DRAFT INTERNATIONAL STANDARD
ISO/IEC DIS 8825-1

ISO/IEC JTC **1**/SC **6** — Secretariat: **KATS**

Voting begins on:
2020-09-28

Voting terminates on:
2020-12-21

Information technology — ASN.1 encoding rules —

Part 1: **Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)**

Technologies de l'information — Règles de codage ASN.1 —

Partie 1: Spécification des règles de codage de base (BER), des règles de codage canoniques (CER) et des règles de codage distinctives (DER)

ICS: 35.100.60

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC DIS 8825-1:2020(E)

CONTENTS

Page

Introduction

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 (Abstract Syntax Notation One or ASN.1) together specify a notation for the definition of abstract syntaxes, enabling application standards to define the types of information they need to transfer. It also specifies a notation for the specification of values of a defined type.

This Recommendation | International Standard defines encoding rules that may be applied to values of types defined using the ASN.1 notation. Application of these encoding rules produces a transfer syntax for such values. It is implicit in the specification of these encoding rules that they are also to be used for decoding.

There may be more than one set of encoding rules that can be applied to values of types that are defined using the ASN.1 notation. This Recommendation | International Standard defines three sets of encoding rules, called *basic encoding rules*, *canonical encoding rules* and *distinguished encoding rules*. Whereas the basic encoding rules give the sender of an encoding various choices as to how data values may be encoded, the canonical and distinguished encoding rules select just one encoding from those allowed by the basic encoding rules, eliminating all of the sender's options. The canonical and distinguished encoding rules differ from each other in the set of restrictions that they place on the basic encoding rules.

The distinguished encoding rules is more suitable than the canonical encoding rules if the encoded value is small enough to fit into the available memory and there is a need to rapidly skip over some nested values. The canonical encoding rules is more suitable than the distinguished encoding rules if there is a need to encode values that are so large that they cannot readily fit into the available memory or it is necessary to encode and transmit a part of a value before the entire value is available. The basic encoding rules is more suitable than the canonical or distinguished encoding rules if the encoding contains a set value or set-of value and there is no need for the restrictions that the canonical and distinguished encoding rules impose. This is due to the memory and CPU overhead that the latter encoding rules exact in order to guarantee that set values and set-of values have just one possible encoding.

Annex A gives an example of the application of the basic encoding rules. It does not form an integral part of this Recommendation | International Standard.

Annex B summarizes the assignment of object identifier and OID internationalized resource identifier values made in this Recommendation | International Standard. It does not form an integral part of this Recommendation | International Standard.

Annex C gives examples of applying the basic encoding rules for encoding reals. It does not form an integral part of this Recommendation | International Standard.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

1 Scope

This Recommendation | International Standard specifies a set of basic encoding rules that may be used to derive the specification of a transfer syntax for values of types defined using the notation specified in 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, and Rec. ITU-T X.683 | ISO/IEC 8824-4, collectively referred to as Abstract Syntax Notation One or ASN.1. These basic encoding rules are also to be applied for decoding such a transfer syntax in order to identify the data values being transferred. It also specifies a set of canonical and distinguished encoding rules that restrict the encoding of values to just one of the alternatives provided by the basic encoding rules.

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 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.680 (2020) | ISO/IEC 8824-1:2020, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- Recommendation ITU-T X.681 (2020) | ISO/IEC 8824-2:2020, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- Recommendation ITU-T X.682 (2020) | ISO/IEC 8824-3:2020, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*
- Recommendation ITU-T X.683 (2020) | ISO/IEC 8824-4:2020, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*

2.2 Additional references

- ISO *International Register of Coded Character Sets to be used with Escape Sequences.*
- ISO/IEC 2022:1994, *Information technology – Character code structure and extension techniques.*
- ISO/IEC 2375:2003, *Information technology – Procedure for registration of escape sequences and coded character sets.*
- ISO 6093:1985, *Information processing – Representation of numerical values in character strings for information interchange.*
- ISO/IEC 6429:1992, *Information technology – Control functions for coded character sets.*
- ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS).*

3 Definitions

For the purposes of this Recommendation | International Standard, the definitions of Rec. ITU-T X.200 | ISO/IEC 7498-1 and Rec. ITU-T X.680 | ISO/IEC 8824-1 and the following definitions apply.

3.1 canonical encoding: A complete encoding of an abstract value obtained by the application of encoding rules that have no implementation-dependent options. Such rules result in the definition of a 1-1 mapping between unambiguous and unique encodings and values in the abstract syntax.

3.2 constructed encoding: A data value encoding in which the contents octets are the complete encoding of one or more data values.

3.3 contents octets: That part of a data value encoding which represents a particular value, to distinguish it from other values of the same type.

3.4 data value: Information specified as the value of a type; the type and the value are defined using ASN.1.

3.5 dynamic conformance: A statement of the requirement for an implementation to adhere to the prescribed behaviour in an instance of communication.

3.6 encoding (of a data value): The complete sequence of octets used to represent the data value.

3.7 end-of-contents octets: Part of a data value encoding, occurring at its end, which is used to determine the end of the encoding.

NOTE – Not all encodings require end-of-contents octets.

3.8 identifier octets: Part of a data value encoding which is used to identify the type of the value.

NOTE – Some ITU-T Recommendations use the term "data element" for this sequence of octets, but the term is not used in this Recommendation | International Standard, as other Recommendations | International Standards use it to mean "data value".

3.9 length octets: Part of a data value encoding following the identifier octets which is used to determine the end of the encoding.

3.10 primitive encoding: A data value encoding in which the contents octets directly represent the value.

3.11 receiver: An implementation decoding the octets produced by a sender, in order to identify the data value which was encoded.

3.12 sender: An implementation encoding a data value for transfer.

3.13 static conformance: A statement of the requirement for support by an implementation of a valid set of features from among the defined features.

3.14 trailing 0 bit: A 0 in the last position of a bitstring value.

NOTE – The 0 in a bitstring value consisting of a single 0 bit is a trailing 0 bit. Its removal produces an empty bitstring.

4 Abbreviations

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

ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules of ASN.1
CER	Canonical Encoding Rules of ASN.1
DER	Distinguished Encoding Rules of ASN.1
ULA	Upper Layer Architecture
UTF8	Universal Transformation Function 8-bit (see ISO/IEC 10646, Annex D)

5 Notation

This Recommendation | International Standard references the notation defined by Rec. ITU-T X.680 | ISO/IEC 8824-1.

6 Convention

6.1 This Recommendation | International Standard specifies 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 purposes of this Recommendation | International Standard only, 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 purpose of this Recommendation | International Standard, two octet strings can be compared. One octet string is equal to another if they are of the same length and are the same at each octet position. An octet string, S_1, is greater than another, S_2, if and only if either:

a) S_1 and S_2 have identical octets in every position up to and including the final octet in S_2, but S_1 is longer; or

b) S_1 and S_2 have different octets in one or more positions, and in the first such position, the octet in S_1 is greater than that in S_2, considering the octets as unsigned binary numbers whose bit n has weight 2^{n-1}.

7 Conformance

7.1 Dynamic conformance is specified by clauses 8 to 12 inclusive.

7.2 Static conformance is specified by those standards which specify the application of one or more of these encoding rules.

7.3 Alternative encodings are permitted by the basic encoding rules as a sender's option. Receivers who claim conformance to the basic encoding rules shall support all alternatives.

NOTE – Examples of such alternative encodings appear in 8.1.3.2 b) and Table 3.

7.4 No alternative encodings are permitted by the Canonical Encoding Rules or Distinguished Encoding Rules.

8 Basic encoding rules

8.1 General rules for encoding

8.1.1 Structure of an encoding

8.1.1.1 The encoding of a data value shall consist of four components which shall appear in the following order:

a) identifier octets (see 8.1.2);

b) length octets (see 8.1.3);

c) contents octets (see 8.1.4);

d) end-of-contents octets (see 8.1.5).

8.1.1.2 The end-of-contents octets shall not be present unless the value of the length octets requires them to be present (see 8.1.3).

8.1.1.3 Figure 1 illustrates the structure of an encoding (primitive or constructed). Figure 2 illustrates an alternative constructed encoding.

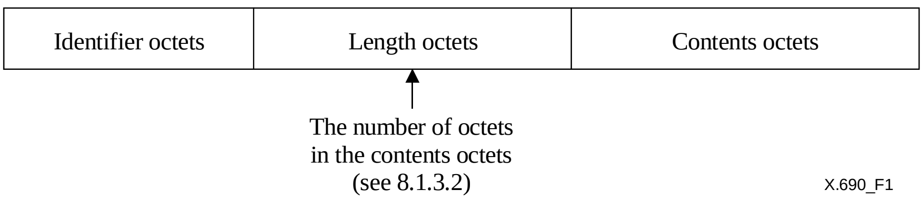

Figure 1 – Structure of an encoding

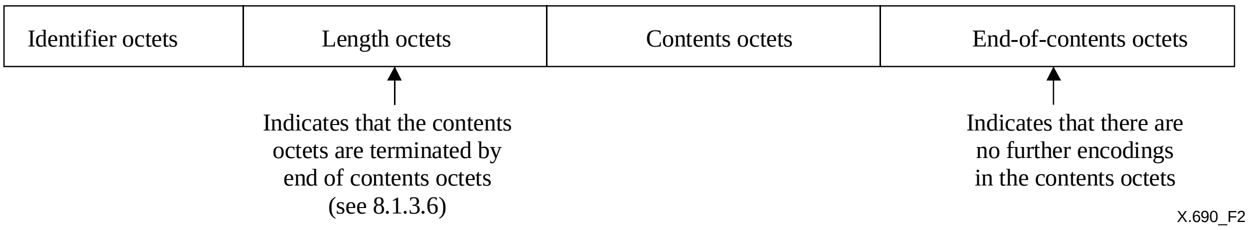

Figure 2 – An alternative constructed encoding

8.1.1.4 Encodings specified in this Recommendation | International Standard are not affected by either the ASN.1 subtype notation or the ASN.1 type extensibility notation.

NOTE – This means that all constraint notation is ignored when determining encodings, and all extensibility markers in **CHOICE**, **SEQUENCE** and **SET** are ignored, with the extensions treated as if they were in the extension root of the type.

8.1.1.5 There are no encoding instructions (see Rec. ITU-T X.680 | ISO/IEC 8824-1, 3.8.27) defined for the encoding rules specified in this Recommendation | International Standard.

8.1.2 Identifier octets

8.1.2.1 The identifier octets shall encode the ASN.1 tag (class and number) of the type of the data value.

8.1.2.2 For tags with a number ranging from zero to 30 (inclusive), the identifier octets shall comprise a single octet encoded as follows:

a) bits 8 and 7 shall be encoded to represent the class of the tag as specified in Table 1;

b) bit 6 shall be a zero or a one according to the rules of 8.1.2.5;

c) bits 5 to 1 shall encode the number of the tag as a binary integer with bit 5 as the most significant bit.

Table 1 – Encoding of class of tag

Class	Bit 8	Bit 7
Universal	0	0
Application	0	1
Context-specific	1	0
Private	1	1

8.1.2.3 Figure 3 illustrates the form of an identifier octet for a type with a tag whose number is in the range zero to 30 (inclusive).

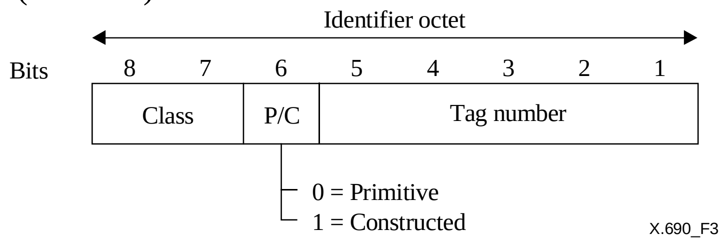

Figure 3 – Identifier octet (low tag number)

8.1.2.4 For tags with a number greater than or equal to 31, the identifier shall comprise a leading octet followed by one or more subsequent octets.

8.1.2.4.1 The leading octet shall be encoded as follows:

a) bits 8 and 7 shall be encoded to represent the class of the tag as listed in Table 1;

b) bit 6 shall be a zero or a one according to the rules of 8.1.2.5;

c) bits 5 to 1 shall be encoded as 11111_2.

8.1.2.4.2 The subsequent octets shall encode the number of the tag as follows:

a) bit 8 of each octet shall be set to one unless it is the last octet of the identifier octets;

b) bits 7 to 1 of the first subsequent octet, followed by bits 7 to 1 of the second subsequent octet, followed in turn by bits 7 to 1 of each further octet, up to and including the last subsequent octet in the identifier octets shall be the encoding of an unsigned binary integer equal to the tag number, with bit 7 of the first subsequent octet as the most significant bit;

c) bits 7 to 1 of the first subsequent octet shall not all be zero.

8.1.2.4.3 Figure 4 illustrates the form of the identifier octets for a type with a tag whose number is greater than 30.

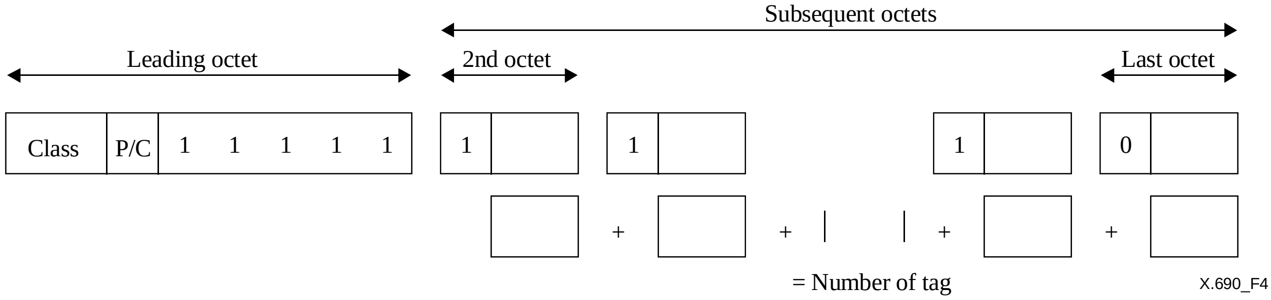

Figure 4 – Identifier octets (high tag number)

8.1.2.5 Bit 6 shall be set to zero if the encoding is primitive, and shall be set to one if the encoding is constructed.

NOTE – Subsequent subclauses specify whether the encoding is primitive or constructed for each type.

8.1.2.6 Rec. ITU-T X.680 | ISO/IEC 8824-1 specifies that the tag of a type defined using the **CHOICE** keyword takes the value of the tag of the type from which the chosen data value is taken.

8.1.2.7 Rec. ITU-T X.681 | ISO/IEC 8824-2, 14.2 and 14.4, specifies that the tag of a type defined using "ObjectClassFieldType" is indeterminate if it is a type field, a variable-type value field, or a variable-type value set field. This type is subsequently defined to be an ASN.1 type, and the complete encoding is then identical to that of a value of the assigned type (including the identifier octets).

8.1.3 Length octets

8.1.3.1 Two forms of length octets are specified. These are:

a) the definite form (see 8.1.3.3); and

b) the indefinite form (see 8.1.3.6).

8.1.3.2 A sender shall:

a) use the definite form (see 8.1.3.3) if the encoding is primitive;

b) use either the definite form (see 8.1.3.3) or the indefinite form (see 8.1.3.6), a sender's option, if the encoding is constructed and all immediately available;

c) use the indefinite form (see 8.1.3.6) if the encoding is constructed and is not all immediately available.

8.1.3.3 For the definite form, the length octets shall consist of one or more octets, and shall represent the number of octets in the contents octets using either the short form (see 8.1.3.4) or the long form (see 8.1.3.5) as a sender's option.

NOTE – The short form can only be used if the number of octets in the contents octets is less than or equal to 127.

8.1.3.4 In the short form, the length octets shall consist of a single octet in which bit 8 is zero and bits 7 to 1 encode the number of octets in the contents octets (which may be zero), as an unsigned binary integer with bit 7 as the most significant bit.

EXAMPLE

L = 38 can be encoded as 00100110_2

8.1.3.5 In the long form, the length octets shall consist of an initial octet and one or more subsequent octets. The initial octet shall be encoded as follows:

a) bit 8 shall be one;

b) bits 7 to 1 shall encode the number of subsequent octets in the length octets, as an unsigned binary integer with bit 7 as the most significant bit;

c) the value 11111111_2 shall not be used.

NOTE 1 – This restriction is introduced for possible future extension.

Bits 8 to 1 of the first subsequent octet, followed by bits 8 to 1 of the second subsequent octet, followed in turn by bits 8 to 1 of each further octet up to and including the last subsequent octet, shall be the encoding of an unsigned binary integer equal to the number of octets in the contents octets, with bit 8 of the first subsequent octet as the most significant bit.

EXAMPLE

L = 201 can be encoded as:

10000001_2

11001001_2

NOTE 2 – In the long form, it is a sender's option whether to use more length octets than the minimum necessary.

8.1.3.6 For the indefinite form, the length octets indicate that the contents octets are terminated by end-of-contents octets (see 8.1.5), and shall consist of a single octet.

8.1.3.6.1 The single octet shall have bit 8 set to one, and bits 7 to 1 set to zero.

8.1.3.6.2 If this form of length is used, then end-of-contents octets (see 8.1.5) shall be present in the encoding following the contents octets.

8.1.4 Contents octets

The contents octets shall consist of zero, one or more octets, and shall encode the data value as specified in subsequent clauses.

NOTE – The contents octets depend on the type of the data value; subsequent clauses follow the same sequence as the definition of types in ASN.1.

8.1.5 End-of-contents octets

The end-of-contents octets shall be present if the length is encoded as specified in 8.1.3.6, otherwise they shall not be present.

The end-of-contents octets shall consist of two zero octets.

NOTE – The end-of-contents octets can be considered as the encoding of a value whose tag is universal class, whose form is primitive, whose number of the tag is zero, and whose contents are absent, thus:

End-of-contents	Length	Contents
00_{16}	00_{16}	Absent

8.2 Encoding of a boolean value

8.2.1 The encoding of a boolean value shall be primitive. The contents octets shall consist of a single octet.

8.2.2 If the boolean value is:

`FALSE`

the octet shall be zero.

If the boolean value is

`TRUE`

the octet shall have any non-zero value, as a sender's option.

EXAMPLE

If of type **`BOOLEAN`**, the value **`TRUE`** can be encoded as:

Boolean	Length	Contents
01_{16}	01_{16}	FF_{16}

8.3 Encoding of an integer value

8.3.1 The encoding of an integer value shall be primitive. The contents octets shall consist of one or more octets.

8.3.2 If the contents octets of an integer value encoding consist of more than one octet, then the bits of the first octet and bit 8 of the second octet:

a) shall not all be ones; and

b) shall not all be zero.

NOTE – These rules ensure that an integer value is always encoded in the smallest possible number of octets.

8.3.3 The contents octets shall be a two's complement binary number equal to the integer value, and consisting of bits 8 to 1 of the first octet, followed by bits 8 to 1 of the second octet, followed by bits 8 to 1 of each octet in turn up to and including the last octet of the contents octets.

NOTE – The value of a two's complement binary number is derived by numbering the bits in the contents octets, starting with bit 1 of the last octet as bit zero and ending the numbering with bit 8 of the first octet. Each bit is assigned a numerical value of 2^N, where N is its position in the above numbering sequence. The value of the two's complement binary number is obtained by summing the numerical values assigned to each bit for those bits which are set to one, excluding bit 8 of the first octet, and then reducing this value by the numerical value assigned to bit 8 of the first octet if that bit is set to one.

8.4 Encoding of an enumerated value

The encoding of an enumerated value shall be that of the integer value with which it is associated.

NOTE – It is primitive.

8.5 Encoding of a real value

8.5.1 The encoding of a real value shall be primitive.

8.5.2 If the real value is the value plus zero, there shall be no contents octets in the encoding.

8.5.3 If the real value is the value minus zero, then it shall be encoded as specified in 8.5.9.

8.5.4 For a non-zero real value, if the base of the abstract value is 10, then the base of the encoded value shall be 10, and if the base of the abstract value is 2 the base of the encoded value shall be 2, 8 or 16 as a sender's option.

8.5.5 If the real value is non-zero, then the base used for the encoding shall be B' as specified in 8.5.4. If B' is 2, 8 or 16, a binary encoding, specified in 8.5.7, shall be used. If B' is 10, a character encoding, specified in 8.5.8, shall be used.

8.5.6 Bit 8 of the first contents octet shall be set as follows:

a) if bit 8 = 1, then the binary encoding specified in 8.5.7 applies;

b) if bit 8 = 0 and bit 7 = 0, then the decimal encoding specified in 8.5.8 applies;

c) if bit 8 = 0 and bit 7 = 1, then either a "SpecialRealValue" (see Rec. ITU-T X.680 | ISO/IEC 8824-1) or the value minus zero is encoded as specified in 8.5.9.

8.5.7 When binary encoding is used (bit 8 = 1), then if the mantissa M is non-zero, it shall be represented by a sign S, a positive integer value N and a binary scaling factor F, such that:

$$M = S \times N \times 2^F$$

$$0 \le F < 4$$

$$S = +1 \text{ or } -1$$

NOTE – The binary scaling factor F is required under certain circumstances in order to align the implied point of the mantissa to the position required by the encoding rules of this subclause. This alignment cannot always be achieved by modification of the exponent E. If the base B' used for encoding is 8 or 16, the implied point can only be moved in steps of 3 or 4 bits, respectively, by changing the component E. Therefore, values of the binary scaling factor F other than zero may be required in order to move the implied point to the required position.

8.5.7.1 Bit 7 of the first contents octets shall be 1 if S is –1 and 0 otherwise.

8.5.7.2 Bits 6 to 5 of the first contents octets shall encode the value of the base B' as follows:

Bits 6 to 5	*Base*
00	base 2
01	base 8
10	base 16
11	Reserved for further editions of this Recommendation | International Standard.

8.5.7.3 Bits 4 to 3 of the first contents octet shall encode the value of the binary scaling factor F as an unsigned binary integer.

8.5.7.4 Bits 2 to 1 of the first contents octet shall encode the format of the exponent as follows: