



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.691

Corrigendum 2
(02/2001)

SERIES X: DATA NETWORKS AND OPEN SYSTEM
COMMUNICATIONS

OSI networking and system aspects – Abstract Syntax
Notation One (ASN.1)

Corrigendum 2:

CAUTION !

PREPUBLISHED RECOMMENDATION

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU [had/had not] received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

All rights reserved. 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 ITU.

Summary

This technical corrigendum to ITU-T Rec. X.691 | ISO/IEC 8825-2:

- a) clarifies the PER visibility of pattern constraints.
- b) clarifies the encoding of the length determinant when the type is constrained and has an upper bound greater than or equal to 64K and the UNALIGNED variant of PER is in use;
- c) clarifies the behavior of compound subtype constraints that involve permitted alphabet constraints.

INFORMATION TECHNOLOGY – ASN.1 ENCODING RULES:
SPECIFICATION OF PACKED ENCODING RULES (PER)

DRAFT TECHNICAL CORRIGENDUM 2

1) *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) *Add a new subclause 9.3.6 bis as follows:*

9.3.6 bis Pattern constraints are not PER-visible.

3) *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) *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 $2^m < \text{"range"} \leq 2^{m+1}$, then the number of bits in the length determinant is $m + 1$.

5) *Change subclause 10.9.4.2 to:*

10.9.4.2 If the length determinant "n" to be encoded is a normally small length, or a constrained whole number with "ub" greater than or equal to 64K, or is a semi-constrained whole number, then "n" shall be encoded as specified in 10.9.3.4 to 10.9.3.8.4.

NOTE – Thus, if "ub" is greater than or equal to 64K, the encoding of the length determinant is the same as it would be if the length were unconstrained.

6) Change the last sentence of subclause 26.4 to:

Change the last sentence of subclause 26.4 before the Note to:

If the value is outside the range of the extension root, then the following encoding shall be as if there was no effective size constraint, and shall have an effective PermittedAlphabet constraint that consists of the set of characters of the unconstrained type.

7) Change the contents of Annex B to:

Certain properties can be observed when PER-visible subtype elements are combined. The following describes by means of examples such properties:

B.1 The effective size constraint for:

```
A ::= IA5String (SIZE(1..4) | SIZE(9..10))
```

is:

```
A ::= IA5String (SIZE(1..4) | 9..10)
```

B.2 When PermittedAlphabet constraints are combined with size constraints, there can be both an effective PermittedAlphabet constraint and an effective size constraint. For example:

```
B ::= IA5String ( FROM ("AB") ^ SIZE(1..2) |  
                FROM ("DE") ^ SIZE(3) |  
                FROM ("AXE") ^ (SIZE(1..5) )
```

has an effective size constraint and effective PermittedAlphabet constraint of:

```
B ::= IA5String (FROM ("ABDEX") ^ SIZE(1..5))
```

B.3 Another example is the following:

```
E ::= IA5String (SIZE(1..4) | SIZE(5..10) ^ FROM("ABCD") | SIZE(6..10))
```

There is an effective size constraint of SIZE(1..10) and an effective permitted alphabet constraint that consists of the entire IA5String alphabet.
