DRAFT INTERNATIONAL STANDARD
ISO/IEC DIS 8825-7

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 7: **Specification of Octet Encoding Rules (OER)**

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

Partie 7: Spécification des règles de codage des octets (OER)

ICS: 35.100.60

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC DIS 8825-7:2020(E)

CONTENTS

Page

Introduction

The publications 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 together describe Abstract Syntax Notation One (ASN.1), a notation for the definition of messages to be exchanged between peer applications.

This Recommendation | International Standard defines encoding rules that may be applied to values of ASN.1 types which have been defined using the notation specified in the above-mentioned publications. 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 are more than one set of encoding rules that can be applied to values of ASN.1 types. This Recommendation | International Standard defines two sets of Octet Encoding Rules, so-called because the encoding of every type takes a whole number of octets. Encoding and decoding data with the Octet Encoding Rules is usually faster than encoding and decoding the same data with the Basic Encoding Rules (described in Rec. ITU-T X.690 | ISO/IEC 8825-1) or the Packed Encoding Rules (described in Rec. ITU-T X.691 | ISO/IEC 8825-2).

NOTE – The encoding rules specified in this Recommendation | International Standard derive from the Octet Encoding Rules (OER) published by American Association of State Highway and Transportation Officials (AASHTO), Institute of Transportation Engineers (ITE) and National Electrical Manufacturers Association (NEMA) as NTCIP 1102:2004. In most practical cases, an implementation of this Recommendation | International Standard can interoperate with an implementation of NTCIP 1102.

Clauses 8 to 30 specify the BASIC-OER encoding of ASN.1 types.

Clause 31 specifies the CANONICAL-OER encoding of ASN.1 types.

Annex A is informative and contains examples of BASIC-OER and CANONICAL-OER encodings.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – ASN.1 encoding rules: Specification of Octet Encoding Rules (OER)

1 Scope

This Recommendation | International Standard specifies a set of Basic Octet Encoding Rules (BASIC-OER) that may be used to derive a transfer syntax for values of the types defined 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, Rec. ITU-T X.683 | ISO/IEC 8824-4. This Recommendation | International Standard also specifies a set of Canonical Octet Encoding Rules (CANONICAL-OER) which provides constraints on the Basic Octet Encoding Rules and produces a unique encoding for any given ASN.1 value. It is implicit in the specification of these encoding rules that they are also to be used for decoding.

The encoding rules specified in this Recommendation | International Standard:

- are used at the time of communication;
- are intended for use in circumstances where encoding/decoding speed is the major concern in the choice of encoding rules;
- allow the extension of an abstract syntax by addition of extra values for all forms of extensibility described in Rec. ITU-T X.680 | ISO/IEC 8824-1.

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 and the Unicode standard version 3.2.0:2002. It cannot be applied using later versions of these two standards.

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*.
- Recommendation ITU-T X.690 (2020) | ISO/IEC 8825-1:2020, *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 (2020) | ISO/IEC 8825-2:2020, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER).*

2.2 Additional references

- ISO/IEC 2375:2003, *Information technology – Procedure for registration of escape sequences and coded character sets.*
- *ISO International Register of Coded Character Sets to be Used with Escape Sequences.*
- ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS).*

3 Definitions

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

3.1 Specification of basic notation

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

3.2 Information object specification

For the purposes of this Recommendation | International Standard, all the definitions in Rec. ITU-T X.681 | ISO/IEC 8824-2 apply.

3.3 Constraint specification

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.682 | ISO/IEC 8824-3:

a) component relation constraint;

b) table constraint.

3.4 Parameterization of ASN.1 specification

This Recommendation | International Standard makes use of the following term defined in Rec. ITU-T X.683 | ISO/IEC 8824-4:

– variable constraint.

3.5 Basic Encoding Rules (BER)

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.690 | ISO/IEC 8825-1:

a) data value;

b) dynamic conformance;

c) encoding (of a data value);

d) receiver;

e) sender;

f) static conformance.

3.6 Packed Encoding Rules (PER)

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.691 | ISO/IEC 8825-2:

a) canonical encoding;

b) composite type;

c) composite value;

d) known-multiplier character string type;

e) outermost type;

f) relay-safe encoding;

g) simple type;

h) textually dependent.

3.7 Additional definitions

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

3.7.1 **abstract syntax value**: A value of an abstract syntax (defined as a set of values of a single ASN.1 type) which is to be encoded by BASIC-OER or CANONICAL-OER, or which is generated by BASIC-OER or CANONICAL-OER decoding.

3.7.2 **effective value constraint** (of an integer type): The smallest integer range that includes all the values of the integer type that are permitted by the OER-visible constraints (see 8.2.7).

3.7.3 **effective size constraint** (of a string type): The smallest integer range that includes the lengths of all the values of the string type that are permitted by the OER-visible constraints (see 8.2.8).

3.7.4 **fixed-size signed number**: A word (see 3.7.13) representing a negative, zero or positive whole number encoded as a signed integer encoding (see 3.7.9).

NOTE 1 – The least significant bit of the whole number is stored in bit 1 of the last octet of the word.

NOTE 2 – The range of integers that can be encoded as fixed-size signed numbers is −128 to 127 for a one-octet word, −32768 to 32767 for a two-octet word, −2147483648 to 2147483647 for a four-octet word, and −9223372036854775808 to 9223372036854775807 for an eight-octet word.

3.7.5 **fixed-size unsigned number**: A word (see 3.7.13) representing a zero or positive whole number encoded as an unsigned integer encoding (see 3.7.10).

NOTE 1 – The least significant bit of the whole number is stored in bit 1 of the last octet of the word.

NOTE 2 – The smallest integer that can be encoded as fixed-size unsigned numbers of any size is 0. The largest integer that can be encoded as a fixed-size unsigned number is 255 for a one-octet word, 65535 for a two-octet word, 4294967295 for a four-octet word, and 18446744073709551615 for an eight-octet word.

3.7.6 **length determinant**: A group of one or more consecutive octets encoding the length of a series of octets (see 8.6).

3.7.7 **octet**: A group of eight consecutive bits, numbered from bit 8 (the most significant bit) to bit 1 (the least significant bit).

NOTE – Within an OER encoding, each octet starts at a location that is a whole multiple of eight bits from the first bit of the encoding.

3.7.8 **OER-visible constraint**: An instance of use of the ASN.1 constraint notation that affects the OER encoding of a value.

3.7.9 **signed integer encoding**: The encoding of a whole number into a group of consecutive octets of a specified length as a 2's-complement binary integer, which provides representations for whole numbers that are equal to, greater than or less than zero.

NOTE – The value of a signed integer encoding is derived by numbering the bits in the octets of the group, starting with bit 1 of the last octet 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 (starting from 0) in the above numbering sequence. The value of the signed integer encoding 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.

3.7.10 **unsigned integer encoding**: The encoding of a whole number into a group of consecutive octets of a specified length as an unsigned binary integer, which provides representations for whole numbers that are equal to or greater than zero.

NOTE – The value of an unsigned integer encoding is derived by numbering the bits in the octets of the group, starting with bit 1 of the last octet 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 (starting from 0) in the above numbering sequence. The value of the unsigned integer encoding is obtained by summing the numerical values assigned to each bit for those bits which are set to one.

3.7.11 **variable-size signed number**: A group of one or more consecutive octets containing a negative, zero, or positive whole number encoded as a signed integer encoding, with the least significant bit of the binary number stored in bit 1 of the last octet of the variable-size signed number.

NOTE – There are no restrictions to the length of such a group of octets. In particular, the Basic Octet Encoding Rules (but not the Canonical Octet Encoding Rules) allow the presence of redundant octets set to 0 (for zero or positive values) or 255 (for negative values) at the beginning of the group.

3.7.12 **variable-size unsigned number**: A group of one or more consecutive octets containing a zero or positive whole number encoded as an unsigned integer encoding, with the least significant bit of the binary number stored in bit 1 of the last octet of the variable-size unsigned number.

NOTE – There are no restrictions to the length of such a group of octets. In particular, the Basic Octet Encoding Rules (but not the Canonical Octet Encoding Rules) allow the presence of redundant octets set to 0 at the beginning of the group.

3.7.13 **word**: A group of one, two, four or eight consecutive octets containing the encoding of a whole number, where the first octet contains the most significant part of the number and the last octet contains the least significant part of the number.

NOTE 1 – A single octet is also a word according to this definition. The octet ordering of words consisting of 2, 4 or 8 octets is big-endian.

NOTE 2 – Within an OER encoding, a word can start at any location within the encoding that is a whole number of octets from the beginning of the encoding (that is, there is no requirement that a word should start on a word boundary).

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
ITS	Intelligent Transportation Systems
NTCIP	National Transportation Communications for ITS Protocol
OER	Octet Encoding Rules of ASN.1
PER	Packed Encoding Rules of ASN.1
PDU	Protocol Data Unit

5 Convention

For the purposes 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 Encodings specified by this Recommendation | International Standard

6.1 This Recommendation | International Standard specifies two sets of encoding rules (together with their associated object identifiers) which can be used to encode and decode the values of an abstract syntax defined as the values of a single (known) ASN.1 type. This clause describes their applicability and properties.

6.2 Without knowledge of the type of the value encoded, it is not possible to determine the structure of the encoding. In particular, the end of the encoding cannot be determined from the encoding itself without knowledge of the type being encoded.

6.3 OER encodings are always relay-safe provided the abstract values of the types **EXTERNAL**, **EMBEDDED PDV** and **CHARACTER STRING** are constrained to prevent the carriage of OSI presentation context identifiers.

6.4 The most general set of encoding rules specified in this Recommendation | International Standard is BASIC-OER, which does not in general produce a canonical encoding.

6.5 A second set of encoding rules specified in this Recommendation | International Standard is CANONICAL-OER, which produces encodings that are canonical. This is defined as a restriction of implementation-dependent choices in the BASIC-OER encoding.

NOTE 1 – CANONICAL-OER produces encodings that have applications when authenticators need to be applied to abstract values.

NOTE 2 – Any implementation conforming to CANONICAL-OER for encoding is conformant to BASIC-OER for encoding. Any implementation conforming to BASIC-OER for decoding is conformant to CANONICAL-OER for decoding. Thus, encodings made according to CANONICAL-OER are encodings that are permitted by BASIC-OER.

6.6 If a type encoded with BASIC-OER or CANONICAL-OER contains **EXTERNAL**, **EMBEDDED PDV** or **CHARACTER STRING** types, then the outer encoding ceases to be relay-safe unless the transfer syntax used for all the **EXTERNAL**, **EMBEDDED PDV** or **CHARACTER STRING** types is relay-safe. If a type encoded with CANONICAL-OER contains **EXTERNAL**, **EMBEDDED PDV** or **CHARACTER STRING** types, then the outer encoding ceases to be canonical unless the encoding used for all the **EXTERNAL**, **EMBEDDED PDV**, and **CHARACTER STRING** types is canonical.

NOTE – The character transfer syntaxes supporting all character abstract syntaxes of the form **{iso standard 10646 level-1(1) …}** are canonical. Those supporting **{iso standard 10646 level-2(2) …}** and **{iso standard 10646 level-3(3) …}** are not always canonical. All the above character transfer syntaxes are relay-safe.

6.7 OER encodings are self-delimiting only with knowledge of the type of the encoded value. Encodings are always a whole multiple of eight bits. When carried in an **EXTERNAL** type, they shall be carried in the **OCTET STRING** choice alternative, unless the **EXTERNAL** type itself is encoded in OER, in which case the value may be encoded as a single ASN.1 type (i.e., an open type). When carried in an OSI presentation protocol, the "full encoding" (as defined in Rec. ITU-T X.226 | ISO/IEC 8823-1) with the **OCTET STRING** alternative shall be used.

6.8 CANONICAL-OER provides an alternative to both the Distinguished Encoding Rules (DER) and Canonical Encoding Rules (CER) specified in Rec. ITU-T X.690 | ISO/IEC 8825-1 where a canonical and relay-safe encoding is required.

7 Conformance

7.1 Dynamic conformance for the Basic Octet Encoding Rules is specified by clauses 8 to 30. Dynamic conformance for the Canonical Octet Encoding Rules is specified by clause 31.

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

7.3 Alternative encodings are permitted by the Basic Octet Encoding Rules as encoder's options. Decoders that claim conformance to BASIC-OER shall support all BASIC-OER encoding alternatives. No alternative encodings are permitted by the CANONICAL-OER for the encoding of an ASN.1 value.

7.4 The rules in this Recommendation | International Standard are specified in terms of an encoding procedure. Implementations are not required to mirror the procedure specified, provided the octet string produced as the complete encoding of an abstract syntax value is identical to one of those specified in this Recommendation | International Standard for the applicable transfer syntax.

7.5 Implementations performing decoding are required to produce the abstract syntax value corresponding to any received octet string which could be produced by a sender conforming to the encoding rules identified in the transfer syntax associated with the material being decoded.

NOTE – When CANONICAL-OER is used to provide a canonical encoding, it is recommended that any resulting encrypted hash value that is derived from it should have associated with it an algorithm identifier that identifies CANONICAL-OER as the transformation from the abstract syntax value to an initial bitstring (which is then hashed).

8 General provisions

8.1 Use of the type notation

8.1.1 These encoding rules make specific use of the ASN.1 type notation as 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, Rec. ITU-T X.683 | ISO/IEC 8824-4, and can only be applied to encode the values of a single ASN.1 type specified using that notation.

8.1.2 In particular, but not exclusively, they are dependent on the following information being retained in the ASN.1 type and value model underlying the use of the notation:

a) the nesting of choice types within choice types;

b) the tags placed on the components in a set type and on the alternatives in a choice type, and the numeric values given to an enumeration;

c) whether a set or sequence type component is optional or not;

d) whether a set or sequence type component has a default value or not;

e) the restricted range of values of a type which arise through the application of OER-visible constraints;

f) whether the type of a component is an open type.

8.2 Constraints

NOTE – The fact that some ASN.1 constraints may not be OER-visible for the purposes of encoding and decoding does not in any way affect the use of such constraints in the handling of errors detected during decoding, nor does it imply that values violating such constraints are allowed to be transmitted by a conforming sender. However, this Recommendation | International Standard makes no use of such constraints in the specification of encodings.

8.2.1 In general, the constraint on a type will consist of individual constraints combined using some or all of set arithmetic, contained subtype constraints, and serial application of constraints.

The following constraints are OER-visible:

a) non-extensible single value constraints and value range constraints on integer types;

b) non-extensible single value constraints on real types where the single value is either plus zero or minus zero or one of the special real values **`PLUS-INFINITY`**, **`MINUS-INFINITY`** and **`NOT-A-NUMBER`**;

c) non-extensible size constraints on known-multiplier character string types, octetstring types, and bitstring types;

d) non-extensible property settings constraints on the time type or on the useful and defined time types;

e) inner type constraints applying OER-visible constraints to real types when used to restrict the mantissa, base, or exponent;

f) inner type constraints applied to CHARACTER STRING or EMBEDDED-PDV types when used to restrict the value of the syntaxes component to a single value, or when used to restrict identification to the fixed alternative;

g) contained subtype constraints in which the constraining type carries an OER-visible constraint.

8.2.2 All other constraints are not OER-visible. In particular, the following constraints are not OER-visible:

a) constraints that are expressed in human-readable text or in ASN.1 comment;

b) variable constraints (see Rec. ITU-T X.683 | ISO/IEC 8824-4, 10.3 and 10.4);

c) user-defined constraints (see Rec. ITU-T X.682 | ISO/IEC 8824-3, 9.1);

d) table constraints (see Rec. ITU-T X.682 | ISO/IEC 8824-3);

e) component relation constraints (see Rec. ITU-T X.682 | ISO/IEC 8824-3, 10.7);

f) constraints whose evaluation is textually dependent on a table constraint or a component relation constraint (see Rec. ITU-T X.682 | ISO/IEC 8824-3);

g) extensible subtype constraints;

h) size constraints on restricted character string types which are not known-multiplier character string types (see clause 27);

i) single value subtype constraints applied to a character string type;

j) permitted alphabet constraints;

k) pattern constraints;

l) constraints on real types except those specified in 8.2.1 (b) and (e);

m) inner type constraints applied to components of unrestricted character string, embedded-pdv or external types, except those specified in 8.2.1 (f);

n) constraints on the useful types.

8.2.3 If a type is specified using a serial application of constraints, each of those constraints may or may not be individually OER-visible. If the last subtype constraint of the series of constraints is OER-visible and contains an extension marker, then that subtype constraint is extensible for the purposes of these encoding rules. Any other constraint is not extensible for the purposes of these encoding rules, even if it contains an extension marker.

NOTE – In a serial application of constraints, each subtype constraint removes the extensibility specified in earlier constraints of the series of constraints (see Rec. ITU-T X.680 | ISO/IEC 8824-1, 50.8).

8.2.4 If a constraint that is OER-visible is part of an **`INTERSECTION`** construction, then the resulting constraint is OER-visible, and consists of the **`INTERSECTION`** of all the OER-visible parts (with the non-OER-visible parts ignored).

8.2.5 If a constraint that is not OER-visible is part of a **`UNION`** construction, then the resulting constraint is not OER-visible.

8.2.6 If a constraint has an **`EXCEPT`** clause, the **`EXCEPT`** keyword and the following value set is completely ignored, whether the value set following the **`EXCEPT`** keyword is OER-visible or not.

8.2.7 The effective value constraint of an integer type is an integer range determined as follows, taking into account all the OER-visible constraints present in the type definition and ignoring any constraints that are not OER-visible:

a) The lower bound of the effective value constraint is the least permitted value of the integer type, if such a value exists; otherwise, the effective value constraint has no finite lower bound.

b) The upper bound of the effective value constraint is the greatest permitted value of the integer type, if such a value exists; otherwise, the effective value constraint has no finite upper bound.

8.2.8 The effective size constraint of a string type (a known-multiplier character string type, an octetstring type, or a bitstring type) is a single integer range determined as follows, taking into account all the OER-visible constraints present in the type definition and ignoring any constraints that are not OER-visible:

a) The lower bound of the effective size constraint is the length of the shortest permitted value of the string type (possibly zero).

b) The upper bound of the effective size constraint is the length of the longest permitted value of the string type, if such length is finite; otherwise, the effective size constraint has no finite upper bound.

8.3 Type and value model used for encoding

8.3.1 An ASN.1 type is either a simple type or a type built using other types. The notation permits the use of type references and tagging of types. For the purpose of these encoding rules, the use of type references and tagging have no effect on the encoding and are invisible in the model, except as stated in 18.2 and 20.1. The notation also permits the application of constraints and of error specifications. OER-visible constraints are present in the model as a restriction of the values of a type. Other constraints and error specifications do not affect encoding and are invisible in the OER type and value model.

8.3.2 A value to be encoded can be considered as either a simple value or as a composite value built using the structuring mechanisms from components which are either simple or composite values, paralleling the structure of the ASN.1 type definition.

8.4 Types to be encoded

8.4.1 Clauses 9 to 30 specify the encoding of the following types: Boolean, integer, enumerated, real, bitstring, octetstring, null, sequence, sequence-of, set, set-of, choice, object identifier, relative object identifier, internationalized resource reference, relative internationalized resource reference, embedded-pdv, external, restricted character string, unrestricted character string, time, and open types.

8.4.1 The selection type shall be encoded as an encoding of the selected type.

8.4.2 This Recommendation | International Standard does not contain specific provisions for the encoding of tagged types, except as stated in 18.2 and 20.1, tagging is not visible in the type and value model used for these encoding rules. Tagged types are thus encoded according to the type which has been tagged.

8.4.3 An encoding prefixed type is encoded according to the type which has been prefixed.

8.4.4 The useful types `GeneralizedTime`, `UTCTime`, and `ObjectDescriptor` shall be encoded as if they had been replaced by their definitions given in Rec. ITU-T X.680 | ISO/IEC 8824-1, clause 45. Constraints on the useful types are not OER-visible.

8.4.5 A type defined using a value set assignment shall be encoded as if the type had been defined using the production specified in Rec. ITU-T X.680 | ISO/IEC 8824-1, 16.8.

8.5 Production of a complete OER encoding

8.5.1 If an ASN.1 type is encoded using OER and the encoding is contained in:

a) an ASN.1 bitstring type or octetstring type; or

b) an ASN.1 open type; or

c) any part of an ASN.1 external or embedded-pdv type; or

d) any carrier protocol that is not defined using ASN.1,

then that ASN.1 type is defined as an outermost type, and clause 8.5.2 shall be applied to all the encodings of its values.

8.5.2 The series of words and groups of octets produced as a result of applying this Recommendation | International Standard to an abstract value of an outermost type shall be concatenated into a string of octets, which is the complete encoding of the abstract value of the outermost type.

8.6 Length determinant

8.6.1 A length determinant occurs at the beginning of the encoding of many types as specified in the respective clauses.