International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.692

Corrigendum 1

(05/2005)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

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

Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN)

Technical Corrigendum 1

ITU-T Recommendation X.692 (2002) – Technical Corrigendum 1

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.369
IP-based networks	X.370–X.379
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	**X.680–X.699**
OSI MANAGEMENT	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.889
Generic applications of ASN.1	X.890–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
TELECOMMUNICATION SECURITY	X.1000–

For further details, please refer to the list of ITU-T Recommendations.

INTERNATIONAL STANDARD ISO/IEC 8825-3
ITU-T RECOMMENDATION X.692

Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN)

Technical Corrigendum 1

Summary

This Technical Corrigendum solves identified defects in ITU-T Rec. X.692 | ISO/IEC 8825-3 which prevented its use in support of ITU-T Rec. X.891 | ISO/IEC 24824-1 "Fast Infoset". The changes introduced in this Corrigendum are summarized as follows:

- Removal of the restriction that prohibited recursive definition of encoding objects. This restriction was an undesirable (and probably unintended) limitation of the previous edition of the Recommendation | International Standard, because encoding objects are the natural means by which new "encoding rules" can be defined by using ECN, and the ASN.1 type definition notation is naturally recursive (as are all the standard encoding rules).
- Removal of ambiguity around the concept and the terminology of "exhibiting an identification handle". This is now clearly specified as a property of an encoding object (rather than as a property of the encodings produced by an encoding object).
- Clarification of the mechanism by which an encoding object (not based on a defined syntax) inherits an identification handle from another encoding object. The previous text was often imprecise or incomplete in this area.
- Removal of the restriction that prohibited multiple bit patterns (handle values) for the identification handle exhibited by an encoding object. This restriction prevented the use of a common and convenient technique whereby, given (for example) two alternatives, the first alternative always produces a well-known bit pattern and the second alternative never produces that same bit pattern.
- Removal of the restriction that prohibited an encoding object applied to the constructor in an "EncodeStructure" to specify structure replacement. This restriction was an undesirable (and probably unintended) limitation of the previous edition of the Recommendation | International Standard, because there is no other way of specifying structure replacement for an encoding structure.
- Resolution of an inconsistency between the declaration of an identification handle (**`EXHIBITS HANDLE &exhibited-handle`**) having an **`OPTIONAL`** handle name, and the use of an identification handle (**`DETERMINED BY handle HANDLE &handle-id`**) having a **`DEFAULT`** handle name (= "**`default-handle`**") with no relationship between these two things.
- Correction of typographical errors.

Source

Corrigendum 1 to ITU-T Recommendation X.692 (2002) was approved on 14 May 2005 by ITU-T Study Group 17 (2005-2008) under the ITU-T Recommendation A.8 procedure. An identical text is also published as Technical Corrigendum 1 to ISO/IEC 8825-3.

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.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

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 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.

© ITU 2006

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN)

Technical Corrigendum 1

NOTE – All new or changed text in this corrigendum is underlined and deleted text is struck through in clauses being replaced. When merging all such text into the base Recommendation, the underlining is to be removed and the struck through text is to be removed.

1) Subclauses 21.15, 21.15.1, 21.15.2, 22.9.1.1, 22.9.1.3, 22.9.1.4, 22.9.1.7, 22.9.1.8, 22.9.1.9, 23.1.1, 23.2.1, 23.3.1, 23.4.1, 23.5.1, 23.7.1, 23.8.1, 23.9.1, 23.11.1, 23.13.1 and 23.14.1, and the Table of Contents

*In the above subclauses, and in the Table of Contents, replace all the occurrences of "*HandleValue*" with "*HandleValueSet*".*

2) Subclauses 22.9.1.1, 22.9.1.2, 23.1.1, 23.2.1, 23.3.1, 23.4.1, 23.5.1, 23.7.1, 23.8.1, 23.9.1, 23.11.1, 23.13.1 and 23.14.1

In the above subclauses, replace all the occurrences of "&handle-value*" with "*&handle-value-set*".*

3) New subclause 3.2.26 bis

Insert a new subclause 3.2.26 bis as follows:

3.2.26 *bis* **handle value set**: The specified set of all possible values of the identification handle that is exhibited by an encoding object.

4) Subclause 3.2.27

Replace subclause 3.2.27 as follows:

3.2.27 **identification handle**: Part of an encoding which serves to distinguish the encodings ~~of~~produced by one encoding object (of a given class) from those ~~of~~produced by other encoding objects (of other classes).

NOTE – The ASN.1 Basic Encoding Rules use tags to provide identification handles in BER encodings.

5) Subclause 3.2.38

Replace the Note in subclause 3.2.38 as follows:

NOTE – Recursive definition of an encoding class (including an encoding structure) or an encoding object is permitted (but see 17.1.4). Recursive definition of an encoding object ~~or an encoding object~~ set is forbidden by ~~17.1.4 and~~ 18.1.3 ~~respectively~~.

6) Subclause 3.2.39

Replace the Note in subclause 3.2.39 as follows:

NOTE – Recursive instantiation of an encoding class (including an encoding structure) ~~is permitted~~or an encoding object is permitted (but see 17.1.4). Recursive instantiation of an encoding object ~~or an encoding object~~ set is forbidden by ~~17.1.4 and~~ 18.1.3 ~~respectively~~.

7) Subclause 9.10.3

Replace subclause 9.10.3 as follows:

9.10.3 A third key feature is that an encoding object may exhibit an **identification handle** in its encodings. This is a part (consisting of a fixed set of bit positions) of all the encodings that it produces and distinguishes ~~its~~those encodings from the encodings ~~of~~produced by other encoding objects (of any class) that exhibit the same identification handle. Identification handles have ~~to be~~a name and are visible to decoders without knowledge of either the encoding class or the abstract value that was encoded (but with knowledge of the name of the identification handle that is being used). This concept models (and generalizes) the use of tags in BER encodings: the tag value in BER can be determined without knowledge of the encoding class, for all BER encodings, and serves to identify the encoding for resolution of optionality, ordering of sets, termination of repetitions, and choice alternatives.

8) Subclause 9.13.3

Replace subclause 9.13.3 as follows:

9.13.3 In addition to terminating repetitions, the identification handle technique can also be used to determine the presence of optional components or of alternatives and the ordering of sets. The mechanism is similar in all these cases. ~~Encodings for all values of any given "possible next class" encoding will have the same bit-pattern (their identification) at some place in their encoding (the handle), but the identification for different "possible next class" encodings will be different for each one.~~ Given an encoding class that is a "possible next class" and an encoding object applied to it, any encoding produced will contain, at some bit positions (the identification handle), a bit pattern that matches a bit pattern within a specified set of bit patterns (the handle value set) characterizing that class, but does not match any bit pattern characterizing any other "possible next class". All such encodings can be interpreted by a decoder as an encoding of ~~any~~a "possible next class", and the ~~identification for the handle~~bit pattern found in the encoding will determine which "possible next class" encoding is present. The concept is similar to that of using tags for such purposes in BER. Identification handles have names that are required to be unique within an ECN specification.

9) Subclause 13.2.12

Replace subclause 13.2.12 as follows:

13.2.12 In the encoding process, encoding objects applied to encoding constructors (and to classes in the optionality category) may require that the encoding objects applied to ~~their~~the components of the constructions defined by those constructors exhibit identification handles (of a given name) to resolve alternatives, or optionality, or termination of a repetition, or order in a set-like concatenation. ~~If in this case the encodings of the components do not exhibit the required identification handles,~~They may also require that the encoding objects applied to other encoding classes (following those constructions) exhibit the same identification handle, and that the handle value sets of all the involved encoding objects (exhibiting the same handle) be all disjoint. If these conditions are not satisfied, then the ECN specification is in error.

NOTE – This problem is most likely to arise if BER encoding objects are applied to encoding constructors and not to their components, as BER is heavily reliant on identification handles. PER encoding objects make no use of identification handles.

10) Subclause 17.1.4

Replace subclause 17.1.4 with the following:

17.1.4 There shall be no recursive definition (see 3.2.38) of an "encodingobjectreference", and there shall be no recursive instantiation (see 3.2.39) of an "encodingobjectreference" if these recursions lead to an infinite recursion in the definition of the encoding.

11) Subclause 17.1.6

Delete the last sentence of subclause 17.1.6. The subclause will become:

17.1.6 "DefinedEncodingObject" identifies an encoding object and is specified in 10.9.2. The "DefinedEncodingObject" shall be of the same encoding class as the governor, or of a class which can be obtained from the governor by de-referencing.

12) Subclause 17.2.6

Replace subclause 17.2.6 with the following:

17.2.6 The "DefinedSyntax" notation specifies whether the encoding object~~reference~~ being defined exhibits an identification handle.

13) Subclause 17.3.6

Replace subclause 17.3.6 with the following:

17.3.6 ~~In the application of encodings specified in clause 13, there is an encoding object (A say) which produces the first bit field in the resulting encoding. The "encodingobjectreference" being defined~~Call E is the encoding object (within the "CombinedEncodings") which is applied to the governor class. If the encoding object E exhibits an identification handle ~~if and only if the encoding object A exhibits that identification~~(with a given handle value set), then the encoding object being defined (see 17.1.5) exhibits the same identification handle as E (with the same handle value set); otherwise, it does not exhibit a handle.

14) Subclause 17.4.8

Replace subclause 17.4.8 with the following:

17.4.8 ~~If the "EncodingObject" alternative of "ValueMappingEncodingObjects" is used, then the "encodingobjectreference"~~ Call E is the encoding object which is applied to the "DefinedOrBuiltinEncodingClass". If the encoding object E exhibits an identification handle (with a given handle value set), then the encoding object being defined (see 17.1.5) exhibits ~~an identification handle if and only if the "EncodingObject" exhibits that identification handle. If~~the same identification handle as E (with the same handle value set); otherwise, it does not exhibit a handle.

NOTE – The encoding object E may be either the "EncodingObject" in the "ValueMappingEncodingObjects", or a member of the "DefinedOrBuiltinEncodingObjectSet"~~ alternative of "ValueMappingEncodingObjects" is used to define the encoding of the "DefinedOrBuiltinEncodingClass", then determination of whether the "encodingobjectreference" exhibits an identification handle is in accordance with 17.3.6.~~.

15) Subclause 17.5.4

Replace subclause 17.5.4 with the following:

17.5.4 ~~The~~If the "ComponentEncodingList" is not empty, then the encoding object applied to the governing encoding constructor (whether from ~~STRUCTURED WITH~~"StructureEncoding" or from "CombinedEncodings") shall not specify any replacement actions.

16) Subclause 17.5.16

Replace subclause 17.5.16 with the following:

17.5.16 Determination of whether the ~~"encodingobjectreference"~~encoding object being defined (see 17.1.5) exhibits an identification handle ~~is in accordance with 17.3.6.~~shall be done as follows:

a) if the "TagEncoding" is present in "StructureEncoding", call E the encoding object which is applied to the encoding class in the tag category; or

b) if the "TagEncoding" is not present in "StructureEncoding", call E the encoding object which is applied to the governing encoding constructor (this may be either the "EncodingObject" in the "EncodingOrUseSet" in the "StructureEncoding", or may be a member of the "CombinedEncodings").

If the encoding object E exhibits an identification handle (with a given handle value set), then the encoding object being defined exhibits the same identification handle as E (with the same handle value set); otherwise, it does not exhibit a handle.

17) Subclause 17.6.5

Replace subclause 17.6.5 with the following:

17.6.5 ~~The "encodingobjectreference" being defined exhibits an identification handle if and only if the same identification handle is being exhibited by~~If the "SpecForEncoding" and ~~by~~ the "SpecForDecoders" exhibit the same identification handle with the same handle value set, then the encoding object being defined (see 17.1.5) exhibits that identification handle (with the same handle value set); otherwise, it does not exhibit a handle.

18) Subclause 17.7.3

Replace subclause 17.7.3 with the following:

17.7.3 The "AlternativesEncodingObject" shall be an encoding object of any class in the alternatives category, and encoders and decoders shall use the encodings and procedures specified by that encoding object as if the encoding options were encodings for ~~components~~alternatives of an instance of that class. The "AlternativesEncodingObject" shall not contain a **REPLACE** specification (see 23.1.1). The **DETERMINED BY** parameter shall be set to **handle**, and an identification handle shall be specified.

NOTE – If the "AlternativesEncodingObject" is parameterized with a reference field parameter, then the "encodingobjectreference" being defined has to be parametized with a dummy reference field parameter that is used as the actual parameter for the AlternativesEncodingObject".

19) Subclause 17.7.4

Replace subclause 17.7.4 with the following:

17.7.4 All "EncodingObject"s in the "EncodingOptionsList" shall exhibit that identification handle, and their handle value sets shall all be disjoint.

20) New subclause 17.7.4 *bis*

Insert a new subclause 17.7.4 bis as follows:

17.7.4 *bis* If the "AlternativesEncodingObject" exhibits an identification handle (with a given handle value set), then the encoding object being defined (see 17.1.5) exhibits the same identification handle (with the same handle value set); otherwise, it does not exhibit a handle.

NOTE – The identification handle exhibited by the "AlternativesEncodingObject" (if any) is unrelated to the identification handle exhibited by the "EncodingObject"s in the "EncodingOptionsList", even if they have the same name.

21) Subclause 17.8.6

Replace subclause 17.8.6 with the following:

17.8.6 An identification handle (with a given handle value set) is exhibited by the ~~"encodingobjectreference"~~encoding object being defined (see 17.1.5) if and only if the "anystringexceptnonecnend" specifies that it does so. The means of such specification is not defined in this Recommendation | International Standard.

22) Subclause 18.1.3

Replace subclause 18.1.3 with the following:

18.1.3 There shall be no recursive definition (see 3.2.38) of an "~~encodingclassreference~~encodingobjectsetreference", and there shall be no recursive instantiation (see 3.2.39) of an "~~encodingclassreference~~encodingobjectsetreference".

23) Subclause 21.5.7

Replace subclause 21.5.7 with the following (splitting the text into a subclause 21.5.7 and a new subclause 21.5.7 bis):

21.5.7 The value "**handle**" requires that an identification handle be specified. This identification handle shall be exhibited both by the encoding object for the optional component and by ~~any~~the encoding object applied to each possible alternative encoding class that can follow if this optional component is absent~~, and the value of the handle shall be different for the encoding of the optional component and all possible alternative encodings that can follow.~~. Each possible alternative encoding class may be a component of the concatenation containing the optional component, or may be an encoding class following the concatenation. The handle value sets specified by all the involved encoding objects (exhibiting the same identification handle) shall all be disjoint.

NOTE – Every abstract value of a given component is required to have a handle value matching the specified handle value set (see 22.9.2.2).

21.5.7 *bis* If the end of any open container (or the end of the PDU) is detected at the time a decoder is attempting to detect the presence or absence of ~~this~~an optional component, then ~~it~~the decoder shall determine that the optional component is absent. Otherwise, ~~a~~the decoder shall determine that the component is present if and only if decoding the remaining parts of the encoding produces a value for the specified identification handle which matches ~~that~~the handle value set of the optional component. It is an ECN specification error if this does not result in correct identification of the presence or absence of an encoding of the optional component, but conforming encoders shall not generate such encodings.

24) Subclause 21.6.6

Replace subclause 21.6.6 with the following (splitting the text into a subclause 21.6.6 and a new subclause 21.6.7):

21.6.6 The value "**handle**" requires that an identification handle be specified. This identification handle shall be exhibited by ~~(the encodings of) all of the alternatives in the class, and the encoding of each alternative shall have a different value for the identification handle~~the encoding objects applied to each of the alternatives in the construction defined by the class in the alternatives category. The handle value sets specified by those encoding objects shall all be disjoint. (Violation of this rule is an ECN specification error, ~~but~~and conforming encoders are required not to generate encodings where this rule is violated.) ~~This value specifies that a~~

21.6.7 A decoder shall determine the alternative that is present by decoding the remaining parts of the encoding to produce a value for the specified identification handle. The alternative whose ~~identification~~ handle value set matches this value is the alternative that is present. If the end of any open container (or the end of the PDU) is reached before the identification handle can be decoded, or if the value of the identification handle does not match ~~that~~the handle value set of any alternative, then this is an encoding error.

NOTE – Every abstract value of a given alternative is required to have a handle value matching the handle value set of the alternative (see 22.9.2.2).

25) Subclause 21.7.10

Replace subclause 21.7.10 with the following:

21.7.10 The value "**handle**" requires that an identification handle be specified. This identification handle shall be exhibited both by the ~~element~~encoding object applied to the component being repeated, and by ~~all~~the encoding object applied to each possible (taking account of optionality) following ~~elements~~encoding class. The ~~value of the identification handle for the element being repeated shall be different from that of all possible following elements~~handle value sets specified by those encoding objects shall all be disjoint.

NOTE – Every abstract value of a given component is required to have a handle value matching the handle value set of the component (see 22.9.2.2).

26) Subclause 21.15.1

Replace subclause 21.15.1 with the following:

21.15.1 The "**HandleValueSet**" type is:

```
HandleValueSet ::= CHOICE {
      bits        BIT STRING,
      octets      OCTET STRING,
      number      INTEGER (0..MAX),
      tag         ENUMERATED {any},
```

```
range       SEQUENCE {
                low   INTEGER(0..MAX),
                high  INTEGER(0..MAX)  },
ranges      SET (SIZE(1..MAX)) OF SEQUENCE {
                low   INTEGER(0..MAX),
                high  INTEGER(0..MAX)  } }
```

27) Subclause 21.15.2

Replace subclause 21.15.2 with the following:

21.15.2 The "**HandleValueSet**" is used to specify the ~~value of~~set of bit patterns (the handle value set) characterizing the encodings produced by an encoding object that exhibits an identification handle~~ that is exhibited by particular encoding objects~~.

28) Subclause 21.15.3

Replace subclause 21.15.3 and its Note with the following:

21.15.3 ~~Values of any identification handle that is exhibited by an encoding object are required to be the same for all abstract values which that encoding object encodes (see 22.9.2.2).~~ The value of an identification handle can be used to identify the presence or absence of optional components, the choice of alternatives, the ordering of sets, or the end of a repetition. There are requirements in such circumstances that the handle ~~values exhibited by~~value sets of the encoding ~~of~~objects applied to the different alternatives or components be ~~distinct (see 21.5.7, 21.6.6 and 21.7.10~~all disjoint (see 21.5.7, 21.6.6, 21.7.10, and 22.10.2.1), and requirements that all the possible values of the identification handle occurring in the encodings of any given alternative or component all match the specified handle value set of the encoding object applied to that alternative or component (see 22.9.2.2).

NOTE – ~~Values of identification handles exhibited by a given encoding object can, in theory, be determined by encoding a trial value. However, to ease the implementation task, the~~The ECN specifier is required to specify the handle value ~~of the handle~~set in all cases except where (for encodings of the tag class) the ~~value of the identification~~ handle value set consists of a single value and depends on the tag number associated with that tag class, either directly through implicit generation from an ASN.1 tag, or by mapping from an implicitly generated structure.

29) Subclause 21.15.4

Replace subclause 21.15.4 with the following:

21.15.4 The "**bits**", "**octets**" and "**number**" alternatives specify ~~the~~a handle value as a bitstring, octetstring or integer value respectively. It is an ECN specification error if this value cannot be encoded within the number of bits specified for the identification handle (see 22.9).

30) Subclause 21.15.5

Replace subclause 21.15.5 with the following:

21.15.5 The "**tag:any**" alternative specifies ~~that the~~a handle value ~~is~~ determined by the number specified in an ECN encoding structure for a class in the tag category, or by the tag number mapped from an ASN.1 tag construction. It shall only be used when specifying the handle identification for the encoding of a class in the tag category.

31) New subclause 21.15.6

Insert a new subclause 21.15.6 as follows:

21.15.6 The "**range**" alternative specifies a range of integer values, with **high** greater than or equal to **low**.

32) New subclause 21.15.7

Insert a new subclause 21.15.7 as follows:

21.15.7 The "**ranges**" alternative specifies a set of ranges of integer values, each with **high** greater than or equal to **low**. One or more such ranges can be specified, and they shall not overlap.

33) New subclause 22.1.1.11

Insert a new subclause 22.1.1.11 as follows:

22.1.1.11 In a full replacement specification, if the encoding object applied to the replacement structure exhibits an identification handle (with a given handle value set), then the encoding object whose defined syntax contains the full replacement specification exhibits the same identification handle (with the same handle value set), otherwise it does not exhibit a handle.

34) Subclause 22.5.2.5

Replace subclause 22.5.2.5 with the following and delete the Note in this subclause.

22.5.2.5 If "**DETERMINED BY**" is "**handle**", then 21.5.7 applies.

~~**22.5.2.5** If "**HANDLE**" is specified, then the component whose presence is being determined, together with all following optional and the next mandatory encoding (if any) shall all be produced by encoding objects whose specifications all exhibit an identification handle with the same name as "**HANDLE**". The next mandatory encoding may be a component of the concatenation containing the optional component, or may be an encoding following the concatenation. The value of the identification handle shall be different for all these components.~~

~~NOTE – It is a requirement that the bits that form an identification handle shall have the same value for all abstract values encoded by an encoding object exhibiting that identification handle (see 22.9.2.2).~~

35) Subclause 22.5.4.4

Replace subclause 22.5.4.4 with the following:

22.5.4.4 If "**DETERMINED BY**" is "**handle**", then the decoder shall determine the value of the specified identification handle. If the value matches ~~match the value of the identification~~the handle value set of the optional component, then the decoder shall set the conceptual value "**element-is-present**" to **TRUE**, otherwise the decoder shall set it to **FALSE**.

36) Subclause 22.6.2.4

Replace subclause 22.6.2.4 with the following and delete the Note in this subclause:

22.6.2.4 If "**DETERMINED BY**" is "**handle**", then 21.6.6 applies.

~~**22.6.2.4** If "**HANDLE**" is specified, then all the alternatives of the encoding class in the alternatives category shall be encoded by encoding objects whose specification exhibits and defines an identification handle with the same name as "**HANDLE**", and with the same value of the identification handle. The value of the identification handle shall be different for all these alternatives.~~

~~NOTE – It is a requirement that an identification handle shall have the same value for all abstract values encoded by an encoding object exhibiting that identification handle (see 22.9.2.2).~~

37) Subclause 22.6.4.4

Replace subclause 22.6.4.4 with the following:

22.6.4.4 If "**DETERMINED BY**" is "**handle**", then the decoder shall determine the value of the identification handle. This value shall be compared to the ~~value of the identification~~ handle value set of each of the alternatives. If none match, then the decoder shall diagnose an encoder's error. Otherwise the conceptual value "alternative-index" shall be set to the matching alternative.

38) Subclause 22.7.2.10

Replace subclause 22.7.2.10 with the following and delete the NOTE in this subclause:

22.7.2.10 If "**DETERMINED BY**" is "**handle**", then 21.7.10 applies.

~~**22.7.2.10** If "**HANDLE**" is specified, then the repeated element, together with any element which (through the use of optionality) may follow the repeated element shall all be encoded by encoding objects whose specification exhibits an identification handle with the same name as "**HANDLE**". The value of the identification handle in the repeating element shall be different from that of any possible following element.~~

~~NOTE – It is a requirement that an identification handle shall have the same value for all abstract values encoded by an encoding object exhibiting that identification handle (see 22.9.2.2).~~

39) Subclause 22.7.4.8

Replace subclause 22.7.4.8 with the following:

22.7.4.8 If "**DETERMINED BY**" is "**handle**", then the decoder shall determine the value of the identification handle and attempt to decode the following ~~element~~encoding (in parallel) as either a further occurrence of the repetition or as a following ~~element~~encoding class, using the value of the identification handle to distinguish these alternatives. If decoding succeeds for more than one of these or for none of these, it is an encoding or a specification error.

40) Subclause 22.9.1.1

Replace subclause 22.9.1.1 with the following:

22.9.1.1 Identification handle specification uses the following encoding properties:

```
&exhibited-handle       PrintableString ~~OPTIONAL~~DEFAULT "default-handle",
&Handle-positions       INTEGER (0..MAX) OPTIONAL,
&handle-value-set       HandleValueSet DEFAULT tag:any
```

41) Subclause 22.9.1.4

Replace subclause 22.9.1.4 with the following:

22.9.1.4 ~~This~~ The purpose of this specification is ~~used~~ to ~~identify~~declare that an encoding object exhibits an identification handle ~~within all its encodings (that is, for all possible abstract values that it encodes). The name of the identification handle is specified, and the bits that are associated with that identification handle. The value of the identification handle is specified by "**HandleValue**".~~and to specify its properties, which are:

a) the name of the handle;

b) the bit positions that form the handle; and

c) the possible bit patterns (for the bit positions forming the handle) occurring in the encodings produced by this encoding object (the handle value set).

42) Subclause 22.9.2.1

Replace subclause 22.9.2.1 and its Note with the following:

22.9.2.1 In any ~~application of~~ ECN specification, all identification handles with the same name shall specify the same set of ~~bits for the location of the identification handle~~bit positions.

NOTE – There is no general requirement that the handle value sets of ~~the identification handle (exhibited by~~ different encoding objects~~) should be distinct, but distinct values~~ defined in an ECN specification be all disjoint, but disjoint handle value sets are required when the identification handle is used to resolve optionality, alternative selection, ~~or~~ repetition termination, or ordering of sets (see 21.5.7, 21.6.6, ~~and~~ 21.7.10 and 22.10.2.1).

43) Subclause 22.9.2.2

Replace subclause 22.9.2.2 with the following:

22.9.2.2 For an encoding object that exhibits an identification handle (with a given handle value set), the value of the identification handle occurring in each of the possible encodings produced by that encoding object (for all possible abstract values) shall be a member of the specified handle value set.

~~**22.9.2.2** The ECN specifier shall ensure that any encoding object exhibiting an identification handle produces the same value of the identification handle for every abstract value that is encoded.~~

44) Subclauses 22.9.2.4, 22.9.2.5 and 22.9.2.6

Delete subclauses 22.9.2.4, 22.9.2.5, and 22.9.2.6.

~~**22.9.2.4** If an encoding object for a class in the repetition category exhibits an identification handle, then that identification handle shall also be exhibited (with the same value) by the encoding of the repeated element.~~

~~**22.9.2.5** If an encoding object for a class in the alternatives category exhibits an identification handle, then that identification handle shall also be exhibited by (the encoding of) all alternatives, and the value of the identification handle shall be the same for all the alternatives.~~

~~NOTE – In this case that identification handle cannot be used for alternative determination in this alternative, and alternative determination has either to be done using a different identification handle or by some other means.~~

~~**22.9.2.6** If an encoding object for a class in the concatenation category exhibits an identification handle, then the first (if any) encoded component (or, if it is tagged, the tag), taking account of optionality, shall exhibit that identification handle with the same value.~~

45) Subclause 22.9.3.1

Replace subclause 22.9.3.1 with the following:

22.9.3.1 If an encoding object exhibits an identification handle, the encoder shall check that the ~~encoding has the~~ value of the identification handle occurring in the encoding produced is a member of the specified handle value set, and shall diagnose a specification or application error otherwise.

46) Subclause 22.10.2.1

Replace subclause 22.10.2.1 with the following:

22.10.2.1 If "**ORDER**" is ~~"random"~~"**random**", then "**HANDLE**" assumes the default value of "**default-handle**" if not set, and the encoding objects applied to all components shall exhibit ~~"**HANDLE**" with distinct values for the~~that identification handle. The handle value sets of those encoding objects shall all be disjoint.

47) Subclause 22.10.4.3

Replace subclause 22.10.4.3 with the following:

22.10.4.3 If "**ORDER**" is ~~"random"~~"**random**", the decoder shall determine the order of the components by examining the value of the ~~bits associated with "HANDLE"~~ identification handle.

48) Subclause 22.10.4.4

Delete the first sentence of subclause 22.10.4.4. The subclause will become:

22.10.4.4 ~~Each component has a distinct value for the bits associated with "**HANDLE**" that enables the component to be identified.~~ Decoding shall proceed until an abstract value for every component has been obtained, and a decoder shall diagnose an encoder's error if more than one encoding is identified for a component, or if unexpected values appear for identification handles during the decoding.

NOTE – Unexpected values can occur as part of extensibility provision, but this is not supported in this version of this Recommendaqtion | International Standard, and such occurrences shall be treated as encoder errors.

49) Subclause 23.1.2.1

Replace subclause 23.1.2.1 with the following:

23.1.2.1 This syntax is used to define the start of the encoding space for an encoding class in the alternatives category, the determination of the alternative that has been encoded, and an optional declaration that ~~all encodings exhibit~~the encoding object exhibits a specified identification handle (with ~~distinct identification~~a given handle value~~s~~ set).

50) Subclause 23.1.2.2

Replace subclause 23.1.2.2 with the following:

23.1.2.2 If "**REPLACE STRUCTURE**" is set, then no other encoding property groups shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

51) Subclause 23.1.2.3

Replace subclause 23.1.2.3 with the following:

23.1.2.3 ~~Encodings~~An encoding object of this class ~~do~~does not exhibit an identification handle unless "**EXHIBITS HANDLE**" is set (even if ~~all~~the components of the defined construction exhibit an identification handle~~, that may or may not be the same~~) or unless "**REPLACE STRUCTURE**" is set and the encoding object of the replacement structure exhibits an identification handle (see 22.1.1.11).

52) Subclause 23.1.2.4

Replace subclause 23.1.2.4 with the following and delete the NOTE in this subclause:

23.1.2.4 If "**EXHIBITS HANDLE**" is set, then ~~encodings of all the alternatives of this class are required to exhibit the defined~~the encoding object exhibits the specified identification handle~~, and to have distinct values for that identification handle.~~.

~~NOTE – This would normally require that every component had an "**EXHIBITS HANDLE**" set to the same value, unless a head-end insertion exhibited the identification handle (see 9.10.3).~~

53) Subclause 23.2.3.1

Replace subclause 23.2.3.1 with the following:

23.2.3.1 This syntax is used to define the start of the encoding space for a class in the bitstring category, the encoding of the abstract values of that class, an optional declaration that ~~all bits encodings exhibit~~the encoding object exhibits a specified identification handle (with a given handle value set), and a specification of how to encode a contained type.

54) Subclause 23.2.3.4

Replace subclause 23.2.3.4 with the following:

23.2.3.4 If there is a "**REPLACE STRUCTURE**" clause in the **#CONDITIONAL-REPETITION** encoding objects, then no other parameters shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

55) Subclause 23.2.3.10

Replace subclause 23.2.3.10 with the following:

23.2.3.10 If "**EXHIBITS HANDLE**" is set, then ~~all encodings of values associated with this class shall exhibit~~the encoding object exhibits the specified identification handle.

NOTE – This will in general require restrictions on the abstract values of the associated type or the addition of redundant bits in the transform into bits, or both.

56) Subclause 23.3.2.1

Replace subclause 23.3.2.1 with the following:

23.3.2.1 This syntax is used to define the start of the encoding space for a class in the boolean category, the encoding of the abstract values of that class, their positioning within the encoding space, an optional declaration that ~~all bits encodings exhibit~~the encoding object exhibits a specified identification handle (with a given handle value set), and possible bit-reversal of the encoding space for the boolean.

57) Subclause 23.4.3.1

Replace subclause 23.4.3.1 with the following:

23.4.3.1 This syntax is used to define the start of the encoding space for a class in the characterstring category, the encoding of the abstract values associated with that class, an optional declaration that ~~all chars encodings exhibit~~ the encoding object exhibits a specified identification handle (with a given handle value set).

58) Subclause 23.4.3.4

Replace subclause 23.4.3.4 with the following:

23.4.3.4 If there is no "**REPLACE STRUCTURE**" clause in the **#CONDITIONAL-REPETITION** encoding objects, then "**TRANSFORMS**" shall be set. If there is a "**REPLACE STRUCTURE**" clause in the **#CONDITIONAL-REPETITION** encoding objects, then no other parameters shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

59) Subclause 23.4.3.9

Replace subclause 23.4.3.9 with the following:

23.4.3.9 If "**EXHIBITS HANDLE**" is set, then ~~all encodings of values associated with this class shall exhibit~~the encoding object exhibits the specified identification handle.

NOTE – This will in general require restrictions on the abstract values of the associated type, or the inclusion of redundant bits in the encoding of each character, or both.

60) Subclause 23.5.2.1

Replace subclause 23.5.2.1 with the following:

23.5.2.1 This syntax is used to define the start of the encoding space for a class in the concatenation category, the way in which the encodings of the components are to be combined, their positioning within the encoding space, an optional declaration that ~~all encodings exhibit~~the encoding object exhibits a specified identification handle (with a given handle value set), and possible bit-reversal of the encoding space.

61) Subclause 23.5.2.2

Replace subclause 23.5.2.2 with the following:

23.5.2.2 If "**REPLACE STRUCTURE**" is set, then no other encoding parameter groups shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

62) Subclause 23.5.2.4

Replace subclause 23.5.2.4 with the following and delete the Note in this subclause:

23.5.2.4 If "**EXHIBITS HANDLE**" is set, then the encoding ~~of all possible abstract values associated with this class shall exhibit the defined~~object exhibits the specified identification handle.

~~NOTE – This would often be achieved by ensuring that the first component of the concatenation, or a head-end insert, exhibited the identification handle.~~

63) Subclause 23.7.2.10

Replace subclause 23.7.2.10 with the following:

23.7.2.10 If "**EXHIBITS HANDLE**" is set, then the ~~specifier asserts that the~~ encoding ~~of all values~~object exhibits the specified identification handle.

NOTE – This will normally require use of "**VALUE-PADDING**" with justification from the left to allow the padding to exhibit the identification handle.

64) Subclause 23.8.2.2

Replace subclause 23.8.2.2 with the following:

23.8.2.2 If "**REPLACE STRUCTURE**" is set, then no other encoding property groups shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

65) Subclause 23.9.3.1

Replace subclause 23.9.3.1 with the following:

23.9.3.1 This syntax is used to define the start of the encoding space for a class in the octetstring category, the encoding of the abstract values associated with that class, an optional declaration that ~~all octetstring encodings exhibit~~the encoding object exhibits a specified identification handle (with a given handle value set), a specification of how to encode a contained type.

66) Subclause 23.9.3.4

Replace subclause 23.9.3.4 with the following:

23.9.3.4 If there is a "**REPLACE STRUCTURE**" clause in the **#CONDITIONAL-REPETITION** encoding objects, then no other parameters shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

67) Subclause 23.9.3.9

Replace subclause 23.9.3.9 with the following:

23.9.3.9 If "**EXHIBITS HANDLE**" is set, then ~~all encodings of values of this class shall exhibit~~the encoding object exhibits the specified identification handle.

NOTE – This will in general require restrictions on the abstract values of the associated type.

68) Subclause 23.10.2.2

Replace subclause 23.10.2.2 with the following:

23.10.2.2 If "**REPLACE STRUCTURE**" is set, then no other encoding property groups shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

69) Subclause 23.11.2.3

Replace subclause 23.11.2.3 with the following:

23.11.2.3 If "**REPLACE STRUCTURE**" is set, then no other encoding property group shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

70) Subclause 23.13.2.3

Replace subclause 23.13.2.3 with the following:

23.13.2.3 If "**REPLACE STRUCTURE**" is set, then no other encoding property groups shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

71) Subclause 23.13.2.4

Replace subclause 23.13.2.4 with the following:

23.13.2.4 If "**EXHIBITS HANDLE**" is set, ~~this asserts that all encodings of this class exhibit~~then the encoding object exhibits the specified identification handle ~~(see also 22.9.2.4)~~.

72) Subclause 23.14.2.2

Replace subclause 23.14.2.2 with the following:

23.14.2.2 If "**REPLACE STRUCTURE**" is set, then no other specifications shall be set. If the encoding object of the replacement structure exhibits a handle (with a given handle value set), the encoding object being defined exhibits the same identification handle (with the same handle value set – see 22.1.1.11).

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2006