INTERNATIONAL STANDARD

ISO/IEC 8825-3

Fourth edition
2021-06

Information technology — ASN.1 encoding rules —

Part 3: Specification of Encoding Control Notation (ECN)

Technologies de l'information — Règles de codage ASN.1 —
Partie 3: Spécification de la notation de contrôle de codage (ECN)

Reference number
ISO/IEC 8825-3:2021(E)

© ISO/IEC 2021



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2021

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs)

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems,* in collaboration with ITU-T. The identical text is published as ITU-T X.692 (02/2021).

This fourth edition cancels and replaces the third edition (ISO/IEC 8825-3:2015), which has been technically revised.

A list of all parts in the ISO/IEC 8825 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO/IEC 2021 – All rights reserved

CONTENTS

Page

Introduction x
1 Scope 1
2 Normative references 1
2.1 Identical Recommendations | International Standards 1
2.2 Additional references 2
3 Definitions 2
3.1 ASN.1 definitions 2
3.2 ECN-specific definitions 2
4 Abbreviations 5
5 Definition of ECN syntax 5
6 Encoding conventions and notation 5
7 The ECN character set 5
8 ECN lexical items 6
8.1 Encoding object references 6
8.2 Encoding object set references 6
8.3 Encoding class references 6
8.4 Reserved word items 7
8.5 Reserved encoding class name items 7
8.6 Non-ECN item 7
9 ECN Concepts 7
9.1 Encoding Control Notation (ECN) specifications 8
9.2 Encoding classes 8
9.3 Encoding structures 9
9.4 Encoding objects 9
9.5 Encoding object sets 9
9.6 Defining new encoding classes 9
9.7 Defining encoding objects 11
9.8 Differential encoding-decoding 11
9.9 Encoders options in encodings 12
9.10 Properties of encoding objects 12
9.11 Parameterization 12
9.12 Governors 13
9.13 General aspects of encodings 13
9.14 Identification of information elements 13
9.15 Reference fields and determinants 14
9.16 Replacement classes and structures 14
9.17 Mapping abstract values onto fields of encoding structures 15
9.18 Transforms and transform composites 15
9.19 Contents of Encoding Definition Modules 16
9.20 Contents of the Encoding Link Module 17
9.21 Defining encodings for primitive encoding classes 17
9.22 Application of encodings 19
9.23 Combined encoding object set 19
9.24 Application point 19
9.25 Conditional encodings 20
9.26 Other conditions for applying encodings 20
9.27 Encoding control for the open type 21
9.28 Changes to ASN.1 Recommendations | International Standards 21
10 Identifying encoding classes, encoding objects, and encoding object sets 21
11 Encoding ASN.1 types 24
11.1 General 24

© ISO/IEC 2021 – All rights reserved

11.2 Built-in encoding classes used for implicitly generated encoding structures ... 24
11.3 Simplification and expansion of ASN.1 notation for encoding purposes ... 25
11.4 The implicitly generated encoding structure ... 27
12 The Encoding Link Module (ELM) ... 27
12.1 Structure of the ELM ... 28
12.2 Encoding types ... 28
13 Application of encodings ... 29
13.1 General ... 29
13.2 The combined encoding object set and its application ... 29
14 The Encoding Definition Module (EDM) ... 31
15 The renames clause ... 33
15.1 Explicitly generated and exported structures ... 33
15.2 Name changes ... 34
15.3 Specifying the region for name changes ... 35
16 Encoding class assignments ... 36
16.1 General ... 36
16.2 Encoding structure definition ... 39
16.3 Alternative encoding structure ... 41
16.4 Repetition encoding structure ... 41
16.5 Concatenation encoding structure ... 42
17 Encoding object assignments ... 42
17.1 General ... 42
17.2 Encoding with a defined syntax ... 43
17.3 Encoding with encoding object sets ... 44
17.4 Encoding using value mappings ... 44
17.5 Encoding an encoding structure ... 45
17.6 Differential encoding-decoding ... 47
17.7 Encoding options ... 48
17.8 Non-ECN definition of encoding objects ... 49
18 Encoding object set assignments ... 49
18.1 General ... 49
18.2 Built-in encoding object sets ... 50
19 Mapping values ... 51
19.1 General ... 51
19.2 Mapping by explicit values ... 52
19.3 Mapping by matching fields ... 53
19.4 Mapping by `#TRANSFORM` encoding objects ... 54
19.5 Mapping by abstract value ordering ... 54
19.6 Mapping by value distribution ... 56
19.7 Mapping integer values to bits ... 57
20 Defining encoding objects using defined syntax ... 58
21 Types used in defined syntax specification ... 59
21.1 The `Unit` type ... 59
21.2 The `EncodingSpaceSize` type ... 59
21.3 The `EncodingSpaceDetermination` type ... 60
21.4 The `UnusedBitsDetermination` type ... 60
21.5 The `OptionalityDetermination` type ... 61
21.6 The `AlternativeDetermination` type ... 62
21.7 The `RepetitionSpaceDetermination` type ... 62
21.8 The `Justification` type ... 63
21.9 The `Padding` type ... 64
21.10 The `Pattern` and `Non-Null-Pattern` types ... 64
21.11 The `RangeCondition` type ... 65

© ISO/IEC 2021 – All rights reserved

21.12 The `Comparison` type 65
21.13 The `SizeRangeCondition` type 66
21.14 The `ReversalSpecification` type 66
21.15 The `ResultSize` type 67
21.16 The `HandleValueSet` type 67
21.17 The `IntegerMapping` type 68

22 Commonly used encoding property groups 68
22.1 Replacement specification 68
22.1.1 Encoding properties, syntax and purpose 68
22.1.2 Specification restrictions 69
22.1.3 Encoder actions 70
22.1.4 Decoder actions 70
22.2 Pre-alignment and padding specification 71
22.2.1 Encoding properties, syntax and purpose 71
22.2.2 Specification constraints 71
22.2.3 Encoder actions 71
22.2.4 Decoder actions 71
22.3 Start pointer specification 72
22.3.1 Encoding properties, syntax and purpose 72
22.3.2 Specification constraints 72
22.3.3 Encoder actions 72
22.3.4 Decoder actions 72
22.4 Encoding space specification 73
22.4.1 Encoding properties, syntax and purpose 73
22.4.2 Specification restrictions 73
22.4.3 Encoder actions 74
22.4.4 Decoder actions 74
22.5 Optionality determination 75
22.5.1 Encoding properties, syntax and purpose 75
22.5.2 Specification restrictions 75
22.5.3 Encoder actions 76
22.5.4 Decoder actions 76
22.6 Alternative determination 77
22.6.1 Encoding properties, syntax and purpose 77
22.6.2 Specification restrictions 77
22.6.3 Encoder actions 77
22.6.4 Decoder actions 78
22.7 Repetition space specification 78
22.7.1 Encoding properties, syntax and purpose 78
22.7.2 Specification constraints 79
22.7.3 Encoder actions 80
22.7.4 Decoder actions 80
22.8 Value padding and justification 81
22.8.1 Encoding properties, syntax, and purpose 81
22.8.2 Specification restrictions 82
22.8.3 Encoder actions 82
22.8.4 Decoder actions 83
22.9 Identification handle specification 83
22.9.1 Encoding properties, syntax and purpose 83
22.9.2 Specification constraints 84
22.9.3 Encoders actions 84
22.9.4 Decoders actions 84
22.10 Concatenation specification 84
22.10.1 Encoding properties, syntax and purpose 84
22.10.2 Specification constraints 85
22.10.3 Encoder actions 85
22.10.4 Decoder actions 85
22.11 Contained type encoding specification 85
22.11.1 Encoding properties, syntax and purpose 85
22.11.2 Encoder actions 86

© ISO/IEC 2021 – All rights reserved

22.11.3 Decoder actions ... 86
22.12 Bit reversal specification ... 86
22.12.1 Encoding properties, syntax, and purpose ... 86
22.12.2 Specification constraints ... 86
22.12.3 Encoder actions ... 87
22.12.4 Decoder actions ... 87
23 Defined syntax specification for bit-field and constructor classes ... 87
23.1 Defining encoding objects for classes in the alternatives category ... 87
23.1.1 The defined syntax ... 87
23.1.2 Purpose and restrictions ... 88
23.1.3 Encoder actions ... 88
23.1.4 Decoder actions ... 88
23.2 Defining encoding objects for classes in the bitstring category ... 88
23.2.1 The defined syntax ... 88
23.2.2 Model for the encoding of classes in the bitstring category ... 89
23.2.3 Purpose and restrictions ... 90
23.2.4 Encoder actions ... 90
23.2.5 Decoder actions ... 91
23.3 Defining encoding objects for classes in the boolean category ... 91
23.3.1 The defined syntax ... 91
23.3.2 Purpose and restrictions ... 92
23.3.3 Encoder actions ... 93
23.3.4 Decoder actions ... 93
23.4 Defining encoding objects for classes in the characterstring category ... 93
23.4.1 The defined syntax ... 93
23.4.2 Model for the encoding of classes in the characterstring category ... 94
23.4.3 Purpose and restrictions ... 94
23.4.4 Encoder actions ... 95
23.4.5 Decoder actions ... 95
23.5 Defining encoding objects for classes in the concatenation category ... 95
23.5.1 The defined syntax ... 95
23.5.2 Purpose and restrictions ... 97
23.5.3 Encoder actions ... 97
23.5.4 Decoder actions ... 98
23.6 Defining encoding objects for classes in the integer category ... 98
23.6.1 The defined syntax ... 98
23.6.2 Purpose and restrictions ... 98
23.6.3 Encoder actions ... 98
23.6.4 Decoder actions ... 98
23.7 Defining encoding objects for the `#CONDITIONAL-INT` class ... 98
23.7.1 The defined syntax ... 98
23.7.2 Purpose and restrictions ... 100
23.7.3 Encoder actions ... 101
23.7.4 Decoder actions ... 102
23.8 Defining encoding objects for classes in the null category ... 102
23.8.1 The defined syntax ... 102
23.8.2 Purpose and restrictions ... 103
23.8.3 Encoder actions ... 103
23.8.4 Decoder actions ... 104
23.9 Defining encoding objects for classes in the octetstring category ... 104
23.9.1 The defined syntax ... 104
23.9.2 Model for the encoding of classes in the octetstring category ... 105
23.9.3 Purpose and restrictions ... 105
23.9.4 Encoder actions ... 106
23.9.5 Decoder actions ... 106
23.10 Defining encoding objects for classes in the open type category ... 106
23.10.1 The defined syntax ... 106
23.10.2 Model for the encoding of classes in the open type category ... 107
23.10.3 Purpose and restrictions ... 108
23.10.4 Encoder actions ... 108
23.10.5 Decoder actions ... 108

© ISO/IEC 2021 – All rights reserved

23.11 Defining encoding objects for classes in the optionality category 109
23.11.1 The defined syntax 109
23.11.2 Purpose and restrictions 109
23.11.3 Encoder actions 110
23.11.4 Decoder actions 110
23.12 Defining encoding objects for classes in the pad category 110
23.12.1 The defined syntax 110
23.12.2 Purpose and restrictions 111
23.12.3 Encoder actions 111
23.12.4 Decoder actions 111
23.13 Defining encoding objects for classes in the repetition category 112
23.13.1 The defined syntax 112
23.13.2 Purpose and restrictions 112
23.13.3 Encoder actions 112
23.13.4 Decoder actions 112
23.14 Defining encoding objects for the #CONDITIONAL-REPETITION class 112
23.14.1 The defined syntax 112
23.14.2 Purpose and restrictions 114
23.14.3 Encoder actions 114
23.14.4 Decoder actions 115
23.15 Defining encoding objects for classes in the tag category 115
23.15.1 The defined syntax 115
23.15.2 Purpose and restrictions 116
23.15.3 Encoder actions 117
23.15.4 Decoder actions 117
23.16 Defining encoding objects for classes in the other categories 117
24 Defined syntax specification for the #TRANSFORM encoding class 118
24.1 Summary of encoding properties and defined syntax 118
24.2 Source and target of transforms 120
24.3 The int-to-int transform 121
24.4 The bool-to-bool transform 122
24.5 The bool-to-int transform 122
24.6 The int-to-bool transform 123
24.7 The int-to-chars transform 123
24.8 The int-to-bits transform 124
24.9 The bits-to-int transform 125
24.10 The char-to-bits transform 125
24.11 The bits-to-char transform 127
24.12 The bit-to-bits transform 128
24.13 The bits-to-bits transform 128
24.14 The chars-to-composite-char transform 129
24.15 The bits-to-composite-bits transform 129
24.16 The octets-to-composite-bits transform 130
24.17 The composite-char-to-chars transform 130
24.18 The composite-bits-to-bits transform 130
24.19 The composite-bits-to-octets transform 130
25 Complete encodings and the #OUTER class 131
25.1 Encoding properties, syntax and purpose for the #OUTER class 131
25.2 Encoder actions for #OUTER 132
25.3 Decoder actions for #OUTER 132
Annex A Addendum to Rec. ITU-T X.680 | ISO/IEC 8824-1 133
A.1 Exports and imports clauses 133
A.2 Addition of REFERENCE 133
A.3 Notation for character string values 134
Annex B Addendum to Rec. ITU-T X.681 | ISO/IEC 8824-2 135
B.1 Definitions 135
B.2 Additional lexical items 135

© ISO/IEC 2021 – All rights reserved

B.2.1 Ordered value list field references ... 135
B.2.2 Ordered encoding object list field references ... 135
B.2.3 Encoding class field references ... 135
B.3 Addition of "ENCODING-CLASS" ... 135
B.4 FieldSpec additions ... 135
B.5 Fixed-type ordered value list field spec ... 136
B.6 Fixed-class encoding object field spec ... 136
B.7 Variable-class encoding object field spec ... 136
B.8 Fixed-class encoding object set field spec ... 137
B.9 Fixed-class ordered encoding object list field spec ... 137
B.10 Encoding class field spec ... 137
B.11 Ordered value list notation ... 137
B.12 Ordered encoding object list notation ... 137
B.13 Primitive field names ... 138
B.14 Additional reserved words ... 138
B.15 Definition of encoding objects ... 139
B.16 Additions to "Setting" ... 139
B.17 Encoding class field type ... 139
Annex C Addendum to Rec. ITU-T X.683 | ISO/IEC 8824-4 ... 141
C.1 Parameterized assignments ... 141
C.2 Parameterized encoding assignments ... 141
C.3 Referencing parameterized definitions ... 142
C.4 Actual parameter list ... 142
Annex D Examples ... 144
D.1 General examples ... 144
D.1.1 An encoding object for a boolean type ... 144
D.1.2 An encoding object for an integer type ... 145
D.1.3 Another encoding object for an integer type ... 145
D.1.4 An encoding object for an integer type with holes ... 145
D.1.5 A more complex encoding object for an integer type ... 145
D.1.6 Positive integers encoded in BCD ... 146
D.1.7 An encoding object of class `#BITS` ... 147
D.1.8 An encoding object for an octetstring type ... 148
D.1.9 An encoding object for a character string type ... 148
D.1.10 Mapping character values to bit values ... 148
D.1.11 An encoding object for a sequence type ... 149
D.1.12 An encoding object for a choice type ... 149
D.1.13 Encoding a bitstring containing another encoding ... 150
D.1.14 An encoding object set ... 150
D.1.15 ASN.1 definitions ... 151
D.1.16 EDM definitions ... 151
D.1.17 ELM definitions ... 151
D.2 Specialization examples ... 152
D.2.1 Encoding by distributing values to an alternative encoding structure ... 152
D.2.2 Encoding by mapping ordered abstract values to an alternative encoding structure ... 152
D.2.3 Compression of non-continuous value ranges ... 153
D.2.4 Compression of non-continuous value ranges using a transform ... 153
D.2.5 Compression of an unevenly distributed value set by mapping ordered abstract values ... 154
D.2.6 Presence of an optional component depending on the value of another component ... 154
D.2.7 The presence of an optional component depends on some external condition ... 155
D.2.8 A variable length list ... 155
D.2.9 Equal length lists ... 156
D.2.10 Uneven choice alternative probabilities ... 157
D.2.11 A version 1 message ... 158
D.2.12 The encoding object set ... 159
D.2.13 ASN.1 definitions ... 159
D.2.14 EDM definitions ... 160
D.2.15 ELM definitions ... 160
D.3 Explicitly generated structure examples ... 160

© ISO/IEC 2021 – All rights reserved

D.3.1 Sequence with optional components defined by a pointer 161
D.3.2 Addition of a boolean type as a presence determinant 161
D.3.3 Sequence with optional components identified by a unique tag and delimited by a length field 163
D.3.4 Sequence-of type with a count 164
D.3.5 Encoding object sets 164
D.3.6 ASN.1 definitions 165
D.3.7 EDM definitions 165
D.3.8 ELM definitions 165
D.4 A more-bit encoding example 166
D.4.1 Description of the problem 166
D.4.2 Use of ASN.1 to provide the more-bit determinant 166
D.4.3 Use of value mappings to provide the more-bit determinant 167
D.4.4 Use of the replacement mechanism to provide the more-bit determinant 168
D.5 Legacy protocol specified with tabular notation 168
D.5.1 Introduction 168
D.5.2 Encoding definition for the top-level message structure 170
D.5.3 Encoding definition for a message structure 170
D.5.4 Encoding for the sequence type "B" 171
D.5.5 Encoding for an octet-aligned sequence-of type with a length determinant 171
D.5.6 Encoding for an octet-aligned sequence-of type which continues to the end of the PDU 171
D.5.7 EDM definitions 171
D.5.8 ELM definitions 172
Annex E Support for Huffman encodings 173
Annex F Additional information on the Encoding Control Notation (ECN) 175
Annex G Summary of the ECN notation 176

© ISO/IEC 2021 – All rights reserved

Introduction

The Encoding Control Notation (ECN) is a notation for specifying encodings of ASN.1 types that differ from those provided by standardized encoding rules. ECN can be used to encode all types of an ASN.1 specification, but can also be used with standardized encoding rules such as BER or PER (Rec. ITU-T X.690 | ISO/IEC 8825-1 and Rec. ITU-T X.691 | ISO/IEC 8825-2) to specify only the encoding of types that have special requirements.

An ASN.1 type specifies a set of abstract values. Encoding rules specify the representation of these abstract values as a series of bits. ECN is designed to meet the following encoding needs:

a) The need to write ASN.1 types (and get the support of ASN.1 tools in implementations) for established ("legacy") protocols where the encoding is already determined and differs from all standardized encoding rules.

b) The need to produce encodings that are minor variations on standardized rules.

The linkage provided in an ECN specification to an ASN.1 specification is well-defined and machine processable, so encoders and decoders can be automatically generated from the combined specifications. This is a significant factor in reducing both the amount of work and the possibility of errors in making interoperable systems. Another significant advantage is the ability to provide automatic tool support for testing.

These advantages are available with ASN.1 alone when standardized encoding rules suffice, but the ECN work provides these advantages in circumstances where the standardized encoding rules are not sufficient.

NOTE 1 – Currently ECN support only binary-based encodings, but could be extended in the future to cover character-based encodings.

Annex A forms an integral part of this Recommendation | International Standard, and details modifications to be made to Rec. ITU-T X.680 | ISO/IEC 8824-1 to support the notation used in this Recommendation | International Standard.

Annex B forms an integral part of this Recommendation | International Standard, and details modifications to be made to Rec. ITU-T X.681 | ISO/IEC 8824-2 to support the notation used in this Recommendation | International Standard.

Annex C forms an integral part of this Recommendation | International Standard, and details modifications to be made to Rec. ITU-T X.683 | ISO/IEC 8824-4 to support the notation used in this Recommendation | International Standard.

NOTE 2 – It is not intended that Annexes A, B and C be progressed as amendments to the referenced Recommendations | International Standards. The modifications are solely for the purpose of ECN definition (see clause 5 and 9.28).

Annex D does not form an integral part of this Recommendation | International Standard, and contains examples of the use of ECN.

Annex E does not form an integral part of this Recommendation | International Standard and provides more detail on the support for Huffman encodings in ECN.

Annex F does not form an integral part of this Recommendation | International Standard, and identifies a Web site providing access to further information and links relevant to ECN.

Annex G does not form an integral part of this Recommendation | International Standard, and provides a summary of ECN using the notation of clause 5.

© ISO/IEC 2021 – All rights reserved

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN)

1 Scope

This Recommendation | International Standard defines a notation for specifying encodings of ASN.1 types or of parts of types.

It provides several mechanisms for such specification, including:

- direct specification of the encoding using standardized notation;
- specification of the encoding by reference to standardized encoding rules;
- specification of the encoding of an ASN.1 type by reference to an encoding structure;
- specification of the encoding using non-ECN notation.

It also provides the means to link the specification of encodings to the type definitions to which they are to be applied.

ECN does not currently provide any support for specifications using the OID internationalized resource identifier type or the relative OID internationalized resource identifier type (see Rec. ITU-T X.680 | ISO/IEC 8824-1), and these are not referred to further in this Standard.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and International Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

NOTE – This Recommendation | International Standard is based on ISO/IEC 10646:2003. It cannot be applied using later versions of this standard.

2.1 Identical Recommendations | International Standards

- Recommendation ITU-T X.660 (2011) | ISO/IEC 9834-1:2012, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures and top arcs of the international object identifier tree.*
- Recommendation ITU-T X.680 (2021) | ISO/IEC 8824-1:2021, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*
- Recommendation ITU-T X.681 (2021) | ISO/IEC 8824-2:2021, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- Recommendation ITU-T X.682 (2021) | ISO/IEC 8824-3:2021, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*
- Recommendation ITU-T X.683 (2021) | ISO/IEC 8824-4:2021, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*
- Recommendation ITU-T X.690 (2021) | ISO/IEC 8825-1:2021, *Information technology – ASN.1 encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER).*
- Recommendation ITU-T X.691 (2021) | ISO/IEC 8825-2:2021, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER).*

© ISO/IEC 2021 – All rights reserved

NOTE – The above references shall be interpreted as references to the identified Recommendations | International Standards together with all their published amendments and technical corrigenda.

2.2 Additional references

– ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*.

NOTE – The above reference shall be interpreted as a reference to ISO/IEC 10646 together with all its published amendments and technical corrigenda.

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 ASN.1 definitions

This Recommendation | International Standard uses the terms defined in clause 3 of Rec. ITU-T X.680 | ISO/IEC 8824-1, Rec. ITU-T X.681 | ISO/IEC 8824-2, Rec. ITU-T X.682 | ISO/IEC 8824-3, Rec. ITU-T X.683 | ISO/IEC 8824-4, Rec. ITU-T X.690 | ISO/IEC 8825-1 and Rec. ITU-T X.691 | ISO/IEC 8825-2.

3.2 ECN-specific definitions

3.2.1 alignment point: The point in an encoding (usually its start) which serves as a reference point when an encoding specification requires alignment to some boundary.

3.2.2 auxiliary field: A field of a replacement structure (that is added in the ECN specification) whose value is set directly by the encoder without the use of any abstract value provided by the application.

NOTE – An example of an auxiliary field is a length determinant for an integer encoding or for a repetition.

3.2.3 bit-field: Contiguous bits or octets in an encoding which are decoded as a whole, and which either represent an abstract value, or provide information (such as a length determinant for some other field – see 3.2.31) needed for successful decoding, or both.

NOTE – It is in legacy protocols that "or both" sometimes occurs.

3.2.4 bit-field class: An encoding class whose objects specify the encoding of abstract values (of some ASN.1 type) into bits.

NOTE – Other encoding classes are concerned with more general encoding procedures, such as those required to determine the end of repetitions of bit-field class encodings, or to determine which of a set of alternative bit-field encodings is present.

3.2.5 bounds condition: A condition on the existence of bounds of an integer field (and whether they allow negative values or not) which, if satisfied, means that specified encoding rules are to be applied.

3.2.6 choice determinant: A bit-field which determines which of several possible encodings (each representing different abstract values) is present in some other bit-field.

3.2.7 combined encoding object set: A temporary set of encoding objects produced by the combination of two sets of encoding objects for the purpose of applying encodings.

3.2.8 conditional encoding: An encoding which is to be applied only if some specified condition is satisfied.

NOTE – The condition may be a bounds condition or a size range condition, or other more complex conditions.

3.2.9 containing type: An ASN.1 type (or encoding structure field) where a contents constraint has been applied to the values of that type (or to the values associated with that encoding structure field).

NOTE – The ASN.1 types to which a contents constraint (using **`CONTAINING/ENCODED BY`**) can be applied are the bitstring and the octetstring types.

3.2.10 current application point: The point in an encoding structure at which a combined encoding object set is being applied.

3.2.11 differential encoding-decoding: The specification of rules for a decoder that require the acceptance of encodings that cannot be produced by an encoder conforming to the current specification.

NOTE – Differential encoding-decoding supports the specification of decoding by a decoder (conforming to an initial version of a standard) which is intended to enable it to successfully decode encodings produced by a later version of that standard. This is sometimes referred to as support for extensibility.

3.2.12 encoding class: The set of all possible encodings for a specific part of the procedures needed to perform the encoding or decoding of an ASN.1 type.

© ISO/IEC 2021 – All rights reserved

NOTE – Encoding classes are defined for the encoding of primitive ASN.1 types, but are also defined for the procedures associated with ASN.1 tag notation, the use of **OPTIONAL** and for encoding constructors.

3.2.13 encoding class category: Encoding classes with some common characteristics.

NOTE – Examples are the integer category, the boolean category, and the concatenation category.

3.2.14 encoding constructor: An encoding class whose encoding objects define procedures for combining, selecting, or repeating parts of an encoding. (Examples are the **#ALTERNATIVES**, **#CHOICE**, **#CONCATENATION**, **#SEQUENCE**, etc. classes.)

3.2.15 Encoding Definition Modules (EDM): Modules that define encodings for application in the Encoding Link Module.

3.2.16 Encoding Link Module (ELM): The (unique, for any given application) module that assigns encodings to ASN.1 types.

3.2.17 encoding object: The specification of some part of the procedures needed to perform the encoding or decoding of an ASN.1 type.

NOTE – Encoding objects can specify the encoding of primitive ASN.1 types, but can also specify the procedures associated with ASN.1 tag notation, the use of **OPTIONAL** and with encoding constructors.

3.2.18 encoding object set: A set of encoding objects.

NOTE – An encoding object set is normally used in the Encoding Link Module to determine the encoding of all the top-level types used in an application.

3.2.19 encoding property: A piece of information used to define an encoding using the notation specified in clauses 23, 24 and 25.

3.2.20 encoding space: The number of bits (or octets, words or other units) used to encode an abstract value into a bit-field (see 9.21.5).

3.2.21 encoding structure: The structure of an encoding, defined either from the structure of an ASN.1 type definition, or in an EDM using bit-field classes and encoding constructors.

NOTE 1 – Use of an encoding structure is only one of several mechanisms (but an important one) that the Encoding Control Notation provides for the definition of encodings for ASN.1 types.

NOTE 2 – Definition of an encoding structure is also the definition of a corresponding encoding class.

3.2.22 explicitly generated encoding structure: An encoding structure derived from an implicitly generated encoding structure by use of the renames clause in an EDM.

3.2.23 extensibility: Provisions in an early version of a standard that are designed to maximize the interworking of implementations of that early version with the expected implementations of a later version of that standard.

3.2.24 fully-qualified name: A reference to an encoding class, object, or object set that includes either the name of the EDM module in which that encoding class, object, or object set was defined, or (in the case of an implicitly generated encoding class) the name of the ASN.1 module in which it was generated. (See also 3.2.43.)

NOTE – A fully-qualified name (see production "ExternalEncodingClassReference" in 10.6) has to be used in the body of a module if the encoding class is an implicitly generated encoding structure whose name is the same as a reserved class name, or if use of the name alone would produce ambiguity due to multiple imports of classes with that name. (See A.1/13.16).

3.2.25 generated encoding structure: An implicitly or explicitly generated encoding structure whose purpose is to define the encodings of the corresponding ASN.1 type through application of encodings in the ELM.

3.2.26 governor: A part of an ECN specification which determines the syntactic form (and semantics) of some other part of the ECN specification.

NOTE – A governor is an encoding class reference, and it determines the syntax to be used for the definition of an encoding object (of that class). The concept is the same as the concept of a type reference in ASN.1 acting as the governor for ASN.1 value notation.

3.2.27 handle value set: The specified set of all possible values of the identification handle that is exhibited by an encoding object.

3.2.28 identification handle: Part of an encoding which serves to distinguish the encodings produced by one encoding object (of a given class) from those produced by other encoding objects (of other classes).

NOTE – The ASN.1 Basic Encoding Rules use tags to provide identification handles in BER encodings.

3.2.29 implicitly generated encoding structure: The encoding structure that is implicitly generated and exported whenever a type is defined in an ASN.1 module.

3.2.30 initial application point: The point in an encoding structure at which any given combined encoding object set is first applied (in the ELM and in EDMs) .

© ISO/IEC 2021 – All rights reserved

3.2.31 length determinant: A bit-field that determines the length of some other bit-field.

3.2.32 negative integer value: A value less than zero.

3.2.33 non-negative integer value: A value greater than or equal to zero.

3.2.34 non-positive integer value: A value less than or equal to zero.

3.2.35 optional bit-field: A bit-field that is sometimes included (to encode an abstract value) and is sometimes omitted.

3.2.36 positive integer value: A value greater than zero.

3.2.37 presence determinant: A bit-field that determines whether an optional bit-field is present or not.

3.2.38 primitive class: An encoding class which is not an encoding structure, and which cannot be de-referenced to some other class (see 16.1.14).

3.2.39 recursive definition (of a reference name): A reference name for which resolution of the reference name, or of the governor of the definition of the reference name, requires resolution of the original reference name.

NOTE – Recursive definition of an encoding class (including an encoding structure) or an encoding object is permitted (but see 17.1.4). Recursive definition of an encoding object set is forbidden by 18.1.3.

3.2.40 recursive instantiation (of a parameterized reference name): An instantiation of a reference name, where resolution of the actual parameters requires resolution of the original reference name.

NOTE – Recursive instantiation of an encoding class (including an encoding structure) or an encoding object is permitted (but see 17.1.4). Recursive instantiation of an encoding object set is forbidden by 18.1.3.

3.2.41 replacement structure: A parameterized structure used to replace some or all parts of a construction before encoding the construction.

3.2.42 self-delimiting encoding: An encoding for a set of abstract values such that there is no abstract value that has an encoding that is an initial sub-string of the encoding of any other abstract value in the set.

NOTE – This includes not only fixed-length encodings of a bounded integer, but also encodings generally described as "Huffman encodings" (see Annex E).

3.2.43 simple reference name: A reference to an encoding class, object, or object set that includes neither the name of the EDM module in which that encoding class, object, or object set was defined, nor (in the case of an implicitly generated encoding class) the name of the ASN.1 module in which it was generated.

NOTE – A simple reference name can only be used when the reference to the encoding class is unambiguous, otherwise a fully-qualified name (see 3.2.24) has to be used in the body of a module.

3.2.44 size range condition: A condition on the existence of effective size constraints on a string or repetition field (and whether the constraint includes zero, and/or allows multiple sizes) which, if satisfied, means that specified encoding rules are to be applied.

3.2.45 source governor (or source class): The governor that determines the notation for specifying abstract values associated with a source class when mapping them to a target class.

3.2.46 start pointer: An auxiliary field indicating the presence or absence of an optional bit-field, and in the case of presence, containing the offset from the current position to the bit-field.

3.2.47 target governor (or target class): The governor that determines the notation for specifying abstract values associated with a target class when mapping to them from a source class.

3.2.48 top-level type(s): Those ASN.1 type(s) in an application that are used by the application in ways other than to define the components of other ASN.1 types.

NOTE 1 – Top-level types may also be used (but usually are not) as components of other ASN.1 types.

NOTE 2 – Top-level types are sometimes referred to as "the application's messages", or "PDUs". Such types are normally treated specially by tools, as they form the top-level of programming language data-structures that are presented to the application.

3.2.49 transforms: Encoding objects of the class **`#TRANSFORM`** which specify that the encoding of the abstract values associated with some class (or of transform composites – see 3.2.50) is to be the encoding of different abstract values associated with the same or a different class (or of transform composites).

NOTE – Transforms can be used, for example, to specify simple arithmetic operations on integer values, or to map integer values into characterstrings or bitstrings.

3.2.50 transform composites: An ordered list of elements that can itself be the source or the result of transforms.

NOTE – All the elements of a composite are required to have the same classification (see 9.18.2).

3.2.51 value encoding: The way in which an encoding space is used to represent an abstract value (see 9.21.5).

 © ISO/IEC 2021 – All rights reserved