



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.680

Amendement 1
(10/2003)

SÉRIE X: RÉSEAUX DE DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Réseautage OSI et aspects systèmes – Notation de
syntaxe abstraite numéro un (ASN.1)

Technologies de l'information – Notation de syntaxe
abstraite numéro un: spécification de la notation de
base

**Amendement 1: Prise en charge des règles de
codage XML étendues (EXTENDED-XER)**

Recommandation UIT-T X.680 (2002) – Amendement 1

RECOMMANDATIONS UIT-T DE LA SÉRIE X
RÉSEAUX DE DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

RÉSEAUX PUBLICS DE DONNÉES	
Services et fonctionnalités	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalisation et commutation	X.50–X.89
Aspects réseau	X.90–X.149
Maintenance	X.150–X.179
Dispositions administratives	X.180–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200–X.209
Définitions des services	X.210–X.219
Spécifications des protocoles en mode connexion	X.220–X.229
Spécifications des protocoles en mode sans connexion	X.230–X.239
Formulaires PICS	X.240–X.259
Identification des protocoles	X.260–X.269
Protocoles de sécurité	X.270–X.279
Objets gérés des couches	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion et fonctions ODMA	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base

Amendement 1

Prise en charge des règles de codage XML étendues (EXTENDED-XER)

Résumé

Le présent Amendement 1 s'applique aux Rec. UIT-T X.680 | ISO/CEI 8824-1, UIT-T X.681 | ISO/CEI 8824-2, UIT-T X.690 | ISO/CEI 8825-1, UIT-T X.691 | ISO/CEI 8825-2 et UIT-T X.693 | ISO/CEI 8825-4. Il permet de:

- Rectifier une erreur dans les règles de codage en langage de balisage extensible canonique (CXER, *canonical extensible markup language (XML) encoding rules*), résultant de la présence d'un espace entre le signe moins et la valeur INTEGER ou REAL qui le suit (la règle CXER n'est pas canonique). Ceci n'est plus autorisé dans la notation des valeurs, ni dans la notation des valeurs en langage XML, ni dans les règles XER et CXER. **Il s'agit d'une modification et non d'une adjonction.**
- Ajouter des instructions de codage dans un module en notation ASN.1, soit au moyen d'un préfixe de type, soit dans une section de commande de codage, afin de spécifier les variantes de codage selon les règles BASIC-XER. Ces instructions de codage visent à prendre en charge des projections d'une spécification conforme à une description schématique en langage XML (XSD, *XML schema definition*) sur une spécification en notation ASN.1. Cette disposition correspond à une modification de terminologie dans laquelle un type commençant par "[...]" est un type préfixé, et la notation "[...]" peut être une étiquette ou non. Cette modification de terminologie conduit à des modifications du texte (mais non du contenu) des spécifications selon les règles de codage de base (BER, *basic encoding rules*) et les règles de codage compact (PER, *packed encoding rules*).
- Ajouter les nouvelles valeurs NaN (*not-a-number*, pas un nombre) et moins zéro pour la valeur REAL (la prise en charge du codage de ces nouvelles valeurs est assurée au moyen de l'Amendement 1 aux Rec. UIT-T X.690 | ISO/CEI 8825-1 et UIT-T X.691 | ISO/CEI 8825-2, ainsi que de l'Amendement 1 à la Rec. UIT-T X.693 | ISO/CEI 8825-4).
- Ajouter les nouvelles notations des valeurs en langage XML pour les valeurs **REAL**, **BOOLEAN**, **ENUMERATED** et **INTEGER** qui emploient du texte plutôt que des étiquettes contenant l'unité lexicale vide pour ces valeurs. Elles s'expriment en langage XML et selon les règles étendues de codage en langage XML (EXTENDED-XER, *extended XML encoding rule*), mais non selon les règles BASIC-XER (pour des raisons de compatibilité avec ce qui précède).
- Modifier la notation des valeurs en langage XML des types sequence-of (et des codages selon les règles XER) afin de déterminer les valeurs pour lesquelles ceux-ci ne sont plus des unités lexicales en langage XML (cela se produit pour les notations supplémentaires des valeurs en langage XML et n'affecte que l'emploi de celles-ci). Cette modification ne concerne que l'emploi des notations des valeurs en langage XML qui ont été ajoutées conformément au présent Amendement, et n'est pas autorisée dans les codages selon les règles BASIC-XER, qui ne sont pas touchés.

Ainsi est assurée la prise en charge de base nécessaire des règles EXTENDED-XER.

Source

L'Amendement 1 de la Recommandation X.680 (2002) de l'UIT-T a été approuvé le 29 octobre 2003 par la Commission d'études 17 (2001-2004) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8. Un texte identique est publié comme Norme internationale ISO/CEI 8824-1, Amendement 1.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 2004

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

TABLE DES MATIERES

	<i>Page</i>
1) Introduction	1
2) Paragraphe 2.2	1
3) Nouveau paragraphe 3.6.18 <i>bis</i>	1
4) Nouveaux paragraphes 3.6.22 <i>bis</i> à 3.6.22 <i>quat</i>	1
5) Paragraphe 3.6.54	2
6) Paragraphe 3.6.69	2
7) Paragraphe 3.6.71	2
8) Nouveau paragraphe 3.6.74 <i>bis</i>	2
9) Paragraphe 8.1	2
10) Paragraphe 8.3	3
11) Nouveau paragraphe 8 <i>bis</i>	3
12) Nouveau paragraphe 11.20 <i>bis</i>	3
13) Paragraphe 11.23.2	3
14) Nouveau paragraphe 11.23 <i>bis</i>	4
15) Paragraphe 11.24.2	4
16) Nouveaux paragraphes 11.24 <i>bis</i> à 11.24 <i>quat</i>	4
17) Paragraphe 11.25.5	5
18) Tableau 4	5
19) Paragraphe 11.27	5
20) Paragraphe 12.1	5
21) Paragraphe 12.2	5
22) Nouveau paragraphe 12.4 <i>bis</i>	6
23) Nouveau paragraphe 12.21	6
24) Paragraphe 13.3	6
25) Paragraphe 16.2	6
26) Paragraphe 16.9	6
27) Paragraphe 16.10	6
28) Paragraphe 17.3	6
29) Nouveau paragraphe 17.4	7
30) Paragraphe 18.9	7
31) Nouveau paragraphe 18.9 <i>bis</i>	7
32) Paragraphe 18.10	7
33) Nouveau paragraphe 18.12	7
34) Paragraphe 19.8	8
35) Nouveau paragraphe 19.8 <i>bis</i>	8
36) Paragraphe 19.9	8
37) Paragraphes 20.3 à 20.6	8
38) Nouveau paragraphe 20.6 <i>bis</i>	9
39) Paragraphe 20.7	10
40) Paragraphe 21.9	10
41) Nouveau paragraphe 21.9 <i>bis</i>	10
42) Paragraphe 21.12	10
43) Nouveau paragraphe 24.1 <i>bis</i>	11
44) Paragraphe 24.2	11

	<i>Page</i>
45) Paragraphe 24.3	11
46) Paragraphe 24.5.1	11
47) Paragraphe 24.8	11
48) Paragraphe 24.9	11
49) Paragraphe 25.3	11
50) Paragraphe 25.4	11
51) Tableau 5	12
52) Nouveaux paragraphes 25.5 <i>bis</i> et 25.5 <i>ter</i>	12
53) Paragraphes 25.6, 25.7 et 25.8.....	12
54) Paragraphe 25.11	12
55) Nouveaux paragraphes 25.11.1 à 25.11.3.....	13
56) Paragraphe 26.3	13
57) Paragraphe 27.3	13
58) Paragraphe 28.2	13
59) Paragraphe 28.3	14
60) Paragraphe 28.4	14
61) Paragraphe 28.5	14
62) Paragraphe 30	14
63) Paragraphe 38.....	16
64) Nouveaux paragraphes 38.1.4 <i>bis</i> et 38.1.4 <i>ter</i>	16
65) Paragraphe 45.4	17
66) Paragraphe 47.4.2	17
67) Paragraphe 48.7.1	18
68) Paragraphe 50.....	18
69) Paragraphe A.2.4	18
70) Paragraphe B.3.2.2	18
71) Paragraphe B.4.2	19
72) Nouvelle Annexe C <i>bis</i>	19
73) Paragraphe E.2.12.2.....	19
74) Annexe H.....	20

**NORME INTERNATIONALE
RECOMMANDATION UIT-T**

**Technologies de l'information – Notation de syntaxe abstraite numéro un:
spécification de la notation de base**

Amendement 1

Prise en charge des règles de codage XML étendues (EXTENDED-XER)

NOTE – Dans le présent amendement, tout texte nouveau ou modifié a été souligné dans les paragraphes à modifier. Lors de l'incorporation de ce texte dans la Recommandation de base, il convient de supprimer ce soulignement.

1) Introduction

- a) *Dans l'introduction, insérer le paragraphe suivant immédiatement avant le paragraphe commençant par la phrase: "Une spécification ASN.1 sera produite initialement avec un ensemble de types ASN.1 complètement définis":*

Il est aussi possible d'attribuer des instructions de codage à un type, modifiant son codage. Cela peut se faire soit en plaçant un préfixe de type avant la définition du type ou en employant une référence de type, soit au moyen d'une section de commande de codage placée à la fin d'un module ASN.1. La syntaxe générique des préfixes de type et les sections de commande de codage sont spécifiées dans la présente Recommandation | Norme internationale et comportent une référence de codage permettant d'indiquer les règles de codage qui sont modifiées par l'instruction de codage. La sémantique et la syntaxe détaillée des instructions de codage sont définies dans la Recommandation | Norme internationale relative aux règles de codage identifiées par la référence de codage.

- b) *Dans l'introduction, insérer le paragraphe suivant entre les paragraphes commençant par "Annexe C" et "Annexe D":*

L'Annexe C bis, qui fait partie intégrante de la présente Recommandation | Norme internationale, spécifie les références de codage actuellement définies et la Recommandation | Norme internationale qui définit la sémantique et la syntaxe détaillée des instructions de codage avec ces références de codage.

2) Paragraphe 2.2

Au 2.2, remplacer la Note 1 par la note suivante:

NOTE 1 – La référence ci-dessus a été introduite parce qu'elle fournit les noms des caractères de commande et spécifie les catégories de caractères.

3) Nouveau paragraphe 3.6.18 bis

Ajouter le nouveau 3.6.18 bis libellé comme suit:

3.6.18 bis référence de codage par défaut (pour un module): référence de codage qui est spécifiée dans l'en-tête de module et est supposée figurer dans tous les préfixes de type qui ne contiennent pas de référence de codage.

NOTE – Si aucune référence de codage par défaut n'est spécifiée dans l'en-tête de module, tous les préfixes de type qui ne contiennent pas de référence de codage attribuent des étiquettes.

4) Nouveaux paragraphes 3.6.22 bis à 3.6.22 quat

Ajouter les nouveaux 3.6.22 bis à quat comme suit:

3.6.22 bis section de commande de codage: partie d'un module ASN.1 qui permet d'attribuer des instructions de codage aux types définis ou employés avec ce module ASN.1.

3.6.22 ter instruction de codage: information pouvant être associée à un type au moyen d'un préfixe de type ou d'une section de commande de codage, qui affecte le codage de ce type à travers une ou plusieurs règles de codage ASN.1.

NOTE – Une instruction de codage n'affecte pas les valeurs abstraites d'un type, et n'est pas censée être vue par une application.

3.6.22 quat référence de codage: nom (voir l'Annexe C bis) qui spécifie les règles de codage affectées par une instruction de codage dans un préfixe de type ou une section de commande de codage.

NOTE – La référence de codage **TAG** peut être employée pour indiquer qu'un préfixe de type attribue une étiquette plutôt qu'une instruction de codage (voir 30.2).

5) Paragraphe 3.6.54

Remplacer le 3.6.54 par le texte suivant:

3.6.54 type réel: type simple dont les valeurs distinctives (spécifiées au 20) contiennent l'ensemble des nombres réels (nombres réels numériques) associés à des valeurs spéciales telles que NOT-A-NUMBER.

6) Paragraphe 3.6.69

Remplacer le 3.6.69 par le texte suivant:

3.6.69 étiquette: information complémentaire, distincte des valeurs abstraites du type, qui est associée à tous les types ASN.1 et peut être modifiée ou précédée d'un préfixe de type.

NOTE – L'information des étiquettes sert dans certaines règles de codage à garantir la non-ambiguïté des codages. Elle diffère des instructions de codage parce qu'elle est associée à tous les types ASN.1, même s'ils ne possèdent pas de préfixe de type.

7) Paragraphe 3.6.71

Remplacer le 3.6.71 par le texte suivant:

3.6.71 étiquetage: attribution d'une nouvelle étiquette, remplaçant l'étiquette existante (éventuellement celle placée par défaut) ou s'ajoutant à elle.

8) Nouveau paragraphe 3.6.74 bis

Ajouter le nouveau 3.6.74 bis, comme suit:

3.6.74 bis préfixe de type: partie de la notation ASN.1 pouvant servir à attribuer une instruction de codage ou une étiquette à un type.

9) Paragraphe 8.1

Remplacer le 8.1 par le texte suivant:

8.1 On spécifie une étiquette (soit dans le texte de la présente Recommandation | Norme internationale, soit au moyen d'un préfixe de type) en indiquant une classe et un numéro dans cette classe. L'étiquette peut appartenir à l'une des classes suivantes:

- universelle;
- application;
- privée;
- propre au contexte.

10) Paragraphe 8.3

Au 8.3 remplacer "30" par "30.2", et, dans la Note, remplacer "30" par "30.2".

11) Nouveau paragraphe 8 bis

Ajouter le nouveau 8 bis libellé comme suit:

8 bis Instructions de codage

8 bis.1 Une instruction de codage est attribuée à un type au moyen soit d'un préfixe de type (voir 30.3), soit d'une section de commande de codage (voir 50).

8 bis.2 Un préfixe de type peut contenir une référence de codage. Si ce n'est pas le cas, la référence de codage est déterminée par la référence de codage par défaut pour le module (voir 12.4 bis).

8 bis.3 Une section de commande de codage contient toujours une référence de codage. Il peut y avoir plusieurs sections de commande de codage, mais chacune d'elles doit posséder une référence de codage distincte.

8 bis.4 Une instruction de codage comporte une séquence d'unités lexicales spécifiées dans la Recommandation | Norme internationale indiquée par la référence de codage (voir l'Annexe C bis).

8 bis.5 Plusieurs instructions de codage ayant les mêmes ou différentes références de codage peuvent être attribuées à un type (au moyen d'un ou de deux préfixes de type et d'une section de commande de codage). Les instructions attribuées dans une référence de codage donnée sont indépendantes de celles attribuées avec une référence de codage différente, ainsi que des préfixes de type destinés à l'étiquetage.

8 bis.6 Les conséquences de l'attribution de plusieurs instructions de codage avec la même référence de codage (au moyen d'un ou de deux préfixes de type et d'une section de commande de codage) sont spécifiées dans la Recommandation | Norme internationale indiquée par la référence de codage (voir l'Annexe C bis) mais ne sont pas spécifiées dans la présente Recommandation | Norme internationale.

8 bis.7 Si une instruction de codage est attribuée au "Type" dans la notation "TypeAssignment", elle est associée au type et s'applique dès que la notation "typereference" est utilisée dans la notation "TypeAssignment". Cela inclut l'emploi dans d'autres modules à travers les déclarations d'importation et d'exportation.

12) Nouveau paragraphe 11.20 bis

Ajouter le nouveau 11.20 bis libellé comme suit:

11.20 bis Références de codage

Nom de l'unité lexicale – encodingreference

La notation "encodingreference" comportera une séquence de caractères comme spécifiés dans la notation "typereference" au 11.2, mais sans minuscule.

NOTE – Les références de codage actuellement définies sont énumérées à l'Annexe C bis avec la Recommandation | Norme internationale qui spécifie la syntaxe et la sémantique des instructions de codage correspondantes. La notation "encodingreference" ne comportera que les séquences énumérées à l'Annexe C bis de la présente Recommandation | Norme internationale ou de ses versions ultérieures.

13) Paragraphe 11.23.2

Remplacer le 11.23.2 par le texte suivant:

11.23.2 En analysant un exemple de l'emploi de cette notation, on distingue à partir du contexte dans lequel elle est employée, la valeur "true" de la référence "valuereference" ou "identifier" ou d'une instance de la valeur booléenne en langage de balisage extensible (XML, extensible markup language) "extended-true".

NOTE – Cette séquence ne contient pas de blanc (voir 11.1.2).

14) Nouveau paragraphe 11.23 bis

Ajouter le nouveau 11.23 bis comme suit:

11.23 bis Unité lexicale booléenne extended-true en langage de balisage extensible

Nom de l'unité lexicale – extended-true

11.23 bis.1 Cette unité lexicale comportera soit la séquence de caractères suivants:

true

soit le seul caractère:

1 (CHIFFRE UN)

11.23 bis.2 En analysant un exemple de l'emploi de cette notation, on distingue à partir du contexte dans lequel elle est employée, la valeur "extended-true" de la référence "valuereference" ou "identifier" ou d'une instance de la valeur booléenne en langage XML "true".

NOTE – Cette séquence ne contient pas de blanc (voir 11.1.2).

15) Paragraphe 11.24.2

Remplacer le 11.24.2 par le texte suivant:

11.24.2 En analysant un exemple de l'emploi de cette notation, on distingue à partir du contexte dans lequel elle est employée, la valeur "false" de la référence "valuereference" ou "identifier" ou d'une instance de la valeur booléenne en langage XML "extended-false".

NOTE – Cette séquence ne contient pas de blanc (voir 11.1.2).

16) Nouveaux paragraphes 11.24 bis à 11.24 quat

Ajouter les nouveaux 11.24 bis, ter et quat libellés comme suit:

11.24 bis Unité lexicale booléenne extended-false en langage de balisage extensible

Nom de l'unité lexicale – extended-false

11.24 bis.1 Cette unité lexicale comportera soit la séquence de caractères suivants:

false

soit le seul caractère:

0 (CHIFFRE ZERO)

11.24 bis.2 En analysant un exemple de l'emploi de cette notation, on distingue à partir du contexte dans lequel elle est employée, la valeur "false" de la référence "valuereference" ou "identifier" ou d'une instance de la valeur booléenne en langage XML "false".

NOTE – Cette séquence ne contient pas de blanc (voir 11.1.2).

11.24 ter Unité lexicale réelle pas un nombre en langage de balisage extensible

Nom de l'unité lexicale – "NaN" (*not a number*, pas un nombre)

11.24 ter.1 Cette unité lexicale comporte la séquence de caractères suivants:

NaN

11.24 ter.2 En analysant un exemple de l'emploi de cette notation, on distingue à partir du contexte dans laquelle elle est employée, l'unité lexicale "NaN" de toute autre unité lexicale commençant par une majuscule.

NOTE – Cette séquence ne contient pas de blanc (see 11.1.2).

11.24 quat Unité lexicale réelle infinité en langage de balisage extensible

Nom de l'unité lexicale – "INF"

11.24 quat.1 Cette unité lexicale comporte la séquence de caractères suivants:

INF

11.24 quat.2 En analysant un exemple de l'emploi de cette notation, on distingue à partir du contexte dans laquelle elle est employée, l'unité lexicale "INF" de toute autre unité lexicale commençant par une majuscule.

NOTE – Cette séquence ne contient pas de blanc (voir 11.1.2).

17) Paragraphe 11.25.5

Remplacer le 11.25.5 par le texte suivant:

11.25.5 Si le type prédéfini en ASN.1 est un type "PrefixedType", alors le type qui définit l'unité lexicale "xmlasn1typename" sera le type "Type" dans le type "PrefixedType" (voir 30.1.5). S'il est lui-même un type "PrefixedType", alors le 11.25.5 s'appliquera de façon récurrente.

NOTE – Les sous-paragraphe du 25.11 spécifient le type "Type" à employer pour les types "SelectionType" et "ConstrainedType".

18) Tableau 4

Dans le Tableau 4, remplacer le type "TaggedType" par le type "PrefixedType".

19) Paragraphe 11.27

Au 11.27, introduire, dans de nouvelles cellules du tableau, en position alphabétique appropriée, les 3 nouveaux mots réservés suivants:

ENCODING-CONTROL

INSTRUCTIONS

NOT-A-NUMBER

20) Paragraphe 12.1

Au 12.1, remplacer la production "ModuleDefinition" par la production suivante:

```

ModuleDefinition ::=
    ModuleIdentif
    DEFINITIONS
    EncodingReferenceDefault
    TagDefault
    ExtensionDefault
    " : : ="
    BEGIN
    ModuleBody
    EncodingControlSections
    END
  
```

et introduire la nouvelle production immédiatement avant la production "TagDefault":

```

EncodingReferenceDefault ::=
    encodingreference INSTRUCTIONS
    | empty
  
```

21) Paragraphe 12.2

Dans la Note du 12.2, remplacer "30" par "30.2".

22) Nouveau paragraphe 12.4 bis

Ajouter le nouveau 12.4 bis libellé comme suit:

12.4 bis La notation "EncodingReferenceDefault" spécifie que la référence "encodingreference" est la référence de codage par défaut pour le module. Si la notation "EncodingReferenceDefault" est vide ("empty"), la référence par défaut pour le module est **TAG**.

NOTE – L'Annexe C bis contient une liste des références de codage autorisées, indiquant aussi la Recommandation | Norme internationale qui définit la forme et la signification des instructions de codage correspondantes.

23) Nouveau paragraphe 12.21

Ajouter le nouveau 12.21 comme suit:

12.21 Les sections de commande de codage "EncodingControlSections" sont spécifiées au 50.

24) Paragraphe 13.3

Dans la Note du 13.3, remplacer les étiquettes "XML tags" par les noms "XML tag names".

25) Paragraphe 16.2

Au 16.2, remplacer les 2 occurrences de type "TaggedType" par le type "PrefixedType".

26) Paragraphe 16.9

Au 16.9, remplacer la valeur "TaggedValue" par la valeur "PrefixedValue".

27) Paragraphe 16.10

Au 16.10, remplacer la valeur "XMLTaggedValue" par la valeur "XMLPrefixedValue".

28) Paragraphe 17.3

Remplacer le 17.3 par le texte suivant:

17.3 La valeur d'un type booléen (voir 3.6.73 et 3.6.38) sera définie au moyen de la notation "BooleanValue", ou si elle est employée comme valeur "XMLValue", au moyen de la notation "XMLBooleanValue". Ces productions sont les suivantes:

BooleanValue ::= TRUE | FALSE

XMLBooleanValue ::=
EmptyElementBoolean
TextBoolean

EmptyElementBoolean ::=
"<" & "true" "/>"
"<" & "false" "/>"

TextBoolean ::=
extended-true
extended-false

29) Nouveau paragraphe 17.4

Ajouter le nouveau 17.4 comme suit:

17.4 Si la notation "EmptyElementBoolean" figure dans une notation "XMLValueAssignment", il n'y aura pas d'occurrence de la notation "TextBoolean" dans cette notation "XMLValueAssignment".

30) Paragraphe 18.9

Remplacer le 18.9 par le texte suivant:

18.9 La valeur d'un type entier sera définie au moyen de la notation "IntegerValue", ou lorsqu'elle est utilisée comme valeur "XMLValue", au moyen de la notation "XMLIntegerValue". Ces productions sont les suivantes:

```

IntegerValue ::=
    SignedNumber
    |identifieur

XMLIntegerValue ::=
    XMLSignedNumber
    | EmptyElementInteger
    | TextInteger

XMLSignedNumber ::=
    number
    | "-" & number

EmptyElementInteger ::=
    "<" & identifieur ">"

TextInteger ::=
    identifieur
  
```

31) Nouveau paragraphe 18.9 bis

Ajouter le nouveau 18.9 bis libellé comme suit:

18.9 bis Si la notation "EmptyElementInteger" figure dans une notation "XMLValueAssignment", il n'y a pas d'occurrence de la notation "TextInteger" dans cette notation "XMLValueAssignment".

32) Paragraphe 18.10

Remplacer le 18.10 et sa Note par le texte suivant:

18.10 L'identificateur "identifieur" dans la valeur "IntegerValue" et dans les deux autres formes possibles de "XMLIntegerValue" sera l'un des indicateurs "identifieur" dans le type "IntegerType" auquel la valeur est associée, et représentera le nombre correspondant.

NOTE – Lors de la référence à une valeur entière pour laquelle un identificateur "identifieur" a été défini, il est préférable d'employer la forme de l'identificateur "identifieur" de la valeur "IntegerValue" et l'une des formes de l'identificateur "identifieur" de la valeur "XMLIntegerValue".

33) Nouveau paragraphe 18.12

Ajouter le nouveau 18.12 libellé comme suit:

18.12 La deuxième possibilité pour la notation "XMLSignedNumber" ne sera pas employée si le nombre "number" est zéro.

34) Paragraphe 19.8

Remplacer le 19.8 par le texte suivant:

19.8 La valeur d'un type énuméré sera définie au moyen de la notation "EnumeratedValue", ou lorsqu'elle est utilisée comme valeur "XMLValue", au moyen de la notation "XMLEnumeratedValue". Ces productions sont les suivantes:

EnumeratedValue ::= identifieur

XMLEnumeratedValue ::=
EmptyElementEnumerated
TextEnumerated

EmptyElementEnumerated ::= "<" & identifieur ">"

TextEnumerated ::= identifieur

35) Nouveau paragraphe 19.8 bis

Ajouter le nouveau 19.8 bis comme suit:

19.8 bis Si la notation "EmptyElementEnumerated" figure dans une notation "XMLValueAssignment", il n'y a pas d'occurrence de la notation "TextEnumerated" dans cette notation "XMLValueAssignment".

36) Paragraphe 19.9

Remplacer le 19.9 par le texte suivant:

19.9 L'identificateur "identifieur" dans la valeur "EnumeratedValue" et dans les deux autres formes possibles de "XMLEnumeratedValue" sera l'un des indicateurs "identifieur" dans le type "EnumeratedType" auquel la valeur est associée, et représentera le nombre correspondant.

37) Paragraphes 20.3 à 20.6

Remplacer les 20.3, 20.4, 20.5 et 20.6 par le texte suivant:

20.3 Les valeurs abstraites du type réel sont les valeurs spéciales PLUS-INFINITY, MINUS-INFINITY, et NOT-A-NUMBER avec des nombres numériques réels étant soit plus zéro, soit moins zéro, ou pouvant être spécifiés au moyen de la formule suivante qui met en jeu les trois entiers M, B et E:

$$M \times B^E$$

où M (non nul) est nommé la mantisse, B (2 ou 10) la base et E l'exposant. Les valeurs pour lesquelles B = 2 (valeurs abstraites en "base" 2) et B = 10 (valeurs abstraites en "base" 10) sont définies comme des valeurs abstraites distinctes. Sinon, les valeurs de $M \times B^E$ qui correspondent à la même valeur numérique sont une seule valeur abstraite.

NOTE – Moins zéro et plus zéro sont deux valeurs abstraites distinctes pour un zéro mathématique, et les valeurs abstraites en "base" 2 et "base" 10 sont distinctes pour tous les autres nombres numériques réels.

20.4 Au type réel est associé un type destiné à prendre en charge les notations des valeurs et des sous-types pour les valeurs numériques du type réel (s'ajoutant à la notation pour les valeurs spéciales du type réel et pour plus zéro et moins zéro).

NOTE – Les règles de codage peuvent définir un type différent servant à spécifier les codages; elles peuvent spécifier les codages sans faire référence à un type associé. En particulier, le codage à l'aide des règles de codage de base BER et de codage compact PER assure un codage décimal codé binaire (BCD, *binary-coded decimal*) si la "base" est 10, et un codage permettant une transformation efficace vers et depuis les représentations en virgule flottante matérielles si la "base" est 2.

20.5 Le type associé pour la définition (et l'attribution des sous-types) des valeurs numériques est le suivant (des commentaires normatifs ayant été ajoutés):

```

SEQUENCE {
    mantissa    INTEGER (ALL EXCEPT 0),
    base        INTEGER (2|10),
    exponent    INTEGER

    -- The associated mathematical real number is "mantissa"
    -- multiplied by "base" raised to the power "exponent"
}

```

NOTE 1 – Les valeurs représentées en "base" 2 et en "base" 10 sont considérées comme des valeurs abstraites distinctes même si elles correspondent à la même valeur réelle, et peuvent employer des sémantiques d'application différentes.

NOTE 2 – La notation **REAL (WITH COMPONENTS { ... , base (10) }**) peut être employée pour restreindre l'ensemble des valeurs aux valeurs numériques en "base" 10 (ou de manière semblable aux valeurs numériques en "base" 2). Cette notation n'incorpore pas les valeurs (valeurs réelles spéciales et plus et moins zéro) qui ne peuvent être représentées par le type associé. Si nécessaire, celles-ci peuvent être ajoutées au moyen d'un ensemble arithmétique.

NOTE 3 – Ce type peut véhiculer une représentation finie exacte de tout nombre qui peut être enregistré sur un support matériel classique à virgule flottante et de tout nombre dont la représentation décimale possède un nombre de caractères fini.

20.6 La valeur d'un type réel sera définie au moyen de la notation "RealValue", ou lorsqu'elle est utilisée dans une valeur "XMLValue", au moyen de la notation "XMLRealValue".

```

RealValue ::=
    NumericRealValue
  | SpecialRealValue

NumericRealValue ::=
    realnumber
  | "-" realnumber
  | SequenceValue    -- Valeur du type associé de séquence

SpecialRealValue ::=
    PLUS-INFINITY
  | MINUS-INFINITY
  | NOT-A-NUMBER

```

NOTE – La troisième possibilité pour les valeurs "NumericRealValue" ne peut être utilisée pour les valeurs plus zéro ou moins zéro. Ces valeurs abstraites sont spécifiées en employant soit la première, soit la deuxième possibilité, respectivement, avec un seul caractère "0" pour le nombre "realnumber".

```

XMLRealValue ::=
    XMLNumericRealValue | XMLSpecialRealValue

XMLNumericRealValue ::=
    realnumber
  | "-" & realnumber

XMLSpecialRealValue ::=
    EmptyElementReal
  | TextReal

EmptyElementReal ::=
    "<" & PLUS-INFINITY ">"
  | "<" & MINUS-INFINITY ">"
  | "<" & NOT-A-NUMBER ">"

TextReal ::=
    "INF"
  | "-" & "INF"
  | "NaN"

```

38) Nouveau paragraphe 20.6 bis

Ajouter le nouveau 20.6 bis comme suit:

20.6 bis Si la notation "EmptyElementReal" figure dans une notation "XMLValueAssignment", il n'y a pas d'occurrence de la notation "TextReal" dans cette notation "XMLValueAssignment".

39) Paragraphe 20.7

Remplacer le 20.7 par le texte suivant:

20.7 Lorsque la notation "realnumber" est employée, elle définit la valeur abstraite correspondante en "base" 10 ou plus zéro. Lorsqu'une notation "realnumber" est précédée du signe "-", elle définit les valeurs abstraites correspondantes en "base" 10, qui sont des nombres négatifs ou moins zéro. Si le type "RealType" est assujetti à la "base" 2, la notation "realnumber" ou "-" "realnumber" définit la valeur abstraite en "base" 2 correspondant soit à la valeur décimale spécifiée par "realnumber" soit à une précision localement définie, si une représentation exacte n'est pas possible.

40) Paragraphe 21.9

Remplacer le 21.9 par le texte suivant:

21.9 La valeur d'un type chaîne binaire sera définie au moyen de la notation "BitStringValue", ou lorsqu'elle est utilisée dans une valeur "XMLValue", au moyen de la notation "XMLBitStringValue". Ces productions sont les suivantes:

```

BitStringValue ::=
    bstring
  | hstring
  | "{" IdentifierList "}"
  | "{" "}"
  | CONTAINING Value

IdentifierList ::=
    identifieur
  | IdentifierList "," identifieur

XMLBitStringValue ::=
    XMLTypedValue
  | xmlbstring
  | XMLIdentifierList
  | empty

XMLIdentifierList ::=
    EmptyElementList
  | TextList

EmptyElementList ::=
    "<" & identifieur ">"
  | EmptyElementList "<" & identifieur ">"

TextList ::=
    identifieur
  | TextList identifieur

```

41) Nouveau paragraphe 21.9 bis

Ajouter le nouveau 21.9 bis libellé comme suit:

21.9 bis Si la notation "EmptyElementList" figure dans une notation "XMLValueAssignment", il n'y a pas d'occurrence de la notation "TextList" dans cette notation "XMLValueAssignment".

42) Paragraphe 21.12

Remplacer le 21.12 par le texte suivant:

21.12 Chaque identificateur "identifieur" dans la valeur "BitStringValue" ou dans les autres formes possibles de "XMLBitStringValue" sera identique à l'identificateur "identifieur" dans la séquence de production "BitStringType" à laquelle la valeur est associée.

43) Nouveau paragraphe 24.1 bis

Ajouter le nouveau 24.1 bis comme suit:

24.1 bis Aux fins des paragraphes suivants, un type "PrefixedType" est défini comme étant un type étiqueté textuellement si:

- a) le type "PrefixedType" est un type "TaggedType"; ou
- b) le "Type" dans le type "PrefixedType" est un type étiqueté textuellement.

44) Paragraphe 24.2

Remplacer le texte du 24.2 par le texte suivant (en conservant les deux Notes):

24.2 Lorsque la production "ComponentTypeLists" figure dans la définition d'un module pour lequel l'étiquetage automatique est sélectionné (voir 12.3), et qu'aucune des occurrences du type "NamedType" dans les trois premières formes possibles du type "ComponentType" n'est un type étiqueté textuellement (voir 24.1 bis), alors la transformation automatique d'étiquetage est sélectionnée pour la liste entière "ComponentTypeLists"; sinon elle ne l'est pas.

45) Paragraphe 24.3

Dans la Note du 24.3, remplacer "explicitement" par "textuellement".

46) Paragraphe 24.5.1

Au 24.5.1, remplacer "30" par "30.2".

47) Paragraphe 24.8

Remplacer le 24.8 par le texte suivant:

24.8 Si l'étiquetage automatique est en vigueur et que les types "ComponentType" dans la racine d'extension n'ont pas d'étiquettes, aucun type "ComponentType" de la liste "ExtensionAdditionList" ne sera un type étiqueté textuellement.

48) Paragraphe 24.9

Dans la Note 1 du 24.9, remplacer "30.6" par "30.2.7".

49) Paragraphe 25.3

Au 25.3, enlever la possibilité "XMLSpaceSeparatedList" de la production "XMLSequenceOfValue", enlever la production "XMLSpaceSeparatedList", supprimer le caractère "1" dans la Note 1 et supprimer la Note 2.

50) Paragraphe 25.4

Ajouter la Note suivante à la suite du 25.4:

NOTE – Cela ne se produit que pour la SEQUENCE OF NULL.

51) Tableau 5

Remplacer le Tableau 5 par le tableau suivant:

Tableau 5 – Notations "XMLSequenceOfValue" et "XMLSetOfValue" pour les types ASN.1

Type ASN.1	Notation des valeurs en langage XML
BitStringType	XMLDelimitedItemList
BooleanType	<i>Voir 25.5 bis</i>
CharacterStringType	XMLDelimitedItemList
ChoiceType	XMLValueList
EmbeddedPDVType	XMLDelimitedItemList
EnumeratedType	<i>Voir 25.5 ter</i>
ExternalType	XMLDelimitedItemList
InstanceOfType	<i>Voir la Rec. UIT-T X.681 ISO/CEI 8824-2, C.9</i>
IntegerType	XMLDelimitedItemList
NullType	XMLValueList
ObjectClassFieldType	<i>Voir la Rec. UIT-T X.681 ISO/CEI 8824-2, 14.10 et 14.11</i>
ObjectIdentifierType	XMLDelimitedItemList
OctetStringType	XMLDelimitedItemList
RealType	XMLDelimitedItemList
RelativeOIDType	XMLDelimitedItemList
SequenceType	XMLDelimitedItemList
SequenceOfType	XMLDelimitedItemList
SetType	XMLDelimitedItemList
SetOfType	XMLDelimitedItemList
PrefixedType	<i>Voir 25.11.1</i>
UsefulType (GeneralizedTime)	XMLDelimitedItemList
UsefulType (UTCTime)	XMLDelimitedItemList
UsefulType (ObjectDescriptor)	XMLDelimitedItemList
TypeFromObject	<i>Voir la Rec. UIT-T X.681 ISO/CEI 8824-2, 15.6</i>
ValueSetFromObjects	<i>Voir la Rec. UIT-T X.681 ISO/CEI 8824-2, 15.6</i>

52) Nouveaux paragraphes 25.5 bis et 25.5 ter

Ajouter les nouveaux 25.5 bis et 25.5 ter libellés comme suit:

25.5 bis Si la valeur "EmptyElementBoolean" est employée pour un type booléen, alors la notation "XMLValueList" sera utilisée, sinon ce sera la notation "XMLDelimitedItemList".

25.5 ter Si la valeur "EmptyElementEnumerated" est employée pour un type énuméré, alors la notation "XMLValueList" sera utilisée, sinon ce sera la notation "XMLDelimitedItemList".

53) Paragraphes 25.6, 25.7 et 25.8

Supprimer les 25.6, 25.7 et 25.8. Il faut noter que ces paragraphes ont été insérés avec des modifications mineures en tant que 25.11.1, 25.11.2 et 25.11.3, respectivement.

54) Paragraphe 25.11

Remplacer le 25.11 par le texte suivant:

25.11 Lorsque la première forme possible pour la notation "XMLDelimitedItem" est employée, et si la composante du type sequence-of (sans tenir compte d'éventuelles occurrences du préfixe "TypePrefix") est une référence

"typereference" ou "ExternalTypeReference", le nom "NonParameterizedTypeName" sera la référence "typereference" ou la référence "typereference" dans la référence "ExternalTypeReference", respectivement, sinon il sera le nom "xmlasnltypename" spécifié dans le Tableau 4, correspondant au type prédéfini de la composante.

55) Nouveaux paragraphes 25.11.1 à 25.11.3

Ajouter les nouveaux 25.11.1, 25.11.2 et 25.11.3 comme suit:

25.11.1 Si le "Type" de la composante est le type "PrefixedType", alors le type qui détermine l'autre forme possible de la valeur "XMLSequenceOfValue" et le nom "xmlasnltypename" (s'il est requis) sera le "Type" dans le type "PrefixedType" (voir 30.1.5). Si celui-ci est lui-même un type "PrefixedType", un type "ConstrainedType" ou un type "SelectionType" alors le présent 25.11 s'appliquera de manière récurrente.

25.11.2 Si le "Type" de la composante est le type "ConstrainedType", alors le type qui détermine l'autre forme possible de la valeur "XMLSequenceOfValue" et le nom "xmlasnltypename" (s'il est requis) sera le "Type" dans le type "ConstrainedType" (voir 45.1). Si celui-ci est lui-même un type "PrefixedType", un type "ConstrainedType" ou un type "SelectionType" alors le présent 25.11 s'appliquera de manière récurrente.

25.11.3 Si le "Type" de la composante est le type "SelectionType", alors le type qui détermine l'autre forme possible de la valeur "XMLSequenceOfValue" et le nom "xmlasnltypename" (s'il est requis) sera le "Type" dans le type "SelectionType" (voir 29). Si celui-ci est lui-même un type "PrefixedType", un type "ConstrainedType" ou un type "SelectionType" alors le présent 25.11 s'appliquera de manière récurrente.

56) Paragraphe 26.3

Au 26.3, remplacer "30" par "30.2".

57) Paragraphe 27.3

Remplacer le 27.3 par le texte suivant:

27.3 La notation permettant de définir une valeur du type set-of sera la valeur "SetOfValue", ou "XMLSetOfValue" lorsqu'elle est employée comme une valeur "XMLValue". Ces productions sont les suivantes:

```

SetOfValue ::=
    "{" ValueList "}"
  |   "{" NamedValueList "}"
  |   "{" "}"

XMLSetOfValue ::=
    XMLValueList
  |   XMLDelimitedItemList
  |   empty
  
```

Les listes "ValueList", "NamedValueList" et les autres valeurs possibles "XMLSetOfValue" sont spécifiées au 25.3, et le choix des autres valeurs possibles est le même que si la valeur "XMLSequenceOfValue" avait été employée. La notation "{" "}" ou "empty" est employée lorsque la valeur "SetOfValue" ou "XMLSetOfValue" est une liste vide.

NOTE 1 – L'ordre de ces valeurs ne devrait pas avoir de signification sémantique.

NOTE 2 – Il n'est pas demandé aux règles de codage de préserver l'ordre de ces valeurs.

NOTE 3 – Le type set-of n'est pas un ensemble mathématique de valeurs, donc, à titre d'exemple, pour les valeurs SET OF INTEGER, les valeurs { 1 } et { 1 1 } sont distinctes.

58) Paragraphe 28.2

Remplacer le 28.2 par le texte suivant:

28.2 Lorsque la production "AlternativeTypeLists" figure dans la définition d'un module pour lequel l'étiquetage automatique est sélectionné (voir 12.3), et qu'aucune des occurrences du type "NamedType" dans les listes "AlternativeTypeList" n'est un type étiqueté textuellement (voir 24.1 bis), la transformation automatique d'étiquetage est sélectionnée pour toutes les listes "AlternativeTypeLists", sinon ce n'est pas le cas.

59) Paragraphe 28.3

Au 28.3, remplacer "30" par "30.2".

60) Paragraphe 28.4

Au 28.4, remplacer le type "tagged type" par le type "textually tagged type".

61) Paragraphe 28.5

Dans la Note 1 du 28.5, remplacer "30.6" par "30.2.7".

62) Paragraphe 30

Remplacer le 30 par le texte suivant:

30 Notation des types préfixés

30.1 Généralités

30.1.1 Un type préfixé est un nouveau type qui est isomorphe à un type existant, mais possède une étiquette différente ou supplémentaire et éventuellement des instructions associées de codage différentes ou supplémentaires.

30.1.2 Un type préfixé est soit un type "TaggedType" ou un type "EncodingPrefixedType".

30.1.3 Un type préfixé qui est un type étiqueté est principalement utilisé là où la présente Recommandation | Norme internationale exige l'utilisation de types munis d'étiquettes distinctes (voir 24.5 à 24.6, 26.3 et 28.3). L'emploi de la valeur **AUTOMATIC TAGS** pour la notation "TagDefault" dans un module permet qu'un tel étiquetage soit effectué sans que la notation "TaggedType" figure explicitement dans ce module.

NOTE – Lorsqu'un protocole détermine qu'à un certain moment des valeurs de différents types de données peuvent être transmises, des étiquettes distinctes peuvent être nécessaires afin de permettre au destinataire de les décoder correctement.

30.1.4 L'attribution d'une instruction de codage en employant le type "EncodingPrefixedType" n'est pertinente que pour les codages identifiés par la référence de codage associée et n'a pas d'effet sur les valeurs abstraites du type.

30.1.5 Un type préfixé est défini par la notation "PrefixedType":

PrefixedType ::=
TaggedType
|
EncodingPrefixedType

NOTE – La spécification de la syntaxe pour le type "PrefixedType" serait plus simple et plus claire si l'étiquetage était décrit comme l'attribution d'une instruction de codage. Toutefois, historiquement, l'étiquetage a été introduit dans les précédentes versions des spécifications en ASN.1 et il peut affecter la légitimité d'une définition de type. Lorsque les types de codage préfixés ont été introduits, on a minimisé les modifications apportées aux concepts de l'étiquetage (et aux descriptions syntaxiques associées). L'étiquetage diffère aussi syntaxiquement de l'attribution des instructions de codage: la spécification indiquant que le codage est **EXPLICIT** ou **IMPLICIT** est faite après la fermeture "]" de l'étiquette qui n'est pas contenue entre les symboles "[" et "]" comme c'est le cas pour les instructions habituelles de codage.

30.1.6 La notation permettant de définir une valeur du type "PrefixedType" sera la valeur "PrefixedValue", ou "XMLPrefixedValue" lorsqu'elle est employée comme une valeur "XMLValue". Ces productions sont les suivantes:

PrefixedValue ::= Value
XMLPrefixedValue ::= XMLValue

où la valeur "Value" ou "XMLValue" est la notation d'une valeur du "Type" dans le type "TaggedType" ou du type "EncodingPrefixedType" dans le type "PrefixedType".

NOTE 1 – Ni l'étiquette "Tag" ni une partie du préfixe "EncodingPrefix" ne figure dans cette notation.

NOTE 2 – Les instructions de codage peuvent aussi être attribuées à un type dans une section de commande de codage (voir 50). Une telle attribution est sans effet sur la notation des valeurs du type.

30.2 Type étiqueté

30.2.1 Un type étiqueté est défini par la notation "TaggedType":

```

TaggedType ::=
    Tag Type
  | Tag IMPLICIT Type
  | Tag EXPLICIT Type

Tag ::= "[" EncodingReference Class ClassNumber "]"

EncodingReference ::=
    encodingreference ":"
  | empty

ClassNumber ::=
    number
  | DefinedValue

Class ::=
    UNIVERSAL
  | APPLICATION
  | PRIVATE
  | empty

```

30.2.2 Lorsqu'elle est employée dans l'étiquette "Tag", la référence "encodingreference" aura une valeur **TAG**. La référence "EncodingReference" dans l'étiquette "Tag" ne sera pas vide "empty" à moins que la référence de codage par défaut ne soit **TAG** (voir 12.4 bis).

30.2.3 Le type de la référence "valuereference" dans la notation "DefinedValue" sera entier, et une valeur non négative lui sera attribuée.

30.2.4 Le nouveau type est isomorphe au type existant, mais possède une étiquette indiquant la "Class" et le numéro "ClassNumber", sauf lorsque la "Class" est vide "empty", auquel cas la classe va dépendre du contexte, tandis qu'est indiqué le numéro "ClassNumber".

30.2.5 Seuls les types définis dans la présente Recommandation | Norme internationale se verront attribuer une "Class" **UNIVERSAL**.

NOTE 1 – L'emploi d'étiquettes de classe universelle fait régulièrement l'objet d'une entente entre l'UIT-T et l'ISO.

NOTE 2 – Le E.2.12 contient des lignes directrices et des indications sur l'emploi stylistique des classes d'étiquette.

30.2.6 L'étiquetage est toujours implicite ou explicite. L'étiquetage implicite indique, pour les règles de codage qui offrent le choix, qu'une identification explicite de l'étiquette originale du "Type" dans la notation "TaggedType" n'est pas nécessaire au cours du transfert.

NOTE – Il peut être utile de conserver l'étiquette existante lorsqu'elle est de classe universelle, et définit donc sans ambiguïté le type existant, sans que soit connue la définition en ASN.1 du nouveau type. Le transfert d'octets est toutefois normalement minimal en employant un étiquetage **IMPLICIT**. Un exemple de codage **IMPLICIT** est donné dans la Rec. UIT-T X.690 | ISO/CEI 8825-1.

30.2.7 L'opération d'étiquetage spécifie un étiquetage explicite si l'une des conditions suivantes est vérifiée:

- la forme possible "Tag **EXPLICIT** Type" est employée;
- la forme possible "Tag Type" est employée et la valeur de l'étiquetage "TagDefault" pour le module a une valeur **EXPLICIT TAGS** ou est vide;
- la forme possible "Tag Type" est employée et l'étiquetage "TagDefault" pour le module a une valeur **IMPLICIT TAGS** ou **AUTOMATIC TAGS**, mais le "Type" est un type choix non étiqueté, un type ouvert non étiqueté ou une référence "DummyReference" non étiquetée (voir la Rec. UIT-T X.683 | ISO/CEI 8824-4, 8.3).

Dans tous les autres cas, l'opération d'étiquetage spécifie un étiquetage implicite.

30.2.8 Si la "Class" est vide "empty", l'emploi de l'étiquette "Tag" n'est soumis à aucune restriction, autre que celles concernant les étiquettes distinctes qui sont spécifiées dans les 24.5 à 24.6, 26.3 et 28.3.

30.2.9 La valeur possible **IMPLICIT** ne sera pas employée si le "Type" est un type choix non étiqueté, un type ouvert non étiqueté ou une référence "DummyReference" non étiquetée (voir la Rec. UIT-T X.683 | ISO/CEI 8824-4, 8.3).

30.3 Type de codage préfixé

30.3.1 Le type de codage préfixé est défini par la notation "EncodingPrefixedType":

EncodingPrefixedType ::=
EncodingPrefix Type

EncodingPrefix ::=
"[" EncodingReference EncodingInstruction "]"

La référence "EncodingReference" est définie au 30.2.1.

30.3.2 La production "EncodingInstruction" est spécifiée dans la Recommandation | Norme internationale indiquée par la référence "EncodingReference" (voir l'Annexe C *bis*) et peut comporter une séquence quelconque d'unités lexicales ASN.1 (y compris des commentaires, des chaînes de caractères et des blancs).

NOTE 1 – Les unités lexicales "[" et "]" ne figurent jamais dans les instructions "EncodingInstruction".

NOTE 2 – Dans les versions ultérieures de la présente Recommandation | Norme internationale, on pourra ajouter d'autres références de codage à l'Annexe C *bis*. Il est recommandé que les outils en ASN.1 donnent (seulement) des avertissements lorsqu'une référence n'est pas celle qui est spécifiée à l'Annexe C *bis*, puis ne tiennent pas compte de l'ensemble du préfixe "EncodingPrefix" au moyen d'un symbole "]" indiquant la fin (voir la Note 1 ci-dessus).

30.3.3 Si la référence "EncodingReference" est vide, la référence de codage pour le préfixe de codage est la référence de codage par défaut pour le module.

NOTE – Si la référence de codage par défaut pour le module a une valeur TAG (voir 30.2.2) et que la référence "EncodingReference" est vide "empty", alors le type "PrefixedType" est un type "TaggedType", et non un type "EncodingPrefixedType".

30.3.4 Des restrictions existent en général concernant les instructions de codage (ayant la même référence de codage) qui peuvent être employées ensemble, et concernant les types auxquels des instructions particulières ou des instructions combinées peuvent s'appliquer. Ces restrictions sont spécifiées dans la Recommandation | Norme internationale associée à la référence de codage (voir l'Annexe C *bis*) et ne sont pas spécifiées dans la présente Recommandation | Norme internationale.

63) Paragraphe 38

Remplacer le 38 par le texte suivant:

38 Désignation des caractères, des ensembles de caractères et des ensembles de catégories de propriétés

Le présent paragraphe donne la description d'un module prédéfini en ASN.1 qui contient la définition des noms de référence des valeurs pour chacun des caractères définis dans la norme ISO/CEI 10646-1, chaque nom renvoyant à une valeur de chaîne `UniversalString` de dimension 1. Ce module contient aussi la définition des noms de référence des types pour chacun des ensembles de caractères définis dans la norme ISO/CEI 10646-1, chacun des noms renvoyant à un sous-ensemble de type `UniversalString`. Enfin, il contient les définitions des noms de référence "typereference" pour l'ensemble des caractères dont les propriétés appartenant aux catégories générales sont énumérées au 4.5 de la norme Unicode, où chaque nom renvoie à un sous-ensemble de type `UniversalString`.

NOTE – Ces valeurs peuvent être utilisées dans la notation des valeurs du type `UniversalString` et des types qui en sont déduits. Toutes les références aux valeurs et aux types, définies dans le module spécifié au 38.1, sont exportées et doivent être importées par tout module qui les emploie.

64) Nouveaux paragraphes 38.1.4 *bis* et 38.1.4 *ter*

Ajouter les nouveaux 38.1.4 *bis* et 38.1.4 *ter*, comme suit:

38.1.4 *bis* Pour chacune des abréviations et des descriptions dont la liste figure dans le Tableau 4-5 de la norme Unicode, deux spécifications sont introduites dans le module, sous la forme suivante:

```

<categoryabbreviation> ::= UniversalString (FROM (<alternativelist>))
    -- représente l'ensemble des caractères possédant cette propriété
    -- catégorie <categoryabbreviation>.

<categorydescription> UniversalString ::= <categoryabbreviation>

```

où:

- a) la notation <categoryabbreviation> est l'abréviation de la catégorie générale des propriétés de caractères, énumérées au Tableau 4-5 de la norme Unicode (par exemple, **Lu** ou **Nd** ou **Pi**);
- b) la notation <categorydescription> donne la description de la même catégorie générale de caractères, chacun des mots commençant par une majuscule, les virgules et les blancs étant supprimés, ainsi que toutes les descriptions entre parenthèses (par exemple, **LetterUppercase** ou **NumericDigit** ou **PunctuationInitialQuote**);
- c) la notation <alternativelist> pour chacune des notations <categoryabbreviation> est une liste de noms <namedcharacter> produits selon le 38.2 pour chacun des caractères mentionnés dans la base de données Unicode des caractères (version 3.2.0), qui possèdent la notation correspondante <categoryabbreviation>.

NOTE – Le nom Unicode pour un caractère est le même que celui de la notation <iso10646name> pour ce caractère.

38.1.4 ter Pour la première lettre de chacune des abréviations énumérées au Tableau 4-5 de la norme Unicode, deux spécifications sont introduites dans le module, sous la forme suivante:

```

<categoryabbreviationletter> ::= UniversalString (FROM (<alternativelist>))
    -- représente l'ensemble des caractères de propriété de catégorie
    -- quelconque avec la première lettre <categoryabbreviationletter>.

<maincategorydescription> UniversalString ::= <categoryabbreviationletter>

```

où:

- a) la notation <categoryabbreviationletter> est la première lettre de l'abréviation de la catégorie générale des propriétés de caractères, énumérées au Tableau 4-5 de la norme Unicode (par exemple, **L** ou **N** ou **P**);
- b) la notation <categorydescription> est le premier mot de la description de la même catégorie de caractères (par exemple, **Letter** ou **Numeric** ou **Punctuation**);
- d) la notation <alternativelist> pour chacune des notations <categoryabbreviationletter> est une liste de noms <namedcharacter> produits selon 38.2 pour chacun des caractères mentionnés dans la base de données Unicode des caractères (version 3.2.0), qui possèdent la notation correspondante <categoryabbreviationletter>.

NOTE – Le nom Unicode pour un caractère est le même que celui de la notation <iso10646name> pour ce caractère.

65) Paragraphe 45.4

Au 45.4, remplacer les deux occurrences de "TaggedType" par "PrefixedType".

66) Paragraphe 47.4.2

Remplacer la Note dans le 47.4.2 par le texte suivant:

NOTE – Aux fins de l'attribution d'un sous-type, la valeur **NOT-A-NUMBER** est supérieure à toutes les valeurs réelles, la valeur **PLUS-INFINITY** est supérieure à toutes les valeurs réelles à l'exception de la valeur **NOT-A-NUMBER**, la valeur **moins zéro** est supérieure à toutes les valeurs réelles négatives et est inférieure à plus zéro, et la valeur **MINUS-INFINITY** est inférieure à toutes les valeurs réelles. Sinon, l'ordre mathématique est appliqué.

67) Paragraphe 48.7.1

Au 48.7.1 a), remplacement sans objet dans la version française.

68) Paragraphe 50

Ajouter le nouveau 50 libellé comme suit:

50 Sections de commande de codage

50.1 La section "EncodingControlSections" est définie au moyen des productions suivantes:

```
EncodingControlSections ::=
  EncodingControlSection EncodingControlSections
| empty

EncodingControlSection ::=
  ENCODING-CONTROL
  encodingreference
  EncodingInstructionAssignmentList
```

50.2 Chacune des sections "EncodingControlSection" incorporées dans un module ASN.1 aura une référence "encodingreference" différente, et attribuera des instructions de codage pour cette référence de codage à l'un ou à plusieurs types dans le module.

50.3 La référence "encodingreference" n'aura pas la valeur **TAG**.

50.4 La production "EncodingInstructionAssignmentList" et la sémantique associée sont spécifiées dans la Recommandation | Norme internationale indiquée par la référence "encodingreference" (voir l'Annexe C *bis*) et peuvent comporter une séquence quelconque d'unités lexicales ASN.1 (y compris des commentaires, des chaînes de caractères et des blancs), à l'exception des unités lexicales **END** et **ENCODING-CONTROL**, qui ne figureront pas dans une notation "EncodingInstructionAssignmentList".

NOTE 1 – Dans les versions ultérieures de la présente Recommandation | Norme internationale, on pourra ajouter d'autres références de codage à l'Annexe C *bis*. Il est recommandé que les outils en ASN.1 donnent (seulement) des avertissements lorsqu'une référence "encodingreference" dans une section "EncodingControlSection" n'est pas celle qui est spécifiée à l'Annexe C *bis*, puis ne tiennent pas compte de la section "EncodingControlSection" jusqu'à l'occurrence suivante des valeurs **END** ou **ENCODING-CONTROL**, selon celle de ces valeurs qui se produira la première.

NOTE 2 – La référence "encodingreference" dans une section "EncodingControlSection" ne peut être omise. La référence de codage par défaut pour le module n'a aucune incidence sur la section "EncodingControlSection".

50.5 L'attribution à un type des instructions de codage au moyen d'un préfixe de type et d'une section "EncodingControlSection" est soumise à des liens d'interdépendance et à des restrictions. Il est toujours possible (c'est une question de style) de n'employer que des sections "EncodingControlSection", mais en général certaines des instructions de codage (en particulier celles qui s'appliquent à tous les types dans un module) ne peuvent être attribuées qu'à la section "EncodingControlSection". Les types auxquels s'appliquent des instructions particulières ou des instructions combinées sous aussi soumis à des restrictions. Ces liens d'interdépendance et ces restrictions sont spécifiés dans la Recommandation | Norme internationale associée à la référence de codage (voir l'Annexe C *bis*) et ne sont pas spécifiés dans la présente Recommandation | Norme internationale.

69) Paragraphe A.2.4

Ajouter le texte suivant à la fin du A.2.4:

Les expressions régulières "\N{LetterUppercase}" et "\N{Lu}" concordent avec tout caractère (unique) de la catégorie générale "Letter, uppercase" (dont l'abréviation est "Lu") telle que définie par la norme Unicode.

70) Paragraphe B.3.2.2

Au B.3.2.2, remplacer les deux occurrences de "30" par "30.2".

71) Paragraphe B.4.2

Au B.4.2, remplacer "30" par "30.2".

72) Nouvelle Annexe C bis

Ajouter la nouvelle Annexe C bis comme suit:

Annexe C bis**Références de codage**

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

C bis.1 Dans la présente annexe sont spécifiées les références de codage actuellement définies et la Recommandation | Norme internationale où est spécifiée la forme syntaxique (et sémantique) des instructions de codage associées à une référence de codage particulière (sauf pour la référence de codage **TAG** à laquelle aucune instruction de codage n'est associée).

NOTE – Il est recommandé que, si une référence de codage qui n'est pas spécifiée ici figure dans une spécification en ASN.1, il ne soit pas tenu compte des instructions de codage associées, le diagnostic émis étant (seulement) un avertissement.

C bis.2 Les références de codage dans la première colonne du Tableau C bis.1 sont actuellement définies. La syntaxe et la sémantique des instructions de codage associées (s'il y a lieu) sont définies dans la Recommandation | Norme internationale à laquelle renvoie la deuxième colonne du Tableau C bis.1.

Tableau C bis.1 – Normes définissant la sémantique associée à une référence de codage donnée

Référence de codage	Référence à la norme
TAG	La présente Recommandation Norme internationale
XER	Rec. UIT-T X.693 (2001)/Amd.1:2003 ISO/CEI 8825-4 (2002)/Amd.1:2004: Prise en charge des règles de codage XML étendues (EXTENDED-XER)

73) Paragraphe E.2.12.2

Remplacer le E.2.12.2 par le texte suivant:

E.2.12.2 Un style qu'il est fréquent de rencontrer lors de l'emploi des étiquettes, est celui qui consiste à attribuer une étiquette de classe d'application une fois exactement dans l'ensemble de la spécification, en l'employant pour identifier un type qui est largement utilisé, de manière dispersée, dans la spécification. Une étiquette de classe d'application est aussi fréquemment employée (une fois seulement) pour étiqueter les types dans les choix **CHOICE** les plus éloignés d'une application, assurant l'identification des messages individuels au moyen de l'étiquette de classes d'application. Ci-après est donné un exemple de l'emploi dans le premier cas:

EXEMPLE

```
FileName ::= [APPLICATION 8] SEQUENCE {
    directoryName                VisibleString,
    directoryRelativeFileName    VisibleString}
```

Dans l'exemple ci-dessus, il est supposé que la référence de codage par défaut est soit vide "empty" soit **TAG**. Sinon, cet exemple serait libellé comme suit:

```
FileName ::= [TAG: APPLICATION 8] SEQUENCE {
    directoryName                VisibleString,
    directoryRelativeFileName VisibleString}
```

Une modification semblable devrait être apportée dans les exemples suivants.

74) **Annexe H**

A l'Annexe H:

- a) *Ajouter les unités suivantes à la liste des unités lexicales:*

encodingreference

extended-true

extended-false

"NaN"

"INF"

ENCODING-CONTROL

INSTRUCTIONS

NOT-A-NUMBER

- b) *Remplacer la production "XMLBooleanValue" par les trois productions suivantes:*

XMLBooleanValue ::=
 EmptyElementBoolean
 | TextBoolean

EmptyElementBoolean ::=
 "<" & "true" ">"
 | "<" & "false" ">"

TextBoolean ::=
 extended-true
 | extended-false

- c) *Remplacer la production "XMLIntegerValue" par les quatre productions suivantes:*

XMLIntegerValue ::=
 XMLSignedNumber
 | EmptyElementInteger
 | TextInteger

XMLSignedNumber ::=
 number
 | "-" & number

EmptyElementInteger ::=
 "<" & identifier ">"

TextInteger ::=
 identifier

- d) *Remplacer la production "XMLEnumeratedValue" par les trois productions suivantes:*

XMLEnumeratedValue ::=
 EmptyElementEnumerated
 | TextEnumerated

EmptyElementEnumerated ::= "<" & identifier ">"

TextEnumerated ::= identifier

- e) *Remplacer les productions "XMLNumericRealValue" et "XMLSpecialRealValue" par les quatre productions suivantes:*

XMLNumericRealValue ::=
 realnumber
 | "-" & realnumber

XMLSpecialRealValue ::=
 EmptyElementReal
 | TextReal

EmptyElementReal ::=
 "<" & PLUS-INFINITY ">"
 | "<" & MINUS-INFINITY ">"
 | "<" & NOT-A-NUMBER ">"

TextReal ::=
 "INF"
 | "-" & "INF"
 | "NaN"

- f) Remplacer la production "XMLIdentifierList" par les trois productions suivantes:

XMLIdentifierList ::=
EmptyElementList
 | **TextList**
EmptyElementList ::=
 "<" & identifi er ">"
 | **EmptyElementList "<" & identifi er ">"**
TextList ::=
identifi er
 | **TextList identifi er**

- g) Dans la production "XMLSequenceOfValue", supprimer l'autre forme possible "XMLSpaceSeparatedList".
 h) Supprimer la production "XMLSpaceSeparatedList".
 i) Dans la production "XMLSetOfValue", supprimer l'autre forme possible "XMLSpaceSeparatedList".
 j) Ajouter les productions suivantes à la suite de la production "SelectionType":

PrefixedType ::=
TaggedType
 | **EncodingPrefixedType**
EncodingPrefixedType ::=
EncodingPrefix Type
EncodingPrefix ::=
 "[" EncodingReference EncodingInstruction]"
PrefixedValue ::= Value
XMLPrefixedValue ::= XMLValue

- k) Remplacer la production "Tag" par les deux productions suivantes:

Tag ::= "[" EncodingReference Class ClassNumber]"
EncodingReference ::=
encodingreference ":"
 | **empty**

- l) Supprimer les productions "TaggedValue" et "XMLTaggedValue".

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de nouvelle génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication