
**Information technology — ASN.1
encoding rules: Specification of Packed
Encoding Rules (PER)**

*Technologies de l'information — Règles de codage ASN.1:
Spécification des règles de codage compact (PER)*

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 8825-2:2008](https://standards.iteh.ai/catalog/standards/sist/a8151f4f-dfe9-4495-a0eb-41e60ed52718/iso-iec-8825-2-2008)

<https://standards.iteh.ai/catalog/standards/sist/a8151f4f-dfe9-4495-a0eb-41e60ed52718/iso-iec-8825-2-2008>

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 8825-2:2008

<https://standards.iteh.ai/catalog/standards/sist/a8151f4f-dfe9-4495-a0eb-41e60ed52718/iso-iec-8825-2-2008>



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	vi
Introduction	vii
1 Scope	1
2 Normative references	1
2.1 Identical Recommendations International Standards	1
2.2 Additional references	1
3 Definitions	2
3.1 Specification of Basic Notation	2
3.2 Information Object Specification	2
3.3 Constraint Specification	2
3.4 Parameterization of ASN.1 Specification	2
3.5 Basic Encoding Rules	2
3.6 PER Encoding Instructions	2
3.7 Additional definitions	2
4 Abbreviations	5
5 Notation	5
6 Convention	5
7 Encoding rules defined in this Recommendation International Standard	5
8 Conformance	6
9 PER encoding instructions	6
10 The approach to encoding used for PER	7
10.1 Use of the type notation	7
10.2 Use of tags to provide a canonical order	7
10.3 PER-visible constraints	7
10.4 Type and value model used for encoding	9
10.5 Structure of an encoding	9
10.6 Types to be encoded	10
11 Encoding procedures	10
11.1 Production of the complete encoding	10
11.2 Open type fields	11
11.3 Encoding as a non-negative-binary-integer	11
11.4 Encoding as a 2's-complement-binary-integer	12
11.5 Encoding of a constrained whole number	12
11.6 Encoding of a normally small non-negative whole number	13
11.7 Encoding of a semi-constrained whole number	13
11.8 Encoding of an unconstrained whole number	14
11.9 General rules for encoding a length determinant	14
12 Encoding the boolean type	16
13 Encoding the integer type	17
14 Encoding the enumerated type	17
15 Encoding the real type	18
16 Encoding the bitstring type	18
17 Encoding the octetstring type	19
18 Encoding the null type	20
19 Encoding the sequence type	20
20 Encoding the sequence-of type	21
21 Encoding the set type	21

22	Encoding the set-of type.....	22
23	Encoding the choice type	22
24	Encoding the object identifier type	23
25	Encoding the relative object identifier type.....	23
26	Encoding the internationalized resource reference type.....	23
27	Encoding the relative internationalized resource reference type.....	23
28	Encoding the embedded-pdv type	23
29	Encoding of a value of the external type.....	24
30	Encoding the restricted character string types.....	25
31	Encoding the unrestricted character string type	26
32	Encoding the time type, the useful time types, the defined time types and the additional time types.....	27
32.1	General	27
32.2	Encoding subtypes with the "Basic=Date" property setting	31
32.3	Encoding subtypes with the "Basic=Time" property setting	33
32.4	Encoding subtypes with the "Basic=Date-Time" property setting.....	36
32.5	Encoding subtypes with the "Basic=Interval Interval-type=SE" property setting.....	36
32.6	Encoding subtypes with the "Basic=Interval Interval-type=D" property setting	37
32.7	Encoding subtypes with the "Basic=Interval Interval-type=SD" or "Basic=Interval Interval-type=DE" property setting.....	38
32.8	Encoding subtypes with the "Basic=Rec-Interval Interval-type=SE" property setting.....	39
32.9	Encoding subtypes with the "Basic=Rec-Interval Interval-type=D" property setting.....	39
32.10	Encoding subtypes with the "Basic=Rec-Interval Interval-type=SD" or "Basic=Rec-Interval Interval-type=DE" property setting.....	40
32.11	Encoding subtypes with mixed settings of the Basic property.....	41
33	Object identifiers for transfer syntaxes	43
Annex A	Example of encoding.....	44
A.1	Record that does not use subtype constraints.....	44
A.1.1	ASN.1 description of the record structure.....	44
A.1.2	ASN.1 description of a record value	44
A.1.3	ALIGNED PER representation of this record value	44
A.1.4	UNALIGNED PER representation of this record value	45
A.2	Record that uses subtype constraints.....	47
A.2.1	ASN.1 description of the record structure.....	47
A.2.2	ASN.1 description of a record value	47
A.2.3	ALIGNED PER representation of this record value	47
A.2.4	UNALIGNED PER representation of this record value	48
A.3	Record that uses extension markers	49
A.3.1	ASN.1 description of the record structure.....	49
A.3.2	ASN.1 description of a record value	50
A.3.3	ALIGNED PER representation of this record value	50
A.3.4	UNALIGNED PER representation of this record value	52
A.4	Record that uses extension addition groups	53
A.4.1	ASN.1 description of the record structure.....	53
A.4.2	ASN.1 description of a record value	54
A.4.3	ALIGNED PER representation of this record value	54
A.4.4	UNALIGNED PER representation of this record value	54
Annex B	Combining PER-visible and non-PER-visible constraints	56
B.1	General.....	56
B.2	Extensibility and visibility of constraints in PER	56
B.2.1	General	56
B.2.2	PER-visibility of constraints	57
B.2.3	Effective constraints.....	58
B.3	Examples.....	59
Annex C	Support for the PER algorithms	61

Annex D Support for the ASN.1 rules of extensibility..... 62

Annex E Tutorial annex on concatenation of PER encodings..... 63

Annex F Identification of Encoding Rules 64

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 8825-2:2008](https://standards.iteh.ai/catalog/standards/sist/a8151f4f-dfe9-4495-a0eb-41e60ed52718/iso-iec-8825-2-2008)
<https://standards.iteh.ai/catalog/standards/sist/a8151f4f-dfe9-4495-a0eb-41e60ed52718/iso-iec-8825-2-2008>

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 8825-2 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.691 (11/2008).

This fourth edition cancels and replaces the third edition (ISO/IEC 8825-2:2002), which has been technically revised. It also incorporates the Amendments ISO/IEC 8825-2:2002/Amd.1:2004, ISO/IEC 8825-2:2002/Amd.2:2007 and ISO/IEC 8825-2:2002/Amd.3:2008, and the Technical Corrigenda ISO/IEC 8825-2:2002/Cor.1:2006 and ISO/IEC 8825-2:2002/Cor.2:2006.

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 publications 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 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 types defined using the notation specified in ITU-T Rec. X.680 | ISO/IEC 8824-1. 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 a set of Packed Encoding Rules (PER), so called because they achieve a much more compact representation than that achieved by the Basic Encoding Rules (BER) and its derivatives described in ITU-T Rec. X.690 | ISO/IEC 8825-1 which is referenced for some parts of the specification of these Packed Encoding Rules.

iTeh STANDARD PREVIEW (standards.iteh.ai)

ISO/IEC 8825-2:2008

<https://standards.iteh.ai/catalog/standards/sist/a8151f4f-dfe9-4495-a0eb-41e60ed52718/iso-iec-8825-2-2008>

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

ISO/IEC 8825-2:2008

<https://standards.iteh.ai/catalog/standards/sist/a8151f4f-dfe9-4495-a0eb-41e60ed52718/iso-iec-8825-2-2008>

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)

1 Scope

This Recommendation | International Standard specifies a set of Packed Encoding Rules that may be used to derive a transfer syntax for values of types defined in ITU-T Rec. X.680 | ISO/IEC 8824-1. These Packed Encoding Rules are also to be applied for decoding such a transfer syntax in order to identify the data values being transferred.

The encoding rules specified in this Recommendation | International Standard:

- are used at the time of communication;
- are intended for use in circumstances where minimizing the size of the representation of values is the major concern in the choice of encoding rules;
- allow the extension of an abstract syntax by addition of extra values, preserving the encodings of the existing values, for all forms of extension described in ITU-T Rec. X.680 | ISO/IEC 8824-1;
- can be modified in accordance with the provisions of ITU-T Rec. X.695 | ISO/IEC 8825-6.

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.

2.1 Identical Recommendations | International Standards

- 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.695 (2008) | ISO/IEC 8825-6:2008, *Information technology – ASN.1 encoding rules: Registration and application of PER encoding instructions*.

2.2 Additional references

- ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange*.
- 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 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 ITU-T Rec. X.680 | ISO/IEC 8824-1 apply.

3.2 Information Object Specification

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

3.3 Constraint Specification

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

- variable constraint.

3.5 Basic Encoding Rules

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

- a) dynamic conformance;
- b) static conformance;
- c) data value;
- d) encoding (of a data value);
- e) sender;
- f) receiver.

3.6 PER Encoding Instructions

This Recommendation | International Standard makes use of the following term defined in ITU-T Rec. X.695 | ISO/IEC 8825-6:

- identifying keyword.

3.7 Additional definitions

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

3.7.1 2's-complement-binary-integer encoding: The encoding of a whole number into a bit-field (octet-aligned in the ALIGNED variant) of a specified length, or into the minimum number of octets that will accommodate that whole number encoded as a 2's-complement-integer, which provides representations for whole numbers that are equal to, greater than, or less than zero, as specified in 11.4.

NOTE 1 – 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.

NOTE 2 – *Whole number* is a synonym for the mathematical term *integer*. It is used here to avoid confusion with the ASN.1 type *integer*.

3.7.2 abstract syntax value: A value of an abstract syntax (defined as the set of values of a single ASN.1 type), which is to be encoded by PER, or which is to be generated by PER decoding.

NOTE – The single ASN.1 type associated with an abstract syntax is formally identified by an object of class **ABSTRACT-SYNTAX**.

3.7.3 bit-field: The product of some part of the encoding mechanism that consists of an ordered set of bits that are not necessarily a multiple of eight.

NOTE – If the use of this term is followed by "octet-aligned in the ALIGNED variant", this means that the bit-field is required to begin on an octet boundary in the complete encoding for the aligned variant of PER.

3.7.4 canonical encoding: A complete encoding of an abstract syntax 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 bitstrings in the transfer syntax and values in the abstract syntax.

3.7.5 composite type: A set, sequence, set-of, sequence-of, choice, embedded-pdv, external or unrestricted character string type.

3.7.6 composite value: The value of a composite type.

3.7.7 constrained whole number: A whole number which is constrained by PER-visible constraints to lie within a range from "lb" to "ub" with the value "lb" less than or equal to "ub", and the values of "lb" and "ub" as permitted values.

NOTE – Constrained whole numbers occur in the encoding which identifies the chosen alternative of a choice type, the length of character, octet and bit string types whose length has been restricted by PER-visible constraints to a maximum length, the count of the number of components in a sequence-of or set-of type that has been restricted by PER-visible constraints to a maximum number of components, the value of an integer type that has been constrained by PER-visible constraints to lie within finite minimum and maximum values, and the value that denotes an enumeration in an enumerated type.

3.7.8 effective size constraint (for a constrained string type): A single finite size constraint that could be applied to a built-in string type and whose effect would be to permit all and only those lengths that can be present in the constrained string type.

NOTE 1 – For example, the following has an effective size constraint:

```
A ::= IA5String (SIZE(1..4) | SIZE(10..15))
```

since it can be rewritten with a single size constraint that applies to all values:

```
A ::= IA5String (SIZE(1..4 | 10..15))
```

whereas the following has no effective size constraint since the string can be arbitrarily long if it does not contain any characters other than 'a', 'b' and 'c':

```
B ::= IA5String (SIZE(1..4) | FROM("abc"))
```

NOTE 2 – The effective size constraint is used only to determine the encoding of lengths.

3.7.9 effective permitted-alphabet constraint (for a constrained restricted character string type): A single permitted-alphabet constraint that could be applied to a built-in known-multiplier character string type and whose effect would be to permit all and only those characters that can be present in at least one character position of any one of the values in the constrained restricted character string type.

NOTE 1 – For example, in:

```
Ax ::= IA5String (FROM("AB") | FROM("CD"))
```

```
Bx ::= IA5String (SIZE(1..4) | FROM("abc"))
```

Ax has an effective permitted-alphabet constraint of "ABCD". **Bx** has an effective permitted-alphabet constraint that consists of the entire **IA5String** alphabet since there is no smaller permitted-alphabet constraint that applies to all values of **Bx**.

NOTE 2 – The effective permitted-alphabet constraint is used only to determine the encoding of characters.

3.7.10 enumeration index: The non-negative whole number associated with an "EnumerationItem" in an enumerated type. The enumeration indices are determined by sorting the "EnumerationItem"s into ascending order by their enumeration value, then by assigning an enumeration index starting with zero for the first "EnumerationItem", one for the second, and so on up to the last "EnumerationItem" in the sorted list.

NOTE – "EnumerationItem"s in the "RootEnumeration" are sorted separately from those in the "AdditionalEnumeration".

3.7.11 extensible for PER encoding: A property of a type which requires that PER identifies an encoding of a value as that of a root value or as that of an extension addition.

NOTE – Root values are normally encoded more efficiently than extension additions.

3.7.12 field-list: An ordered set of bit-fields that is produced as a result of applying these encoding rules to components of an abstract value.

3.7.13 indefinite-length: An encoding whose length is greater than 64K-1 or whose maximum length cannot be determined from the ASN.1 notation.

3.7.14 fixed-length type: A type such that the value of the outermost length determinant in an encoding of this type can be determined (using the mechanisms specified in this Recommendation | International Standard) from the type notation (after the application of PER-visible constraints only) and is the same for all possible values of the type.

3.7.15 fixed value: A value such that it can be determined (using the mechanisms specified in this Recommendation | International Standard) that this is the only permitted value (after the application of PER-visible constraints only) of the type governing it.

3.7.16 known-multiplier character string type: A restricted character string type where the number of octets in the encoding is a known fixed multiple of the number of characters in the character string for all permitted character string values. The known-multiplier character string types are **IA5String**, **PrintableString**, **VisibleString**, **NumericString**, **UniversalString** and **BMPString**.

3.7.17 length determinant: A count (of bits, octets, characters, or components) determining the length of part or all of a PER encoding.

3.7.18 normally small non-negative whole number: A part of an encoding which represents values of an unbounded non-negative integer, but where small values are more likely to occur than large ones.

3.7.19 normally small length: A length encoding which represents values of an unbounded length, but where small lengths are more likely to occur than large ones.

3.7.20 non-negative-binary-integer encoding: The encoding of a constrained or semi-constrained whole number into either a bit-field of a specified length, or into a bit-field (octet-aligned in the ALIGNED variant) of a specified length, or into the minimum number of octets that will accommodate that whole number encoded as a non-negative-binary-integer which provides representations for whole numbers greater than or equal to zero, as specified in 11.3.

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.

3.7.21 outermost type: An ASN.1 type whose encoding is included in a non-ASN.1 carrier or as the value of other ASN.1 constructs (see 11.1.1).

NOTE – PER encodings of an outermost type are always an integral multiple of eight bits.

3.7.22 PER-visible constraint: An instance of use of the ASN.1 constraint notation which affects the PER encoding of a value.

3.7.23 relay-safe encoding: A complete encoding of an abstract syntax value which can be decoded (including any embedded encodings) without knowledge of the environment in which the encoding was performed.

3.7.24 semi-constrained whole number: A whole number which is constrained by PER-visible constraints to exceed or equal some value "lb" with the value "lb" as a permitted value, and which is not a constrained whole number.

NOTE – Semi-constrained whole numbers occur in the encoding of the length of unconstrained (and in some cases constrained) character, octet and bit string types, the count of the number of components in unconstrained (and in some cases constrained) sequence-of and set-of types, and the value of an integer type that has been constrained to exceed some minimum value.

3.7.25 simple type: A type that is not a composite type.

3.7.26 textually dependent: A term used to identify the case where if some reference name is used in evaluating an element set, the value of the element set is considered to be dependent on that reference name, regardless of whether the actual set arithmetic being performed is such that the final value of the element set is independent of the actual element set value assigned to the reference name.

NOTE – For example, the following definition of **Foo** is textually dependent on **Bar** even though **Bar** has no effect on **Foo**'s set of values (thus, according to 10.3.6 the constraint on **Foo** is not PER-visible since **Bar** is constrained by a table constraint and **Foo** is textually dependent on **Bar**).

```
MY-CLASS ::= CLASS { &name PrintableString, &age INTEGER } WITH SYNTAX{&name , &age}
MyObjectSet MY-CLASS ::= { {"Jack", 7} | {"Jill", 5} }
```

```
Bar ::= MY-CLASS.&age ({MyObjectSet})
Foo ::= INTEGER (Bar | 1..100)
```

3.7.27 unconstrained whole number: A whole number which is not constrained by PER-visible constraints.

NOTE – Unconstrained whole numbers occur only in the encoding of a value of the integer type.

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
PER	Packed Encoding Rules of ASN.1
16K	16384
32K	32768
48K	49152
64K	65536

5 Notation

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)

6 Convention

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 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 the "least significant bit".

6.3 The term "octet" is frequently used in this Recommendation | International Standard to stand for "eight bits". The use of this term in place of "eight bits" does not carry any implications of alignment. Where alignment is intended, it is explicitly stated in this Recommendation | International Standard.

7 Encoding rules defined in this Recommendation | International Standard

7.1 This Recommendation | International Standard specifies four 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.

7.2 Without knowledge of the type of the value encoded, it is not possible to determine the structure of the encoding (under any of the PER encoding rule algorithms). In particular, the end of the encoding cannot be determined from the encoding itself without knowledge of the type being encoded.

7.3 PER 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.

7.4 The most general encoding rule algorithm specified in this Recommendation | International Standard is BASIC-PER, which does not in general produce a canonical encoding.

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

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

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

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

7.7 Both BASIC-PER and CANONICAL-PER come in two variants, the **ALIGNED** variant, and the **UNALIGNED** variant. In the **ALIGNED** variant, padding bits are inserted from time to time to restore octet alignment. In the **UNALIGNED** variant, no padding bits are ever inserted.

7.8 There are no interworking possibilities between the **ALIGNED** variant and the **UNALIGNED** variant.

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

7.10 The rules of this Recommendation | International Standard apply to both algorithms and to both variants unless otherwise stated (but see 9.2 and 9.3).

7.11 Annex C is informative, and gives recommendations on which combinations of PER to implement in order to maximize the chances of interworking.

iTech STANDARD PREVIEW
(standards.itech.ai)

8 Conformance

8.1 Dynamic conformance is specified by clause 9 onwards.

8.2 Static conformance is specified by those standards which specify the application of these Packed Encoding Rules.

NOTE – Annex C provides guidance on static conformance in relation to support for the two variants of the two encoding rule algorithms. This guidance is designed to ensure interworking, while recognizing the benefits to some applications of encodings that are neither relay-safe nor canonical.

8.3 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 bit 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.

8.4 Implementations performing decoding are required to produce the abstract syntax value corresponding to any received bit 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 1 – In general there are no alternative encodings defined for the BASIC-PER explicitly stated in this Recommendation | International Standard. The BASIC-PER becomes canonical by specifying relay-safe operation and by restricting some of the encoding options of other ISO/IEC Standards that are referenced. CANONICAL-PER provides an alternative to both the Distinguished Encoding Rules and Canonical Encoding Rules (see ITU-T Rec. X.690 | ISO/IEC 8825-1) where a canonical and relay-safe encoding is required.

NOTE 2 – When CANONICAL-PER 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-PER as the transformation from the abstract syntax value to an initial bitstring (which is then hashed).

9 PER encoding instructions

9.1 PER encoding instructions can be associated with a type in accordance with the provisions of ITU-T Rec. X.680 | ISO/IEC 8824-1 and ITU-T Rec. X.695 | ISO/IEC 8825-6.

NOTE 1 – The application of some PER encoding instructions can make it impossible to encode all the abstract values of the type. Where this can arise, the specific PER encoding instruction identifies the problem. It is a designers decision, based on the possible need to use multiple encoding rules, whether to add an explicit constraint on the type in order to restrict the range of

abstract values to those that can be handled by the encoding using the PER encoding instruction. This can make the specification less readable, but ensures that all encoding rules can encode all allowed abstract values, making relaying possible without errors.

NOTE 2 – Each PER encoding instruction starts with an identifying keyword that unambiguously identifies that encoding instruction.

9.2 If the ALIGNED version of either BASIC-PER or CANONICAL-PER is in use, then all PER encoding instructions shall be silently ignored and have no effect on the encoding.

9.3 If the UNALIGNED version of either BASIC-PER or CANONICAL-PER is in use, then if a type has an associated encoding instruction, the following subclauses shall apply.

9.3.1 If the identifying keyword is not known, then a "not supported" error message shall be issued.

9.3.2 If the identifying keyword is known, the procedures of this Recommendation | International Standard shall be modified by the amendments to those procedures that are specified by the PER encoding instruction (see ITU-T Rec. X.695 | ISO/IEC 8825-6).

NOTE 1 – If multiple PER encoding instructions are associated with a type, then the amendments specified for all of them shall be applied.

NOTE 2 – It is an error in the register of PER encoding instructions if amendments produced by two or more separate encoding instructions conflict and it is not stated that they are mutually exclusive.

10 The approach to encoding used for PER

10.1 Use of the type notation

10.1.1 These encoding rules make specific use of the ASN.1 type notation as specified in ITU-T Rec. X.680 | ISO/IEC 8824-1, and can only be applied to encode the values of a single ASN.1 type specified using that notation.

10.1.2 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 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 PER-visible constraints (only);
- f) whether a component is an open type;
- g) whether a type is extensible for PER encoding.

10.2 Use of tags to provide a canonical order

This Recommendation | International Standard requires components of a set type and a choice type to be canonically ordered independent of the textual ordering of the components. The canonical order is determined by sorting the outermost tag of each component, as specified in ITU-T Rec. X.680 | ISO/IEC 8824-1, 8.6.

10.3 PER-visible constraints

NOTE – The fact that some ASN.1 constraints may not be PER-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.

10.3.1 Constraints that are expressed in human-readable text or in ASN.1 comment are not PER-visible.

10.3.2 Variable constraints are not PER-visible (see ITU-T Rec. X.683 | ISO/IEC 8824-4, 10.3 and 10.4).

10.3.3 User-defined constraints (see ITU-T Rec. X.682 | ISO/IEC 8824-3, 9.1) are not PER visible.

10.3.4 Table constraints are not PER-visible (see ITU-T Rec. X.682 | ISO/IEC 8824-3).

10.3.5 Component relation constraints (see ITU-T Rec. X.682 | ISO/IEC 8824-3, 10.7) are not PER-visible.

10.3.6 Constraints whose evaluation is textually dependent on a table constraint or a component relation constraint are not PER-visible (see ITU-T Rec. X.682 | ISO/IEC 8824-3).