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

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 8825-3:2008](https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008)

<https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008>

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

**iTeh STANDARD PREVIEW
(standards.iteh.ai)**

[ISO/IEC 8825-3:2008](https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008)

<https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published by ISO in 2009

Published in Switzerland

CONTENTS

	<i>Page</i>
Foreword	ix
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.....	6
8 ECN lexical items.....	6
8.1 Encoding object references	6
8.2 Encoding object set references.....	7
8.3 Encoding class references	7
8.4 Reserved word items.....	7
8.5 Reserved encoding class name items.....	7
8.6 Non-ECN item	8
9 ECN Concepts.....	8
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.....	10
9.6 Defining new encoding classes	10
9.7 Defining encoding objects	11
9.8 Differential encoding-decoding	12
9.9 Encoders options in encodings.....	12
9.10 Properties of encoding objects	12
9.11 Parameterization.....	13
9.12 Governors.....	13
9.13 General aspects of encodings	13
9.14 Identification of information elements.....	14
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	16
9.19 Contents of Encoding Definition Modules	17
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	20
9.24 Application point.....	20
9.25 Conditional encodings.....	20
9.26 Other conditions for applying encodings	21
9.27 Encoding control for the open type.....	21
9.28 Changes to ASN.1 Recommendations International Standards	22

10	Identifying encoding classes, encoding objects, and encoding object sets	22
11	Encoding ASN.1 types	25
11.1	General	25
11.2	Built-in encoding classes used for implicitly generated encoding structures.....	25
11.3	Simplification and expansion of ASN.1 notation for encoding purposes	26
11.4	The implicitly generated encoding structure	28
12	The Encoding Link Module (ELM)	28
12.1	Structure of the ELM	28
12.2	Encoding types	29
13	Application of encodings	30
13.1	General	30
13.2	The combined encoding object set and its application.....	30
14	The Encoding Definition Module (EDM).....	32
15	The renames clause	34
15.1	Explicitly generated and exported structures	34
15.2	Name changes	35
15.3	Specifying the region for name changes	36
16	Encoding class assignments	37
16.1	General	37
16.2	Encoding structure definition	39
16.3	Alternative encoding structure	42
16.4	Repetition encoding structure	42
16.5	Concatenation encoding structure	42
17	Encoding object assignments	43
17.1	General	43
17.2	Encoding with a defined syntax	44
17.3	Encoding with encoding object sets	45
17.4	Encoding using value mappings	45
17.5	Encoding an encoding structure	46
17.6	Differential encoding-decoding	48
17.7	Encoding options	48
17.8	Non-ECN definition of encoding objects.....	49
18	Encoding object set assignments.....	50
18.1	General	50
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.....	55
19.6	Mapping by value distribution	56
19.7	Mapping integer values to bits	57
20	Defining encoding objects using defined syntax.....	59
21	Types used in defined syntax specification	59
21.1	The Unit type	59
21.2	The EncodingSpaceSize type.....	60
21.3	The EncodingSpaceDetermination type.....	60
21.4	The UnusedBitsDetermination type.....	61
21.5	The OptionalityDetermination type	61
21.6	The AlternativeDetermination type	62
21.7	The RepetitionSpaceDetermination type	63

ITH STANDARD PREVIEW
 (standards.iteh.ai)

21.8	The Justification type	64
21.9	The Padding type	64
21.10	The Pattern and Non-Null-Pattern types.....	65
21.11	The RangeCondition type.....	65
21.12	The Comparison type.....	66
21.13	The SizeRangeCondition type	66
21.14	The ReversalSpecification type	67
21.15	The ResultSize type	67
21.16	The HandleValueSet type.....	68
21.17	The IntegerMapping type	68
22	Commonly used encoding property groups.....	69
22.1	Replacement specification.....	69
22.1.1	Encoding properties, syntax and purpose.....	69
22.1.2	Specification restrictions	70
22.1.3	Encoder actions	71
22.1.4	Decoder actions.....	71
22.2	Pre-alignment and padding specification	72
22.2.1	Encoding properties, syntax and purpose.....	72
22.2.2	Specification constraints	72
22.2.3	Encoder actions	72
22.2.4	Decoder actions.....	72
22.3	Start pointer specification	73
22.3.1	Encoding properties, syntax and purpose.....	73
22.3.2	Specification constraints	73
22.3.3	Encoder actions	73
22.3.4	Decoder actions.....	73
22.4	Encoding space specification	74
22.4.1	Encoding properties, syntax and purpose.....	74
22.4.2	Specification restrictions	74
22.4.3	Encoder actions	75
22.4.4	Decoder actions.....	75
22.5	Optionality determination	76
22.5.1	Encoding properties, syntax and purpose.....	76
22.5.2	Specification restrictions	76
22.5.3	Encoder actions	77
22.5.4	Decoder actions.....	77
22.6	Alternative determination.....	78
22.6.1	Encoding properties, syntax and purpose.....	78
22.6.2	Specification restrictions	78
22.6.3	Encoder actions	79
22.6.4	Decoder actions.....	79
22.7	Repetition space specification.....	79
22.7.1	Encoding properties, syntax and purpose.....	79
22.7.2	Specification constraints	80
22.7.3	Encoder actions	81
22.7.4	Decoder actions.....	82
22.8	Value padding and justification	82
22.8.1	Encoding properties, syntax, and purpose.....	82
22.8.2	Specification restrictions	83
22.8.3	Encoder actions	83
22.8.4	Decoder actions.....	84
22.9	Identification handle specification	84
22.9.1	Encoding properties, syntax and purpose.....	84
22.9.2	Specification constraints	85
22.9.3	Encoders actions	85
22.9.4	Decoders actions	85
22.10	Concatenation specification	85
22.10.1	Encoding properties, syntax and purpose.....	85
22.10.2	Specification constraints	86

22.10.3	Encoder actions	86
22.10.4	Decoder actions	86
22.11	Contained type encoding specification	87
22.11.1	Encoding properties, syntax and purpose	87
22.11.2	Encoder actions	87
22.11.3	Decoder actions	87
22.12	Bit reversal specification	87
22.12.1	Encoding properties, syntax, and purpose	87
22.12.2	Specification constraints	88
22.12.3	Encoder actions	88
22.12.4	Decoder actions	88
23	Defined syntax specification for bit-field and constructor classes	88
23.1	Defining encoding objects for classes in the alternatives category	88
23.1.1	The defined syntax	88
23.1.2	Purpose and restrictions	89
23.1.3	Encoder actions	89
23.1.4	Decoder actions	90
23.2	Defining encoding objects for classes in the bitstring category	90
23.2.1	The defined syntax	90
23.2.2	Model for the encoding of classes in the bitstring category	90
23.2.3	Purpose and restrictions	91
23.2.4	Encoder actions	91
23.2.5	Decoder actions	92
23.3	Defining encoding objects for classes in the boolean category	92
23.3.1	The defined syntax	92
23.3.2	Purpose and restrictions	93
23.3.3	Encoder actions	94
23.3.4	Decoder actions	94
23.4	Defining encoding objects for classes in the characterstring category	95
23.4.1	The defined syntax	95
23.4.2	Model for the encoding of classes in the characterstring category	95
23.4.3	Purpose and restrictions	96
23.4.4	Encoder actions	96
23.4.5	Decoder actions	96
23.5	Defining encoding objects for classes in the concatenation category	97
23.5.1	The defined syntax	97
23.5.2	Purpose and restrictions	98
23.5.3	Encoder actions	99
23.5.4	Decoder actions	99
23.6	Defining encoding objects for classes in the integer category	99
23.6.1	The defined syntax	99
23.6.2	Purpose and restrictions	99
23.6.3	Encoder actions	100
23.6.4	Decoder actions	100
23.7	Defining encoding objects for the #CONDITIONAL-INT class	100
23.7.1	The defined syntax	100
23.7.2	Purpose and restrictions	101
23.7.3	Encoder actions	102
23.7.4	Decoder actions	103
23.8	Defining encoding objects for classes in the null category	103
23.8.1	The defined syntax	103
23.8.2	Purpose and restrictions	105
23.8.3	Encoder actions	105
23.8.4	Decoder actions	105
23.9	Defining encoding objects for classes in the octetstring category	106
23.9.1	The defined syntax	106
23.9.2	Model for the encoding of classes in the octetstring category	106
23.9.3	Purpose and restrictions	107
23.9.4	Encoder actions	107
23.9.5	Decoder actions	107

23.10	Defining encoding objects for classes in the open type category.....	108
23.10.1	The defined syntax	108
23.10.2	Model for the encoding of classes in the open type category	109
23.10.3	Purpose and restrictions	109
23.10.4	Encoder actions	110
23.10.5	Decoder actions.....	110
23.11	Defining encoding objects for classes in the optionality category	110
23.11.1	The defined syntax	110
23.11.2	Purpose and restrictions	111
23.11.3	Encoder actions	111
23.11.4	Decoder actions.....	111
23.12	Defining encoding objects for classes in the pad category	112
23.12.1	The defined syntax	112
23.12.2	Purpose and restrictions	113
23.12.3	Encoder actions	113
23.12.4	Decoder actions.....	113
23.13	Defining encoding objects for classes in the repetition category	113
23.13.1	The defined syntax	113
23.13.2	Purpose and restrictions	114
23.13.3	Encoder actions	114
23.13.4	Decoder actions.....	114
23.14	Defining encoding objects for the #CONDITIONAL-REPETITION class.....	114
23.14.1	The defined syntax	114
23.14.2	Purpose and restrictions	116
23.14.3	Encoder actions	116
23.14.4	Decoder actions.....	117
23.15	Defining encoding objects for classes in the tag category	117
23.15.1	The defined syntax	117
23.15.2	Purpose and restrictions	118
23.15.3	Encoder actions	118
23.15.4	Decoder actions.....	119
23.16	Defining encoding objects for classes in the other categories	119
24	Defined syntax specification for the #TRANSFORM encoding class	119
24.1	Summary of encoding properties and defined syntax	119
24.2	Source and target of transforms	122
24.3	The int-to-int transform.....	122
24.4	The bool-to-bool transform.....	124
24.5	The bool-to-int transform.....	124
24.6	The int-to-bool transform.....	124
24.7	The int-to-chars transform.....	125
24.8	The int-to-bits transform	126
24.9	The bits-to-int transform	127
24.10	The char-to-bits transform.....	127
24.11	The bits-to-char transform.....	129
24.12	The bit-to-bits transform	130
24.13	The bits-to-bits transform.....	130
24.14	The chars-to-composite-char transform	131
24.15	The bits-to-composite-bits transform.....	131
24.16	The octets-to-composite-bits transform	131
24.17	The composite-char-to-chars transform	132
24.18	The composite-bits-to-bits transform.....	132
24.19	The composite-bits-to-octets transform	132
25	Complete encodings and the #OUTER class.....	132
25.1	Encoding properties, syntax and purpose for the #OUTER class	132
25.2	Encoder actions for #OUTER	133
25.3	Decoder actions for #OUTER.....	134
	Annex A Addendum to ITU-T Rec. X.680 ISO/IEC 8824-1	135

A.1	Exports and imports clauses.....	135
A.2	Addition of REFERENCE	136
A.3	Notation for character string values	136
Annex B	Addendum to ITU-T Rec. X.681 ISO/IEC 8824-2	137
B.1	Definitions.....	137
B.2	Additional lexical items	137
B.2.1	Ordered value list field references	137
B.2.2	Ordered encoding object list field references.....	137
B.2.3	Encoding class field references.....	137
B.3	Addition of "ENCODING-CLASS"	137
B.4	FieldSpec additions	138
B.5	Fixed-type ordered value list field spec	138
B.6	Fixed-class encoding object field spec.....	138
B.7	Variable-class encoding object field spec	138
B.8	Fixed-class encoding object set field spec	139
B.9	Fixed-class ordered encoding object list field spec.....	139
B.10	Encoding class field spec	139
B.11	Ordered value list notation.....	140
B.12	Ordered encoding object list notation	140
B.13	Primitive field names	140
B.14	Additional reserved words	140
B.15	Definition of encoding objects.....	140
B.16	Additions to "Setting"	141
B.17	Encoding class field type	141
Annex C	Addendum to ITU-T Rec. X.683 (ISO/IEC 8824-4)	142
Annex C	Addendum to ITU-T Rec. X.683 ISO/IEC 8824-4	142
C.1	Parameterized assignments	142
C.2	Parameterized encoding assignments.....	142
C.3	Referencing parameterized definitions.....	143
C.4	Actual parameter list	143
Annex D	Examples.....	145
Annex E	Support for Huffman encodings.....	171
Annex F	Additional information on the Encoding Control Notation (ECN).....	173
Annex G	Summary of the ECN notation.....	174

Iteh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 8825-3:2008

<https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008>

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 8825-3 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 Rec. ITU-T X.692 (11/2008).

This second edition cancels and replaces the first edition (ISO/IEC 8825-3:2002), which has been technically revised. It also incorporates the Amendments ISO/IEC 8825-3:2002/Amd.1:2005 and ISO/IEC 8825-3:2002/Amd.2:2006, and the Technical Corrigendum ISO/IEC 8825-3:2002/Cor.1:2006.

ISO/IEC 8825 consists of the following parts, under the general title *Information technology — ASN.1 encoding rules*:

- *Part 1: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*
- *Part 2: Specification of Packed Encoding Rules (PER)*
- *Part 3: Specification of Encoding Control Notation (ECN)*
- *Part 4: XML Encoding Rules (XER)*
- *Part 5: Mapping W3C XML schema definitions into ASN.1*
- *Part 6: Registration and application of PER encoding instructions*

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 (ITU-T Rec. X.690 | ISO/IEC 8825-1 and ITU-T Rec. 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 ITU-T Rec. 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 ITU-T Rec. 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 ITU-T Rec. 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.

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 ITU-T Rec. X.680 | ISO/IEC 8824-1), and these are not referred to further in this Standard.

[ISO/IEC 8825-3:2008](https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008)

<https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008>

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.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.660 (1992) | ISO/IEC 9834-1:1993, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures. (plus amendments)*.
- ITU-T Recommendation X.680 (2008) | ISO/IEC 8824-1:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*
- ITU-T Recommendation X.681 (2008) | ISO/IEC 8824-2:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- ITU-T Recommendation X.682 (2008) | ISO/IEC 8824-3:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*
- ITU-T Recommendation X.683 (2008) | ISO/IEC 8824-4:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*
- ITU-T Recommendation X.690 (2008) | ISO/IEC 8825-1:2008, *Information technology – ASN.1 encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER).*

- ITU-T Recommendation X.691 (2008) | ISO/IEC 8825-2:2008, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

NOTE 1 – Notwithstanding the ISO publication date, the above specifications are normally referred to as "ASN.1:2008".

NOTE 2 – 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 ITU-T Rec. X.680 | ISO/IEC 8824-1, ITU-T Rec. X.681 | ISO/IEC 8824-2, ITU-T Rec. X.682 | ISO/IEC 8824-3, ITU-T Rec. X.683 | ISO/IEC 8824-4, ITU-T Rec. X.690 | ISO/IEC 8825-1 and ITU-T Rec. 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.

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) .

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).

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
BCD	Binary Coded Decimal
BER	Basic Encoding Rules of ASN.1
CER	Canonical Encoding Rules of ASN.1
DER	Distinguished Encoding Rules of ASN.1
ECN	Encoding Control Notation for ASN.1
EDM	Encoding Definition Module
ELM	Encoding Link Module
PDU	Protocol Data Unit
PER	Packed Encoding Rules of ASN.1

ITeH STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 8825-3:2008](https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008)

<https://standards.iteh.ai/catalog/standards/sist/9706ffa9-ed99-4e4a-8b8b-fa57ba6e63a7/iso-iec-8825-3-2008>

5 Definition of ECN syntax

5.1 This Recommendation | International Standard employs the notational convention defined in ITU-T Rec. X.680 | ISO/IEC 8824-1, clause 5.

5.2 This Recommendation | International Standard employs the notation for information object classes defined in ITU-T Rec. X.681 | ISO/IEC 8824-2 as modified by Annex B.

5.3 This Recommendation | International Standard references productions defined in ITU-T Rec. X.680 | ISO/IEC 8824-1 as modified by Annex A, ITU-T Rec. X.681 | ISO/IEC 8824-2 as modified by Annex B, and ITU-T Rec. X.683 | ISO/IEC 8824-4 as modified by Annex C.

6 Encoding conventions and notation

6.1 This Recommendation | International Standard defines the value of each octet in an encoding by use of the terms "most significant bit" and "least significant bit".

NOTE – Lower layer specifications use the same notation to define the order of bit transmission on a serial line, or the assignment of bits to parallel channels.

6.2 For the purpose of this Recommendation | International Standard, the bits of an octet are numbered from 8 to 1, where bit 8 is the "most significant bit" and bit 1 is the "least significant bit".

6.3 For the purposes of this Recommendation | International Standard, encodings are defined as a string of bits starting from a "leading bit" through to a "trailing bit". On transmission, the first eight bits of this string of bits starting with the "leading bit" shall be placed in the first transmitted octet with the leading bit as the most significant bit of that octet. The next eight bits shall be placed in the next octet, and so on. If the encoding is not a multiple of eight bits, then the remaining bits shall be transmitted as if they were bits 8 downwards of a subsequent octet.

NOTE – A complete ECN encoding is not necessarily always a multiple of eight bits, but an ECN specification can determine the addition of padding to ensure this property.