



INTERNATIONAL STANDARD ISO/IEC 8825-3:2002
TECHNICAL CORRIGENDUM 1

Published 2006-06-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

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

TECHNICAL CORRIGENDUM 1

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

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 8825-3:2002 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 Rec. X.692 (2002)/Cor.1 (05/2005).

ICS 35.100.60

Ref. No. ISO/IEC 8825-3:2002/Cor.1:2006(E)

© ISO/IEC 2006 – All rights reserved

Published in Switzerland

AVA Documents
Document Sharing Hub

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

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

Technical Corrigendum 1

NOTE – All new or changed text in this corrigendum is underlined and deleted text is struck through in clauses being replaced. When merging all such text into the base Recommendation, the underlining is to be removed and the struck through text is to be removed.

- 1) **Subclauses 21.15, 21.15.1, 21.15.2, 22.9.1.1, 22.9.1.3, 22.9.1.4, 22.9.1.7, 22.9.1.8, 22.9.1.9, 23.1.1, 23.2.1, 23.3.1, 23.4.1, 23.5.1, 23.7.1, 23.8.1, 23.9.1, 23.11.1, 23.13.1 and 23.14.1, and the Table of Contents**

In the above subclauses, and in the Table of Contents, replace all the occurrences of "HandleValue" with "HandleValueSet".

- 2) **Subclauses 22.9.1.1, 22.9.1.2, 23.1.1, 23.2.1, 23.3.1, 23.4.1, 23.5.1, 23.7.1, 23.8.1, 23.9.1, 23.11.1, 23.13.1 and 23.14.1**

In the above subclauses, replace all the occurrences of "&handle-value" with "&handle-value-set".

- 3) **New subclause 3.2.26 bis**

Insert a new subclause 3.2.26 bis as follows:

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

- 4) **Subclause 3.2.27**

Replace subclause 3.2.27 as follows:

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

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

- 5) **Subclause 3.2.38**

Replace the Note in subclause 3.2.38 as follows:

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 ~~or an encoding object~~ set is forbidden by ~~17.1.4 and~~ 18.1.3 ~~respectively~~.

- 6) **Subclause 3.2.39**

Replace the Note in subclause 3.2.39 as follows:

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

7) Subclause 9.10.3

Replace subclause 9.10.3 as follows:

9.10.3 A third key feature is that an encoding object may exhibit an **identification handle** in its encodings. This is a part (consisting of a fixed set of bit positions) of all the encodings that it produces and distinguishes ~~it~~ those encodings from the encodings ~~of~~ produced by other encoding objects (of any class) that exhibit the same identification handle. Identification handles have ~~to be a name and are~~ visible to decoders without knowledge of either the encoding class or the abstract value that was encoded (but with knowledge of the name of the identification handle that is being used). This concept models (and generalizes) the use of tags in BER encodings: the tag value in BER can be determined without knowledge of the encoding class, for all BER encodings, and serves to identify the encoding for resolution of optionality, ordering of sets, termination of repetitions, and choice alternatives.

8) Subclause 9.13.3

Replace subclause 9.13.3 as follows:

9.13.3 In addition to terminating repetitions, the identification handle technique can also be used to determine the presence of optional components or of alternatives and the ordering of sets. The mechanism is similar in all these cases. ~~Encodings for all values of any given "possible next class" encoding will have the same bit pattern (their identification) at some place in their encoding (the handle), but the identification for different "possible next class" encodings will be different for each one. Given an encoding class that is a "possible next class" and an encoding object applied to it, any encoding produced will contain, at some bit positions (the identification handle), a bit pattern that matches a bit pattern within a specified set of bit patterns (the handle value set) characterizing that class, but does not match any bit pattern characterizing any other "possible next class". All such encodings can be interpreted by a decoder as an encoding of any "possible next class", and the identification for the handle bit pattern found in the encoding will determine which "possible next class" encoding is present. The concept is similar to that of using tags for such purposes in BER. Identification handles have names that are required to be unique within an ECN specification.~~

9) Subclause 13.2.12

Replace subclause 13.2.12 as follows:

13.2.12 In the encoding process, encoding objects applied to encoding constructors (and to classes in the optionality category) may require that the encoding objects applied to ~~their~~ the components of the constructions defined by those constructors exhibit identification handles (of a given name) to resolve alternatives, or optionality, or termination of a repetition, or order in a set-like concatenation. ~~If in this case the encodings of the components do not exhibit the required identification handles, They may also require that the encoding objects applied to other encoding classes (following those constructions) exhibit the same identification handle, and that the handle value sets of all the involved encoding objects (exhibiting the same handle) be all disjoint. If these conditions are not satisfied, then the ECN specification is in error.~~

NOTE – This problem is most likely to arise if BER encoding objects are applied to encoding constructors and not to their components, as BER is heavily reliant on identification handles. PER encoding objects make no use of identification handles.

10) Subclause 17.1.4

Replace subclause 17.1.4 with the following:

17.1.4 There shall be no recursive definition (see 3.2.38) of an "encodingobjectreference", and there shall be no recursive instantiation (see 3.2.39) of an "encodingobjectreference" if these recursions lead to an infinite recursion in the definition of the encoding.

11) Subclause 17.1.6

Delete the last sentence of subclause 17.1.6. The subclause will become:

17.1.6 "DefinedEncodingObject" identifies an encoding object and is specified in 10.9.2. The "DefinedEncodingObject" shall be of the same encoding class as the governor, or of a class which can be obtained from the governor by de-referencing.

12) Subclause 17.2.6

Replace subclause 17.2.6 with the following:

17.2.6 The "DefinedSyntax" notation specifies whether the `encoding_objectreference` being defined exhibits an identification handle.

13) Subclause 17.3.6

Replace subclause 17.3.6 with the following:

17.3.6 ~~In the application of encodings specified in clause 13, there is an encoding object (A say) which produces the first bit field in the resulting encoding. The "encodingobjectreference" being defined~~ Call E is the encoding object (within the "CombinedEncodings") which is applied to the governor class. If the encoding object E exhibits an identification handle if and only if the encoding object A exhibits that identification (with a given handle value set), then the encoding object being defined (see 17.1.5) exhibits the same identification handle as E (with the same handle value set); otherwise, it does not exhibit a handle.

14) Subclause 17.4.8

Replace subclause 17.4.8 with the following:

17.4.8 ~~If the "EncodingObject" alternative of "ValueMappingEncodingObjects" is used, then the "encodingobjectreference"~~ Call E is the encoding object which is applied to the "DefinedOrBuiltinEncodingClass". If the encoding object E exhibits an identification handle (with a given handle value set), then the encoding object being defined (see 17.1.5) exhibits an identification handle if and only if the "EncodingObject" exhibits that identification handle. If the same identification handle as E (with the same handle value set); otherwise, it does not exhibit a handle.

NOTE – The encoding object E may be either the "EncodingObject" in the "ValueMappingEncodingObjects", or a member of the "DefinedOrBuiltinEncodingObjectSet" alternative of "ValueMappingEncodingObjects" is used to define the encoding of the "DefinedOrBuiltinEncodingClass", then determination of whether the "encodingobjectreference" exhibits an identification handle is in accordance with 17.3.6.

15) Subclause 17.5.4

Replace subclause 17.5.4 with the following:

17.5.4 ~~The~~ If the "ComponentEncodingList" is not empty, then the encoding object applied to the governing encoding constructor (whether from ~~STRUCTURED WITH~~ "StructureEncoding" or from "CombinedEncodings") shall not specify any replacement actions.

16) Subclause 17.5.16

Replace subclause 17.5.16 with the following:

17.5.16 Determination of whether the ~~"encodingobjectreference"~~ encoding object being defined (see 17.1.5) exhibits an identification handle ~~is in accordance with 17.3.6.~~ shall be done as follows:

- a) if the "TagEncoding" is present in "StructureEncoding", call E the encoding object which is applied to the encoding class in the tag category; or
- b) if the "TagEncoding" is not present in "StructureEncoding", call E the encoding object which is applied to the governing encoding constructor (this may be either the "EncodingObject" in the "EncodingOrUseSet" in the "StructureEncoding", or may be a member of the "CombinedEncodings").

If the encoding object E exhibits an identification handle (with a given handle value set), then the encoding object being defined exhibits the same identification handle as E (with the same handle value set); otherwise, it does not exhibit a handle.