INTERNATIONAL STANDARD ISO/IEC 8825-2:1998
TECHNICAL CORRIGENDUM 2

Published 2002-04-15

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

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

TECHNICAL CORRIGENDUM 2

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

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to International Standard ISO/IEC 8825-2:1998 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems.*

ICS 35.100.60 **Ref. No. ISO/IEC 8825-2:1998/Cor.2:2002(E)**

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

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

TECHNICAL CORRIGENDUM 2

1) Subclause 3.7.9

Change subclause 3.7.9 to:

3.7.9 effective PermittedAlphabet constraint (for a constrained restricted character string type): A single PermittedAlphabet 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 – For example, in:

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

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

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

2) New subclause 9.3.6 *bis*

Add a new subclause 9.3.6 bis as follows:

9.3.6 *bis* Pattern constraints are not PER-visible.

3) Subclause 9.3.10

Change subclause 9.3.10 to:

9.3.10 The effective PermittedAlphabet constraint for a constrained type is a single PermittedAlphabet constraint which allows a character if and only if there is some value of the constrained type that contains that character. If all characters of the type being constrained can be present in some value of the constrained type, then the effective PermittedAlphabet constraint is the set of characters defined for the unconstrained type.

NOTE 1 – In the definition of a constrained type, multiple PER-visible constraints may be applied either directly or through the use of "ContainedSubtype"s.

NOTE 2 – See Annex B for observations on the effect of combining constraints that individually are PER-visible.

4) Subclause 10.9.4.1

Change subclause 10.9.4.1 to:

10.9.4.1 If the length determinant "n" to be encoded is a constrained whole number with "ub" less than 64K, then ("n" – "lb") shall be encoded as a non-negative-binary-integer (as specified in 10.3) using the minimum number of bits necessary to encode the "range" ("ub" – "lb" + 1), unless "range" is 1, in which case there shall be no length encoding. If "n" is non-zero this shall be followed by an associated field or list of fields, completing these procedures. If "n" is zero there shall be no further addition to the field-list, completing these procedures.

NOTE – If "range" satisfies the inequality $2^m <$ "range" $\leq 2^{m+1}$, then the number of bits in the length determinant is m + 1.