

# UIT-T

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

# Z.146

(03/2006)

SÉRIE Z: LANGAGES ET ASPECTS GÉNÉRAUX  
LOGICIELS DES SYSTÈMES DE  
TÉLÉCOMMUNICATION

Techniques de description formelle – Notation de test et  
de commande de test

---

**Notation de test et de commande de test  
version 3 (TTCN-3): utilisation de la notation  
ASN.1 avec la notation TTCN-3**

Recommandation UIT-T Z.146

RECOMMANDATIONS UIT-T DE LA SÉRIE Z  
LANGAGES ET ASPECTS GÉNÉRAUX LOGICIELS DES SYSTÈMES DE TÉLÉCOMMUNICATION

<b>TECHNIQUES DE DESCRIPTION FORMELLE</b>	
Langage de description et de spécification (SDL)	Z.100–Z.109
Application des techniques de description formelle	Z.110–Z.119
Diagrammes des séquences de messages	Z.120–Z.129
Langage étendu de définition d'objets	Z.130–Z.139
<b>Notation de test et de commande de test</b>	<b>Z.140–Z.149</b>
Notation de prescriptions d'utilisateur	Z.150–Z.159
<b>LANGAGES DE PROGRAMMATION</b>	
CHILL: le langage de haut niveau de l'UIT-T	Z.200–Z.209
<b>LANGAGE HOMME-MACHINE</b>	
Principes généraux	Z.300–Z.309
Syntaxe de base et procédures de dialogue	Z.310–Z.319
LHM étendu pour terminaux à écrans de visualisation	Z.320–Z.329
Spécification de l'interface homme-machine	Z.330–Z.349
Interfaces homme-machine orientées données	Z.350–Z.359
Interfaces homme-machine pour la gestion des réseaux de télécommunication	Z.360–Z.379
<b>QUALITÉ</b>	
Qualité des logiciels de télécommunication	Z.400–Z.409
Aspects qualité des Recommandations relatives aux protocoles	Z.450–Z.459
<b>MÉTHODES</b>	
Méthodes de validation et d'essai	Z.500–Z.519
<b>INTERGICIELS</b>	
Environnement de traitement réparti	Z.600–Z.609

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

## **Recommandation UIT-T Z.146**

### **Notation de test et de commande de test version 3 (TTCN-3): utilisation de la notation ASN.1 avec la notation TTCN-3**

#### **Résumé**

La présente Recommandation définit la façon d'utiliser la notation ASN.1, telle que définie dans les Recommandations UIT-T X.680, X.681, X.682 et X.683, avec la notation de test et de commande de test version 3 (TTCN-3, *testing and test control notation 3*), telle que définie dans la Rec. UIT-T Z.140.

Le contenu de la présente Recommandation figurait auparavant dans la Rec. UIT-T Z.140 (04/2003) dont il a été extrait. Des corrections et des modifications d'ordre rédactionnel ont été apportées au texte tel qu'il figurait dans la version 04/2003.

#### **Source**

La Recommandation UIT-T Z.146 a été approuvée le 16 mars 2006 par la Commission d'études 17 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

## 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 développeurs de consulter la base de données des brevets du TSB sous <http://www.itu.int/ITU-T/ipr/>.

© UIT 2007

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 MATIÈRES

	<i>Page</i>
1	Domaine d'application ..... 1
2	Références ..... 1
3	Définitions et abréviations ..... 1
3.1	Définitions..... 1
3.2	Abréviations ..... 2
4	Introduction..... 2
5	Généralités ..... 2
6	Modifications apportées au langage noyau..... 3
7	Types TTCN-3 additionnels..... 5
7.1	Généralités..... 5
7.2	Simple types et valeurs de base additionnels..... 5
7.3	Sous-typage de types additionnels ..... 6
8	Equivalences entre types ASN.1 et TTCN-3 ..... 6
8.1	Généralités..... 6
8.2	Identificateurs ..... 7
9	Types et valeurs de données ASN.1..... 7
9.1	Généralités..... 7
9.2	Portée des identificateurs ASN.1 ..... 14
10	Paramétrage en notation ASN.1 ..... 14
11	Définition de modèles de message ASN.1 ..... 15
11.1	Généralités..... 15
11.2	Messages de réception ASN.1 utilisant la syntaxe de modèle TTCN-3..... 15
11.3	Séquencement des champs de modèle ..... 16
12	Informations de codage..... 16
12.1	Généralités..... 16
12.2	Attributs de codage ASN.1..... 16
12.3	Attributs de variante ASN.1 ..... 17
	Annexe A – Formalisme BNF et sémantique statique additionnels ..... 18
	A.1 Prise en charge de la notation ASN.1 ..... 18
	Annexe B – Fonctions TTCN-3 prédéfinies ..... 19
	B.1 La fonction Décomposition ..... 19
	Annexe C – Composants d'identificateur d'objet prédéfinis..... 20
	BIBLIOGRAPHIE ..... 21



## Recommandation UIT-T Z.146

### Notation de test et de commande de test version 3 (TTCN-3): utilisation de la notation ASN.1 avec la notation TTCN-3

#### 1 Domaine d'application

La présente Recommandation définit une façon normative d'utiliser, avec la notation TTCN-3, la notation ASN.1 définie dans les Recommandations UIT-T X.680 [2], X.681 [3], X.682 [4] et X.683 [5]. Elle ne traite pas de l'harmonisation d'autres langages avec la notation TTCN-3.

#### 2 Références

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

- [1] Recommandation UIT-T Z.140 (2006), *Notation de test et de commande de test version 3 (TTCN-3): langage noyau.*
- [2] Recommandation UIT-T X.680 (2002), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- [3] Recommandation UIT-T X.681 (2002), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels.*
- [4] Recommandation UIT-T X.682 (2002), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes.*
- [5] Recommandation UIT-T X.683 (2002), *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un.*
- [6] Recommandation UIT-T X.690 (2002), *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*
- [7] Recommandation UIT-T X.691 (2002), *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact.*
- [8] Recommandation UIT-T X.693 (2001), *Technologies de l'information – Règles de codage ASN.1: règles de codage XML (XER).*
- [9] Vide.
- [10] Recommandation UIT-T T.100 (1988), *Echange international d'informations pour le Vidéotex interactif.*
- [11] Recommandation UIT-T T.101 (1994), *Interfonctionnement international pour les services Vidéotex.*
- [12] Recommandation UIT-T X.660 (2004), *Technologies de l'information – Interconnexion des systèmes ouverts – Procédures opérationnelles des organismes d'enregistrement de l'OSI: procédures générales et arcs sommitaux de l'arborescence des identificateurs d'objet ASN.1.*

#### 3 Définitions et abréviations

##### 3.1 Définitions

Dans la présente Recommandation, les termes et définitions figurant dans la Rec. UIT-T Z.140 [1] s'appliquent.

## 3.2 Abréviations

Dans la présente Recommandation, les abréviations figurant dans la Rec. UIT-T Z.140 [1] s'appliquent en plus de ce qui suit:

ASN.1            notation de syntaxe abstraite numéro un (*abstract syntax notation one*)

## 4 Introduction

Lorsque l'on utilise la notation ASN.1 avec la notation TTCN-3, toutes les caractéristiques de la notation TTCN-3 et les indications données au § 4/Z.140 [1] continuent de s'appliquer. En outre, lorsque la présente Recommandation est prise en charge, la notation TTCN-3 est alors pleinement harmonisée avec la notation ASN.1, qui peut être utilisée avec des modules TTCN-3 sous la forme d'une variante syntaxique pour les types et valeurs de données. La présente Recommandation définit l'utilisation de la notation ASN.1 dans des modules TTCN-3. La méthode employée pour associer les notations ASN.1 et TTCN-3 pourrait être appliquée pour prendre en charge l'utilisation, avec la notation TTCN-3, d'autres systèmes de types et de valeurs. Les détails n'en sont pas cependant définis dans la présente Recommandation.

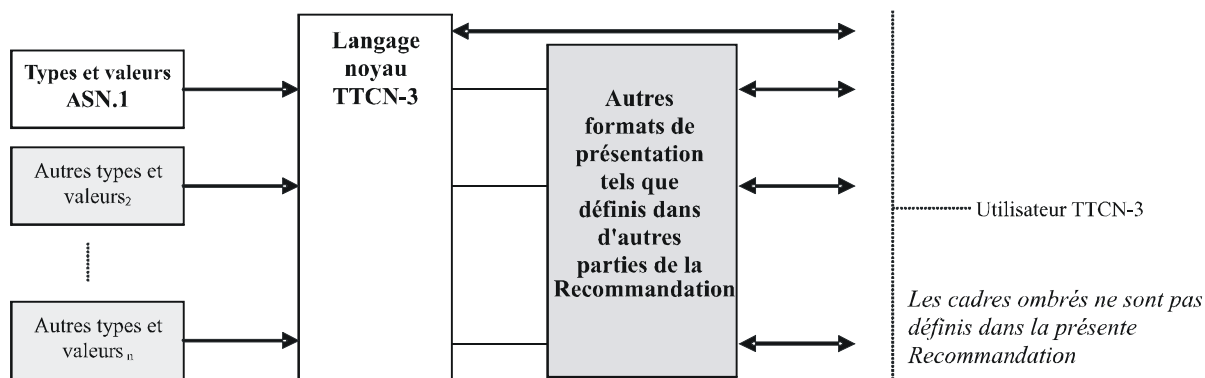


Figure 1/Z.146 – Vue d'utilisateur du langage noyau et des divers formats de présentation

## 5 Généralités

La notation TTCN-3 fournit une interface claire pour l'utilisation, dans des modules TTCN-3, de la version ASN.1 2002 (telle que définie dans les Recommandations UIT-T X.680 [2], X.681 [3], X.682 [4] et X.683 [5]). Les capacités XML de la notation ASN.1 ne sont pour l'instant pas considérées. En cas d'importation depuis un module ASN.1, l'une des chaînes d'identification de langage suivantes doit être utilisée:

- "ASN.1:2002"            pour la version ASN.1 2002;
- "ASN.1:1997"            pour la version ASN.1 1997;
- "ASN.1:1994"            pour la version ASN.1 1994;
- "ASN.1:1988"            pour la version du Livre bleu de l'ASN.1.

NOTE 1 – Les identificateurs de langage "ASN.1:1997", "ASN.1:1994" et "ASN.1:1988" se rapportent à des versions de l'ASN.1 fondées sur des Recommandations UIT-T obsolètes (document de base et tous les amendements et corrigenda techniques publiés jusqu'à présent pour modifier le document de base). La seule finalité de leur inclusion dans la présente Recommandation est d'affecter des identificateurs uniques si des modules de protocole fondés sur ces versions de l'ASN.1 sont utilisés avec la notation TTCN-3. Lorsque la version ASN.1 1997 est prise en charge, la prise en charge de l'Amendement 3 à la Rec. UIT-T X.680 [2] n'est pas considérée.

NOTE 2 – Lorsque la version "ASN.1:1988" est prise en charge, les items ASN.1 doivent être importés de tels modules conformément aux règles syntaxiques et sémantiques de la Rec. UIT-T X.208 (*Livre Bleu*).

Lorsque la notation ASN.1 est utilisée avec la notation TTCN-3, les mots clés énumérés au § 11.18/X.680 [2] ne doivent pas être utilisés comme identificateurs dans un module TTCN-3. Les mots clés ASN.1 doivent suivre les exigences de la Rec. UIT-T X.680 [2].



## 6 Modifications apportées au langage noyau

Lorsqu'on utilise des définitions ASN.1 en notation TTCN-3, les modifications apportées dans le présent paragraphe au langage noyau TTCN-3 tel qu'il est défini dans la Rec. UIT-T Z.140 [1] doivent également s'appliquer. **Les numéros de paragraphe ci-dessous sont ceux de la Rec. UIT-T Z.140 [1].**

### Paragraphe 3.1 Définitions

Modifier comme suit la définition de "types connus":

**types connus:** ensemble de types définis, de types ASN.1 importés et d'autres types externes importés.

Insérer la note suivante après la définition de "type radical":

NOTE – Dans le cas de types fondés sur des types ASN.1 importés, le type radical est déterminé à partir du type TTCN-3 associé (voir § 8).

### Paragraphe 3.2 Abréviations

Ajouter l'abréviation suivante:

ASN.1 notation de syntaxe abstraite numéro un (*abstract syntax notation one*)

### Paragraphe 7.1 Nommage des modules

Ajouter à la fin de la première phrase du premier alinéa: "facultativement suivi par un identificateur d'objet".

Ajouter la nouvelle note suivante après la NOTE 1:

NOTE 2 – Il est possible que des noms de module ne diffèrent que par la partie relative à l'identificateur d'objet. Des précautions appropriées doivent cependant être prises dans ce cas, lors de l'importation, afin d'éviter tout conflit de noms car l'adjonction de préfixes aux identificateurs (voir § 7.5.8) n'est pas en mesure de résoudre ce type de conflits.

### Paragraphe 7.5.0 Généralités

Ajouter le nouvel alinéa suivant après le 2<sup>e</sup> alinéa:

Si l'identificateur d'objet est fourni comme partie du nom de module (à partir duquel les définitions sont importées) dans l'instruction d'importation, cet identificateur d'objet doit servir à identifier le module correct.

### Paragraphe 7.5.8

Ajouter la note suivante après le 1<sup>er</sup> alinéa:

NOTE 1 – La règle permettant de résoudre des conflits de noms dans des valeurs d'identificateurs d'objet est donnée au point d) du § 6.1.0.

### Paragraphe 14.3 Mécanismes d'appariement de modèles

Ajouter `objid` à la liste de types figurant dans la note qui suit le 3<sup>e</sup> alinéa.

NOTE – Les types suivants peuvent être omis: ...

### Paragraphe 15.3 Opérateurs relationnels

Ajouter `objid` à la liste de types dans le 1<sup>er</sup> alinéa pour lesquels des opérateurs relationnels sont autorisés (tous les autres opérateurs relationnels ne doivent avoir des opérandes que du type `integer` (y compris les types dérivés de celui-ci), du type `float` (y compris les dérivés de ce type), du type `objid...`).

Ajouter un nouvel alinéa après le 2<sup>e</sup> alinéa:

Deux valeurs `objid` sont égales si elles ont le même nombre de composants et si les valeurs numériques sont les mêmes à toutes les positions. Les opérateurs d'infériorité (<), de supériorité (>), de supériorité ou égalité (>=) et d'infériorité ou égalité (<=) doivent utiliser les valeurs numériques des composants des valeurs `objid` pour le processus de décision, qui doit satisfaire aux règles suivantes:

- il faut commencer par comparer les premiers composants des valeurs `objid`;
- si la valeur numérique du composant comparé de la première valeur `objid` est inférieure à la valeur numérique du composant correspondant de la seconde valeur `objid`, la première valeur `objid` est inférieure à la seconde;
- si les valeurs numériques des deux composants comparés sont égales, la comparaison se poursuit en considérant la paire suivante de composants des deux valeurs `objid`; la valeur `objid` dans laquelle on trouve le premier composant ayant une valeur numérique plus petite est inférieure à l'autre valeur `objid`;

- si toutes les paires comparées de composants des deux valeurs objid sont égales et que l'une des valeurs objid a des composants supplémentaires par rapport à l'autre, la valeur objid la plus courte est inférieure à la valeur objid la plus longue.

EXEMPLE:

```
// si l'on a
const objid c_etsiMobNet := objid{itu_t identified_organization etsi(0)
                                mobile_domain(0) umts_Network(1)}
const objid c_etsiINNet  := objid{itu_t identified_organization etsi(0)
                                inDomain(1) in_Network(1)}
const objid c_etsiIN     := objid{itu_t identified_organization etsi(0)
                                inDomain(1)}
var objid v_etsiInIso    := objid{ iso identified_organization dod(6)
                                internet(1) private(4) enterprise(1) etsi(13019)}

// alors
c_etsiMobNet == c_etsiINNet // retourne la valeur "false"
c_etsiMobNet < c_etsiINNet // retourne la valeur "true" car le composant mobile_domain(0)
                           // est numériquement inférieur au composant inDomain(1)
c_etsiINNet == c_etsiIN     // retourne la valeur "false" car c_etsiINNet a plus de composants
c_etsiINNet > c_etsiIN     // retourne la valeur "true" car c_etsiINNet a plus de composants
v_etsiInIso <= c_etsiMobNet // retourne la valeur "false" car le composant itu_t(0) est
                           // numériquement inférieur au composant iso(1)
```

### Paragraphe 16.1.3 Fonctions prédéfinies

Ajouter la ligne ci-dessous à la section "Autres fonctions" du Tableau 10 ("Liste des fonctions prédéfinies en notation TTCN-3"):

<b>Autres fonctions</b>	Décompose une valeur objid	<b>decomp</b>
-------------------------	----------------------------	---------------

### Paragraphe A.1.5 Symboles terminaux de la notation TTCN-3

Ajouter **objid** au Tableau A.2 ("Liste des symboles terminaux spéciaux en notation TTCN-3").

#### Paragraphe A.1.6.0 Module TTCN-3

Ajouter '[DefinitiveIdentifieur]' à la fin de la production de TTCN3ModuleId.

NOTE – TTCN3ModuleId devient, en l'absence d'autres modifications dues à d'autres parties de la norme:

```
'3. TTCN3ModuleId ::= ModuleIdentifieur DefinitiveIdentifieur'
```

#### Paragraphe A.1.6.1.8 Définitions d'importation

Ajouter '[Dot ObjectIdentifieurValue]' à la fin de la production de GlobalModuleId.

NOTE – GlobalModuleId devient, en l'absence d'autres modifications dues à d'autres parties de la norme:

```
'223. GlobalModuleId ::= ModuleIdentifieur Dot ObjectIdentifieurValue'
```

#### Paragraphe A.1.6.3 Type

Ajouter 'ObjectIdentifieurKeyword' à la fin de la production de PredefinedType.

NOTE – PredefinedType devient, en l'absence d'autres modifications dues à d'autres parties de la norme:

```
'410. PredefinedType ::= BitStringKeyword |
                          BooleanKeyword |
                          CharStringKeyword |
                          UniversalCharString |
                          IntegerKeyword |
                          OctetStringKeyword |
                          HexStringKeyword |
                          VerdictTypeKeyword |
                          FloatKeyword |
                          AddressKeyword |
                          DefaultKeyword |
                          AnyTypeKeyword |
                          ObjectIdentifieurKeyword'
```

## Paragraphe A.1.6.4 Valeur

Ajouter 'ObjectIdentifierValue' à la fin de la production de PredefinedValue.

NOTE – PredefinedValue devient, en l'absence d'autres modifications dues à d'autres parties de la norme:

```
'434. PredefinedValue ::= BitStringValue |
                          BooleanValue |
                          CharStringValue |
                          IntegerValue |
                          OctetStringValue |
                          HexStringValue |
                          VerdictTypeValue |
                          EnumeratedValue |
                          FloatValue |
                          AddressValue |
                          OmitValue |
                          ObjectIdentifierValue'
```

## Paragraphe C.14 Nombre d'éléments contenus dans un type structuré

Ajouter **objid** à la liste de types dans la 1ère phrase du première alinéa ('Cette fonction retourne le nombre effectif d'éléments d'un paramètre, d'une constante, d'une variable ou d'un modèle (**template**) de module de type **record**, **record of**, **set**, **set of** ou **objid** ...').

Ajouter le nouvel alinéa suivant après le premier alinéa:

Dans le cas de valeurs, de modèles ou de tableaux **objid**, la valeur effective à retourner est le numéro séquentiel du dernier composant de la valeur **objid**.

Ajouter le nouvel exemple suivant à la fin de la section "EXEMPLE:"

```
// si l'on a
var objid v_etsiMobNet := objid(itu_t identified_organization etsi(0)
                               mobile_domain(0) umts_Network (1))
// alors
numElements := sizeof(v_etsiMobNet); // retourne 5
```

## 7 Types TTCN-3 additionnels

### 7.1 Généralités

Des types TTCN-3 additionnels nécessaires à la prise en charge de l'utilisation de la notation ASN.1 sont récapitulés dans le Tableau 1.

Tableau 1/Z.146 – Aperçu général des types TTCN-3

Classe de type	Mot clé	Sous-type
Simple types de base	<b>objid</b>	liste

### 7.2 Simples types et valeurs de base additionnels

Pour pouvoir utiliser la notation ASN.1 dans la notation TTCN-3, les simples types et valeurs suivants doivent être pris en charge en plus de ceux spécifiés au § 6.1 de la Recommandation sur le langage noyau (Rec. UIT-T Z.140 [1]):

- a) **objid**: type dont les valeurs distinctives sont formées par l'ensemble de:
  - toutes les valeurs d'identificateurs d'objet conformes à l'Annexe A/X.660 [12]; et
  - toutes les valeurs d'identificateurs d'objet syntaxiquement correctes n'appartenant pas à l'ensemble des valeurs définies dans la Rec. UIT-T X.660 [12] (par exemple, avec un nœud sous la racine non défini dans la Rec. UIT-T X.660 [12]).

Les notations de valeur pour le type **objid** doivent être conformes aux règles données au § 31/X.680 [2] à ceci près que les tirets sont remplacés par des soulignements dans les identificateurs d'objet.

NOTE 1 – La forme de nom des composants d'identificateurs d'objet ne doit être utilisée que pour les composants définis dans la Rec. UIT-T X.660 [12]. Ces composants d'identificateurs d'objet prédéfinis sont donnés dans l'Annexe C. En cas d'incohérence entre la Rec. UIT-T X.660 [12] et l'Annexe C, c'est la Rec. UIT-T X.660 qui prévaut.

Lorsque l'identificateur d'une valeur référencée dans une notation de valeur d'identificateur d'objet est identique à l'un quelconque des noms de composants prédéfinis (c'est-à-dire indépendamment de la position du composant prédéfini ou de la valeur référencée dans la notation), le nom de la valeur référencée doit avoir pour préfixe le nom du module dans lequel il est défini (voir la définition des modules TTCN-3 au § 7/Z.140 [1]). Le préfixe et l'identificateur doivent être séparés par un point (.). Les noms de composants d'identificateurs d'objet prédéfinis peuvent également avoir pour préfixe le nom 'X660'.

NOTE 2 – Pour une meilleure lisibilité, il est recommandé d'utiliser également le préfixe 'X660' dans les valeurs d'identificateurs d'objet qui font référence à un identificateur de valeur provoquant un conflit de nom par rapport à l'un quelconque des noms de composants prédéfinis.

NOTE 3 – Des règles permettant de résoudre des conflits de noms dus à des importations sont définies au § 7.5.8/Z.140 [1].

EXEMPLE:

```
objid{itu_t(0) identified_organization(4) etsi(0)}
// ou bien
objid {itu_t identified_organization etsi(0)}
// ou bien
objid { 0 4 0 }

// ou bien
const integer etsi := 0;
const objid itu_idOrg := objid{ itu_t identified_organization }
objid{ itu_idOrg etsi } // on notera que les deux noms font référence à des définitions de
                        valeurs

const integer x := 162;
objid{ itu_t recommendation x A.x } // il est obligatoire d'utiliser
// le nom du module ('A')
// comme préfixe avant l'identificateur ambigu
// ou bien
objid{ itu_t recommendation X660.x A.x } // le nom du module doit être présent même
// si le préfixe "X660" est utilisé
```

## 7.3 Sous-typage de types additionnels

### 7.3.1 Généralités

Les types définis par l'utilisateur doivent être indiqués par le mot clé **type**. Avec les types définis par l'utilisateur, il est possible de créer des sous-types (comme des listes) à partir de types, conformément au Tableau 1.

### 7.3.2 Listes de valeurs

La notation TTCN-3 permet de spécifier une liste de valeurs distinctives du type **objid**. Le sous-typage s'applique comme il est spécifié au § 6.2.1/Z.140 [1].

EXEMPLE:

```
type objid MyListOfObjids (objid{0 4 0 0 1}, objid{0 4 0 1 1});
```

## 8 Equivalences entre types ASN.1 et TTCN-3

### 8.1 Généralités

Les types ASN.1 énumérés dans le Tableau 2 sont considérés comme équivalents à leurs homologues TTCN-3.

**Tableau 2/Z.146 – Liste des types ASN.1 et TTCN-3 équivalents**

Type ASN.1	Type TTCN-3 équivalent
BOOLEAN	<code>boolean</code>
INTEGER	<code>integer</code>
REAL (Note)	<code>float</code>
OBJECT IDENTIFIER	<code>objid</code>
BIT STRING	<code>bitstring</code>
OCTET STRING	<code>octetstring</code>
SEQUENCE	<code>record</code>
SEQUENCE OF	<code>record of</code>
SET	<code>set</code>
SET OF	<code>set of</code>
ENUMERATED	<code>enumerated</code>
CHOICE	<code>union</code>
VisibleString	<code>charstring</code>
IA5String	<code>charstring</code>
UniversalString	<code>universal charstring</code>
<p>NOTE – Le type ASN.1 REAL est équivalent au type TTCN-3 <code>float</code> tant que la base est non restreinte ou restreinte à la base 10 explicitement ou implicitement. La notation ASN.1 permet une restriction explicite, par exemple par un sous-typage interne; mais du point de vue du mappage des types ASN.1-TTCN-3, une restriction explicite est une notation de valeur ASN.1. Une restriction implicite peut être définie par la description textuelle du protocole indiqué, c'est-à-dire à l'extérieur du ou des modules ASN.1. Cependant, dans les deux cas, la notation de valeur TTCN-3 peut être utilisée quelle que soit la base en ASN.1.</p>	

En notation TTCN-3, tous les opérateurs, toutes les fonctions, tous les mécanismes d'appariement, toutes les notations de valeur, etc., qui peuvent être utilisés avec un type TTCN-3 figurant dans le Tableau 2 peuvent également être utilisés avec le type ASN.1 correspondant.

## 8.2 Identificateurs

Lors de la conversion d'identificateurs ASN.1 en identificateurs TTCN-3, tous les symboles de tiret "-" doivent être remplacés par un soulignement "\_".

EXEMPLE:

```

MyASN1module DEFINITIONS ::=
BEGIN
    Missleading-ASN1-Name ::=    INTEGER -- Identificateur de type ASN.1 utilisant un tiret '-'

END

module MyTTCNModule
{
    import from MyASN1module language "ASN.1:2002" all;

    const Missleading_ASN1_Name ExampleConst := 1;           // référence TTCN-3 au type ASN.1
                                                                // utilisant des soulignements
}
    
```

## 9 Types et valeurs de données ASN.1

### 9.1 Généralités

Les types et valeurs ASN.1 peuvent être utilisés dans des modules TTCN-3. Pour les définitions ASN.1, on utilise un module ASN.1 distinct. Les types et valeurs ASN.1 sont désignés par leurs références de type et de valeur, produites conformément aux § 9.3 et 9.4/X.680 [2] dans le ou les modules ASN.1.

## EXEMPLE 1:

```
MyASN1module DEFINITIONS ::=
BEGIN
  Z ::=      INTEGER          -- Définition de type simple

  Bmessage ::= SET           -- Définition de type ASN.1
  {
    name      IA5String,
    title     VisibleString,
    date      IA5String
  }

  johnValues Bmessage ::=   -- Définition de valeur ASN.1
  {
    name      "John Doe",
    title     "Mr",
    date      "April 12th"
  }

  DefinedValuesForField1 Z ::= {0 | 1} -- Définition de sous-type ASN.1
END
```

Le module ASN.1 doit être conforme à la syntaxe des Recommandations UIT-T X.680 [2], X.681 [3], X.682 [4] et X.683 [5]. Une fois déclarés, les types et valeurs ASN.1 peuvent être utilisés dans des modules TTCN-3 de la même façon que les types et valeurs TTCN-3 issus d'autres modules TTCN-3 sont utilisés (c'est-à-dire que les définitions requises doivent être importées). Lors de l'importation d'items ASN.1 dans un module TTCN-3, un type ou une valeur associé est produit pour chaque item ASN.1 importé. Toutes les définitions ou affectations TTCN-3 fondées sur des items ASN.1 importés doivent être effectuées conformément aux règles imposées par le type ou la valeur associé. De plus, le mécanisme d'appariement doit utiliser le type associé lors de l'appariement de constantes, variables, modèles ou expressions en ligne fondées sur des déclarations ASN.1.

Les types et valeurs associés sont obtenus à partir d'items ASN.1 par application des règles de transformation indiquées ci-après. Pour les transformations, il faut partir d'un module ASN.1 valide et aboutir à un module TTCN-3 valide. L'ordre correspond à celui de l'exécution des transformations individuelles:

- 1) ignorer les éventuels marqueurs d'extension et spécifications d'exception.
- 2) Ignorer les éventuelles contraintes définies par l'utilisateur (voir § 9/X.682 [4]).
- 3) Ignorer les éventuelles contraintes de contenu (voir § 9/X.682 [4]).
- 4) Ignorer les éventuelles contraintes de structure (voir § 48.9/X.680 [2]).
- 5) Créer des sous-types TTCN-3 équivalents pour tous les types ASN.1 contraints au moyen d'un sous-typage contenu en remplaçant les types inclus par l'ensemble de valeurs qu'ils représentent. Des informations plus détaillées sur la conversion des contraintes de type ASN.1 en sous-types TTCN-3 sont données au Tableau 3. Le Tableau 3 donne les mécanismes de contraintes de type ASN.1 applicables aux différents types ASN.1. Lorsque "Non" figure dans la cellule, la contrainte de type n'est pas autorisée pour le type considéré. Les cellules ombrées identifient les contraintes de type applicables à un type donné et le texte dans la cellule définit les mécanismes de sous-typage TTCN-3 à utiliser lors de la transformation de types ASN.1 contraints.
- 6) Exécuter la transformation COMPONENTS OF conformément au § 24.4/X.680 [2] sur tout type SEQUENCE et conformément au § 26.2/X.680 [2] sur tout type SET contenant les mots clés "COMPONENTS OF".
- 7) Remplacer tout type EMBEDDED PDV par son type associé obtenu par expansion de sous-typage interne dans le type associé au type EMBEDDED PDV (voir § 33.5/X.680 [2]) afin d'obtenir une définition complète du type.
- 8) Remplacer le type EXTERNAL par son type associé obtenu par expansion de sous-typage interne dans le type associé au type EXTERNAL (voir § 34.5/X.680 [2]) afin d'obtenir une définition complète du type (voir Note 3).
- 9) Remplacer le type CHARACTER STRING par son type associé obtenu par expansion de sous-typage interne dans le type associé au type CHARACTER STRING (voir § 40.5/X.680 [2]) afin d'obtenir une définition complète du type.
- 10) Remplacer le type INSTANCE OF par son type associé obtenu en remplaçant INSTANCE OF DefinedObjectClass par son type ASN.1 associé (voir § C.7/X.681 [3]) et remplacer tous les types ASN.1 par leurs types TTCN-3 équivalents conformément au Tableau 2. Le type résultant est le type TTCN-3 associé.

- 11) Ignorer l'éventuel sous-typage interne restant (voir Note 4).
- 12) Ignorer les éventuels nombres nommés et bits nommés dans des types ASN.1. Dans les valeurs ASN.1, remplacer tout nombre nommé par sa valeur et remplacer tout bit nommé ou toute séquence de bits nommés par une chaîne binaire sans zéros à droite, où les positions binaires identifiées par des noms présents sont remplacées par des "1" et où les autres positions binaires sont remplacées par des "0".
- 13) Remplacer tout type sélection par le type désigné par ce type sélection; si le type choix indiqué (le "Type" dans le § 29.1/X.680 [2]) est un type contraint, la sélection doit être effectuée sur le type parent du type choix indiqué.
- 14) Convertir tout type ou valeur RELATIVE-OID en un type ou en une valeur **objid** (voir Note 5).
- 15) Remplacer tout type chaîne de caractères restreints ci-après par son type associé obtenu comme suit (voir Note 6):
  - BMPString: **universal charstring** (char ( 0,0,0,0 ) .. char ( 0,0,255,255)).
  - UTF8String: **universal charstring**.
  - NumericString: **charstring** contraint à l'ensemble de caractères indiqué dans le § 37.2/X.680 [2].
  - PrintableString: **charstring** contraint à l'ensemble de caractères indiqué dans le § 37.4/X.680 [2].
  - TeletexString et T61String: **universal charstring** contraint à l'ensemble de caractères indiqué dans la Rec. UIT-T T.61 (voir Bibliographie).
  - VideotexString: **universal charstring** contraint à l'ensemble de caractères indiqué dans les Recommandations UIT-T T.100 [10] et T.101 [11].
  - GraphicString: **universal charstring**.
  - GeneralString: **universal charstring**.
- 16) Remplacer les éventuels types ou valeurs GeneralizedTime et UTCTime par le type ou la valeur **charstring**.
- 17) Remplacer les éventuels types ou valeurs ObjectDescriptor par le type ou la valeur **universal charstring**.
- 18) Remplacer les éventuelles notations des types de champ de classe d'objets (voir § 14/X.681 [3]) par l'item ASN.1 auquel elles font référence (voir Note 8); les types ouverts doivent être remplacés par le métatype "OPEN TYPE" aux fins de la transformation (et seulement à cette fin).
- 19) Remplacer toutes les informations issues de notations d'objet (voir § 15/X.681 [3]) par l'item ASN.1 auquel elles font référence.
- 20) Transformer les contraintes tabulaires (voir § 10/X.682 [4]) en sous-typage énuméré et ignorer toutes les contraintes relationnelles (voir Note 7).
- 21) Remplacer toutes les occurrences du type NULL par le type TTCN-3 associé suivant:
  - **type enumerated** <identificateur> { NULL }, où <identificateur> est la référence de type ASN.1 convertie conformément au § 7.2.
- 22) Remplacer toutes les références à des types ouverts par le métatype "OPEN TYPE".
- 23) Remplacer les types ASN.1 par leurs équivalents conformément au Tableau 2 et remplacer les valeurs ASN.1 par les valeurs TTCN-3 équivalentes d'après les types associés. Les champs facultatifs manquants (c'est-à-dire implicitement omis) dans les valeurs ASN.1 structurées de type (SET, SEQUENCE, etc.) doivent être explicitement omis dans les valeurs TTCN-3 structurées résultantes (voir Note 9). Le métatype "OPEN TYPE" doit être remplacé par **anytype**.

NOTE 1 – Les seuls types associés ne contiennent pas toutes les informations nécessaires pour le codage correct des valeurs fondées sur les types ASN.1. Le traitement des informations complémentaires qui sont requises par le système pour assurer un codage correct dépend de l'implémentation et reste caché pour l'utilisateur. Cette connaissance n'est pas requise pour effectuer des déclarations ou affectations TTCN-3 valides faisant intervenir des types et valeurs ASN.1 importés.

NOTE 2 – Lors de l'importation de types ENUMERATED, les nombres entiers affectés par l'utilisateur à des énumérations sont également importés.

NOTE 3 – Le champ de valeur de données du type EXTERNAL peut être codé comme un type ASN.1 isolé, être aligné au niveau des octets ou être arbitraire (voir § 8.18.1/X.690 [6]) à la discrétion du codeur. Si l'utilisateur souhaite appliquer une certaine forme de codage ou ne permettre qu'une seule forme de codage spécifique lors de l'appariement, il doit utiliser l'attribut de codage approprié au type ou à la constante, à la variable, au modèle ou au champ de modèle considéré (voir § 11.3).

NOTE 4 – Le sous-typage interne doit être pris en considération par l'utilisateur lors de la définition de valeurs ou modèles TTCN-3 fondés sur un type ASN.1 contraint par sous-typage interne.

NOTE 5 – L'équivalence avec le type **objid** est limitée à la seule syntaxe à utiliser pour les notations de valeur. Lors du codage/décodage d'une valeur **objid** extraite d'une valeur RELATIVE-OID ASN.1 à l'aide d'une règle de codage ASN.1, le codage/décodage doit se faire conformément aux règles spécifiées pour le type RELATIVE-OID.

NOTE 6 – Les types VisibleString, IA5String et UniversalString ont un type TTCN-3 équivalent et sont remplacés directement.

NOTE 7 – Les contraintes relationnelles doivent être prises en considération par l'utilisateur lors de la déclaration de valeurs et de modèles (elles peuvent également être traitées implicitement par des utilitaires).

NOTE 8 – Pour ce remplacement, il n'est pas tenu compte des contraintes appliquées à la "notation du type de champ de classe d'objets" proprement dite.

NOTE 9 – Les champs facultatifs faisant défaut dans des valeurs de types ASN.1 structurés (SET, SEQUENCE, EXTERNAL, etc.) sont équivalents aux champs explicitement omis dans les valeurs TTCN-3 structurées.

#### EXEMPLE 2:

```
module MyTTCNModule
{
    import from MyASN1module language "ASN.1:2002" all;

    const Bmessage MyTTCNConst:= johnValues;
    const DefinedValuesForField1 Value1:= 1;
}
```

NOTE 10 – Les définitions ASN.1 autres que de types et valeurs (c'est-à-dire les classes ou ensembles d'objets informationnels) ne sont pas directement accessibles à partir de la notation TTCN-3. De telles définitions doivent être ramenées à un type ou à une valeur dans le module ASN.1 avant de pouvoir être désignées depuis le module TTCN-3.



**Tableau 3/Z.146 – Conversions des contraintes de type ASN.1 en sous-type TTCN-3**

Type (ou déduit d'un tel type par étiquetage ou sous-typage)	Valeur unique	Sous-type contenu <sup>b)</sup>	Etendue de valeurs	Contrainte de taille	Alphabet autorisé	Contrainte de type	Sous-typage interne <sup>l)</sup>	Contrainte de structure	Contrainte définie par l'utilisateur	Contrainte tabulaire <sup>k)</sup>	Contrainte relationnelle <sup>k)</sup>	Contrainte de contenu
Chaîne binaire	liste	Valeur unique: liste, taille: longueur	Non	longueur	Non	Non	Non	Non	ignorer	Non	Non	ignorer
Booléen	liste	liste	Non	Non	Non	Non	Non	Non	ignorer	Non	Non	Non
Choix	liste	liste	Non	Non	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Valeur pdv imbriquée <sup>a)</sup>	liste	Non	Non	Non	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Énuméré	liste	liste	Non	Non	Non	Non	Non	Non	ignorer	Non	Non	Non
Externe <sup>a)</sup>	liste	Non	Non	Non	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Instance-de <sup>a), b)</sup>	liste	liste	Non	Non	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Entier	liste	valeur unique: liste, étendue de valeurs: étendue	étendue	Non	Non	Non	Non	Non	ignorer	Non	Non	Non
Néant	ignorer	ignorer	Non	Non	Non	Non	Non	Non	ignorer	Non	Non	Non
Type champ de classe d'objets	<sup>c)</sup>	<sup>c)</sup>	Non	Non	Non	Non	Non	Non	ignorer	liste	ignorer	Non
Descripteur d'objets <sup>e)</sup>	liste	valeur unique: liste, taille: longueur, alphabet autorisé: étendue	Non	longueur	étendue	Non	Non	Non	ignorer	Non	Non	Non
Identificateur d'objet	liste	liste	Non	Non	Non	Non	Non	Non	ignorer	Non	Non	Non
Chaîne d'octets	liste	valeur unique: liste, taille: longueur	Non	longueur	Non	Non	Non	Non	ignorer	Non	Non	ignorer
Type ouvert <sup>f)</sup>	Non	Non	Non	Non	Non	anytype avec contrainte de liste	Non	Non	ignorer	Non <sup>m)</sup>	Non <sup>m)</sup>	Non

**Tableau 3/Z.146 – Conversions des contraintes de type ASN.1 en sous-type TTCN-3**

Type (ou déduit d'un tel type par étiquetage ou sous-typage)	Valeur unique	Sous-type contenu <sup>h)</sup>	Étendue de valeurs	Contrainte de taille	Alphabet autorisé	Contrainte de type	Sous-typage interne <sup>i)</sup>	Contrainte de structure	Contrainte définie par l'utilisateur	Contrainte tabulaire <sup>k)</sup>	Contrainte relationnelle <sup>k)</sup>	Contrainte de contenu
Réel	liste	valeur unique: liste, étendue de valeurs: étendue	étendue	Non	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Identificateur d'objet relatif <sup>d)</sup>	liste	liste	Non	Non	Non	Non	Non	Non	ignorer	Non	Non	Non
Type chaîne de caractères à alphabet restreint	liste	valeur unique: liste, taille: longueur, alphabet permis: étendue	étendue	longueur	étendue	Non	Non	ignorer <sup>e)</sup>	ignorer	Non	Non	Non
Séquence	liste	liste	Non	Non	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Séquence-de	liste	valeur unique: liste, étendue de valeurs: étendue	Non	longueur	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Ensemble	liste	liste	Non	Non	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Ensemble-de	liste	valeur unique: liste, étendue de valeurs: étendue	Non	longueur	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non
Types temps <sup>a)</sup>	liste	liste	Non	Non	Non	Non	Non	Non	ignorer	Non	Non	Non
Type chaîne de caractères à alphabet non restreint <sup>a)</sup>	liste	Non	Non	longueur (appliquée au champ "string-value")	Non	Non	convertir en type complet	Non	ignorer	Non	Non	Non

**Tableau 3/Z.146 – Conversions des contraintes de type ASN.1 en sous-type TTCN-3**

- |    |  |
|----|--|
| a) | Ces types sont vus depuis la notation TTCN-3 comme étant équivalents à leurs types associés.   |
| b) | Le champ Type-id du type associé à instance-de doit être remplacé par le type du champ &id dont la valeur est anytype (Annexe C/X.681 [3]).  |
| c) | Remplacé par le type référencé, et par conséquent applicable comme s'il s'agissait du type référencé.  |
| d) | Vu comme identificateur d'objet depuis la notation TTCN-3.   |
| e) | Son type associé est un type chaîne de caractères à alphabet restreint.  |
| f) | Type ouvert est remplacé par <b>anytype</b> .  |
| g) | Les structures de caractères ne peuvent être utilisées que dans des constantes, des variables, des modèles et des paramètres de module en notation TTCN-3 mais ne peuvent pas être utilisées pour le sous-typage.  |
| h) | Les contraintes de sous-type contenu doivent être remplacées par des contraintes littérales lors de l'importation.   |
| i) | Les informations dans cette colonne se rapportent aux vues TTCN-3 d'items ASN.1. Le codage/décodage doit être conforme au type radical; il faut donc également conserver des informations supplémentaires pour le codage, qui ne sont pas indiquées dans ce tableau. |
| k) | Applicable uniquement aux notations pour le type de champ de classe d'objets.  |
| m) | Applicable lorsque le type ouvert est défini en utilisant la notation du type de champ de classe d'objets (voir plus haut).  |

## 9.2 Portée des identificateurs ASN.1

Les identificateurs ASN.1 importés suivent les mêmes règles de portée que les types et valeurs TTCN-3 importés (voir § 5.3/Z.140 [1]).

## 10 Paramétrage en notation ASN.1

Il est permis de faire référence à des définitions paramétrées de type et valeur ASN.1 à partir du module TTCN-3. Cependant, toutes les définitions paramétrées ASN.1 utilisées dans un module TTCN-3 doivent être fournies avec des paramètres effectifs (les types ouverts ne sont pas autorisés) et les paramètres effectifs ainsi fournis doivent pouvoir être résolus au moment de la compilation.

Le langage noyau de la notation TTCN-3 ne prend pas en charge l'importation d'items ASN.1, qui font appel uniquement à des objets propres à la notation ASN.1 en tant que paramètre(s) formel(s) ou effectif(s). Le paramétrage propre à la notation ASN.1, qui fait intervenir des objets ne pouvant pas être définis directement dans le langage noyau de la notation TTCN-3, doit donc être résolu dans la partie ASN.1 avant d'être utilisé dans la notation TTCN-3. Les objets propres à la notation ASN.1 sont les suivants:

- a) classes d'objets informationnels;
- b) objets informationnels;
- c) ensembles d'objets informationnels.

Par exemple, la notation qui suit n'est pas légale parce qu'elle définit un type TTCN-3 qui prend un ensemble d'objets ASN.1 en tant que paramètre effectif.

```
MyASN1module DEFINITIONS ::=
BEGIN
  -- Définition de module ASN.1

  -- Définition de classe d'objets informationnels
  MESSAGE ::= CLASS { &msgTypeValue  INTEGER UNIQUE,
                    &MsgFields}

  -- Définition d'objets informationnels
  setupMessage MESSAGE ::= { &msgTypeValue  1,
                            &MsgFields      OCTET STRING}

  setupAckMessage MESSAGE ::= { &msgTypeValue  2,
                              &MsgFields      BOOLEAN}

  -- Définition d'ensemble d'objets informationnels
  MyProtocol MESSAGE ::= { setupMessage | setupAckMessage}

  -- Type ASN.1 contraint par un ensemble d'objets
  MyMessage{ MESSAGE : MsgSet} ::= SEQUENCE
  {
    code      MESSAGE.&msgTypeValue({ MsgSet}),
    Type      MESSAGE.&MsgFields({ MsgSet})
  }
END

module MyTTCNModule
{
  // Définition de module TTCN-3
  import from MyASN1module language "ASN.1:2002" all;

  // Type TTCN-3 illégal avec un ensemble d'objets comme paramètre
  type record Q(MESSAGE MyMsgSet) ::= { Z          field1,
                                       MyMessage(MyMsgSet) field2}
}
```

Afin de rendre cette définition légale, le type ASN.1 complémentaire My Message1 doit être défini comme indiqué ci-dessous. Cette notation résout le paramétrage d'ensemble d'objets informationnels et peut donc être directement utilisée dans le module TTCN-3.

```
MyASN1module DEFINITIONS ::=
BEGIN
  -- Définition de module ASN.1
  ...
```

```

MyProtocol MESSAGE ::= { setupMessage | setupAckMessage}

-- Type ASN.1 complémentaire pour supprimer le paramétrage d'ensemble d'objets
MyMessage1 ::= MyMessage{ MyProtocol}
END

module MyTTCNModule
{
  // Définition de module TTCN-3
  import from MyASN1module language "ASN.1:2002" all;

  // Type TTCN-3 légal sans ensemble d'objets comme paramètre
  type record Q := { Z field1,
                    MyMessage1 field2}
}

```

## 11 Définition de modèles de message ASN.1

### 11.1 Généralités

Si des messages sont définis en notation ASN.1 au moyen, par exemple, d'un type SEQUENCE (ou éventuellement SET) alors les messages effectifs, pour les deux événements **send** et **receive**, peuvent être spécifiés au moyen de la syntaxe de valeur ASN.1.

#### EXEMPLE:

```

MyASN1module DEFINITIONS ::=
BEGIN
  -- Définition de module ASN.1

  -- La définition du message
  MyMessageType ::= SEQUENCE
  { field1 [1] IA5STRING, // comme le type TTCN-3 character string
    field2 [2] INTEGER OPTIONAL, // comme le type TTCN-3 integer
    field3 [4] Field3Type, // comme le type TTCN-3 record
    field4 [5] Field4Type // comme le type TTCN-3 array
  }

  Field3Type ::= SEQUENCE {field31 BIT STRING, field32 INTEGER, field33 OCTET STRING},
  Field4Type ::= SEQUENCE OF BOOLEAN

  -- peut avoir la valeur suivante
  myValue MyMessageType ::=
  {
    field1 "A string",
    field2 123,
    field3 {field31 '11011'B, field32 456789, field33 'FF'O},
    field4 {true, false}
  }
END

```

### 11.2 Messages de réception ASN.1 utilisant la syntaxe de modèle TTCN-3

Les mécanismes d'appariement ne sont pas pris en charge dans la syntaxe ASN.1 normale. Donc, si l'on souhaite utiliser des mécanismes d'appariement avec un message de réception ASN.1, la syntaxe TTCN-3 pour les modèles de réception doit être utilisée. Il est à noter que cette syntaxe comprend des références de composant afin de pouvoir faire référence aux différents composants contenus dans des types ASN.1 SEQUENCE, SET, etc.

#### EXEMPLE:

```
import from MyASN1module language "ASN.1:2002" {
  type myMessageType
}

// Un modèle de message utilisant des mécanismes d'appariement en notation
// TTCN-3 pourrait être
template myMessageType MyValue:=
{
  field1 := "A"<?>"tr"<*>"g",
  field2 := *,
  field3.field31 := '110??'B,
  field3.field32 := ?,
  field3.field33 := 'F?'O,
  field4.[0] := true,
  field4.[1] := false
}

// la syntaxe suivante est également valide
template myMessageType MyValue:=
{
  field1 := "A"<?>"tr"<*>"g", // chaîne avec caractères génériques
  field2 := *, // entier quelconque ou absence de caractère
  field3 := {'110??'B, ?, 'F?'O},
  field4 := {?, false}
}
```

### 11.3 Séquencement des champs de modèle

Lorsque des modèles TTCN-3 sont utilisés pour des types ASN.1, l'importance de l'ordre des champs dans le modèle dépend du type de construction ASN.1 utilisé pour définir le type de message. Par exemple: si le type SEQUENCE ou SEQUENCE OF est utilisé, les champs de message doivent être envoyés ou appariés dans l'ordre spécifié dans le modèle. Si le type SET ou SET OF est utilisé, les champs de message peuvent être envoyés ou appariés dans un ordre quelconque.

## 12 Informations de codage

### 12.1 Généralités

La notation TTCN-3 permet d'associer à divers éléments de langage TTCN-3 des références à des règles de codage ou à des variantes de règles de codage. Il est également possible de définir des codages non valides. Ces informations de codage sont spécifiées au moyen de l'instruction **with** conformément à la syntaxe suivante:

#### EXEMPLE:

```
module MyModule
{
  :
  import from MyASN1module language "ASN.1:2002" {
    type myMessageType
  }
  with {
    encode "PER-BASIC-ALIGNED:1997" // Toutes les instances de MyMessageType devraient être
    // codées au moyen des règles PER:1997
  }
  :
} // fin du module
with { encode "BER:1997" } // Le codage par défaut pour le module entier
// (suite de tests) est BER:1997
```

### 12.2 Attributs de codage ASN.1

Les chaînes suivantes sont les attributs de codage prédéfinis (normalisés) pour la version actuelle de la notation ASN.1:

- "BER:2002" signifie: codé conformément à la Rec. UIT-T X.690 [6] (BER);
- "CER:2002" signifie: codé conformément à la Rec. UIT-T X.690 [6] (CER);
- "DER:2002" signifie: codé conformément à la Rec. UIT-T X.690 [6] (DER);

- d) "PER-BASIC-UNALIGNED:2002" signifie: codé conformément à la Rec. UIT-T X.691 [7] (règles PER sans alignement);
- e) "PER-BASIC-ALIGNED:2002" signifie: codé conformément à la Rec. UIT-T X.691 [7] (règles PER avec alignement);
- f) "PER-CANONICAL-UNALIGNED:2002" signifie: codé conformément à la Rec. UIT-T X.691 [7] (règles PER canoniques sans alignement);
- g) "PER-CANONICAL-ALIGNED:2002" signifie: codé conformément à la Rec. UIT-T X.691 [7] (règles PER canoniques avec alignement);
- h) "BASIC-XER:2002" signifie: codé conformément à la Rec. UIT-T X.693 [8] (règles de codage XML de base);
- i) "CANONICAL-XER:2002" signifie: codé conformément à la Rec. UIT-T X.693 [8] (règles de codage XML canoniques);
- j) "EXTENDED-XER:2002" signifie: codé conformément à l'Amd.1 à la Rec. UIT-T X.693 [8] (règles de codage XML étendues).

Les codages des versions ASN.1 précédentes (par exemple 1988, 1994 ou 1997) peuvent également être utilisés. La date doit alors être remplacée en conséquence. Par exemple, dans le cas de la notation ASN.1 1997, les attributs de codage suivants s'appliquent: "BER:1997", "CER:1997", "DER:1997", "PER-BASIC-UNALIGNED:1997", "PER-BASIC-ALIGNED:1997", "PER-CANONICAL-UNALIGNED:1997" et "PER-CANONICAL-ALIGNED:1997".

### 12.3 Attributs de variante ASN.1

Les chaînes suivantes sont des attributs de variante prédéfinis (normalisés). Ils n'ont une signification prédéfinie que quand ils sont appliqués conjointement avec des attributs de codage ASN.1 prédéfinis (voir § 11.2). Le traitement de ces attributs prédéfinis quand ils sont appliqués conjointement avec d'autres attributs ou appliqués à un objet TTCN-3 sans attribut est hors du domaine d'application de la présente Recommandation (voir Note 1):

- a) "format de longueur 1" signifie que la valeur indiquée ne doit être codée et décodée qu'au moyen du format court du champ de longueur (voir § 8.1.3/X.690 [6]) dans le cas des codages BER, CER et DER, ou qu'au moyen du déterminant de longueur d'un octet (voir § 10.9/X.691 [7]) dans le cas d'un format quelconque du codage PER (voir Note 2).
- b) "format de longueur 2" signifie que la valeur indiquée ne doit être codée et décodée qu'au moyen du format long du champ de longueur (voir § 8.1.3/X.690 [6]) dans le cas des codages BER, CER et DER, ou qu'au moyen du déterminant de longueur de deux octets (voir § 10.9/X.691 [7]) dans le cas d'un format quelconque du codage PER (voir Note 2).
- c) "format de longueur 3" signifie que la valeur indiquée ne doit être codée et décodée qu'au moyen du format indéfini du champ de longueur (voir § 8.1.3/X.690 [6]) dans le cas des codages BER, CER et DER.
- d) "REAL base 2" signifie que la valeur indiquée doit être codée ou comparée conformément à la forme de codage REAL binaire. Cet attribut ne peut être utilisé que sur des constantes, des variables ou des modèles. Quand il est appliqué à une sorte quelconque de groupage (par exemple, à des groupes ou à toute l'instruction d'importation), il n'aura d'effet que sur ces objets TTCN-3.
- e) Les expressions "type ASN1 isolé", "aligné à l'octet" et "arbitraire" signifient que le champ de valeur, basé sur un type ASN.1 EXTERNAL, sera codé selon le format de codage spécifié par l'attribut, ou ne sera comparé à la réception que si l'émetteur l'a codé selon le format de codage spécifié (voir § 8.18/X.690 [6]). Cet attribut ne sera appliqué qu'à des types EXTERNAL importés de la notation ASN.1, ainsi qu'à des constantes, sera des variables, des modèles ou des champs de modèle basés sur ces types; quand cet attribut est appliqué à une sorte quelconque de groupage (par exemple, à des groupes TTCN-3, à l'importation d'un type de définitions, aux déclarations d'importation de tout, etc.), il n'aura d'effet que sur les objets TTCN-3 basés sur les types ASN.1 EXTERNAL. Si les conditions fixées dans les § 8.18.6 à 8.18.8/X.690 [6] et l'attribut spécifié ne correspondent pas, cela provoquera une erreur d'exécution.
- f) "TeletexString" signifie que la valeur indiquée doit être codée et décodée comme le type ASN.1 TeletexString (voir § 8.20/X.690 [6] et 26/X.691 [7]).
- g) "VideotexString" signifie que la valeur indiquée doit être codée et décodée comme le type ASN.1 VideotexString (voir § 8.20/X.690 [6] et 26/X.691 [7]).
- h) "GraphicString" signifie que la valeur indiquée doit être codée et décodée comme le type ASN.1 GraphicString (voir § 8.20/X.690 [6] et 26/X.691 [7]).

- i) "GeneralString" signifie que la valeur indiquée doit être codée et décodée comme le type ASN.1 GeneralString (voir § 8.20/X.690 [6] et 26/X.691 [7]).

NOTE 1 – Ces attributs peuvent être réutilisés dans des règles de codage propres à l'implémentation avec une signification différente de celle qui est spécifiée dans le présent paragraphe, peuvent être ignorés ou une indication d'avertissement/d'erreur peut être donnée. Cependant, la stratégie à appliquer dépend de l'implémentation.

NOTE 2 – L'application de ces attributs de variante peut conduire à un codage ASN.1 non valide (par exemple si l'on applique la forme indéfinie de longueur à des valeurs de primitive dans les règles BER ou si l'on n'utilise pas le nombre minimal nécessaire d'octets de longueur). Cela est autorisé intentionnellement et les utilisateurs doivent attribuer avec précaution ces attributs de variante à des constantes, des variables, des modèles ou des champs de modèle utilisés pour la réception.

## Annexe A

### Formalisme BNF et sémantique statique additionnels

Pour prendre en charge l'utilisation de la notation ASN.1 dans la notation TTCN-3, la syntaxe TTCN-3 de l'Annexe A/Z.140 [1] doit être complétée par le formalisme BNF et les règles sémantiques spécifiés dans la présente annexe.

#### A.1 Prise en charge de la notation ASN.1

```
xxx. DefinitiveIdentifieur ::= Dot ObjectIdentifieurKeyword "{" DefinitiveObjIdComponentList "}"
xxx. ObjectIdentifieurKeyword ::= "objid"
xxx. DefinitiveObjIdComponentList ::= {DefinitiveObjIdComponent}+
xxx. DefinitiveObjIdComponent ::= NameForm |
                                DefinitiveNumberForm |
                                DefinitiveNameAndNumberForm
xxx. DefinitiveNumberForm ::= Number
xxx. DefinitiveNameAndNumberForm ::= Identifieur "(" DefinitiveNumberForm ")"
xxx. ObjectIdentifieurValue ::= ObjectIdentifieurKeyword "{" ObjIdComponentList "}"
xxx. ObjIdComponentList ::= {ObjIdComponent}+
xxx. ObjIdComponent ::= NameForm |
                       NumberForm |
                       NameAndNumberForm |
                       ReferencedValue
/* STATIC SEMANTICS - ReferencedValue doit être une valeur d'identificateur d'object */
xxx. NameForm ::= Identifieur
xxx. NumberForm ::= Number | ReferencedValue
/* STATIC SEMANTICS - ReferencedValue doit être une valeur d'entier non négative */
xxx. NameAndNumberForm ::= Identifieur "(" NumberForm ")"
```



## Annexe B

### Fonctions TTCN-3 prédéfinies

#### B.1 La fonction Décomposition

```
decomp (in objid inval, in integer index, in integer count) return objid
```

Cette fonction retourne une valeur d'identificateur d'objet contenant un fragment (séquence de composants) de la valeur d'identificateur d'objet d'entrée (inval). Le début du fragment est défini par le deuxième paramètre d'entrée (index). La valeur effective du paramètre 'index' doit être un entier non négatif. L'indexation commence à zéro, qui identifie le premier composant de la valeur d'identificateur d'objet entrée. Le troisième paramètre d'entrée (count) définit le nombre de composants dans la valeur objid retournée. La valeur effective du paramètre 'count' doit être un entier positif non nul. La somme des paramètres 'index' et 'count' doit être inférieure ou égale au nombre de composants de la valeur objid d'entrée moins 1.

EXEMPLE:

```
var objid v_etsiMobNet := objid{itu_t identified_organization etsi(0)
                               mobile_domain(0) umts_Network (1)}

decomp (v_etsiMobNet, 0, 2) // retourne {itu_t identified_organization}

decomp (v_etsiMobNet, 2, 3) // retourne {etsi(0) mobile_domain(0) umts_Network (1)}

decomp (v_etsiMobNet, 0, 0) // provoque une erreur car le nombre de composants à retourner
                             // doit être supérieur à 0

decomp (v_etsiMobNet, 0, 6) // provoque une erreur car la valeur objid d'entrée
                             // contient moins de 6 composants
```

## Annexe C

### Composants d'identificateur d'objet prédéfinis

La Rec. UIT-T X.660 [12] définit l'arborescence de composants d'identificateur d'objet indiquée ci-après. Seuls les composants d'identificateur d'objet définis dans la Rec. UIT-T X.660 [12] doivent utiliser la forme de nom (sans définir la valeur numérique du composant) dans des notations de valeurs d'identificateur d'objet. Ces composants prédéfinis n'ont de valeurs numériques spécifiées que lorsqu'ils sont utilisés à leurs positions prédéfinies. Les noms en *italique* étant réservés pour des raisons historiques, leur utilisation dans les codes TTCN-3 est déconseillée; il est toutefois recommandé que les utilitaires TTCN-3 puissent les reconnaître et les remplacer par la valeur numérique correcte.

NOTE – Les noms ci-dessous sont donnés conformément à la syntaxe TTCN-3, c'est-à-dire que les caractères tiret sont remplacés par des caractères soulignement.

```
itu_t(0), ccitt(0), itu_r(0)
  recommendation(0)
    a(1)
    d(4)
    e(5)
    f(6)
    g(7)
    h(8)
    i(9)
    j(10)
    k(11)
    l(12)
    m(13)
    n(14)
    o(15)
    p(16)
    q(17)
    r(18)
    s(19)
    t(20)
    u(21)
    v(22)
    x(24)
    y(25)
    z(26)
  question(1)
  administration(2)
  network_operator(3)
  identified_organization(4)
  r_recommendation(5)
iso(1)
  standard(0)
  registration_authority(1)
  member_body(2)
  identified_organization(3)
joint_iso_itu_t(2), joint_iso_ccitt(2)
```

## BIBLIOGRAPHIE

- ISO/CEI 6429:1992, *Technologies de l'information – Fonctions de commande pour les jeux de caractères codés.*
- Recommandation UIT-T T.50 (1992), *Alphabet international de référence (ancien alphabet international n° 5 ou A15) – Technologies de l'information – Jeux de caractères codés à 7 bits pour l'échange d'informations.*
- Recommandation UIT-T X.208, *Spécification de la syntaxe abstraite numéro un (ASN.1).*
- ISO/CEI 8859-1:1998, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 1: Alphabet latin n° 1.*
- Recommandation UIT-T T.61 (1993), *Répertoire de caractères et jeux de caractères codés pour le service international télétext.*
- Un répertoire d'identificateurs d'objets (OID) est accessible librement à l'adresse suivante: <http://oid.elibel.tm.fr>.

NOTE – Il est prévu de transférer le contenu du site Web ci-dessus sur le site Web de l'UIT-T.





## SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
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	Gestion des télécommunications y compris le RGT et maintenance des réseaux
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, communication entre systèmes ouverts et sécurité
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de prochaine génération
<b>Série Z</b>	<b>Langages et aspects généraux logiciels des systèmes de télécommunication</b>