

DIN EN 61158-6-14

ICS 35.100.70; 35.160

Ersatz für
DIN EN 61158-6-14:2013-03
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 6-14: Protokollspezifikation des Application Layer
(Anwendungsschicht) – Typ 14-Elemente
(IEC 61158-6-14:2014);
Englische Fassung EN 61158-6-14:2014, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 6-14: Application layer protocol specification – Type 14 elements
(IEC 61158-6-14:2014);
English version EN 61158-6-14:2014, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 6-14: Spécification du protocole de la couche application – Éléments de type 14
(CEI 61158-6-14:2014);
Version anglaise EN 61158-6-14:2014, seulement en CD-ROM

Gesamtumfang 100 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2014-09-23 angenommene Europäische Norm als DIN-Norm ist 2015-09-01.

Für DIN EN 61158-6-14:2013-03 besteht eine Übergangsfrist bis 2017-09-23.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 61158-600:2013-08.

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

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

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

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

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine 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 61158-6-14 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt, und Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Einführung

Ein Feldbus ist ein digitaler serieller Datenbus für die Kommunikation zwischen Geräten der industriellen Automatisierungstechnik wie beispielsweise Messeinrichtungen, Stellgliedern, Reglern und speicherprogrammierbaren Steuerungen. Die Normen der Reihe DIN EN 61158 legen eine Anzahl von Feldbus-Protokolltypen fest. Die vorliegende Norm legt das Protokoll des Application Layer für Typ 14 fest.

Änderungen

Gegenüber DIN EN 61158-6-14:2013-03 wurden folgende Änderungen vorgenommen:

- a) redaktionelle Überarbeitung;
- b) Änderungen der Spezifikation für CPF4;
- c) Anforderungen an alle Konformitätsklassen und -dienste 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 50170/A1: 2002-11
DIN EN 50170/A2: 2002-11
DIN EN 50170/A3 2002-11
DIN EN 50254: 1999-07
DIN EN 61158-6: 2000-08
DIN EN 61158-6-14: 2008-09, 2013-03

Nationaler Anhang NA (informativ)

1 Anwendungsbereich^{N1)}

1.1 Allgemeines

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

Diese Norm stellt für Typ 14-Feldbusse maßgebliche allgemeine Elemente für zeitkritischen und nicht zeitkritischen Datentransfer zwischen Anwendungsprogrammen in einer Automatisierungsumgebung bereit. Der Begriff „zeitkritisch“ (en: time-critical) wird verwendet, um die Präsenz eines Zeitfensters darzustellen, in dem eine oder mehrere festgelegte Aktivitäten mit einem vorbestimmten Grad an Gewissheit erledigt werden. Das Scheitern der Erledigung der festgelegten Aktivität innerhalb des Zeitfensters trägt das Risiko des Betriebsausfalls der Anwendung mit und somit das aufgeförderte Eingreifen mit dazugehörigem Risiko für Geräte, Anlage und möglicherweise Menschenleben.

Diese Norm legt Interaktionen zwischen entfernten Anwendungen fest und definiert das von außen sichtbare Verhalten, das vom Typ 14 des Feldbus-Application Layer bereitgestellt wird, in Form von:

- a) einer formellen abstrakten Syntax zur Festlegung der Application Layer-Protokoll-Dateneinheiten, die zwischen kommunizierenden Anwendungseinheiten übermittelt werden;
- b) der übertragenen Syntax zur Festlegung der Kodierungsregeln, welche auf die Application Layer-Dateneinheiten angewendet sind;
- c) dem Anwendungskontext der Zustandsmaschine zur Festlegung des sichtbaren Dienstverhalten zwischen kommunizierenden Anwendungseinheiten;
- d) der Anwendungsbeziehung der Zustandsmaschine zur Festlegung des sichtbaren Kommunikationsverhalten zwischen kommunizierenden Anwendungseinheiten.

Der Zweck dieser Norm ist es, das Protokoll zu definieren, welches bereitgestellt wird, zur

- Festlegung der Leitungsdarstellung der Dienstprimitiven, welche in IEC 61158-5-14 beschrieben sind und
- zur Festlegung des von außen sichtbaren Verhaltens verbunden mit ihrer Übertragung.

Diese Norm beschreibt das Protokoll des Typ 14 Feldbus-Application Layer in Übereinstimmung mit dem OSI-Basis-Referenzmodell (ISO/IEC 7498-1) und der OSI-Application Layer-Struktur (ISO/IEC 9545).

1.2 Spezifikationen

Das hauptsächliche Ziel dieser Norm ist das Festlegen der Syntax und des Verhaltens des Application Layer-Protokolls, welches die in IEC 61158-5-13 beschriebenen Application Layer-Dienste vermittelt.

Ein zweites Ziel ist das Bereitstellen von Migrationspfaden von bisherigen bestehenden industriellen Kommunikationsprotokollen. Es ist dieses letztere Ziel, welches zur Steigerung von unterschiedlichen Protokollen, die in IEC 61158-6 genormt werden, führt.

^{N1)} Die Nummerierung aus IEC 61158-6-14 wurde beibehalten.

1.3 Konformität

Diese Norm legt keine individuellen Implementierungen oder Produkte fest, noch beschränkt sie die Implementierungen von Application Layer-Einheiten innerhalb eines industriellen Automatisierungssystems. Konformität wird erreicht durch das Implementieren dieser Application Layer-Protokoll-Spezifikation.

3.2 Spezifische Begriffe für den Feldbus-Application Layer

3.2.1

Zugriffssteuerung

(en: access control)

Steuerung des Lesens und Schreibens eines Objekts

3.2.2

Zugriffspfad

(en: access Path)

Verknüpfung eines symbolischen Namens mit einer Variablen zum Zweck der offenen Kommunikation

3.2.3

Makrozyklus der Kommunikation

(en: communication macrocycle)

Satz von grundlegenden Zyklen, der für eine konfigurierte Kommunikationsaktivität in einem Makro-Netzwerksegment erforderlich ist

3.2.4

Kommunikationsplanung

(en: communication scheduling)

Algorithmen und Operationen für Datentransfers, die deterministisch und wiederholt auftreten

3.2.5

Konfiguration (eines Systems oder Gerätes)

(en: configuration (of a system or device))

Schritt während des Systementwurfs: Auswahl funktionaler Einheiten, Zuweisung ihrer Positionen und Festlegung ihrer Verbindungen

3.2.6

zyklisch

(en: cyclic)

sich regelmäßig wiederholend

3.2.7

FB-Zielinstanz

(en: destination FB Instance)

FB-Instanz, die die festgelegten Parameter empfängt

3.2.8

Domäne

(en: domain)

Teil des Speichers, der für die Speicherung von Code oder Daten verwendet wird

3.2.9

Domänen-Download

(en: domain download)

Operation zum Schreiben von Daten in eine Domäne

3.2.10

Domänen-Upload

(en: domain upload)

Operation zum Lesen von Daten aus einer Domäne

3.2.11

Entität

(en: entity)

bestimmte Sache, wie z. B. eine Person, ein Ort, ein Prozess, ein Objekt, ein Konzept, eine Verknüpfung oder ein Ereignis

3.2.12

Typ 14-Bridge

(en: Type 14 bridge)

DL-Relais-Einheit, die die Synchronisation zwischen Verbindungszweigen (Bussen) durchführt und selektive Store-and-Forward- sowie Routing-Funktionen zum Verbinden zweier Mikro-Netzwerksegmente ausführen kann

3.2.13

Kennung

(en: identifier)

16-Bit-Wort, das einer Systemvariablen zugeordnet ist

3.2.14

Index

(en: index)

Adresse eines Objekts innerhalb eines Anwendungsprozesses

3.2.15

Instanz

(en: instance)

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

3.2.16

Instanziierung

(en: instantiation)

Erzeugung einer Instanz eines bestimmten Typs

3.2.17

Verwaltungsinformationen

(en: management information)

im Netzwerk sichtbare Informationen zum Zweck der Verwaltung des Feldsystems

3.2.18

Verwaltungsinformationsbasis

(en: management information base)

organisierte Liste von Verwaltungsinformationen

3.2.19

Zuordnung

(en: mapping)

Satz von Werten, der eine festgelegte Verbindung zu den Größen oder Werten eines anderen Satzes aufweist

3.2.20

Member

(en: member)

Teil eines Attributs, das als Element eines Feldes strukturiert ist

3.2.21

Nachrichtenfilterung

(en: message filtering)

Entscheidung über eine Nachricht entsprechend einer bestimmten Regel

3.2.22**Mikrosegment**

(en: micro segment)

Teil eines Netzwerks, in dem eine besondere Planung implementiert ist

3.2.23**Abstand**

(en: offset)

Anzahl von Oktetten von einer besonders gekennzeichneten Position

3.2.24**Phase**

(en: phase)

abgelaufener Teil eines Zyklus, gemessen von einem beliebigen festgelegten Ausgangspunkt

3.2.25**Prozessschnittstelle**

(en: process interface)

Datenaustausch und Informationszuordnung zwischen einem physischen Prozess und einer Anwendungseinheit

3.2.26**Echtzeit**

(en: real-time)

Fähigkeit eines Systems, ein erforderliches Ergebnis in einer begrenzten Zeit bereitzustellen

3.2.27**Echtzeit-Kommunikation**

(en: real-time communication)

Transfer von Daten in Echtzeit

3.2.28**Echtzeit-Ethernet**

RTE

(en: Real-Time Ethernet; RTE)

Netzwerk, das auf ISO/IEC 8802-3 basiert und Echtzeit-Kommunikation unterstützt

Anmerkung 1 zum Begriff: Sofern die Echtzeit-Kommunikation nicht beeinträchtigt wird, können andere Kommunikationsarten unterstützt werden.

Anmerkung 2 zum Begriff: Diese Definition bezieht sich auf ISO/IEC 8802-3, ist aber nicht darauf beschränkt. Sie könnte auch auf andere IEEE-802-Spezifikationen, z. B. IEEE 802.1Q, anwendbar sein.

3.2.29**Plan**

(en: schedule)

zeitliche Einteilung einer Menge von in Beziehung stehenden Operationen

3.2.30**Makrozyklus der Planung**

(en: scheduling macrocycle)

Zeitintervall zur Implementierung eines spezifischen Plans

3.2.31**FB-Quellinstanz**

(en: source FB Instance)

FB-Instanz, die einen spezifischen Parameter sendet

3.2.32

Zeitabstand

(en: time offset)

Zeitunterschied zu einer besonders gekennzeichneten Zeit

English Version

**Industrial communication networks - Fieldbus specifications -
Part 6-14: Application layer protocol specification - Type 14
elements
(IEC 61158-6-14:2014)**

Réseaux de communication industriels - Spécifications des
bus de terrain - Partie 6-14: Spécification du protocole de la
couche application - Éléments de type 14
(CEI 61158-6-14:2014)

Industrielle Kommunikationsnetze - Feldbusse - Teil 6-14:
Protokollspezifikation des Application Layer
(Anwendungsschicht) - Typ 14-Elemente
(IEC 61158-6-14:2014)

This European Standard was approved by CENELEC on 2014-09-23. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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



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

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Foreword

The text of document 65C/764/FDIS, future edition 3 of IEC 61158-6-14, 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 61158-6-14:2014.

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) 2015-06-23
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2017-09-23

This document supersedes EN 61158-6-14:2012.

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

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 61158-6-14:2014 was approved by CENELEC as a European Standard without any modification.

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

IEC 61158-1	NOTE	Harmonized as EN 61158-1.
IEC 61784-1	NOTE	Harmonized as EN 61784-1.
IEC 61784-2	NOTE	Harmonized as EN 61784-2.

CONTENTS

INTRODUCTION.....	9
1 Scope	10
1.1 General.....	10
1.2 Specifications	10
1.3 Conformance.....	11
2 Normative references	11
3 Terms, definitions, symbols, abbreviations and conventions	12
3.1 Referenced terms and definitions	12
3.2 Fieldbus application layer specific terms and definitions	13
3.3 Abbreviations and symbols	15
3.4 Conventions	17
4 Abstract syntax.....	18
4.1 Fixed format PDU description	18
4.2 Object definitions in FAL management ASE.....	27
4.3 Definition of objects used in Type 14 application access entity	33
5 Transfer syntax	36
5.1 Encoding of basic data types	36
5.2 Encoding of Type 14 APDU header	42
5.3 Encoding of FAL management entity service parameters	42
5.4 Encoding of AAE Services	48
6 Structure of FAL protocol state machines	57
7 AP-Context state machine	58
7.1 Primitives exchanged between ALU and ALE	58
7.2 Protocol state machine descriptions	59
7.3 State transitions	59
7.4 Function descriptions.....	65
8 FAL management state machines	65
8.1 Primitives	65
8.2 Protocol state machine descriptions	66
8.3 State transitions	67
8.4 Function descriptions.....	69
9 Application access entity protocol machine	73
9.1 Primitives	73
9.2 AAE state machine	75
9.3 Event ASE protocol machine.....	77
9.4 Domain ASE protocol machine	78
9.5 Block ASE protocol machine	82
10 Application relationship state machine	85
10.1 Primitives	85
10.2 AREP state description	86
10.3 State transitions	86
10.4 Function descriptions.....	87
11 DLL mapping protocol machine	88
11.1 Concept	88

11.2 Primitives	88
11.3 State description	88
11.4 State transitions	89
11.5 Function description	89
Bibliography	90
Annex ZA (normative) Normative references to international publications with their corresponding European publications	91
Figure 1 – State transition diagram	17
Figure 2 – Exchanged primitives of protocol state machine	58
Figure 3 – ACE protocol state machine	59
Figure 4 – FME protocol state machine	67
Figure 5 – AAE state transition diagrams.....	75
Figure 6 – Event ASE state transition diagrams	78
Figure 7 – Domain ASE state transition diagram.....	79
Figure 8 – Block ASE state transition diagrams	83
Figure 9 – AREP state transition diagrams	86
Figure 10 – ESME state transition	89
Table 1 – State machine description elements.....	17
Table 2 – Definition of Type 14 MOB header object	27
Table 3 – Definition of Type 14 device descriptor object	27
Table 4 – Definition of the time synchronization object	28
Table 5 – Definition of maximum response time object	28
Table 6 – Definition of the Type 14 communication scheduling management object.....	29
Table 7 – Definition of the device application information object	29
Table 8 – Definition of FB application information header	29
Table 9 – Definition of domain application information header.....	30
Table 10 – Definition of Type 14 link object header	30
Table 11 – Definition of Type 14 FRT link object header	30
Table 12 – Definition of FB application information object	31
Table 13 – Definition of Type 14 link object	31
Table 14 – Definition of Type 14 FRT link object	32
Table 15 – Definition of domain application information object	32
Table 16 – Definition of domain object	33
Table 17 – Definition of simple variable object.....	34
Table 18 – Definition of event object	34
Table 19 – Definition of Type 14 socket mapping object	35
Table 20 – Definition of Type 14 socket timer object.....	35
Table 21 – Definition of ErrorType object	36
Table 22 – Encoding of Boolean value TRUE	36
Table 23 – Encoding of Boolean value FALSE.....	36
Table 24 – Encoding of Unsigned8 data type.....	36

Table 25 – Encoding of Unsigned16 data type	37
Table 26 – Encoding of Unsigned32 data type	37
Table 27 – Encoding of Unsigned64 data type	37
Table 28 – Encoding of Int8 data type	38
Table 29 – Encoding of Int16 data type	38
Table 30 – Encoding of Int32 data type	38
Table 31 – Encoding of Int64 data type	38
Table 32 – Encoding of Real type	39
Table 33 – Encoding of VisibleString data type	39
Table 34 – Encoding of OctetString data type	39
Table 35 – Encoding of BitString data type	40
Table 36 – Encoding of TimeOfDay data type	40
Table 37 – Encoding of BinaryDate data type	41
Table 38 – Encoding of PrecisionTimeDifference data type	42
Table 39 – Encoding of Type 14 application layer service message header	42
Table 40 – Encoding of EM_DetectingDevice request parameters	43
Table 41 – Encoding of EM_OnlineReply request parameters	43
Table 42 – Encoding of EM_GetDeviceAttribute request parameters	43
Table 43 – Encoding of EM_GetDeviceAttribute positive response parameters	44
Table 44 – Encoding of EM_GetDeviceAttribute negative response parameters	45
Table 45 – Encoding of EM_ActiveNotification request parameters	45
Table 46 – Encoding of EM_ConfiguringDevice request parameters	46
Table 47 – Encoding of EM_ConfiguringDevice positive response parameters	47
Table 48 – Encoding of EM_ConfiguringDevice negative response parameters	47
Table 49 – Encoding of EM_SetDefaultValue request parameters	47
Table 50 – Encoding of EM_SetDefaultValue positive response parameters	47
Table 51 – Encoding of clear device attribute service refuse packet	48
Table 52 – Encoding of DomainDownload request parameters	48
Table 53 – Encoding of domain download service response packet	48
Table 54 – Encoding of DomainDownload negative response parameters	48
Table 55 – Encoding of DomainUpload request parameters	49
Table 56 – Encoding of DomainUpload positive response parameters	49
Table 57 – Encoding of DomainUpload negative response parameters	49
Table 58 – Encoding of EventReport request parameters	50
Table 59 – Encoding of EventReportAcknowledge request parameters	50
Table 60 – Encoding of EventReportAcknowledge positive response parameters	50
Table 61 – Encoding of EventReportAcknowledge negative response parameters	50
Table 62 – Encoding of ReportConditionChanging request parameters	51
Table 63 – Encoding of ReportConditionChanging positive response parameters	51
Table 64 – Encoding of ReportConditionChanging negative response parameters	51
Table 65 – Encoding of Read request parameters	51
Table 66 – Encoding of Read positive response parameters	52

Table 67 – Encoding of Read negative response parameters	52
Table 68 – Encoding of Write request parameters	52
Table 69 – Encoding of Write positive response parameters	52
Table 70 – Encoding of Write negative response parameters	53
Table 71 – Encoding of VariableDistribute request parameters	53
Table 72 – Encoding of FRTRead request parameters	53
Table 73 – Encoding of FRTRead positive response parameters	53
Table 74 – Encoding of FRTRead negative response parameters	54
Table 75 – Encoding of FRTWrite request parameters	54
Table 76 – Encoding of FRTWrite positive response parameters	54
Table 77 – Encoding of FRTWrite negative response parameters	54
Table 78 – Encoding of FRTVariableDistribute request parameters	55
Table 79 – Encoding of BlockTransmissionOpen request parameters	55
Table 80 – Encoding of BlockTransmissionOpen positive response parameters	55
Table 81 – Encoding of BlockTransmissionOpen negative response parameters	55
Table 82 – Encoding of BlockTransmissionClose request parameters	56
Table 83 – Encoding of BlockTransmissionClose positive response parameters	56
Table 84 – Encoding of BlockTransmissionClose negative response parameters	56
Table 85 – Encoding of BlockTransmit request parameters	57
Table 86 – Encoding of BlockTransmissionHeartbeat request parameters	57
Table 87 – Primitives delivered by ALU to ALE	58
Table 88 – Primitives delivered by ALE to ALU	58
Table 89 – ACE state descriptions	59
Table 90 – ACE state transitions (sender)	59
Table 91 – ACE state transitions (receiver)	62
Table 92 – APSERVICEType() descriptions	65
Table 93 – Primitives delivered by application layer user to FME	65
Table 94 – Primitives delivered by FME to application layer user	65
Table 95 – Primitive parameters exchanged between FME and application layer user	66
Table 96 – Primitives delivered by FME to ESME	66
Table 97 – Primitives delivered by ESME to FME	66
Table 98 – Primitive parameters exchanged between FME and ESME	66
Table 99 – State transitions of Type 14 FME	67
Table 100 – RcvNewIpAddress() descriptions	69
Table 101 – Attribute_Set() descriptions	70
Table 102 – RestoreDefaults() descriptions	70
Table 103 – NewAddress() descriptions	70
Table 104 – Restart_Type 14RepeatTimer() descriptions	70
Table 105 – Clear_DuplicatePdTagFlag() descriptions	71
Table 106 – Type 14RepeatTimerExpire() descriptions	71
Table 107 – Send_EM_ReqRspMessage() descriptions	71
Table 108 – Send_EM_CommonErrorRsp() descriptions	71

Table 109 – SntpSyncLost() descriptions	72
Table 110 – IPAddressCollision() descriptions	72
Table 111 – RecvMsg() descriptions	72
Table 112 – QueryMatch() descriptions	72
Table 113 – MessageIDMatch() descriptions	72
Table 114 – DevId_Match() descriptions	73
Table 115 – PdTag_Match() descriptions	73
Table 116 – Set_Attribute_Data() descriptions	73
Table 117 – Set_DuplicatePdTagFlag() descriptions	73
Table 118 – Primitives issued by ALU to AAE	74
Table 119 – Primitives issued by AAE to ALU	74
Table 120 – Primitives parameters exchanged between AAE and ALU	74
Table 121 – Primitives issued by AAE to ESME	74
Table 122 – Primitives issued by ESME to AAE	75
Table 123 – Primitive parameters exchanged between AAE and ESME	75
Table 124 – AAE state descriptions	75
Table 125 – AAE state transitions (sender)	76
Table 126 – AAE state transitions (receiver)	76
Table 127 – ServiceType() descriptions	77
Table 128 – State value of event management	77
Table 129 – Event ASE state transition table	78
Table 130 – Domain state value	79
Table 131 – Domain ASE state transition table	79
Table 132 – Domain_DownloadSucceed() description	82
Table 133 – Domain_WriteBuffer() description	82
Table 134 – IncrementInvokeDomainCounter() description	82
Table 135 – DecreamentInvokeDomainCounter() description	82
Table 136 – State value of Block transmission	82
Table 137 – Block ASE state transition table	83
Table 138 – BlockTransmissionOpenSucceed() descriptions	84
Table 139 – BlockTransmissionCloseSucceed() descriptions	84
Table 140 – ReceiveBlockTransmissionHeartbeat_timeout() description	84
Table 141 – Primitives issued by FME(or AAE) to AREP	85
Table 142 – Primitives issued by AREP to FME(or AAE)	85
Table 143 – Primitives parameters exchanged between AREP and FME(or AAE)	85
Table 144 – Primitives issued by AREP to ESME	85
Table 145 – Primitives issued by ESME to AREP	86
Table 146 – Primitive parameters exchanged between AREP and ESME	86
Table 147 – AREP state descriptions	86
Table 148 – AREP state transitions	86
Table 149 – AREPType() descriptions	87
Table 150 – ServiceType() descriptions	88

Table 151 – The primitives exchanged between transport layer and ESME88

Table 152 – Primitives parameters exchanged between Transport Layer and ESME..... 88

Table 153 – ESME state description.....89

Table 154 – ECFME state transitions89

Table 155 – ServiceType()description89

INTRODUCTION

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

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

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-14: Application layer protocol specification – Type 14 elements

1 Scope

1.1 General

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

This 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 14 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 specifies interactions between remote applications and defines the externally visible behavior provided by the Type 14 fieldbus application layer in terms of

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

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

- a) define the wire-representation of the service primitives defined in IEC 61158-5-14, and
- b) define the externally visible behavior associated with their transfer.

This standard specifies the protocol of the Type 14 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498) and the OSI application layer structure (ISO/IEC 9545).

1.2 Specifications

The principal objective of this standard is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-14.

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 protocols standardized in the IEC 61158-6 series.

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. Conformance is achieved through implementation of this application layer protocol specification.

2 Normative references

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

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

IEC 61158-3-14, *Industrial communication networks – Fieldbus specifications – Part 3-14: Data-link layer service definition – Type 14 elements*

IEC 61158-4-14, *Industrial communication networks – Fieldbus specifications – Part 4-14: Data-link layer protocol specification – Type 14 elements*

IEC 61158-5-14, *Industrial communication networks – Fieldbus specifications – Part 5-14: Application layer service definition – Type 14 elements*

IEC 61158-6 (all parts), *Industrial communication networks – Fieldbus specifications – Part 6: Application layer protocol specification*

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

ISO/IEC 2375, *Information technology – Procedure for registration of escape sequences and coded character sets*

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

ISO/IEC 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

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

ISO/IEC 8824:1990, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*¹

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

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

ISO/IEC/IEEE 60559, *Information technology – Microprocessor Systems – Floating-Point arithmetic*

IEEE 754-2008, *IEEE Standard for Floating-Point Arithmetic*

¹ Withdrawn.

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 Referenced terms and definitions

3.1.1 ISO/IEC 7498-1 terms

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

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

3.1.2 ISO/IEC 8822 terms

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

- a) abstract syntax
- b) presentation context

3.1.3 ISO/IEC 9545 terms

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

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

3.1.4 ISO/IEC 8824 terms

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

- a) object identifier
- b) type

3.1.5 Fieldbus data-link Layer terms

For the purposes of this document, the following terms as defined in IEC 61158-3-14 and IEC 61158-4-14 apply.

- a) DL-Time
- b) DL-Scheduling-policy
- c) DLCEP
- d) DLC
- e) DL-connection-oriented mode
- f) DLPDU
- g) DLSDU
- h) DLSAP
- i) link
- j) network address
- k) node address
- l) node
- m) scheduled

3.2 Fieldbus application layer specific terms and definitions

3.2.1

access control

control on the reading and writing of an object

3.2.2

access path

association of a symbolic name with a variable for the purpose of open communication

3.2.3

communication macrocycle

set of basic cycles needed for a configured communication activity in a macro network segment

3.2.4

communication scheduling

algorithms and operation for data transfers occurring in a deterministic and repeatable manner

3.2.5

configuration (of a system or device)

step in system design: selecting functional units, assigning their locations and defining their interconnections

3.2.6

cyclic

repetitive in a regular manner

3.2.7

destination FB Instance

FB instance that receives the specified parameters

3.2.8

domain

part of memory used to store code or data

3.2.9

domain download

operation to write data in a domain

3.2.10

domain upload

operation to read data from a domain

3.2.11

entity

particular thing, such as a person, place, process, object, concept, association, or event

3.2.12

Type 14 bridge

DL-relay entity which performs synchronization between links (buses) and may perform selective store-and-forward and routing functions to connect two micro network segments

3.2.13

identifier

16-bit word associated with a system variable

3.2.14

index

address of an object within an application process

3.2.15

instance

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

3.2.16

instantiation

creation of an instance of a specified type

3.2.17

management information

network-visible information for the purpose of managing the field system

3.2.18

management information base

organized list of management information

3.2.19

mapping

set of values having defined correspondence with the quantities or values of another set

3.2.20

member

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

3.2.21

message filtering

decision on a message according to a special rule

3.2.22

micro segment

part of a network, where special scheduling is implemented

3.2.23

offset

number of octets from a specially designated position

3.2.24

phase

elapsed fraction of a cycle, measured from some fixed origin

3.2.25

process interface

data exchange and information mapping between physical process and application unit

3.2.26

real-time

ability of a system to provide a required result in a bounded time

3.2.27

real-time communication

transfer of data in real-time

3.2.28

Real-Time Ethernet

RTE

ISO/IEC 8802-3-based network that includes real-time communication

Note 1 to entry: Other communication can be supported, providing the real-time communication is not compromised.

Note 2 to entry: This definition is dedicated, but not limited, to ISO/IEC 8802-3. It could be applicable to other IEEE 802 specifications, for example IEEE 802.1Q.

3.2.29

schedule

temporal arrangement of a number of related operations

3.2.30

scheduling macrocycle

time interval to implement a specific schedule

3.2.31

source FB Instance

FB instance that sends a specific parameter

3.2.32

time offset

time difference from a specially designated time

3.3 Abbreviations and symbols

AAE	Application Access Entity
AE	Application Entity
AL	Application Layer
ALE	Application Layer Entity
ALP	Application Layer Protocol
APO	Application Object
AP	Application Process
APDU	Application Protocol Data Unit

API	Application Process Identifier
AR	Application Relationship
ARP	Address Resolution Protocol
AREP	Application Relationship End Point
ASE	Application Service Element
Cnf	Confirmation
CR	Communication Relationship
CREP	Communication Relationship End Point
CSMA/CD	Carrier Sense Multiple Access Protocol with Collision Detection
DD	Device Description
DHCP	Dynamic Host Configuration Protocol
DL-	(as a prefix) data-link-
DLCEP	Data-link Connection End Point
DLL	Data-link Layer
DLE	Data-link Entity
DLM	Data-link-management
DLS	Data-link Service
DLSAP	Data-link Service Access Point
DLSDU	DL-service-data-unit
ECSME	Type 14 communication scheduling management entity
EM_	(as a prefix) Type 14 Management
ESME	Type 14 Socket Mapping Entity
FB	Function Block
FBAP	Function Block Application Process
FME	FAL Management Entity
FRT	Fast Real-time
Ind	Indication
IP	Internet Protocol
LLC	Logical Link Control
LMP	Link Management Protocol
MAC	Medium Access Control
MAU	Medium Attachment Unit
MOB	Management Object Base
PAD	Pad (bits)
PDU	Protocol Data Unit
P/S	Publisher/Subscriber
Req	Request
Rsp	Response
RTE	Real-Time Ethernet
RT-Ethernet	Real-Time Ethernet
SAP	Service Access Point
SDU	Service Data Unit
SME	System Management Entity
SNTP	Simple Network Time Protocol
TCP	Transmission Control Protocol

UDP	User Datagram Protocol
.cnf	Confirm Primitive
.ind	Indication Primitive
.req	Request Primitive
.rsp	Response Primitive

3.4 Conventions

3.4.1 General concept

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of three parts: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5-14. The protocol specification for each of the ASEs is defined in this standard.

The class definitions define the attributes of the classes supported by each ASE. The attributes are accessible from instances of the class using the Management ASE services specified in IEC 61158-5-14. The service specification defines the services that are provided by the ASE.

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

3.4.2 Conventions for state machines for Type 14

A state machine describes the state sequence of an entity and can be represented by a state transition diagram and/or a state table.

In a state transition diagram (Figure 1), the transition between two states represented by circles is illustrated by an arrow beside which the transition events or conditions are presented.

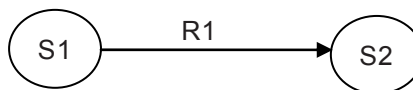


Figure 1 – State transition diagram

Table 1 – State machine description elements

#	Current state	Events or conditions that trigger this state transaction => Action	Next state
Name of this transition	The current state to which this state transition applies	Events or conditions that trigger this state transaction. => The actions that are taken when the above events or conditions are met. The actions are always indented below events or conditions	The next state after the actions in this transition is taken

The conventions used in the state transition table (Table 1) are as follows.

:= Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event.

xxx A parameter name.

Example:

Identifier := reason

means value of a 'reason' parameter is assigned to a parameter called 'Identifier.'
"xxx" Indicates fixed value.

Example:

Identifier := "abc"

means value "abc" is assigned to a parameter named 'Identifier.'

= A logical condition to indicate an item on the left is equal to an item on the right.

< A logical condition to indicate an item on the left is less than the item on the right.

> A logical condition to indicate an item on the left is greater than the item on the right.

<> A logical condition to indicate an item on the left is not equal to an item on the right.

&& Logical "AND"

|| Logical "OR"

Service.req represents a Request Primitive; Service.req{} indicates that a request primitive is sent;

Service.ind represents an Indication Primitive; Service.ind{} indicates that an Indication Primitive is received;

Service.rsp represents a Response Primitive; Service.rsp{} indicates that a Response Primitive is sent;

Service.cnf represents a Confirm Primitive; Service.cnf{} indicates that a Confirm Primitive is received.

4 Abstract syntax

4.1 Fixed format PDU description

Type 14 PDU consists of fixed-length PDU header and variable-length PDU body. The former contains service type, message type and message length, etc.

```
Type 14 PDU ::= CHOICE {
    confirmed-RequestPDU      [0]    IMPLICIT  Confirmed-RequestPDU,
    confirmed-ResponsePDU    [1]    IMPLICIT  Confirmed-ResponsePDU,
    confirmed-ErrorPDU       [2]    IMPLICIT  Confirmed-ErrorPDU,
    unconfirmed-RequestPDU   [3]    IMPLICIT  Unconfirmed-RequestPDU
}
Confirmed-RequestPDU ::= SEQUENCE {
    pduHeader          PDUHeader,
    confirmed-request   Confirmed-Request
}
Confirmed-ResponsePDU ::= SEQUENCE {
    pduHeader          PDUHeader,
    confirmed-response  Confirmed-Response
}
Confirmed-ErrorPDU ::= SEQUENCE {
    pduHeader          PDUHeader,
    confirmed-error     Confirmed-Error
}
Unconfirmed-RequestPDU ::= SEQUENCE {
    pduHeader          PDUHeader,
    unconfirmed-request Unconfirmed-Request
}
```

4.1.1 Confirmed request service

```
Confirmed-Request ::= CHOICE {
    EM_GetDeviceAttribute      [0]    IMPLICIT  EM_GetDeviceAttribute-RequestPDU,
    EM_ConfiguringDevice      [1]    IMPLICIT  EM_ConfiguringDevice-RequestPDU,
    EM_SetDefaultValue         [2]    IMPLICIT  EM_SetDefaultValue-RequestPDU,
```

DomainDownload	[3]	IMPLICIT	DomainDownload-RequestPDU,
DomainUpload	[4]	IMPLICIT	DomainUpload-RequestPDU,
AcknowledgeEventRoport	[5]	IMPLICIT	AcknowledgeEventNotifi-RequestPDU,
ReportConditionChanging	[6]	IMPLICIT	AlterEventConditionMon-RequestPDU,
Read	[7]	IMPLICIT	Read-RequestPDU,
Write	[8]	IMPLICIT	Write-RequesPDU,
FRTRead	[9]	IMPLICIT	FRTRead-RequestPDU,
FRTWrite	[10]	IMPLICIT	FRTWrite-RequesPDU
BlockTransmissionOpen	[11]	IMPLICIT	OpenBlockTransmission-RequestPDU,
BlockTransmissionClose	[12]	IMPLICIT	CloseBlockTransmission-RequestPDU,

}

4.1.2 Confirmed response service

Confirmed- Response : : = CHOICE {

EM_GetDeviceAttribute	[0]	IMPLICIT	EM_GetDeviceAttribute-ResponsePDU,
EM_ConfiguringDevice	[1]	IMPLICIT	EM_ConfiguringDevice-ResponsePDU,
EM_SetDefaultValue	[2]	IMPLICIT	EM_SetDefaultValue-ResponsePDU,
DomainDownload	[3]	IMPLICIT	DomainDownload-ResponsePDU,
DomainUpload	[4]	IMPLICIT	DomainUpload-ResponsePDU,
AcknowledgeEventRoport	[5]	IMPLICIT	AcknowledgeEventNotifi-ResponsePDU,
ReportConditionChanging	[6]	IMPLICIT	AlterEventConditionMon-ResponsePDU,
Read	[7]	IMPLICIT	Read-ResponsePDU,
Write	[8]	IMPLICIT	Write-ResponsePDU,
FRTRead	[9]	IMPLICIT	FRTRead-ResponsePDU,
FRTWrite	[10]	IMPLICIT	FRTWrite-ResponsePDU
BlockTransmissionOpen	[11]	IMPLICIT	OpenBlockTransmission-ResponsePDU,
BlockTransmissionClose	[12]	IMPLICIT	CloseBlockTransmission-ResponsePDU,

}

4.1.3 Confirmed error

Confirmed- Error : : = CHOICE {

EM_GetDeviceAttribute	[0]	IMPLICIT	Error-Type,
EM_ConfiguringDevice	[1]	IMPLICIT	Error-Type,
EM_SetDefaultValue	[2]	IMPLICIT	Error-Type,
DomainDownload	[3]	IMPLICIT	Error-Type,
DomainUpload	[4]	IMPLICIT	Error-Type,
AcknowledgeEventRoport	[5]	IMPLICIT	Error-Type,
ReportConditionChanging	[6]	IMPLICIT	Error-Type,
Read	[7]	IMPLICIT	Error-Type,
Write	[8]	IMPLICIT	Error-Type,
FRTRead	[9]	IMPLICIT	Error-Type,
FRTWrite	[10]	IMPLICIT	Error-Type
BlockTransmissionOpen	[11]	IMPLICIT	Error-Type
BlockTransmissionClose	[12]	IMPLICIT	Error-Type

}

4.1.4 Error type

ErrorType : : = SEQUENCE {

ErrorClass	[0]	IMPLICIT	Integer8,
ErrorCode	[1]	IMPLICIT	Integer8,
AdditionalCode	[2]	IMPLICIT	Integer8,
Reserved	[3]	IMPLICIT	OctetString,
AdditionalDescription	[4]	IMPLICIT	VisibleString

}

4.1.5 Error class

ErrorClass ::= CHOICE {			ErrorCode
Resource	[0]	IMPLICIT Integer8 { memory-unavailable Other	(0), (1)
}, Service	[1]	IMPLICIT Integer8 { object-state-conflict object-constraint-conflict parameter-inconsistent illegal-parameter Size Error Other	(0), (1), (2), (3), (4), (5)
}			
Access	[2]	IMPLICIT Integer8 { object-access-unsupported object-non-existent object-access-denied hardware-fault type-conflict object-attribute-inconsistent Access-to-element-unsupported Other	(0), (1), (2), (3), (4), (5), (6), (7)
}			
Timer	[3]	IMPLICIT Integer8 { Timer-Expire Timer-Error Other	(0), (1), (2)
}, Other	[4]	IMPLICIT Integer8 { Other	(0)
}			

4.1.6 Unconfirmed request

```

Unconfirmed-Request: := CHOICE {
    EM_DetectingDevice      [0]    IMPLICIT EM_DetectingDevice-RequestPDU,
    EM_OnlineReply          [1]    IMPLICIT EM_OnlineReply-RequestPDU,
    EM_ActiveNotification   [2]    IMPLICIT EM_ActiveNotification-RequestPDU,
    EventRoport             [3]    IMPLICIT EventRoport-RequestPDU,
    VariableDistribute       [4]    IMPLICIT VariableDistribute-RequestPDU,
    FRTVariableDistribute    [5]    IMPLICIT FRTVariableDistribute-RequestPDU,
    BlockTransmit           [6]    IMPLICIT BlockTransmit-RequestPDU,
    BlockTransmissionHeartbeat[7]   IMPLICIT RequestPDU-RequestPDU
}

```

4.1.7 Type 14 application layer PDU

```

ApplicationLayerPDU ::= SEQUENCE {
    PDUHeader,
    PDUBody          CHOICE {
        Confirmed-Request,
        Confirmed-Response,
        Confirmed-Error,
        Unconfirmed-Request
    }
}

```

4.1.8 APDU header format

```

PDUHeader ::= SEQUENCE {
    ServiceID      [0]    IMPLICIT Unsigned8,
    Reserved       [1]    IMPLICIT OctetString,
    Length         [2]    IMPLICIT Unsigned16,
    MessageID      [3]    IMPLICIT Unsigned16
}

```

4.1.9 FAL Management Entity services

4.1.9.1 EM_DetectingDevice service

```
EM_DetectingDevice-RequestPDU ::= SEQUENCE {
    QueryType                [0]    IMPLICIT  Unsigned8,
    Reserved                  [1]    IMPLICIT  OctetString,
    PDTag                     [2]    IMPLICIT  VisibleString,
    FBTag                     [3]    IMPLICIT  VisibleString,
    ElementID                 [4]    IMPLICIT  Unsigned16
}
```

4.1.9.2 EM_OnlineReply service

```
EM_OnlineReply -RequestPDU ::= SEQUENCE {
    QueryType                [0]    IMPLICIT  Unsigned8,
    DuplicateTagDetected     [1]    IMPLICIT  Boolean,
    Reserved                  [2]    IMPLICIT  OctetString,
    QueriedObjectIpAddress  [3]    IMPLICIT  Unsigned32,
    QueriedObjectDeviceID    [4]    IMPLICIT  VisibleString,
    QueriedObjectPDTag       [5]    IMPLICIT  VisibleString
}
```

4.1.9.3 EM_GetDeviceAttribute service

```
EM_GetDeviceAttribute-RequestPDU ::= SEQUENCE {
    DestinationIPAddress      [0]    IMPLICIT  Unsigned32,
}
EM_GetDeviceAttribute-ResponsePDU ::= CHOICE {
    EM_GetDeviceAttribute-PositiveResponsePDU,
    EM_GetDeviceAttribute-NegativeResponsePDU
}
EM_GetDeviceAttribute-PositiveResponsePDU ::= SEQUENCE {
    DeviceID                  [0]    IMPLICIT  VisibleString,
    PdTag                     [1]    IMPLICIT  VisibleString,
    Status                    [2]    IMPLICIT  Unsigned8,
    DeviceType                [3]    IMPLICIT  Unsigned8,
    Annunciation Interval    [4]    IMPLICIT  Unsigned16,
    Annunciation Version Number [5]    IMPLICIT  Unsigned16,
    DuplicateTagDetected     [6]    IMPLICIT  Boolean,
    DeviceRedundancyNumber   [7]    IMPLICIT  Unsigned8,
    LANRedundancyPort        [8]    IMPLICIT  Unsigned16,
    DeviceRedundancy State   [9]    IMPLICIT  Unsigned8,
    MaxRedundancyNumber      [10]   IMPLICIT  Unsigned8,
    ActiveIPAddress          [11]   IMPLICIT  Unsigned32
}
EM_GetDeviceAttribute-NegativeResponsePDU ::= SEQUENCE {
    DestinationIPAddress      [0]    IMPLICIT  Unsigned32,
    Error-Type                [1]    IMPLICIT  Error-Type
}
```

4.1.9.4 EM_ActiveNotification service

```
EM_ActiveNotification-RequestPDU ::= SEQUENCE {
    DeviceID                  [0]    IMPLICIT  VisibleString,
    PdTag                     [1]    IMPLICIT  VisibleString,
    Status                    [2]    IMPLICIT  Unsigned8,
    DeviceType                [3]    IMPLICIT  Unsigned8,
    AnnunciationVersionNumber [4]    IMPLICIT  Unsigned16,
    Device Redundancy Number  [5]    IMPLICIT  Unsigned8,
    DeviceRedundancyState     [6]    IMPLICIT  Unsigned8,
    LANRedundancyPort         [7]    IMPLICIT  Unsigned16,
    DuplicateTagDetected     [8]    IMPLICIT  Boolean,
    MaxRedundancyNumber      [9]    IMPLICIT  Unsigned8,
    Reserved                  [10]   IMPLICIT  OctetString,
    ActiveIPAddress          [11]   IMPLICIT  Unsigned32
}
```

4.1.9.5 EM_ConfiguringDevice service

```

EM_ConfiguringDevice-RequestPDU ::= SEQUENCE {
    DestinationIPAddress      [0]    IMPLICIT  Unsigned32,
    DeviceID                  [1]    IMPLICIT  VisibleString,
    PdTag                     [2]    IMPLICIT  VisibleString,
    AnnunciationInterval      [3]    IMPLICIT  Unsigned16,
    DuplicateTagDetected      [4]    IMPLICIT  Boolean,
    DeviceRedundancyNumber    [5]    IMPLICIT  Unsigned8,
    LANRedundancyPort         [6]    IMPLICIT  Unsigned16,
    DeviceRedundancyState     [7]    IMPLICIT  Unsigned8,
    MaxRedundancyNumber       [8]    IMPLICIT  Unsigned8,
    ActiveIPAddress           [9]    IMPLICIT  Unsigned32
}
EM_ConfiguringDevice-ResponsePDU ::= CHOICE {
    EM_ConfiguringDevice-PositiveResponsePDU,
    EM_ConfiguringDevice-NegativeResponsePDU
}
EM_ConfiguringDevice-PositiveResponsePDU ::= SEQUENCE {
    DestinationIPAddress      [0]    IMPLICIT  Unsigned32,
    MaxRedundancyNumber      [1]    IMPLICIT  Unsigned8
}
EM_ConfiguringDevice-NegativeResponsePDU ::= SEQUENCE {
    DestinationIPAddress      [0]    IMPLICIT  Unsigned32,
    ErrorType                  [1]    IMPLICIT  ErrorType
}

```

4.1.9.6 EM_SetDefaultValue service

```

EM_SetDefaultValue-RequestPDU ::= SEQUENCE {
    DestinationIPAddress      [0]    IMPLICIT  Unsigned32,
    DeviceID                  [1]    IMPLICIT  VisibleString,
    PdTag                     [2]    IMPLICIT  VisibleString
}
EM_SetDefaultValue-ResponsePDU ::= CHOICE {
    EM_SetDefaultValue-PositiveResponsePDU,
    EM_SetDefaultValue-NegativeResponsePDU
}
EM_SetDefaultValue-PositiveResponsePDU ::= SEQUENCE {
    DestinationIPAddress      [0]    IMPLICIT  Unsigned32
}
EM_SetDefaultValue-NegativeResponsePDU ::= SEQUENCE {
    DestinationIPAddress      [0]    IMPLICIT  Unsigned32,
    ErrorType                  [1]    IMPLICIT  ErrorType
}

```

4.1.10 Application Access Entity (AAE) services

4.1.10.1 DomainDownload service

```

DomainDownload-RequestPDU ::= SEQUENCE {
    SourceAppID               [0]    IMPLICIT  Unsigned16,
    DestinationAppID          [1]    IMPLICIT  Unsigned16,
    DestinationObjectID       [2]    IMPLICIT  Unsigned16,
    DataNumber                 [3]    IMPLICIT  Unsigned16,
    MoreFollows                [4]    IMPLICIT  Boolean,
    Reserved                   [5]    IMPLICIT  OctetString,
    DataLength                 [6]    IMPLICIT  Unsigned16,
    LoadData                  [7]    IMPLICIT  OctetString
}
DomainDownload-ResponsePDU ::= CHOICE {
    DomainDownload-PositiveResponsePDU,
    DomainDownload-NegativeResponsePDU
}

```