clock model	76
Figure 11 – CPF2 time synchronization system with offset clock model	77
Figure 12 – CPF2 time synchronization group startup sequence	79
Figure 13 – Parameter object state transition diagram	85

Figure 14 – Example of Find_Next_Object_Instance service	110
Figure 15 – Transmission trigger timer	145
Figure 16 – Inactivity watchdog timer	146
Figure 17 – Using tools for configuration	147
Figure 18 – Production inhibit timer	148
Figure 19 – Context of transport services within the connection model	154
Figure 20 – Application-to-application view of data transfer	154
Figure 21 – Data flow diagram for a link producer	155
Figure 22 – Data flow diagram for a link consumer	156
Figure 23 – Triggers	157
Figure 24 – Binding transport instances to the producer and consumer of a transport connection that does not have a reverse data path	158
Figure 25 – Binding transport instances to the producers and consumers of a transport connection that does have a reverse data path	158
Figure 26 – Binding transport instances to the producer and consumers of a multipoint connection when the transport connection does not have a reverse data path	159
Figure 27 – Binding transport instances to the producers and consumers of a multipoint connection when the transport connection does have reverse data paths	159
Table 1 – Valid IANA MIB printer codes for character set selection	29
Table 2 – Common elements	35
Table 3 – ST language elements	36
Table 4 – Type conversion operations	36
Table 5 – Values of implementation-dependent parameters	37
Table 6 – Extensions to IEC 61131-3:2003	38
Table 7 – Identity object state event matrix	50
Table 8 – Static Assembly state event matrix	54
Table 9 – Dynamic Assembly state event matrix	55
Table 10 – Message Router object Forward_Open parameters	58
Table 11 – Acknowledge Handler object state event matrix	61
Table 12 – Producing I/O application object state event matrix	62
Table 13 – Profile identification	74
Table 14 – Profile default settings and ranges	74
Table 15 – Default PTP clock settings	75
Table 16 – Hand_Set clock quality management	75
Table 17 – Parameter object state event matrix	85
Table 18 – Status codes	87
Table 19 – Get_Attribute_All service parameters	90
Table 20 – Set_Attribute_All service parameters	91
Table 21 – Get_Attribute_List service parameters	93
Table 22 – Set_Attribute_List service parameters	95
Table 23 – Reset service parameters	97
Table 24 – Start service parameters	99
Table 25 – Stop service parameters	101

Table 26 – Create service parameters.....	102
Table 27 – Delete service parameters	104
Table 28 – Get_Attribute_Single service parameters	105
Table 29 – Set_Attribute_Single service parameters	107
Table 30 – Find_Next_Object_Instance service parameters	108
Table 31 – NOP service parameters.....	111
Table 32 – Apply_Attributes service parameters.....	112
Table 33 – Save service parameters	114
Table 34 – Restore service parameters.....	115
Table 35 – Group_Sync service parameters.....	117
Table 36 – Add_AckData_Path service parameters.....	119
Table 37 – Remove_AckData_Path service parameters	120
Table 38 – Get-Enum_String service parameters	121
Table 39 – CM_Open service parameters	129
Table 40 – CM_Close service parameters.....	131
Table 41 – CM_Unconnected_Send service parameters	133
Table 42 – CM_Get_Connection_Data service parameters	134
Table 43 – CM_Search_Connection_Data service parameters.....	136
Table 44 – CM_Get_Connection_Data service parameters	137
Table 45 – I/O Connection object attribute access.....	142
Table 46 – Bridged Connection object attribute access.....	143
Table 47 – Explicit messaging object attribute access	144
Table 48 – Connection_Bind service parameters.....	149
Table 49 – Service_Name service parameters	150
Table 50 – How production trigger, transport class, and CM_RPI determine when data is produced	153
Table 51 – Transport classes	163
Table 52 – UCMM_Create service parameters.....	180
Table 53 – UCMM_Delete service parameters	181
Table 54 – UCMM_Write service parameters	182
Table 55 – UCMM_Abort service parameters.....	183
Table 56 – TR_Write service parameters	184
Table 57 – TR_Trigger service parameters.....	184
Table 58 – TR_Packet_arrived service parameters	185
Table 59 – TR_Ack_received service parameters.....	186
Table 60 – TR_Verify service parameters.....	186
Table 61 – TR_Status_updated service parameters	187
Table 62 – FAL class summary	187
Table 63 – FAL services by AR type.....	188

INTRODUCTION

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

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELD BUS SPECIFICATIONS –

Part 5-2: Application layer service definition – Type 2 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 2 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the Type 2 fieldbus application layer in terms of

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

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

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

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

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 the Type 2 application layer services as defined in this standard.

2 Normative references

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

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

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

IEC/TR 61158-1:2010¹, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-3-2:2007, *Industrial communication networks – Fieldbus specifications - Part 3-2: Data-link layer service definition – Type 2 elements*

IEC 61158-4-2:2010¹, *Industrial communication networks – Fieldbus specifications – Part 4-2: Data-link layer protocol specification – Type 2 elements*

IEC 61158-6-2:2010¹, *Industrial communication networks – Fieldbus specifications – Part 6-2: Application layer protocol specification – Type 2 elements*

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

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

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

¹ To be published.

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

ISO/IEC 8859 (all parts), *Information technology – 8-bit single-byte coded graphic character sets*

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

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

ISO/IEC 10646, *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*

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

ISO 11898:19932, *Road vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication*

IETF RFC 1759, *Printer MIB*, available at <<http://www.ietf.org>>

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 ISO/IEC 7498-1 terms

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

3.2 ISO/IEC 8822 terms

- a) abstract syntax
- b) presentation context

3.3 ISO/IEC 9545 terms

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

² A newer edition of this standard has been published, but only the cited edition applies.

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

3.4 ISO/IEC 8824 terms

- a) object identifier
- b) type

3.5 Type 2 fieldbus data-link layer terms

The following terms, defined in IEC 61158-3-2:2007 and IEC 61158-4-2:2010, apply.

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

3.6 Type 2 fieldbus application-layer specific definitions

3.6.1

allocate

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

3.6.2

application

function or data structure for which data is consumed or produced

3.6.3

application objects

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

3.6.4

application process

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

3.6.5

application process object

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

3.6.6

application process object class

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

3.6.7

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

3.6.8

application relationship application service element

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

3.6.9

application relationship endpoint

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

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

3.6.10

attribute

description of an externally visible characteristic or feature of an object

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

3.6.11

behavior

indication of how an object responds to particular events

3.6.12

boundary clock

clock with more than a single PTP port, with each PTP port providing access to a separate PTP communication path

NOTE Boundary clocks are used to eliminate fluctuations produced by routers and similar network elements.

3.6.13

class

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

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

3.6.14

class attributes

attribute that is shared by all objects within the same class

3.6.15

class code

unique identifier assigned to each object class

3.6.16

class specific service

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

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

3.6.17

client

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

3.6.18

clock

device providing a measurement of the passage of time since a defined epoch

NOTE There are two types of clocks in IEC 61588:2009, boundary clocks and ordinary clocks.

3.6.19

communication objects

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

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

3.6.20

connection

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

NOTE 1 Connections may be either point-to-point or multipoint.

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

3.6.21

connection ID (CID)

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

3.6.22

connection path

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

3.6.23

connection point

buffer which is represented as a subinstance of an Assembly object

3.6.24

consume

act of receiving data from a producer

3.6.25

consumer

node or sink that is receiving data from a producer

3.6.26

consuming application

application that consumes data

3.6.27

cyclic

repetitive in a regular manner

3.6.28

device

physical hardware connected to the link

NOTE A device may contain more than one node.

3.6.29

device profile

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

3.6.30

end node

producing or consuming node

3.6.31

endpoint

one of the communicating entities involved in a connection

3.6.32

epoch

reference time defining the origin of a time scale
[IEC 61588:2009]

3.6.33

error

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

3.6.34

event

an instance of a change of conditions

3.6.35

frame

denigrated synonym for DLPDU

3.6.36

grandmaster clock

clock which serve as the primary source of time to which all others (within a collection of IEC 61588 clocks) are ultimately synchronized.

3.6.37

group

- a) <general> a general term for a collection of objects. Specific uses:
- b) <addressing> when describing an address, an address that identifies more than one entity

3.6.38

interface

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

3.6.39

invocation

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

NOTE Each invocation represents a separate thread of control that may be described by its context. Once the service completes, or use of the resource is released, the invocation ceases to exist. For service invocations, a service that has been initiated but not yet completed is referred to as an outstanding service invocation. Also for service invocations, an Invoke ID may be used to unambiguously identify the service invocation and differentiate it from other outstanding service invocations.

3.6.40

instance

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

EXAMPLE California is an instance of the object class state.

NOTE The terms object, instance, and object instance are used to refer to a specific instance.

3.6.41

instance attributes

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

3.6.42

instantiated

object that has been created in a device

3.6.43

logical device

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

3.6.44

Lpacket (or Link packet)

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

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

3.6.45

management information

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

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

3.6.46

master clock

single clock which serves as the primary source of time within each region, and which will in turn synchronize to other master clocks and ultimately to the grandmaster clock

NOTE A system of IEC 61588 clocks may be segmented into regions separated by boundary clocks.

3.6.47

member

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

3.6.48

Message Router

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

3.6.49

method

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

**3.6.50
module**

hardware or logical component of a physical device

**3.6.51
multipoint connection**

connection from one node to many

NOTE Multipoint connections allow messages from a single producer to be received by many consumer nodes.

**3.6.52
network**

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

**3.6.53
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

**3.6.54
object specific service**

service unique to the object class which defines it

**3.6.55
ordinary clock**

IEC 61588 clock with a single PTP port

**3.6.56
originator**

client responsible for establishing a connection path to the target

**3.6.57
peer**

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

**3.6.58
physical device**

an automation or other network device

**3.6.59
point-to-point connection**

connection that exists between exactly two application objects

**3.6.60
Precision Time Protocol (PTP)**

name used for the time synchronization protocol.

**3.6.61
produce**

act of sending data to be received by a consumer

**3.6.62
producer**

node that is responsible for sending data

3.6.63

property

general term for descriptive information about an object

3.6.64

PTP message

IEC 61588 time synchronization message

NOTE There are five designated messages types defined by IEC 61588:2009: Sync, Delay_Req, Follow-up, Delay_Resp, and Management.

3.6.65

PTP port

logical access point for IEC 61588 communications to the clock containing the port

3.6.66

resource

a processing or information capability of a subsystem

3.6.67

serial number

a unique 32-bit integer value assigned by each manufacturer/vendor to every device having Type 2 communication capabilities

NOTE The Vendor ID and serial number jointly form a unique identifier for each device.

3.6.68

server

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

3.6.69

service

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

3.6.70

synchronized clocks

(to a specified uncertainty) two clocks which have the same epoch and for which measurements of any time interval by both clocks differ by no more than the specified uncertainty

NOTE The timestamps generated by two synchronized clocks for the same event will differ by no more than the specified uncertainty.

3.6.71

System Time

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

NOTE In the context of CPF2, System Time is a 64-bit integer value in units of nanoseconds with a value of 0 corresponding to the date 1972-01-01.

3.6.72

target

end-node to which a connection is established

3.6.73

Unconnected Message Manager (UCMM)

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

3.6.74

unconnected service

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

3.6.75

vendor ID

identification of each product manufacturer/vendor by a unique number

NOTE Vendor IDs are assigned by ODVA, Inc.

3.7 Type 2 abbreviations and symbols

AE	Application Entity	
AL	Application layer	
APO	Application object	
AP	Application process	
APDU	Application protocol data unit	
AR	Application relationship	
AREP	Application relationship end point	
ASCII	American standard code for information interchange	
ASE	Application service element	
CID	Connection ID	
CM_API	Actual packet interval	
CM_RPI	Requested packet interval	
Cnf	Confirmation	
CR	Communication relationship	
DL-	(as a prefix) data-link-	
DLC	Data-link connection	
DLCEP	Data-link connection end point	
DLL	Data-link layer	
DLM	Data-link-management	
DLSDU	DL-service-data-unit	
FAL	Fieldbus application layer	
FIFO	First-in first-out	
ID	Identifier	
IEC	International Electrotechnical Commission	
Ind	Indication	
IP	Internet protocol	
ISO	International Organization for Standardization	
I/O	Input/output	
LME	Layer management entity	
O2T	Originator to target (connection characteristics)	
O⇒T	Originator to target (connection characteristics)	
OSI	Open systems interconnect	
PDU	Protocol data unit	
PL	Physical layer	
PTP	Precision Time Protocol	[IEC 61588:2009]
QoS	Quality of service	
REP	Route endpoint	
Req	Request	

Rsp	Response
SAP	Service access point
SDU	Service data unit
SEM	State event matrix
STD	State transition diagram, used to describe object behavior
T2O	Target to originator (connection characteristics)
T⇒O	Target to originator (connection characteristics)

3.8 Conventions

3.8.1 Overview

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

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

3.8.2 General conventions

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

3.8.3 Conventions for class definitions

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

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

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

- (2) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this standard, or by a user of this standard.
- (3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 99, 240 and 767 are reserved by this standard to identify standardized classes. CLASS IDs between 100 and 199, 768 and 1 279 are allocated for identifying user defined classes.
- (4) The "PARENT CLASS:" entry is the name of the parent class for the class being specified. All attributes defined for the parent class and inherited by it are inherited for the class being defined, and therefore do not have to be redefined in the template for this class.

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

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

3.8.4 Conventions for service definitions

3.8.4.1 General

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

3.8.4.2 Service parameters

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

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

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

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

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

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

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

3.8.4.3 Service procedures

The procedures are defined in terms of

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

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

4 Common concepts

The common concepts and templates used to describe the application layer service in this standard are detailed in IEC/TR 61158-1:2010, Clause 9.

5 Data type ASE

5.1 General

An overview of the data type ASE and the relationships between data types is provided in IEC/TR 61158-1:2010, 10.2.

5.2 Formal definition of data type objects

The template used to describe the data type class in this clause is detailed in IEC/TR 61158-1:2010, 10.2. This includes the specific ASE structure and the definition of its attributes.

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Boolean types

5.3.1.1.1 Boolean

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

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

5.3.1.1.2 BOOL

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

5.3.1.2 Bitstring types

5.3.1.2.1 BitString8

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

This type contains 1 element of type BitString.

5.3.1.2.2 SWORD

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

5.3.1.2.3 BitString16

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

5.3.1.2.4 WORD

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

5.3.1.2.5 BitString32

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

5.3.1.2.6 DWORD

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

5.3.1.2.7 BitString64

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

5.3.1.2.8 LWORD

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

5.3.1.3 Date types

5.3.1.3.1 DATE

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

This IEC 61131-3 type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of two octets. It expresses the date as a number of days, starting from 1972-01-01 (January 1st, 1972), the start of the Coordinated Universal Time (UTC) era, until 2151-06-06 (June 6th, 2151), i.e. a total range of 65 536 days.

5.3.1.3.2 TimeOfDay

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

This data type is composed of two elements of unsigned values and expresses the time of day and the date. The first element is an Unsigned32 data type and gives the time after the midnight in milliseconds. The second element is an Unsigned16 data type and gives the date counting the days from 1984-01-01 (January 1st, 1984).

5.3.1.3.3 TimeOfDay without date indication

CLASS:	Data type
ATTRIBUTES:	
1 Data type Numeric Identifier	= 52
2 Data type Name	= TimeOfDay without date indication
4 Format	= FIXED LENGTH
4.1 Octet Length	= 4

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

5.3.1.3.4 TIME_OF_DAY

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

5.3.1.4 Numeric types

5.3.1.4.1 Floating Point types

5.3.1.4.1.1 Float32

CLASS:	Data type
ATTRIBUTES:	
1 Data type Numeric Identifier =	8
2 Data type Name =	Float32
4 Format =	FIXED LENGTH
4.1 Octet Length =	4

This type has a length of four octets. The format for Float32 is that defined by IEC 60559 as single precision.

5.3.1.4.1.2 REAL

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

5.3.1.4.1.3 Float64

CLASS:	Data type
ATTRIBUTES:	
1 Data type Numeric Identifier =	15
2 Data type Name =	Float64
3 Format =	FIXED LENGTH
4.1 Octet Length =	8

This type has a length of eight octets. The format for Float64 is that defined by IEC 60559 as double precision.

5.3.1.4.1.4 LREAL

This IEC 61131-3 type is the same as Float64.

5.3.1.4.2 Integer types

5.3.1.4.2.1 Integer8

CLASS:	Data type
ATTRIBUTES:	
1 Data type Numeric Identifier =	2
2 Data type Name =	Integer8
3 Format =	FIXED LENGTH
4.1 Octet Length =	1

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

5.3.1.4.2.2 SINT

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

5.3.1.4.2.3 Integer16

CLASS:	Data type
ATTRIBUTES:	

1	Data type Numeric Identifier	=	3
2	Data type Name	=	Integer16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

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

5.3.1.4.2.4 INT

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

5.3.1.4.2.5 Integer32

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	4
2	Data type Name	=	Integer32
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

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

5.3.1.4.2.6 DINT

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

5.3.1.4.2.7 Integer64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	55
2	Data type Name	=	Integer64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

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

5.3.1.4.2.8 LINT

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

5.3.1.4.3 Unsigned types

5.3.1.4.3.1 Unsigned8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	5
2	Data type Name	=	Unsigned8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

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

5.3.1.4.3.2 USINT

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

5.3.1.4.3.3 Unsigned16

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 6
2	Data type Name	= Unsigned16
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

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

5.3.1.4.3.4 UINT

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

5.3.1.4.3.5 Unsigned32

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 7
2	Data type Name	= Unsigned32
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

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

5.3.1.4.3.6 UDINT

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

5.3.1.4.3.7 Unsigned64

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 56
2	Data type Name	= Unsigned64
3	Format	= FIXED LENGTH
4.1	Octet Length	= 8

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

5.3.1.4.3.8 ULINT

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

5.3.1.5 Time types

5.3.1.5.1 TIME

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= not used
2	Data type Name	= TIME
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This IEC 61131-3 type is a two's complement binary number with a length of four octets. The unit of time for this type is 1 ms.

5.3.1.5.2 ITIME

CLASS: Data type

ATTRIBUTES:

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

This IEC 61131-3 type extension is a two's complement binary number with a length of two octets. The unit of time for this type is 1 ms.

5.3.1.5.3 FTIME

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
2	Data type Name	=	FTIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This IEC 61131-3 type extension is a two's complement binary number with a length of four octets. The unit of time for this type is 1 μ s.

5.3.1.5.4 LTIME

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
2	Data type Name	=	LTIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This IEC 61131-3 type extension is a two's complement binary number with a length of eight octets. The unit of time for this type is 1 μ s.

5.3.2 String types

5.3.2.1 BitString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	14
2	Data type Name	=	Bitstring
3	Format	=	STRING
5.1	Octet Length	=	1 to n

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

5.3.2.2 OctetString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	10
2	Data type Name	=	OctetString
3	Format	=	STRING
4.1	Octet Length	=	1 to n

An OctetString is an ordered sequence of octets, numbered from 1 to n. For the purposes of discussion, octet 1 of the sequence is referred to as the first octet. IEC 61158-6-2:2010 defines the order of transmission.

5.3.2.3 EPATH

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|----------|
| 1 | Data type Numeric Identifier | = | not used |
| 2 | Data type Name | = | EPATH |
| 3 | Format | = | STRING |
| 4.1 | Octet Length | = | 1 to n |

An EPATH is an ordered sequence of octets, numbered from 1 to n. Its format is further specified in IEC 61158-6-2:2010, 4.1.9.

5.3.3 Structure types

5.3.3.1 DATE_AND_TIME

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-------|------------------------------|---|---------------------|
| 1 | Data type Numeric Identifier | = | not used |
| 2 | Data type Name | = | DATE_AND_TIME |
| 3 | Format | = | STRUCTURE |
| 5.1 | Number of Fields | = | 2 |
| 5.2.1 | Field Name | = | Time_Of_Day_Element |
| 5.2.2 | Field Data type | = | TIME_OF_DAY |
| 5.3.1 | Field Name | = | Date_Element |
| 5.3.2 | Field Data type | = | DATE |

This IEC 61131-3 type extension is a structure which expresses both the date (as a number of days starting from 1972-01-01 until 2151-06-06), and the time of day as a number of ms starting from midnight.

5.3.3.2 SHORT_STRING

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-------|------------------------------|---|------------------------|
| 1 | Data type Numeric Identifier | = | not used |
| 2 | Data type Name | = | SHORT_STRING |
| 3 | Format | = | STRUCTURE |
| 5.1 | Number of Fields | = | 2 |
| 5.2.1 | Field Name | = | Charcount_Element |
| 5.2.2 | Field Data type | = | USINT |
| 5.3.1 | Field Name | = | Stringcontents_Element |
| 5.3.2 | Field Data type | = | OctetString |

This IEC 61131-3 type extension is composed of two elements. Charcount_Element gives the current number of characters in the Stringcontents_Element (one octet per character). Characters shall be as specified in ISO 8859-1.

5.3.3.3 STRING

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-------|------------------------------|---|-------------------|
| 1 | Data type Numeric Identifier | = | not used |
| 2 | Data type Name | = | STRING |
| 3 | Format | = | STRUCTURE |
| 5.1 | Number of Fields | = | 2 |
| 5.2.1 | Field Name | = | Charcount_Element |

5.2.2	Field Data type	=	UINT
5.3.1	Field Name	=	Stringcontents_Element
5.3.2	Field Data type	=	OctetString

This IEC 61131-3 type is composed of two elements. Charcount_Element gives the current number of characters in the Stringcontents_Element (one USINT per character). Characters shall be as specified in ISO 8859-1.

5.3.3.4 STRING2

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= not used
2	Data type Name	= STRING2
3	Format	= STRUCTURE
5.1	Number of Fields	= 2
5.2.1	Field Name	= Charcount_Element
5.2.2	Field Data type	= UINT
5.3.1	Field Name	= String2contents_Element
5.3.2	Field Data type	= OctetString

This IEC 61131-3 data type extension is composed of two elements. Charcount_Element gives the current number of characters in the String2contents_Element (one UINT per character). Characters shall be as specified in ISO/IEC 10646.

5.3.3.5 STRINGN

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= not used
2	Data type Name	= STRINGN
3	Format	= STRUCTURE
5.1	Number of Fields	= 3
5.2.1	Field Name	= Charsize_Element
5.2.2	Field Data type	= UINT
5.3.1	Field Name	= Charcount_Element
5.3.2	Field Data type	= UINT
5.4.1	Field Name	= StringNcontents_Element
5.4.2	Field Data type	= OctetString

This IEC 61131-3 type extension is composed of three elements. Charsize_Element gives the size of a character in StringNcontents_Element (N = number of USINT). Charcount_Element gives the current number of characters in the StringNcontents_Element (N USINT per character). Characters shall be as specified in ISO/IEC 10646.

5.3.3.6 STRINGI

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= not used
2	Data type Name	= STRINGI
3	Format	= STRUCTURE
5.1	Number of Fields	= 2
5.2.1	Field Name	= Stringnum_Element
5.2.2	Field Data type	= USINT
5.3.1	Field Name	= International_String_Array
5.3.2	Field Data type	= STRINGI_ARRAY

This IEC 61131-3 type extension is a structured data type which allocates a USINT variable (Stringnum_Element) containing the number of internationalized character strings and an array of structures (International_String_Array) containing the internationalized character strings.

The international character string structure (STRINGI_ELEMENT) contains a USINT (Language1_Element) indicating the first ASCII character of the ISO 639-2/T language, a USINT (Language2_Element) indicating the second character of the ISO 639-2/T language, a USINT (Language3_Element) indicating the third character of the ISO 639-2/T language, an EPATH (Datatype_Element, limited to the values 0xD0, 0xD5, 0xD9, and 0xDA) indicating the structure of the character string, a UINT (Charset_Element) indicating the character set which the character string is based on, and an array of octet elements which is the actual international character string. The three characters for the language come from ISO 639-2/T (Alpha-3 Terminology Code), and the character set values come from IANA MIB printer codes (IETF RFC 1759). The character set values are limited to those values that are provided in Table 1.

Table 1 – Valid IANA MIB printer codes for character set selection

Character Set	Value
ISO 8859-1:1987	4
ISO 8859-2:1987	5
ISO 8859-3:1988	6
ISO 8859-4:1988	7
ISO 8859-5:1988	8
ISO 8859-6:1987	9
ISO 8859-7:1987	10
ISO 8859-8:1989	11
ISO 8859-9:1989	12
ISO/IEC 10646-UCS-2	1 000

5.3.3.7 SHORT_STRING

CLASS:

Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
2	Data type Name	=	SHORT_STRING
3	Format	=	STRUCTURE
5.1	Number of Fields	=	2
5.2.1	Field Name	=	Charcount_Element
5.2.2	Field Data type	=	USINT
5.3.1	Field Name	=	Stringcontents_Element
5.3.2	Field Data type	=	OctetString

This IEC 61131-3 type extension is composed of a single elements. Charcount_Element gives the current number of characters in the Stringcontents_Element (one octet per character). Characters shall be as specified in ISO 8859-1.

5.3.3.8 STRINGI_ELEMENT

CLASS:	Data type
ATTRIBUTES:	
1 Data type Numeric Identifier =	not used
2 Data type Name =	STRINGI_ELEMENT
3 Format =	STRUCTURE
5.1 Number of Fields =	6
5.2.1 Field Name =	Language1_Element
5.2.2 Field Data type =	USINT
5.3.1 Field Name =	Language2_Element
5.3.2 Field Data type =	USINT
5.4.1 Field Name =	Language3_Element
5.4.2 Field Data type =	USINT
5.5.1 Field Name =	Datatype_Element
5.5.2 Field Data type =	EPATH
5.6.1 Field Name =	Charset_Element
5.6.2 Field Data type =	UINT
5.7.1 Field Name =	Stringcontents_Element
5.7.2 Field Data type =	SHORT_STRING STRING STRING2 STRINGN

5.4 Data type ASE service specification

There are no operational services defined for the type object.

6 Communication model specification

6.1 Concepts

6.1.1 General

This is a serial communication system for communication between devices that wish to exchange both time critical and messaging type application information. These devices include simple I/O devices, such as sensors/actuators as well as complex control devices such as robots, programmable logic controllers, welders, process controllers.

Unlike some general purpose communication systems that rely on the destination delivery model, this network uses a variant of the publisher/subscriber push model, called the producer/consumer model. The producer/consumer model allows the exchange of time critical application information between a sending device (i.e. the producer) and many receiving devices (i.e. the consumers) without the need to send the data separately to each destination. This is accomplished by attaching a unique identifier to each piece of application information that is being produced onto the network medium. Any device that requires a specific piece of application information simply filters the data on the network medium for the appropriate identifier. Many devices can receive the same piece of application information from a single producing device.

The data-link layer provides a high degree of protocol efficiency by utilizing an implied token passing mechanism. This mechanism allows all devices equal access to the network without the network overhead associated with passing a "token" to each device granting it permission to send data. The protocol utilizes a time based scheduling mechanism which provides network devices with deterministic and predictable access to the medium while preventing network collisions. This scheduling mechanism allows time critical data, which is required on a periodic, repeatable and predictable basis, to be produced on a predefined schedule without the loss of efficiency associated with continuously requesting or "polling" for the required data. The protocol supports an additional mechanism which allows data that is not time critical in nature or which is only required on an occasional basis to utilize any available network time. This unscheduled data is transmitted after the production of the time critical data has been completed and before the beginning of the next scheduled production of time critical data.

6.1.2 General concepts

Most of the general concepts described in Clause 4 apply, with some restrictions or additions as noted:

- the application layer includes those functionalities from the intermediate layers which are necessary for proper mapping onto the data-link layer,
- Client/Server relationships can be one-to-one, but also one-to-many (see AR model in 6.3.1),
- a variant of the Publisher/Subscriber Push model, the Producer/Consumer model, is used as the base of all AR's (see AR model in 6.3.1),
- an overview of the ASE/APO types and their relationships is provided in 6.1.3,
- AR follow a model which is detailed in 6.3.1.
- specific naming and addressing is specified in 6.1.4,
- common FAL attributes and parameters listed in IEC/TR 61158-1:2010, 9.7 are not used; instead they are specified in relevant Type-specific subclauses.
- there are no Abort or Reject services, only negative result confirmations, so the error procedure described in Annex A does not apply.

6.1.3 Relationships between ASEs

Every node shall contain as a minimum instance one of the following ASE object classes in its application layer.

Object Management ASE:

- Identity (identification and general information about the device)
- Message Router (messaging connection point for communications within the device)

Connection Manager ASE:

- Connection Manager (establishment and maintenance of connections)

or Connection ASE:

- Connection (messaging connection point for communications within the device)

AR ASE:

- Unconnected Message Manager (queued messaging services on a single link, optional for CP 2/3)

Other application layer objects may be implemented in some nodes only:

Object Management ASE:

- Assembly object (binds data from multiple objects to be sent through one connection)
- Acknowledge Handler object (handles acknowledge messages from the application objects)
- TimeSync object (interface to the IEC 61588:2009 precision clock synchronization protocol)
- Parameter object (public interface to device configuration data)
- additional application-specific objects constructed according to the general Type 2 formal model

AR ASE:

- Transports (connected messaging and data services).

Yet more objects belong to system management, primarily concerned with the data-link layer:

- ControlNet object (station management interfaces to lower layers, required in all devices using Type 2 data-link layer),
- Keeper object (holds and distributes attributes of all devices on the link, one required per link, with optional backup(s) – used in conjunction with Type 2 data-link layer),
- Scheduling object (holds information on link schedules, one required in each connection originator – used in conjunction with Type 2 data-link layer),

- TCP/IP Interface object (provides the mechanism to configure the TCP/IP interface of a device, required in all devices with a TCP/IP interface),
- Ethernet Link object (maintains link-specific counters and status information for an Ethernet port, required in all devices with an Ethernet port),
- DeviceNet object (station management interfaces to lower layers, required in all devices using ISO 11898:1993 data-link layer),
- Connection Configuration object (interface to create, configure and control connections in a device),
- DLR object (configuration and status information interface for the DLR protocol),
- QoS object (configuration interface for QoS-related behaviors in devices).

NOTE These lower layer management objects are described in IEC 61158-4-2:2010.

The ASE and object interactions for a device using both Application and data-link Type 2 protocols are illustrated in Figure 1:

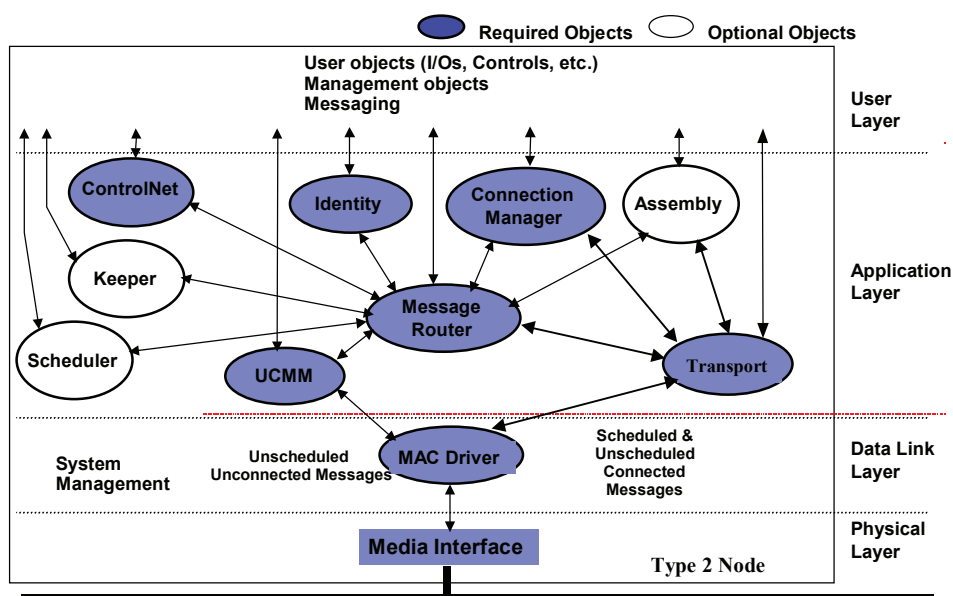


Figure 1 – Overview of ASEs and object classes

6.1.4 Naming and addressing

The information presented here provides a common basis for logically addressing separate physical components across the network. These addressing terms are used in the specifications. Figure 2 should be referenced in the following discussion.

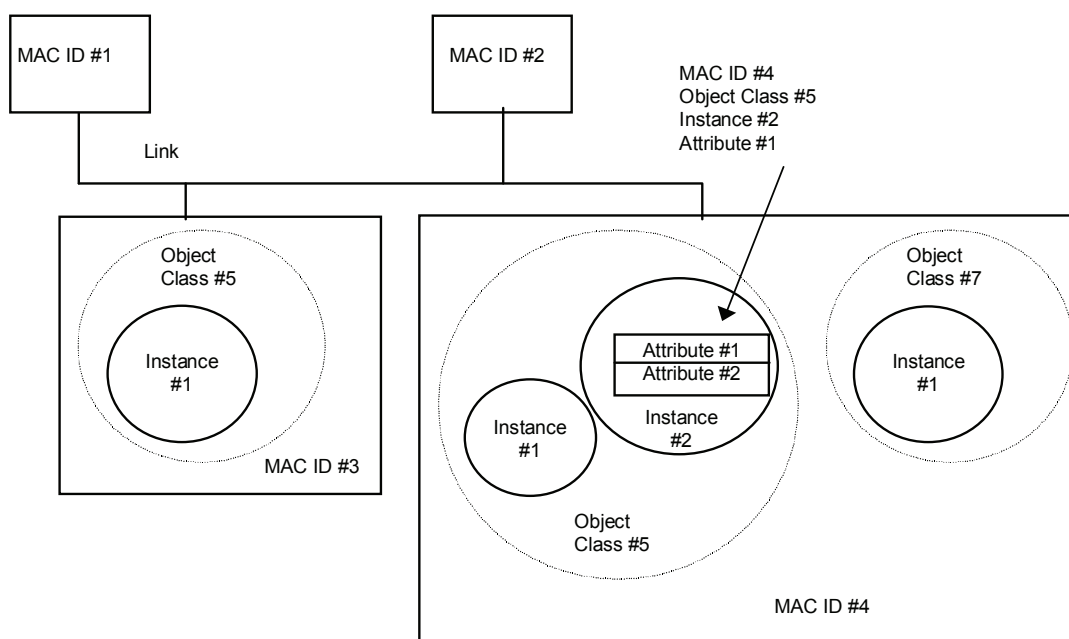


Figure 2 – Addressing format using MAC, class, instance and attribute IDs

The term “node” is used to refer to that portion of a device which contains a link interface and responds to a single MAC ID. The term “device” refers to a complete physical device connecting to the network. A device is able to contain multiple nodes.

The Class ID is a unique integer identification value assigned to each object class accessible from the network. The object class may be referenced with this Class ID. In all IEC 61158 specifications for Type 2, class code is synonymous with Class ID.

The Instance ID is an integer identification value assigned to an object instance when it is created that identifies it among all Instances of the same Class. This integer is unique within the <Node:Class> in which it resides.

The Attribute ID is an integer identification value that is unique among all other attributes of that object.

The address of attribute number 1 of instance 2 in class 5 in the node with a MAC ID of 4 is MAC ID#4: Object Class #5: Instance #2: Attribute #1 as shown in Figure 2. This nomenclature is referred to as Class/Instance/Attribute addressing.

Information on how to address an object through multiple links or using a name is provided in IEC 61158-6-2:2010 (Path definition).

6.1.5 Data types

6.1.5.1 Data type general specification

The specification of a data type is comprised of the range of values that variables of the type may take on and the operations performed on these variables.

Data is made up of elementary (primitive) data types. These elementary data types are used to construct derived (constructed) data types.

NOTE 1 Elementary and derived data type specifications correspond to the notation of IEC 61131-3. In addition, since function blocks as defined in IEC 61131-3 have both an associated data structure and a set of defined operations, these elements are also specified below as additional data types.

The derived data types are the following:

directly derived;
enumerated;
subrange;
structured;
array.

NOTE 2 These derived data types are defined in IEC 61131-3:2003, 1.3 and 2.3.3. The means of specifying these data types and their default initial values is defined in IEC 61131-3:2003, 2.3.3.1 and 2.3.3.2. The usage of variables of these data types is defined in IEC 61131-3:2003, 2.3.3.3.

6.1.5.2 Supported elementary data types

The following basic data types defined in Clause 5 are supported:

BOOL
SINT
INT
DINT
LINT
USINT
UINT
UDINT
ULINT
REAL
LREAL
ITIME
TIME
FTIME
LTIME
DATE
TIME_OF_DAY or TOD
DATE_AND_TIME or DT
STRING
STRING2
STRINGN
STRINGI
SHORT_STRING
SWORD
WORD
DWORD
LWORD
EPATH

6.1.5.3 Compliance with IEC 61131-3

6.1.5.3.1 Compliance statement

NOTE 1 This subclause provides information with respect to only the data types and associated operations defined in this part of IEC 61158.

NOTE 2 IEC 61131-3:2003, 1.5.1, defines the requirements which are met by programmable controller systems claiming compliance to IEC 61131-3. This provides the data type-related information to be included in the documentation of functional units which support the data types defined in this part of IEC 61158. Subsets or extensions of this documentation are provided as appropriate to the specific compliant functional unit.

An implementation shall comply with the requirements of IEC 61131-3 for language features as specified in Table 2 through Table 4.

Table 2 – Common elements

IEC 61131-3:2003 Table/feature	Feature description
10/1	BOOL data type
10/2	SINT data type
10/3	INT data type
10/4	DINT data type
10/5	LINT data type
10/6	USINT data type
10/7	UINT data type
10/8	UDINT data type
10/9	ULINT data type
10/10	REAL data type
10/11	LREAL data type
10/12	TIME data type
10/13	DATE data type
10/14	TIME_OF_DAY or TOD data type
10/15	DATE_AND_TIME or DT data type
10/16	STRING data type
10/17	SWORD data type
10/18	WORD data type
10/19	DWORD data type
10/20	LWORD data type
12/1	Directly derived data types
12/2	Enumerated data types
12/3	Subrange data types
12/4	Array data types
12/5	Structured data types
13	Standard default initial values
Subclause 2.5.1.3	User-declared functions (no table entry)
22/1	Type conversions (see Table 4)
22/2	TRUNC function
22/3	BCD_TO_** functions
22/4	*_TO_BCD functions
23/1-11	Standard functions of one numeric variable: ABS, SQRT, LN, LOG, EXP, SIN, COS, TAN, ASIN, ACOS, ATAN
24/12n-18n	Standard named arithmetic functions: ADD, MUL, SUB, DIV, MOD, EXPT, MOVE
24/12s-15s, 17s, 18s	Standard symbolic arithmetic functions: +, *, -, /, **, :=
25/1-4	Standard bit string functions: SHL, SHR, ROR, ROL
26/5s-8s	Standard named bitwise Boolean functions: AND, OR, XOR, NOT
26/5s-7s	Standard symbolic bitwise Boolean functions: &, >=1, =2k+1
27/1-4	Standard selection functions: SEL, MAX, MIN, LIMIT, MUX
28/5n-10n	Standard named comparison functions: GT, GE, EQ, LE, LT, NE
28/5s-10s	Standard symbolic comparison functions: >, >=, =, <=, <, <>
29/1-9	Standard character string functions: LEN, LEFT, RIGHT, MID, CONCAT, INSERT, DELETE, REPLACE, FIND
30/1-14	Standard functions of time data types: (see NOTE) ADD, SUB, MUL, DIV, CONCAT, DATE_AND_TIME_TO_TIME_OF_DAY, TIME_OF_DAY_TO_DATE_AND_TIME
31/1-4	Standard functions of enumerated data types: SEL, MUX, EQ, NE
32	Standard access mechanisms to function block I/O parameters
33/8a, 8b, 9a, 9b	User-defined function blocks per IEC 61131-3:2003, 2.5.2.2, with graphical or textual rising or falling edge input options
34/1-3	Standard bistable function blocks: SR, RS, SEMA
35/1, 2	Standard edge detection function blocks: R_EDGE, F_EDGE
36/1-3	Standard counter function blocks: CTU, CTD, CTUD
37/1, 2a, 3a, 4	Standard timer function blocks: TP, TON, TOF, RTC
55/1-17	Standard operators: (), function evaluation, **, -, NOT, *, /, MOD, +, -, <, >, <=, >=, =, <>, &, AND, XOR, OR
NOTE IEC 61131-3:2003, Table 30, limits the data types to which these operations apply.	

Table 3 – ST language elements

IEC 61131-3:2003 Table/feature	Feature description
55/1-17	Standard operators: (), function evaluation, **, -, NOT, *, /, MOD, +, -, <, >, <=, >=, =, <>, &, AND, XOR, OR
56/2	Function block invocation and output usage

Table 4 – Type conversion operations

Operation	Result	Error conditions
ANY_BIT_TO_ANY_BIT	See note 4	None
ANY_BIT_TO_ANY_INT	OUT _{min} + Sbk _{2k} (note 5)	Result > OUT _{max}
ANY_BIT_TO_BOOL	FALSE if IN = 0 TRUE otherwise	None
ANY_BIT_TO_STRING	See note 6	None
ANY_DATE_TO_STRING	See note 7	None
ANY_INT_TO_BOOL	FALSE if IN = 0 TRUE otherwise	None
ANY_NUM_TO_ANY_INT	IN (note 8)	(IN > OUT _{max}) or (IN < OUT _{min})
ANY_NUM_TO_ANY_REAL	IN	See note 9
ANY_NUM_TO_DATE	See note 10	
ANY_NUM_TO_STRING	See note 11	
ANY_NUM_TO_TIME	See note 12	
ANY_NUM_TO_TOD	See note 13	
ANY_REAL_TO_BOOL	FALSE if IN = 0,0 TRUE otherwise	None
BOOL_TO_ANY_BIT BOOL_TO_ANY_INT	0 if IN = FALSE 1 if IN = TRUE	None
BOOL_TO_ANY_REAL	0,0 if IN = FALSE 1,0 if IN = TRUE	None
BOOL_TO_STRING	'FALSE' if IN = FALSE 'TRUE' if IN = TRUE	None
DATE_TO_ANY_NUM	See note 14	Result > OUT _{max}
STRING_TO_ANY	Converted data	See note 15
TIME_TO_ANY_NUM	See note 16	Result > OUT _{max}
TOD_TO_ANY_NUM	See note 17	Result > OUT _{max}

NOTE 1 Use of the generic data types ANY_NUM, ANY_REAL, ANY_INT, and ANY_BIT defined in IEC 61131-3:2003, 2.3.2, is intended to imply a family of conversions. For instance, the conversion BOOL_TO_ANY_REAL is intended to imply BOOL_TO_REAL and BOOL_TO_LREAL.

NOTE 2 IN refers to the value of the input variable of the type conversion function.

NOTE 3 OUT_{min} and OUT_{max} refer to the minimum and maximum values of the output data type of the conversion function, as defined in Clause 5.

NOTE 4 In conversions of bit string types, if the number of bits in the input variable IN is less than the number of the bits in the output variable OUT, the bits of the input is copied to the corresponding least significant bits of the result and the remainder of the result is zero-filled. If the number of bits of IN is greater than the number of bits of OUT, the least significant bits of IN is copied to the corresponding bits of the result. For instance:

SWORD_TO_WORD(16#FF) = 16#00FF
and
WORD_TO_SWORD (16#0FF0) = 16#F0

NOTE 5 Bit numbering in this expression is as specified in IEC 61158-6-2:2010.

NOTE 6 The result of conversion of a bit string variable to type STRING consists of a string containing the base 16 literal representation of the variable value, as defined in IEC 61131-3:2003, 2.2.1, in characters taken from the ISO/IEC 646 character set.

NOTE 7 The result of conversion of a date and/or time of day variable to type STRING consists of a string containing the literal representation of the variable value, as defined in IEC 61131-3:2003, 2.2.1, in characters taken from the ISO/IEC 646 character set.

NOTE 8 Conversion of REAL and LREAL to integer types is accomplished by rounding as specified in IEC 60559.

NOTE 9 Rounding errors may occur if the number of significant bits in the input variable is larger than the number of significant bits in the output floating-point representation. Also, range errors of the type noted for ANY_NUM_TO_ANY_INT may occur in LREAL_TO_REAL.

NOTE 10 Conversion of a variable of a numeric type to DATE has the same result as conversion of the variable to UINT, with the result being interpreted as the number of days since 1972-01-01.

NOTE 11 Conversion of a variable of a numeric type to type STRING consists of a string containing the literal representation of the variable value, as defined in IEC 61131-3:2003, 2.2.1, in characters taken from the ISO/IEC 646 character set.

NOTE 12 Conversion of a variable of a numeric type to TIME has the same result as conversion of the variable to DINT, with the result being interpreted as a duration in milliseconds.

NOTE 13 Conversion of a variable of a numeric type to TOD has the same result as conversion of the variable to UDINT, with the result being interpreted as a time since midnight in milliseconds.

NOTE 14 Conversion of a variable of type DATE to a numerical type is the same as the conversion of a variable of type UINT to the corresponding numerical type, with the result being the numerical equivalent of the days since 1972-01-01.

NOTE 15 It is an error if the STRING data to be converted is not in the format for external representation of the output data type as specified in IEC 61131-3:2003, 2.2, or if the result of the conversion is outside the range {OUTmin..OUTmax}.

NOTE 16 Conversion of a variable of type TIME to a numerical type is the same as the conversion of a variable of type DINT to the corresponding numerical type, with the result being the numerical equivalent of the corresponding time interval expressed in milliseconds.

NOTE 17 Conversion of a variable of type TOD (TIME_OF_DAY) to a numerical type is the same as the conversion of a variable of type UDINT to the corresponding numerical type, with the result being the numerical equivalent of the time since midnight expressed in milliseconds.

6.1.5.3.2 Implementation dependant parameters

The values of implementation dependent parameters from IEC 61131-3:2003, Table D.1, are as shown in Table 5.

NOTE Values of other implementation dependent parameters are defined in other standards or in the specifications of individual functional units as appropriate.

Table 5 – Values of implementation-dependent parameters

Subclause of IEC 61131-3:2003	Parameter	Value
2.2.3.1	Range of values of duration	Same as LINT in microseconds
2.3.1	Range of values for variables of type TIME	Same as DINT in milliseconds
	Precision of representation of seconds in types TIME_OF_DAY and DATE_AND_TIME	1 ms
2.3.3.1	Maximum number of enumerated values	256
2.3.3.2	Default maximum length of STRING variables	256
	Maximum allowed length of STRING variables	65 536
2.4.1.2	Maximum number of subscripts	8
	Maximum range of subscript values	0 – 255
	Maximum number of levels of structures	8
2.5.1.5	Maximum inputs of extensible functions	8
2.5.1.5.1	Effects of type conversions on accuracy	As defined in Table 4
2.5.1.5.2	Accuracy of functions of one variable	As defined in IEC 60559
2.5.2.3.3	PVmin, PVmax of counters	0, 65 535

6.1.5.3.3 Language extensions

The extensions to IEC 61131-3 defined in this part are listed in Table 6. When these extensions are used in a particular device, the subclause references in this table shall be replaced by references to the descriptions of the corresponding extensions in the functional unit's documentation.

Table 6 – Extensions to IEC 61131-3:2003

Subclause	Description
2.1.1	ITIME data type FTIME data type LTIME data type STRING2 data type STRINGN data type SHORT_STRING data type STRINGI data type EPATH data type
2.1.4	Structured bit string types
2.2	Operations on STRING2 variables
2.2.4.1	Numbered bit string access
2.2.4.2	Structured bit string access

6.2 ASEs

6.2.1 Object management ASE

6.2.1.1 Overview

The FAL Object Management ASE is used to access all network accessible application elements or system management elements.

An access rule is associated with each attribute of an object class, which specifies how a requester can access this attribute. The definitions for access rules are as follows:

- Settable (Set) – The attribute may be accessed by one of the Set_Attribute services (if the behavior of the device does not require a Set_Attribute service, then a device implementation of that object is not required to implement the attribute as settable);
- Settable attributes are also Gettable and may be accessed by Get_Attribute services;
- Gettable (Get) – The attribute may be accessed by one of the Get_Attribute services, but shall not be modified by any of the Set_Attribute services.

6.2.1.2 FAL management model class specification

6.2.1.2.1 General formal model

6.2.1.2.1.1 Class definition

The base AP class below specifies the common attributes and services defined for all other AP classes, including those defined by the user (application specific).

(*) in front of an attribute or a service means that this attribute/service is either mandatory or optional, based on some constraints defined in the attribute/service description.

FAL ASE:	FAL Management ASE
CLASS:	Base_Object
CLASS ID:	NULL
PARENT CLASS:	TOP
ACCESS ATTRIBUTES:	
1	(m) Key Attribute: Object Instance number
2	(o) Key Attribute: Symbolic name
SYSTEM MANAGEMENT ATTRIBUTES (CLASS ATTRIBUTES):	
1	(*) Attribute: Revision
2	(o) Attribute: Max Instance
3	(o) Attribute: Number of Instances
4	(o) Attribute: Optional attribute list