INTERNATIONAL STANDARD ISO/IEC 8824-1:1998
TECHNICAL CORRIGENDUM 3

Published 2002-05-01

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

Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation

TECHNICAL CORRIGENDUM 3

Technologies de l'information — Notation de syntaxe abstraite numéro un (ASN.1): Spécification de la notation de base

RECTIFICATIF TECHNIQUE 3

Technical Corrigendum 3 to International Standard ISO/IEC 8824-1:1998 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

Withdrawn
STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 8824-1:1998/COR3:2002

ICS 35.100.60 **Ref. No. ISO/IEC 8824-1:1998/Cor.3:2002(E)**

© ISO/IEC 2002 – All rights reserved

Printed in Switzerland

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 8824-1:1998/COR3:2002
Withdrawn

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation

TECHNICAL CORRIGENDUM 3

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 8824-1:1998/COR3:2002
Withdrawn

1) Introduction

In the Introduction (page vi), replace the second and third paragraphs (beginning with "Although this standard notation" and "Outside the context of OSI", respectively) with the following text :

In some protocol architectures, each message is specified as the binary value of a sequence of octets. However, standards-writers need to define quite complex data types to carry their messages, without concern for their binary representation. In order to specify these data types, they require a notation that does not necessarily determine the representation of each value. ASN.1 is such a notation. This notation is supplemented by the specification of one or more algorithms called **encoding rules** that determine the value of the octets that carry the application semantics (called the **transfer syntax**). ITU-T Rec.X.690 | ISO/IEC 8825-1 and ITU-T Rec. X.691 | ISO/IEC 8825-2 specify two families of standardized encoding rules, called **Basic Encoding Rules (BER)** and **Packed Encoding Rules (PER)**. Some users wish to redefine their legacy protocols using ASN.1, but cannot use standardized encoding rules because they need to retain their existing binary representations. Other users wish to have more complete control over the exact layout of the bits on the wire (the transfer syntax). These requirements are addressed by ITU-T Rec.X.692 | ISO/IEC 8825-3 (expected to be approved late 2001) which specify an **Encoding Control Notation (ECN)** for ASN.1. ECN enables designers to formally specify the abstract syntax of a protocol using ASN.1, but to then (if they so wish) take complete or partial control of the bits on the wire by writing an accompanying ECN specification (which may reference standardized Encoding Rules for some parts of the encoding).

There is increasing recognition of the notion of an abstract value of some class (type) divorced from the details of any particular encoding. In order to correctly interpret the bit-pattern representation of the value, it is necessary to know (usually from the context), the class (type) of the value being represented, as well as the encoding mechanism being employed. Thus, the identification of a type is an important part of this Recommendation | International Standard.

2) Subclause 2.1

Add the following item at the end of subclause 2.1:

– ITU-T Recommendation X.692 (2001) | ISO/IEC 8825-3:2001, *Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN).*

3) Subclause 3.8

Add a new subclause 3.8.15 bis as follows:

3.8.15 *bis* contents constraint: A constraint on a bit string or octet string type that specifies either that the contents are to be an encoding of a specified ASN.1 type, or that specified procedures are to be used to produce the contents.

4) Clause 4

In clause 4 add the following line after the line that defines DNIC:

ECN Encoding Control Notation of ASN.1

5) New subclause 11.8 *bis*

Add a new subclause 11.8 bis as follows:

11.8 *bis* Real number item

Name of item – realnumber

A "realnumber" shall consist of an integer part that is a series of one or more digits, and optionally a decimal point (.). The decimal point can optionally be followed by a fractional part which is one or more digits. The integer part, decimal point or fractional part (which ever is last present) can optionally be followed by an e or E and an optionally signed exponent which is one or more digits. The leading digit of the integer part shall not be zero unless immediately followed by the decimal point. The leading digit of the exponent shall not be zero unless the exponent is a single digit.

NOTE – The "realnumber" item is always mapped to real value by interpreting it as decimal notation.

6) Subclause 11.18

Add the reserved words "CONTAINING" *and* "ENCODED" *to both the list of reserved words and to Note 2, and* "PATTERN" *to the list of reserved words.*

7) Subclause 20.5 Note 1

Change Note 1 to read:

NOTE 1 – Non-zero values represented by "base" 2 and by "base" 10 are considered to be distinct abstract values even if they evaluate to the same real number value, and may carry different application semantics.

8) Subclause 20.6

Change subclause 20.6 to read:

20.6 The value of a real type shall be defined by the notation "RealValue":

```
RealValue ::=  NumericRealValue | SpecialRealValue

NumericRealValue ::=
        realnumber                  |
        "-" realnumber              |
        SequenceValue  -- Value of the associated sequence type

SpecialRealValue ::=
        PLUS-INFINITY | MINUS-INFINITY
```

The "SequenceValue" form for "NumericRealValue" shall not be used for zero values.

9) New subclause 20.7

Add a new subclause 20.7:

20.7 When the "realnumber" notation is used it identifies the corresponding "base" 10 abstract value. If the RealType is constrained to "base" 2, the "realnumber" identifies the "base" 2 abstract value corresponding either to the decimal value specified by the "realnumber" or to a locally-defined precision if an exact representation is not possible.

10) Subclause 38.1

Add "UTF8String, " *after* "UniversalString,".

11) Subclause 38.4

Add " and UTF8String" *after* "UniversalString".

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 8824-1:1998/Cor 3:2002
Withdrawn

12) Subclause 48.1

Add the following to the end of **SubtypeElements**:

PatternConstraint

and at the end of the line "**InnerTypeConstraints**"*, add* "|".

Add a new column to Table 6 with the heading "PatternConstraint"*, with an entry of* "Yes [a)]" *for* "Restricted Character String Types" *and* "No" *for all other rows.*

Add "UTF8String, " *after* "VisibleString" *to footnote [a)] of Table 6.*

13) Subclause 48.9

Add a new subclause 48.9 as follows:

48.9 Pattern constraint

48.9.1 The "PatternConstraint" notation shall be:

PatternConstraint ::= PATTERN Value

48.9.2 "Value" shall be a "cstring" of type UniversalString (or a reference to such a character string) which contains a regular expression as defined in Annex H. The "PatternConstraint" selects those values of the parent type that satisfy the regular expression. The entire value shall satisfy the entire regular expression, i.e. the "PatternConstraint" does not select values whose leading characters match the (entire) regular expression but which contain further trailing characters.

NOTE – "Value" is formally defined as a value of type UniversalString, but the sets of values of type UniversalString and UTF8String are the same (see 36.13). Thus a totally equivalent defintion could have been to say that "Value" is a value of type UTF8String.

14) Annex C, subclause C.2.8

Change the two C.2.8 example object identifiers to use values assigned under the "asn1(1) examples(123)" arc:

```
packedBCDStringAbstractSyntaxId OBJECT IDENTIFIER ::=
        { joint-iso-itu-t asn1(1) examples(123) packedBCD(2) charSet(0) }

packedBCDStringTransferSyntaxId OBJECT IDENTIFIER ::=
        { joint-iso-itu-t asn1(1) examples(123) packedBCD(2) characterTransferSyntax(1) }
```

15) Annex F, subclause F.3.2.2 to Amendment 2 (Semantic model)

Add the following after item c):

c bis) Any occurrence of "realnumber" shall be transformed to a "base" 10 associated "SequenceValue". Any occurrence of the "RealValue" associated with "SequenceValue" shall be transformed to the associated "SequenceValue" of the same "base", such that the last digit of the mantissa is not zero.

16) Annex G

Add the following to the end of **SubtypeElements**:

PatternConstraint

and at the end of the line "**InnerTypeConstraints**" *add a* "|".

Also, add the following to Annex G after the line which begins with "**PresenceConstraint ::=**".

PatternConstraint ::= PATTERN Value

17) New Annex H

Add the following as a new Annex H.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 8824-1:1998/COR3:2002
Withdrawn

Annex H

ASN.1 regular expressions

(This annex forms an integral part of this Recommendation | International Standard)

NOTE – This annex will be assigned a new letter, so that it precedes all informative annexes when this Recommendation | International Standard is re-published.

H.1 Definition

H.1.1 A regular expression is a pattern that describes a set of strings whose format conforms to this pattern. A regular expression is itself a string; it is constructed analogously to arithmetic expressions, by using various operators to combine smaller expressions. The smallest expressions, which are (usually) made of one or two characters, are placeholders that stand for a set of characters. The regular expressions presented here are very similar to those of scripting languages like Perl and to those of XML Schemas, where some other examples of use can be found.

H.1.2 Most characters, including all letters and digits, are regular expressions that match themselves.

EXAMPLE

The regular expression "fred" matches only the string "fred".

H.1.3 Two regular expressions may be concatenated; the resulting regular expression matches any string formed by concatenating two substrings that respectively match the concatenated subexpressions.

H.2 Metacharacters

H.2.1 A metacharacter sequence (or metacharacter) is a set of one or more contiguous characters that have a special meaning in the context of a regular expression. The following list contains all of the metacharacter sequences. Their meaning is explained in the following clauses.

[ ]		Match any character in the set where ranges are denoted by "-". A "^" after the opening bracket complements the set which follows it.
{g,p,r,c}		Quadruple which identifies a character of ISO/IEC 10646-1 (see 36.7)
\N{name}		Match the named character (or any character of the named character set) as defined in 37.1
.		Match any character (unless it is one of the newline characters defined in 11.1.6)
\d		Match any digit (equivalent to "[0-9]")
\w		Match any alphanumeric character (equivalent to "[a-zA-Z0-9]")
\t		Match the HORIZONTAL TABULATION (9) character (see 11.1.6)
\n		Match any one of the newline characters defined in 11.1.6
\r		Match the CARRIAGE RETURN (13) character (see 11.1.6)
\s		Match any one of the white-space characters defined in 11.1.6
\b		Match a word boundary
\	(prefix)	Quote the next metacharacter and cause it to be interpreted literally
\\		Match the BACKSLACH (92) character
""		Match the double quote character (")
|	(infix)	Alternative between two expressions
( )		Grouping of the enclosed expression
*	(postfix)	Match the previous expression zero, one or several times
+	(postfix)	Match the previous expression one or several times
?	(postfix)	Match the previous expression once or not at all
#n	(postfix)	Match the previous expression exactly n times (where n is a single digit)
#(n)	(postfix)	Match the previous expression exactly n times
#(n,)	(postfix)	Match the previous expression at least n times
#(n,m)	(postfix)	Match the previous expression at least n but not more than m times
#(,m)	(postfix)	Match the previous expression not more than m times

NOTE 1 – The characters CIRCUMFLEX ACCENT (94) "^" and HYPHEN-MINUS (45) "-" are additional metacharacters in certain positions of the string defined in H.2.2.

NOTE 2 – The value in round brackets after a character name in this annex is the decimal value of the character in ISO/IEC 10646.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 8824-1:1998/Cor 3:2002
Withdrawn