



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.667

(09/2004)

SÉRIE X: RÉSEAUX DE DONNÉES, COMMUNICATION
ENTRE SYSTÈMES OUVERTS ET SÉCURITÉ

Réseautage OSI et aspects systèmes – Dénomination,
adressage et enregistrement

**Technologies de l'information – Interconnexion
des systèmes ouverts – Procédures
opérationnelles des organismes
d'enregistrement de l'OSI: génération et
enregistrement des identificateurs uniques
universels (UUID) et utilisation de ces
identificateurs comme composants
d'identificateurs d'objets ASN.1**

Recommandation UIT-T X.667

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

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.379
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.889
Applications génériques de l'ASN.1	X.890–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999
SÉCURITÉ DES TÉLÉCOMMUNICATIONS	X.1000–

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

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

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.379
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.889
Applications génériques de l'ASN.1	X.890–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999
SÉCURITÉ DES TÉLÉCOMMUNICATIONS	X.1000–

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

**Technologies de l'information – Interconnexion des systèmes ouverts –
Procédures opérationnelles des organismes d'enregistrement
de l'OSI: génération et enregistrement des identificateurs
uniques universels (UUID) et utilisation de ces identificateurs
comme composants d'identificateurs d'objets ASN.1**

Résumé

La présente Recommandation | Norme internationale spécifie des procédures pour la génération et l'enregistrement des identificateurs UUID, ainsi que pour l'utilisation de ces identificateurs comme composants d'identificateurs d'objets ASN.1 (OID) au-dessous de l'arc `{joint-iso-itu-t uuid(25)}`.

Source

La Recommandation UIT-T X.667 a été approuvée le 13 septembre 2004 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 9834-8.

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 responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 2005

Tous droits réservés. Aucune partie de cette publication ne peut être regénérée, 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 normatives..... 1
2.1	Recommandations Normes internationales identiques 1
2.2	Autres références normatives 1
3	Termes et définitions 2
3.1	Notation ASN.1 2
3.2	organismes d'enregistrement 2
3.3	Termes relatifs au réseau 2
3.4	Définitions additionnelles 2
4	Abréviations 3
5	Notation 3
6	Structure et représentation des identificateurs UUID 3
6.1	Structure de champ d'un identificateur UUID 3
6.2	Représentation binaire..... 4
6.3	Représentation par valeur unique d'entier 4
6.4	Représentation hexadécimale 4
6.5	Syntaxe formelle de la représentation hexadécimale 4
7	Utilisation d'un identificateur UUID afin de former un identificateur OID 5
8	Utilisation d'un identificateur UUID afin de former un nom URN 5
9	Règles de comparaison et d'ordonnancement des identificateurs UUID 5
10	Validation..... 6
11	Les bits de variante..... 6
12	Utilisation des champs d'UUID et ordre des octets pendant la transmission..... 6
12.1	Généralités 6
12.2	Version 7
12.3	Temps 8
12.4	Séquence d'horloge 8
12.5	Nœud 8
13	Réglage des champs d'un identificateur UUID fondé sur le temps 9
14	Réglage des champs d'un identificateur UUID fondé sur le nom 9
15	Réglage des champs d'un identificateur UUID fondé sur un nombre aléatoire 10
16	Enregistrement des identificateurs UUID et utilisation de ces identificateurs comme composants d'identificateur OID 10
16.1	L'arborescence d'identificateurs OID en notation ASN.1 10
16.2	Désignation des organismes d'enregistrement..... 11
16.3	Honoraires..... 11
16.4	Procédures d'enregistrement..... 12
16.4.1	Demande d'enregistrement d'un identificateur UUID 12
16.4.2	Processus de confirmation..... 12
16.4.3	Contenu de la demande 12
16.5	Maintenance d'un registre accessible par protocole IP..... 12
Annexe A	– Algorithmes pour la génération efficace d'identificateurs UUID fondés sur le temps..... 13
A.1	Algorithme de base 13
A.2	Lecture en mémoire stable 13
A.3	Résolution de l'horloge-système..... 14
A.4	Ecriture en mémoire stable..... 14
A.5	Partage d'état entre processus..... 14
Annexe B	– Propriétés des identificateurs UUID fondés sur le nom..... 15
Annexe C	– Génération de nombres aléatoires dans un système..... 16

	<i>Page</i>
Annexe D – Exemple de réalisation	17
D.1 Fichiers fournis	17
D.2 Le fichier <code>copyrt.h</code>	17
D.3 Le fichier <code>uuid.h</code>	17
D.4 Le fichier <code>uuid.c</code>	18
D.5 Le fichier <code>sysdep.h</code>	21
D.6 Le fichier <code>sysdep.c</code>	22
D.7 Le fichier <code>utest.c</code>	23
D.8 Exemple de sortie du fichier <code>utest</code>	24
D.9 Quelques identificateurs d'espace nominatif	24
BIBLIOGRAPHIE.....	25

Introduction

La présente Recommandation | Norme internationale normalise la génération, ainsi que l'enregistrement facultatif, d'identificateurs uniques universels (UUID).

Les identificateurs UUID forment une chaîne de 16 octets (soit 128 bits) qui peut être interprétée comme un codage d'entier non signé. La valeur d'entier résultante peut être utilisée comme un arc de l'arbre d'identificateur OID au-dessous de l'arc `{joint-iso-itu-t uuid(25)}`, ce qui permet aux utilisateurs de générer des identificateurs OID sans aucune procédure d'enregistrement.

Les identificateurs UUID sont également appelés *identificateurs mondialement uniques* (GUID), mais ce terme n'est pas utilisé dans la présente Recommandation | Norme internationale. Les identificateurs UUID ont été initialement utilisés dans le Système de calcul de réseau (NCS) [1] et ultérieurement dans l'Environnement de calcul réparti (DCE) [2] de la Fondation des logiciels ouverts. L'ISO/CEI 11578 [3] contient une brève définition de quelques-uns (mais non de la totalité) des formats d'identificateur UUID spécifiés dans la présente Recommandation | Norme internationale. La spécification régénérée dans la présente Recommandation | Norme internationale est compatible avec toutes ces spécifications antérieures.

Les identificateurs UUID qui sont des éléments constitutifs d'un identificateur OID sont exprimés par notation ASN.1 de valeurs d'entier en base décimale. Mais pour tous les autres objectifs d'affichage, il est plus courant de les représenter au moyen de chiffres hexadécimaux avec un trait d'union séparant les différents champs contenus dans un identificateur UUID de 16 octets. Cette représentation est définie dans la présente Recommandation | Norme internationale.

S'il a été généré conformément à un des mécanismes définis dans la présente Recommandation | Norme internationale, un identificateur UUID est garanti différent de tous les autres identificateurs UUID générés avant l'an 3603 de notre ère, ou est extrêmement susceptible d'en être différent (selon le mécanisme choisi).

Aucune autorité centralisée n'est appelée à administrer les identificateurs UUID mais un enregistrement central des identificateurs UUID générés automatiquement est assuré, avec génération automatique (au moyen de l'algorithme défini dans la présente Recommandation | Norme internationale) et enregistrement automatique des identificateurs UUID. Les UUID à génération centralisée sont garantis différents de tous les autres UUID à génération centralisée. Les UUID enregistrés sont garantis différents de tous les autres UUID enregistrés.

Un identificateur UUID peut être utilisé à de multiples fins, de l'étiquetage d'objets ayant une durée de vie extrêmement brève jusqu'à l'identification fiable d'objets très persistants dans un réseau, particulièrement (mais non nécessairement) en faisant partie d'une valeur d'identificateur d'objet ASN.1 (OID) ou d'un nom uniforme de ressource (URN).

L'algorithme de génération d'identificateurs UUID spécifié dans la présente Recommandation | Norme internationale prend en charge de très hauts débits d'affectation: 10 millions par seconde et par machine si nécessaire, de telle sorte que les identificateurs UUID peuvent également être utilisés comme identificateurs de transaction. Une annexe informative fournit un programme écrit en langage C qui générera des identificateurs UUID conformément à la présente Recommandation | Norme internationale.

Trois algorithmes sont spécifiés pour la génération d'identificateurs UUID uniques au moyen de différents mécanismes, afin d'en garantir l'unicité. Ces algorithmes génèrent différentes versions d'un identificateur UUID.

Le premier (et le plus courant) mécanisme construit la version dite "fondée sur le temps". Ces identificateurs UUID peuvent être générés au débit de 10 millions par seconde. Concernant les identificateurs UUID générés dans un seul système informatique, un marqueur temporel de 60 bits, utilisé comme valeur d'horloge avec une granularité de 100 ns et fondé sur le Temps universel coordonné (UTC), sert à garantir l'unicité sur une période d'approximativement 1 600 ans. Concernant les identificateurs UUID générés avec le même marqueur temporel par différents systèmes, l'unicité est obtenue au moyen d'adresses de commande d'accès au support (MAC) de 48 bits, spécifiées dans l'ISO/CEI 8802-3 (cette adresse est utilisée comme valeur nodale). (Ces adresses sont habituellement déjà disponibles dans la plupart des systèmes mis en réseau, mais sont sinon attribuables par l'organisme d'enregistrement de l'IEEE pour les adresses MAC – voir [4].) D'autres moyens de génération de valeurs temporelles et de valeurs nodales sont spécifiés pour la version "fondée sur le temps" si le temps UTC n'est pas disponible dans un système ou s'il n'y a aucune adresse MAC disponible.

Le second mécanisme génère un identificateur UUID unique qui est une version fondée sur le nom dans laquelle un hachage cryptographique sert à générer les 128 bits de la valeur d'identificateur UUID à partir d'un nom (textuel) mondialement univoque.

Le troisième mécanisme fait appel à la génération d'un nombre pseudo-aléatoire ou vraiment aléatoire afin de générer la plupart des bits contenus dans la valeur de 128 bits.

Le paragraphe 5 spécifie la notation utilisée pour le nommage de l'ordre des octets et de l'ordre des bits, ainsi que pour la spécification de l'ordre de transmission.

Le paragraphe 6 spécifie la structure d'un identificateur UUID et sa représentation en notation binaire, en notation hexadécimale, ou en valeur unique d'entier.

Les § 7 et 8 spécifient respectivement l'utilisation d'un identificateur UUID faisant partie d'un identificateur OID ou faisant partie d'un nom URN.

Le paragraphe 9 spécifie des règles permettant de comparer les identificateurs UUID afin de vérifier leur égalité ou afin d'offrir une relation d'ordonnancement entre deux UUID.

Le paragraphe 10 étudie la possibilité de vérifier la validité d'un identificateur UUID. En général, les identificateurs UUID ont peu de redondance et il y a peu de possibilités de vérifier leur validité. Cependant, si un identificateur UUID est accepté pour enregistrement, alors il est garanti différent de tous les autres identificateurs UUID enregistrés.

Le paragraphe 11 décrit l'utilisation historique de quelques bits contenus dans l'identificateur UUID afin de définir différentes variantes du format d'identificateur UUID. Il spécifie également la valeur de ces bits pour les identificateurs UUID définis conformément à la présente Recommandation | Norme internationale.

Le paragraphe 12 spécifie l'utilisation des champs d'un identificateur UUID dans les différentes versions qui sont définies (versions "fondée sur le temps", "fondée sur le nom" et "fondée sur un nombre aléatoire"). Il définit également l'ordre des octets pendant la transmission.

Le paragraphe 13 spécifie le réglage des champs d'un identificateur UUID fondé sur le temps.

Le paragraphe 14 spécifie le réglage des champs d'un identificateur UUID fondé sur le nom.

Le paragraphe 15 spécifie le réglage des champs d'un identificateur UUID fondé sur un nombre aléatoire.

Le paragraphe 16 traite du fonctionnement d'un organisme d'enregistrement d'identificateurs UUID, permettant leur enregistrement central et fournissant des garanties d'unicité.

Toutes les annexes sont informatives.

L'Annexe A décrit divers algorithmes pour la génération efficace d'identificateurs UUID fondés sur le temps.

L'Annexe B étudie les propriétés qu'un identificateur UUID fondé sur le nom devrait posséder, affectant la sélection des espaces nominatifs à utiliser lors de la génération de tels identificateurs UUID.

L'Annexe C fournit des orientations sur les mécanismes qui peuvent servir à générer des nombres aléatoires dans un système informatique.

L'Annexe D contient un programme complet en langage de programmation C qui peut servir à générer des identificateurs UUID.

**NORME INTERNATIONALE
RECOMMANDATION UIT-T**

**Technologies de l'information – Interconnexion des systèmes ouverts –
Procédures opérationnelles des organismes d'enregistrement
de l'OSI: génération et enregistrement des identificateurs
uniques universels (UUID) et utilisation de ces identificateurs
comme composants d'identificateurs d'objets ASN.1**

1 Domaine d'application

La présente Recommandation | Norme internationale spécifie le format et les règles de génération qui permettent aux utilisateurs de générer des identificateurs de 128 bits qui sont soit garantis mondialement uniques ou sont mondialement uniques avec une haute probabilité.

Les identificateurs UUID générés conformément à la présente Recommandation | Norme internationale sont appropriés soit pour usage temporaire avec génération d'un nouvel identificateur UUID toutes les 100 ns, ou comme identificateurs permanents.

La présente Recommandation | Norme internationale est déduite de spécifications antérieures non normalisées concernant les identificateurs UUID et leur génération. Elle est techniquement identique à ces spécifications antérieures.

La présente Recommandation | Norme internationale spécifie les procédures de fonctionnement d'un organisme d'enregistrement pour identificateurs UUID accessible par protocole IP.

La présente Recommandation | Norme internationale spécifie et permet également l'utilisation d'identificateurs UUID (enregistrés ou non enregistrés) comme composants d'identificateur OID au-dessous de l'arc `{joint-iso-itu-t uuid(25)}`. Cet usage permet aux utilisateurs de générer des identificateurs OID sans aucune procédure d'enregistrement.

La présente Recommandation | Norme internationale spécifie et permet également l'utilisation des identificateurs UUID (enregistrés ou non enregistrés) afin de former un nom URN.

2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT tient à jour une liste des Recommandations de l'UIT-T en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.660 (2004) | ISO/CEI 9834-1:2005, *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*.
- Recommandation UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Technologies de l'information – Notation de syntaxe abstraite numéro un (ASN.1): Spécification de la notation de base*.

2.2 Autres références normatives

- ISO/CEI 8802-3:2000, *Technologies de l'information – Télécommunications et échange d'information entre systèmes – Réseaux locaux et métropolitains – Prescriptions spécifiques – Partie 3: Accès multiple par surveillance du signal et détection de collision (CSMA/CD) et spécifications pour la couche physique*.
- ISO/CEI 10118-3:2004, *Technologies de l'information – Techniques de sécurité – Fonctions de brouillage – Partie 3: Fonctions de brouillage dédiées*.
- ISO/CEI 10646:2003, *Technologies de l'information – Jeu universel de caractères codés sur plusieurs octets (JUC)*.

- FIPS PUB 180-2:2002, *Federal Information Processing Standards Publication, Secure Hash Standard (SHS) (Publication fédérale de normes de traitement de l'information – Norme de hachage sécurisé (SHS))*.
- IETF RFC 1321 (1992), *The MD5 Message-Digest Algorithm (L'algorithme MD5 de condensation de message)*.
- IETF RFC 2141 (1997), *URN Syntax (Syntaxe de nom uniforme de ressource)*.

3 Termes et définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent.

3.1 Notation ASN.1

La présente Recommandation | Norme internationale utilise les termes suivants, qui sont définis dans la Rec. UIT-T X.680 | ISO/CEI 8824-1:

- a) temps universel coordonné (UTC);
- b) identificateur d'objet (ASN.1).

3.2 Organismes d'enregistrement

La présente Recommandation | Norme internationale utilise les termes suivants, qui sont définis dans la Rec. UIT-T X.660 | ISO/CEI 9834-1:

- a) arbre d'identificateurs d'objet (ou arbre d'identificateurs OID);
- b) enregistrement;
- c) organisme d'enregistrement;
- d) procédures d'enregistrement.

3.3 Termes relatifs au réseau

La présente Recommandation | Norme internationale utilise le terme suivant, qui est défini dans l'ISO/CEI 8802-3:

- adresse MAC.

3.4 Définitions additionnelles

3.4.1 nombre aléatoire de qualité cryptographique: nombre aléatoire ou pseudo-aléatoire généré par un mécanisme qui garantit une dispersion suffisante de valeurs générées de façon répétitive pour que ces valeurs soient acceptables pour utilisation dans des travaux cryptographiques (et qui est utilisé dans de tels travaux).

3.4.2 version fondée sur le nom: identificateur UUID qui est généré par hachage cryptographique d'un nom contenu dans un espace nominatif et au moyen d'un nom contenu dans cet espace nominatif.

3.4.3 espace nominatif: système de génération de noms d'objets qui garantit une identification univoque dans cet espace nominatif.

NOTE – Exemples d'espace nominatifs: le système d'adressage par nom de domaine dans un réseau, les noms URN, les identificateurs OID, les noms distinctifs d'annuaire (voir [5]) et les mots réservés dans un langage de programmation.

3.4.4 version fondée sur un nombre aléatoire: identificateur UUID qui est généré au moyen d'un nombre aléatoire ou pseudo-aléatoire.

3.4.5 variante normalisée des identificateurs UUID: variante des formats possibles d'identificateur UUID qui est spécifiée par la présente Recommandation | Norme internationale.

NOTE – Historiquement, il y a eu d'autres spécifications de format d'identificateur UUID qui différaient de la variante spécifiée dans la présente Recommandation | Norme internationale. Les identificateurs UUID générés conformément à tous ces formats de variante sont tous distincts.

3.4.6 version "fondée sur le temps": identificateur UUID dans lequel l'unicité est obtenue au moyen d'une adresse de commande MAC afin d'identifier un système et une valeur d'horloge fondée sur le temps UTC actuel.

3.4.7 identificateur unique universel (UUID): valeur de 128 bits générée conformément à la présente Recommandation | Norme internationale ou conformément à quelques spécifications historiques et fournissant des valeurs uniques entre systèmes et dans le temps (voir également § 3.4.5).

4 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale, les abréviations suivantes s'appliquent:

ASN.1	notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
GUID	identificateur mondialement unique (<i>globally unique identifier</i>)
IEEE	Institut des ingénieurs électriciens et électroniciens
MAC	commande d'accès au support (<i>media access control</i>)
MD5	algorithme de condensation de message 5 (<i>message digest algorithm 5</i>)
OID	identificateur d'objets ASN.1 (<i>ASN.1 object identifier</i>)
RA	organisme d'enregistrement (<i>registration authority</i>)
SHA-1	algorithme de hachage sécurisé 1 (<i>secure hash algorithm 1</i>)
URL	localisateur uniforme de ressource (<i>uniform resource locator</i>)
URN	nom uniforme de ressource (<i>uniform resource name</i>)
UTC	temps universel coordonné (<i>coordinated universal time</i>)
UUID	identificateur unique universel (<i>universally unique identifier</i>)

5 Notation

5.1 La présente Recommandation | Norme internationale spécifie une séquence d'octets pour un identificateur UUID utilisant les termes "premier" et "dernier". Le premier octet est également appelé "octet 15" et le dernier "octet 0".

5.2 Les bits contenus dans un identificateur UUID sont également numérotés du "bit 127" au "bit 0", le bit 127 étant le bit de poids fort de l'octet 15 et le bit 0 étant le bit de poids faible de l'octet 0.

5.3 Quand des figures et des tableaux sont utilisés dans la présente Recommandation | Norme internationale, l'octet de poids fort (et le bit de poids fort) sont affichés à gauche de la page. Cela correspond à un ordre de transmission des octets dans lequel les octets situés à gauche sont transmis en premier.

5.4 Un certain nombre de valeurs utilisées dans la présente spécification sont exprimées par la valeur d'un entier non signé d'une longueur binaire donnée (par exemple N). Les N bits de la valeur d'entier non signé sont numérotés du "bit N-1" au "bit 0", le bit N-1 étant le bit de poids fort et le bit 0 étant le bit de poids faible.

5.5 Ces notations ne sont utilisées qu'aux fins de la présente spécification. Les représentations mises en mémoire informatique ne sont pas normalisées et dépendent de l'architecture du système.

6 Structure et représentation des identificateurs UUID

6.1 Structure de champ d'un identificateur UUID

6.1.1 Un identificateur UUID est spécifié comme une séquence ordonnée de six champs concaténés, qui sont nommés comme suit:

- a) le champ "TimeLow";
- b) le champ "TimeMid";
- c) le champ "VersionAndTimeHigh";
- d) le champ "VariantAndClockSeqHigh";
- e) le champ "ClockSeqLow";
- f) le champ "Node".

6.1.2 Les champs d'identificateur UUID sont définis de façon à avoir un poids dans l'ordre énuméré ci-dessus, "TimeLow" étant le champ de poids fort (le bit 31 du champ "TimeLow" est le bit 127 de l'identificateur UUID) et "Node" étant le champ de poids faible (le bit 0 du champ "Node" est le bit 0 de l'identificateur UUID).

6.1.3 Le contenu de ces champs d'identificateur UUID est spécifié sous la forme d'une valeur d'entier non signé indiquant la version, la variante, l'heure, la séquence d'horloge et le nœud (chaque valeur ayant une longueur binaire fixe). Le réglage de ces valeurs est spécifié au § 12 et leur mappage dans les champs d'identificateur UUID est spécifié dans le § 12.1.

NOTE – Comme l'implique une partie des noms de quelques-uns des champs d'identificateur UUID (par exemple, TimeLow, TimeMid et TimeHigh), l'ordre séquentiel des bits contenus dans un identificateur UUID (du bit 127 au bit 0) qui a été construit à partir d'une valeur particulière d'entier non signé (par exemple à partir des bits 59 à 0 de la valeur temporelle) n'est pas le même que l'ordre séquentiel des bits contenus dans cette valeur d'entier non signé, cela pour des raisons historiques.

6.2 Représentation binaire

6.2.1 Un identificateur UUID doit être représenté en format binaire par 16 octets, formés par la concaténation des codages de longueur fixe des entiers non signés de chacun de ses champs, ceux-ci étant contenus dans un ou plusieurs octets. Le nombre d'octets à utiliser pour chaque champ doit être le suivant:

- a) champ "TimeLow": quatre octets;
- b) champ "TimeMid": deux octets;
- c) champ "VersionAndTimeHigh": deux octets;
- d) champ "VariantAndClockSeqHigh": un octet;
- e) champ "ClockSeqLow": un octet;
- f) champ "Node": six octets.

NOTE – Cet ordre des champs d'UUID est la représentation courante dans un système informatique et en représentation textuelle hexadécimale (voir § 6.4).

6.2.2 Le bit de poids fort du codage d'entier non signé de chaque champ d'identificateur UUID doit être le bit de poids fort de son premier octet (l'octet N, qui est l'octet de poids fort) et le bit de poids faible du codage d'entier non signé doit être le bit de poids faible de son dernier octet (l'octet 0, qui est l'octet de poids faible).

6.2.3 Les champs d'identificateur UUID doivent être concaténés dans l'ordre de leur poids (voir § 6.1.2) avec le champ de poids fort en premier et le champ de poids faible en dernier.

6.3 Représentation par valeur unique d'entier

Les identificateurs UUID peuvent être représentés par une valeur unique d'entier. Afin d'obtenir la valeur unique d'entier de l'identificateur UUID, les 16 octets de la représentation binaire doivent être traités comme un codage d'entier non signé, le bit de poids fort du codage d'entier étant le bit de poids fort (bit 7) du premier des seize octets (octet 15) et le bit de poids faible étant le bit de poids faible (bit 0) du dernier des seize octets (octet 0).

NOTE – La valeur unique d'entier est utilisée quand l'identificateur UUID fait partie d'un identificateur OID comme spécifié dans le § 7.

6.4 Représentation hexadécimale

En format hexadécimal, les octets du format binaire doivent être représentés par une chaîne de chiffres hexadécimaux, avec deux chiffres hexadécimaux pour chaque octet du format binaire, le premier étant la valeur des quatre bits de poids fort de l'octet 15, le second étant la valeur des quatre bits de poids faible de l'octet 15 et ainsi de suite, le dernier chiffre étant la valeur des bits de poids faible de l'octet 0 (voir § 6.5). Un caractère TIRET-MOINS (45) (voir ISO/CEI 10646) doit être inséré entre les représentations hexadécimales de chaque paire de champs adjacents, sauf entre le champ "VariantAndClockSeqHigh" et le champ "ClockSeqLow" (voir l'exemple dans le § 8).

6.5 Syntaxe formelle de la représentation hexadécimale

6.5.1 La définition formelle de la syntaxe de représentation hexadécimale des identificateurs UUID est spécifiée au moyen de la notation BNF étendue qui est définie au § 5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, sauf qu'il ne doit y avoir aucun espace blanc entre les items lexicaux.

6.5.2 L'item lexical "hexdigit" est utilisé dans la spécification BNF et est défini comme suit:

Nom de l'item lexical – hexdigit

Un item "hexdigit" doit consister exactement d'un des caractères suivants:

A B C D E F a b c d e f 0 1 2 3 4 5 6 7 8 9

6.5.3 La représentation hexadécimale d'un identificateur UUID doit être la génération "UUID" suivante:

UUID ::=
TimeLow
"-" TimeMid
"-" VersionAndTimeHigh

"-" VariantAndClockSeqHigh ClockSeqLow

"-" Node

TimeLow ::=
HexOctet HexOctet HexOctet HexOctet

TimeMid ::=
HexOctet HexOctet

VersionAndTimeHigh ::=
HexOctet HexOctet

VariantAndClockSeqHigh ::=
HexOctet

ClockSeqLow ::=
HexOctet

Node ::=
HexOctet HexOctet HexOctet HexOctet HexOctet HexOctet

HexOctet ::=
hexdigit hexdigit

6.5.4 En génération logicielle, la représentation hexadécimale d'un identificateur UUID ne doit pas utiliser de lettres majuscules.

NOTE – Il est recommandé que la représentation hexadécimale utilisée dans tous les formats lisibles par l'œil humain soit limitée aux lettres minuscules. Le traitement logiciel de cette représentation est, cependant, appelé à accepter les lettres aussi bien majuscules que minuscules, comme spécifié dans le § 6.5.2.

7 Utilisation d'un identificateur UUID afin de former un identificateur OID

Un identificateur OID formé au moyen d'un identificateur UUID doit avoir la syntaxe suivante:

{joint-iso-itu-t uuid(25) <uuid-single-integer-value>}

où **<uuid-single-integer-value>** est la valeur unique d'entier de l'identificateur UUID spécifiée au § 6.3.

NOTE – Le § 16 fournit d'autres détails sur l'utilisation d'un identificateur UUID afin de former un identificateur OID et le § 16.1.3 fournit d'importantes indications sur l'unicité des identificateurs OID générés de cette façon.

8 Utilisation d'un identificateur UUID afin de former un nom URN

Un nom URN (voir IETF RFC 2141) formé au moyen d'un identificateur UUID doit être la chaîne "urn:uuid:" suivie par la représentation hexadécimale d'un identificateur UUID défini au § 6.4.

EXEMPLE – Ce qui suit est un exemple de la représentation concaténée d'un identificateur UUID sous forme de nom URN:

urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6

NOTE – Une variante du format de nom URN (voir [6]) est disponible, mais n'est pas recommandée pour les noms URN générés au moyen d'identificateurs UUID. Ce format en variante utilise la valeur unique d'entier de l'identificateur UUID spécifiée au § 6.3 et représente l'exemple ci-dessus sous la forme suivante:

" urn:oid:2.25.329800735698586629295641978511506172918".

9 Règles de comparaison et d'ordonnement des identificateurs UUID

9.1 Afin de comparer une paire d'identificateurs UUID, les valeurs des champs correspondants (voir § 6.1) de chaque UUID sont comparées par ordre de poids (voir § 6.1.2). Deux identificateurs UUID sont égaux si et seulement si tous les champs correspondants sont égaux.

NOTE 1 – Cet algorithme, permettant de comparer deux identificateurs UUID, est équivalent à la comparaison des valeurs exprimées par un seul entier, qui sont spécifiées au § 6.3.

NOTE 2 – Cette comparaison utilise les champs physiques qui sont spécifiés au § 6.1.1 et non pas les valeurs qui sont énumérées au § 6.1.3 et spécifiées au § 12 (heure, séquence d'horloge, variante, version et nœud).

9.2 L'identificateur UUID est considéré comme plus grand qu'un autre identificateur UUID s'il possède une plus grande valeur dans le champ de poids fort par lequel ces deux identificateurs diffèrent.

9.3 Dans un ordonnancement lexicographique de la représentation hexadécimale des identificateurs UUID (voir § 6.4), un plus grand identificateur UUID doit suivre un plus petit identificateur UUID.

10 Validation

En dehors de la détermination de la question de savoir si les bits de variante sont réglés correctement et si la valeur temporelle utilisée dans un identificateur UUID fondé sur le temps est située dans le futur (et donc n'est pas encore assignable), il n'y a aucun mécanisme permettant de déterminer si un identificateur UUID est valide dans un sens réel ou un autre, car toutes les valeurs possibles peuvent apparaître par ailleurs.

11 Les bits de variante

11.1 Les bits de variante sont les trois bits de poids fort (bits 7, 6 et 5) de l'octet 7, qui est l'octet de poids fort du champ "VariantAndClockSeqHigh".

11.2 Tous les identificateurs UUID conformes à la présente Recommandation | Norme internationale doivent avoir des bits de variante avec le bit 7 de l'octet 7 réglé à 1 et le bit 6 de l'octet 7 réglé à 0. Le bit 5 de l'octet 7 est le bit de poids fort de la séquence d'horloge et doit être réglé conformément au § 12.4.

NOTE – Le bit 5 est énuméré ici comme un bit de variante parce que sa valeur distingue les formats historiques. A strictement parler, il ne fait pas partie de la valeur de variante au sens de la présente Recommandation | Norme internationale, qui utilise seulement deux bits pour la variante.

11.3 Le Tableau 1 indique, pour information, l'utilisation d'autres valeurs des bits de variante.

Tableau 1 – Utilisation des bits de variante

Bit 7	Bit 6	Bit 5	Description
0	–	–	Champ réservé afin d'offrir la rétrocompatibilité avec le système NCS
1	0	–	La variante spécifiée dans la présente Recommandation Norme internationale
1	1	0	Champ réservé afin d'offrir la rétrocompatibilité avec le système de Microsoft Corporation
1	1	1	Champ réservé pour future utilisation par la présente Recommandation Norme internationale

12 Utilisation des champs d'UUID et ordre des octets pendant la transmission

12.1 Généralités

12.1.1 Le Tableau 2 positionne et résume l'utilisation des divers champs d'identificateur UUID exprimés en représentation binaire.

Tableau 2 – Position et utilisation des champs d'UUID

Champ	N° d'octet dans l'UUID	Description
"TimeLow"	15-12	Bits de poids faible de la valeur temporelle (32 bits)
"TimeMid"	11-10	Bits médians de la valeur temporelle (16 bits)
"VersionAndTimeHigh"	9-8	Bits de version (4 bits) suivis par les bits de poids fort de la valeur temporelle (12 bits)
"VariantAndClockSeqHigh"	7	Bits de variante (2 bits) suivis par les bits de poids fort de la séquence d'horloge (6 bits)
"ClockSeqLow"	6	Bits de poids faible de la séquence d'horloge (8 bits)
"Node"	5-0	Nœud (voir § 12.5) (48 bits)

12.1.2 La position des champs d'identificateur UUID exprimés en représentation binaire est illustrée dans la Figure 1.

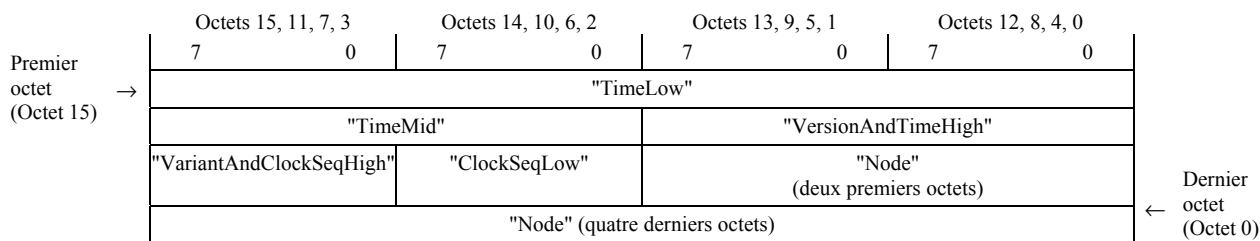


Figure 1 – Position des champs d'UUID dans la représentation binaire

12.1.3 Il est recommandé que la représentation binaire soit utilisée pour la transmission par un mécanisme de communication, avec les seize octets de la représentation binaire transmis comme un ensemble contigu de seize bits, le premier octet (octet 15) précédant le dernier octet (octet 0) dans la transmission.

NOTE 1 – L'ordre des bits contenus dans un octet est déterminé par la spécification du mécanisme de communication.

NOTE 2 – L'utilisation de seize octets consécutifs pour la transmission d'un identificateur UUID, dans l'ordre spécifié ci-dessus, est recommandée; mais les spécifications relatives au protocole peuvent choisir d'autres moyens de transfert d'un identificateur UUID, y compris la fragmentation ou la seule transmission de portions de l'identificateur UUID (telles que les parties qui contribuent à la valeur temporelle).

12.2 Version

12.2.1 Les trois moyens en variante de génération d'un identificateur UUID ("fondé sur le temps", "fondé sur le nom" et "fondé sur un nombre aléatoire") sont identifiés et distingués par les quatre bits de poids fort du champ "VersionAndTimeHigh" (bits 7 à 4 de l'octet 9 de l'identificateur UUID). Les identificateurs UUID générés au moyen de ces différents mécanismes sont appelés "identificateurs UUID des versions différentes".

NOTE – La description de ce type d'identificateur comme une "version différente d'identificateur UUID" prête légèrement à confusion, mais cette désignation est utilisée pour des raisons historiques. Il n'y a aucun concept de "numéro de version" traditionnel pour les formats d'identificateur UUID lorsque les nouvelles versions peuvent être définies comme une révision de la présente Recommandation | Norme internationale. Tout nouveau format d'identificateur UUID qui sera nécessaire ultérieurement sera identifié par une valeur différente des bits de variante.

12.2.2 Le Tableau 3 énumère les "versions d'identificateur UUID" actuellement définies, au moyen des quatre premiers bits du champ "VersionAndTimeHigh" (bits 7 à 4 de l'octet 9 de l'identificateur UUID). Il attribue également une valeur d'entier de "version" à chaque combinaison de bits.

NOTE – La valeur de version égale à 2 n'est pas utilisée pour des raisons de compatibilité avec des définitions historiques de l'identificateur UUID. Les valeurs de version égales à 0 et comprises entre 6 et 15 sont réservées pour utilisation future.

Tableau 3 – Versions d'identificateur UUID actuellement définies

Bit 7	Bit 6	Bit 5	Bit 4	Valeur de version	Description
0	0	0	1	1	La version "fondée sur le temps" qui est spécifiée dans la présente Recommandation Norme internationale (voir § 13)
0	0	1	0	2	Champ réservé pour la version de sécurité de l'environnement DCE, avec imbrication de l'identificateur UUID d'interface POSIX
0	0	1	1	3	La version fondée sur le nom qui est spécifiée dans la présente Recommandation Norme internationale avec hachage MD5 (voir § 14)
0	1	0	0	4	La version fondée sur un nombre aléatoire qui est spécifiée dans cette Recommandation Norme internationale (voir § 15)
0	1	0	1	5	La version fondée sur le nom qui est spécifiée dans la présente Recommandation Norme internationale avec hachage SHA-1 (voir § 14)

12.3 Temps

12.3.1 Le temps doit être représenté par une valeur de 60 bits.

NOTE – L'appellation "Temps" est appropriée pour la version "fondée sur le temps" d'un identificateur UUID (version 1), mais est également utilisée pour le contenu de la valeur correspondante dans les autres versions d'un identificateur UUID (versions 3 et 4).

12.3.2 Pour la version "fondée sur le temps" d'un identificateur UUID, le temps doit être un décompte des intervalles de 100 ns en temps universel coordonné (UTC) qui se sont écoulés depuis l'heure zéro (minuit) du 15 octobre 1582 (date de la réforme grégorienne du calendrier chrétien).

NOTE 1 – Avant l'établissement du Bureau international de l'heure (BIH), chaque minute contenait précisément soixante secondes. Depuis lors, des secondes intercalaires sont apparues quand nécessaire, augmentant (ou réduisant, le cas échéant) le nombre de secondes par année.

NOTE 2 – Les systèmes portatifs peuvent avoir des problèmes de détermination du temps UTC car ils sont souvent calés sur l'heure locale de leur base de rattachement. A condition que ces systèmes continuent à utiliser l'heure locale de leur base de rattachement ou qu'ils modifient la valeur de leur séquence d'horloge (voir § 12.4), l'identificateur UUID qu'ils génèrent restera unique.

NOTE 3 – Dans le cas des systèmes qui n'ont pas accès à des signaux temporels diffusés, une heure locale par enregistrement d'un système d'horloge peut être utilisée avec adjonction d'un différentiel temporel, à condition qu'aucun identificateur UUID ne soit généré lors d'un changement à partir de l'heure d'été ou d'un changement de valeur de la séquence d'horloge (voir § 12.4).

12.3.3 Pour la version fondée sur le nom d'un identificateur UUID, ce doit être une valeur de 60 bits construite à partir d'un nom mondialement unique, comme spécifié au § 14.

NOTE – Exemples de nom mondialement unique: identificateurs OID, noms URN et noms distinctifs de l'Annuaire (voir [5]).

12.3.4 Pour la version fondée sur un nombre aléatoire d'un identificateur UUID, ce doit être une valeur de 60 bits générée de façon aléatoire ou pseudo-aléatoire, comme spécifié au § 15.

12.4 Séquence d'horloge

12.4.1 Pour la version "fondée sur le temps" de l'identificateur UUID, la séquence d'horloge contribue à éviter les doublons qui pourraient apparaître quand la valeur temporelle est retardée ou lorsque la valeur nodale est modifiée.

NOTE – L'expression "séquence d'horloge" est appropriée pour la version "fondée sur le temps" d'un identificateur UUID, mais est également utilisée pour le contenu de la valeur correspondante dans les versions fondées sur le nom et fondées sur un nombre aléatoire d'un identificateur UUID.

12.4.2 Si la valeur temporelle est retardée, ou peut avoir été retardée (par exemple lorsque le système a été déconnecté de son alimentation), alors le générateur d'identificateurs UUID ne peut pas savoir si un identificateur UUID a déjà été généré avec des valeurs de temps plus grandes que la valeur à laquelle l'heure est actuellement réglée. Dans de telles situations, la valeur de la séquence d'horloge doit être modifiée.

NOTE – Si la valeur précédente de la séquence d'horloge est connue, cette valeur peut être seulement augmentée; sinon elle devrait être réglée à une valeur aléatoire ou pseudo-aléatoire, de qualité cryptographique.

12.4.3 De même, si la valeur nodale change (par exemple parce qu'une carte de réseau a été déplacée entre des machines), la valeur de la séquence d'horloge doit être modifiée.

12.4.4 La séquence d'horloge doit être initialement (c'est-à-dire une seule fois pendant la durée de vie d'un système produisant des identificateurs UUID) initialisée à un nombre aléatoire qui n'est pas déduit de la valeur nodale.

NOTE – Cette exigence vise à minimiser la corrélation entre systèmes afin d'assurer une protection maximale contre la possibilité que des adresses MAC puissent être transférées ou commutées rapidement d'un système à un autre.

12.4.5 Pour la version fondée sur le nom de l'identificateur UUID, la séquence d'horloge doit être une valeur de 14 bits construite à partir d'un nom comme spécifié au § 14.

12.4.6 Pour la version fondée sur un nombre aléatoire de l'identificateur UUID, la séquence d'horloge doit être une valeur de 14 bits générée de façon aléatoire ou pseudo-aléatoire comme spécifié au § 15.

12.5 Nœud

12.5.1 Pour la version "fondée sur le temps" d'un identificateur UUID, la valeur nodale doit consister en une adresse de commande MAC (voir ISO/CEI 8802-3), qui est habituellement l'adresse du serveur local d'une interface avec le réseau.

12.5.2 Pour les systèmes ayant de multiples adresses de commande MAC, toute adresse disponible peut être utilisée sauf une adresse de multidiffusion. L'octet 5 de l'identificateur UUID (c'est-à-dire le premier octet du champ "Node") doit être réglé au premier octet de l'adresse de commande MAC qui est transmise par un système conforme à l'ISO/CEI 8802-3.

NOTE 1 – Cet octet contient le bit de communication mondiale/locale et le bit de communication unidiffusée/multidiffusée. Il est requis que le bit de communication unidiffusée/multidiffusée soit réglé à l'unidiffusion afin d'éviter des conflits avec les adresses générées conformément au § 12.5.3.

NOTE 2 – Il est possible d'obtenir un bloc d'adresses de commande MAC auprès de l'organisme d'enregistrement d'adresses de commande MAC (voir [4]).

12.5.3 Pour les systèmes n'ayant aucune adresse MAC, un nombre aléatoire ou pseudo-aléatoire de qualité cryptographique peut être utilisé (voir Annexe C). Le bit de communication multidiffusée doit être activé dans de telles adresses.

NOTE – Cette exigence vise à garantir que les adresses générées n'entreront jamais en conflit avec des adresses obtenues à partir de cartes de réseau, comme spécifié dans le § 12.5.2.

12.5.4 Pour un identificateur UUID fondé sur le nom, la valeur nodale doit être une valeur de 48 bits construite par transformation en nom canonique et hachage à partir d'un nom mondialement unique comme spécifié au § 14.

12.5.5 Pour un identificateur UUID fondé sur un nombre aléatoire, la valeur nodale doit être une valeur de 48 bits générée de façon aléatoire ou pseudo-aléatoire comme spécifié au § 15.

13 Réglage des champs d'un identificateur UUID fondé sur le temps

Les champs d'un identificateur UUID fondé sur le temps doivent être réglés comme suit:

- déterminer les valeurs de l'heure fondée sur le temps UTC et de la séquence d'horloge à utiliser dans l'identificateur UUID, comme spécifié dans les § 12.3 et 12.4;
- pour les besoins de cet algorithme, considérer que le temps est un entier de 60 bits non signé et que la séquence d'horloge est un entier de 14 bits non signé. Numérotier séquentiellement les bits contenus dans chaque valeur, avec zéro pour le bit de poids faible;
- rendre le champ "TimeLow" égal aux 32 bits de poids faible (bits 31 à 0) de l'heure, dans le même ordre de poids;
- rendre le champ "TimeMid" égal aux bits 47 à 32 de l'heure, dans le même ordre de poids;
- rendre les 12 bits de poids faible (bits 11 à 0) du champ "VersionAndTimeHigh" égaux aux bits 59 à 48 de l'heure, dans le même ordre de poids;
- rendre les quatre bits de poids fort (bits 15 à 12) du champ "VersionAndTimeHigh" égaux aux quatre bits de numéro de version spécifiés dans le § 12.2;
- rendre le champ "ClockSeqLow" égal aux huit bits de poids faible (bits 7 à 0) de la séquence d'horloge, dans le même ordre de poids;
- rendre les six bits de poids faible (bits 5 à 0) du champ "VariantAndClockSeqHigh" égaux aux six bits de poids fort (bits 13 à 8) de la séquence d'horloge, dans le même ordre de poids;
- rendre les deux bits de poids fort (bits 7 et 6) du champ "VariantAndClockSeqHigh" égaux à un et à zéro, respectivement;
- rendre le champ "Node" égal aux 48 bits de l'adresse de commande MAC, dans le même ordre de poids que cette adresse.

14 Réglage des champs d'un identificateur UUID fondé sur le nom

Ce paragraphe spécifie les procédures pour la génération d'un identificateur UUID fondé sur le nom. Le § 14.1 spécifie les procédures générales pour toute fonction de hachage (voir également l'ISO/CEI 10118-3). Le § 14.2 spécifie l'utilisation de l'algorithme MD5 et le § 14.3 spécifie l'utilisation de l'algorithme SHA-1.

NOTE – L'utilisation de l'algorithme MD5 est limitée aux cas nécessitant une rétrocompatibilité avec les identificateurs UUID existants car l'algorithme SHA-1 fournit désormais un hachage ayant une plus petite probabilité que la même valeur de hachage apparaisse à partir de différents matériaux hachés (voir § C.4).

14.1 Les champs d'un identificateur UUID fondé sur le nom doivent être réglés comme suit:

- attribuer un identificateur UUID à utiliser comme "identificateur d'espace nominatif" pour tous les identificateurs UUID générés à partir des noms contenus dans cet espace nominatif;
NOTE – La Rec. UIT-T D.9 recommande les identificateurs UUID à utiliser pour quatre espaces nominatifs d'usage courant.
- convertir le nom en une séquence canonique d'octets (comme défini par les normes ou conventions de son espace nominatif);

- calculer la valeur de hachage de 16 octets de l'identificateur d'espace nominatif concaténé avec le nom, au moyen de la fonction de hachage spécifiée dans le § 14.2 ou 14.3. Le numérotage des octets dans la valeur de hachage va de 0 à 15, comme spécifié dans le document IETF RFC 1321 pour l'algorithme MD5 et comme spécifié dans le document FIPS PUB 180-2 pour l'algorithme SHA-1;
- régler les octets 3 à 0 du champ "TimeLow" aux octets 3 à 0 de la valeur de hachage;
- régler les octets 1 et 0 du champ "TimeMid" aux octets 5 et 4 de la valeur de hachage;
- régler les octets 1 et 0 du champ "VersionAndTimeHigh" aux octets 7 et 6 de la valeur de hachage;
- remplacer par surécriture les quatre bits de poids fort (bits 15 à 12) du champ "VersionAndTimeHigh" par les quatre bits de numéro de version du Tableau 3 du § 12.2 relatifs à la fonction de hachage qui a été utilisée;
- rendre le champ "VariantAndClockSeqHigh" égal à l'octet 8 de la valeur de hachage;
- remplacer par surécriture les deux bits de poids fort (bits 7 et 6) du champ "VariantAndClockSeqHigh" par 1 et 0, respectivement;
- rendre le champ "ClockSeqLow" égal à l'octet 9 de la valeur de hachage;
- régler les octets 5 à 0 du champ "Node" à la valeur des octets 15 à 10 de la valeur de hachage.

14.2 Ce paragraphe spécifie un identificateur UUID fondé sur le nom, utilisant l'algorithme MD5 comme fonction de hachage. L'algorithme MD5 ne doit cependant pas être utilisé pour les identificateurs UUID récemment générés (voir § C.4). Pour une fonction de hachage par algorithme MD5, la "valeur de hachage" citée au § 14.1 est la valeur de 16 octets qui est spécifiée par le document IETF RFC 1321 pour les octets de 0 à 15.

NOTE – La présente spécification de l'algorithme MD5, avec le numéro de version associé, n'est incluse qu'aux fins de la rétrocompatibilité avec les spécifications antérieures.

14.3 Ce paragraphe spécifie un identificateur UUID fondé sur le nom, utilisant l'algorithme SHA-1 comme fonction de hachage. Pour une fonction de hachage SHA-1, la "valeur de hachage" citée au § 14.1 doit correspondre aux octets 0 à 15 de la valeur de 20 octets obtenue à partir de la valeur de 160 bits du condensé de message, spécifiée par le document FIPS PUB 180-2. Les octets 16 à 19 de la valeur de 20 octets doivent être rejetés. La valeur de 20 octets doit être obtenue à partir de la valeur de 160 bits du condensé de message selon le document FIPS PUB 180-2 en plaçant le bit de poids fort de la valeur de 160 bits dans le bit de poids fort du premier octet (octet zéro) de la valeur de 20 octets et le bit de poids faible dans le dernier octet (octet 19) de la valeur de 20 octets.

15 Réglage des champs d'un identificateur UUID fondé sur un nombre aléatoire

15.1 Les champs d'un identificateur UUID fondé sur un nombre aléatoire doivent être réglés comme suit:

- régler les deux bits de poids fort (bits 7 et 6) du champ "VariantAndClockSeqHigh" à 1 et 0, respectivement;
- rendre les quatre bits de poids fort (bits 15 à 12) du champ "VersionAndTimeHigh" égaux aux quatre bits du numéro de version spécifié au § 12.2;
- régler tous les autres bits de l'identificateur UUID à des valeurs générées de façon aléatoire (ou pseudo-aléatoire).

NOTE – Des nombres pseudo-aléatoires peuvent générer la même valeur plusieurs fois. L'utilisation de nombres aléatoires de qualité cryptographique est fortement recommandée afin de réduire la probabilité de répétition de valeurs.

15.2 L'Annexe C fournit des orientations sur la façon de générer des nombres aléatoires dans un système.

16 Enregistrement des identificateurs UUID et utilisation de ces identificateurs comme composants d'identificateur OID

16.1 L'arborescence d'identificateurs OID en notation ASN.1

NOTE – Ce paragraphe résume les principales dispositions de la Rec. UIT-T X.660 | ISO/CEI 9834-1.

16.1.1 La présente Recommandation | Norme internationale définit les procédures de fonctionnement d'un organisme d'enregistrement qui enregistre des identificateurs UUID. Cet enregistrement permet également d'utiliser ces identificateurs UUID comme arcs de l'arbre d'identificateurs OID au-dessous de l'arc `{joint-iso-itu-t uuid(25)}` (voir également le § 7).

NOTE – Les identificateurs UUID peuvent également servir à identifier des arcs situés au-dessous de cet arc `uuid(25)` sans avoir été enregistrés. Mais l'identification de tels arcs n'est alors pas garantie univoque au niveau mondial.

16.1.2 Les identificateurs OID sont une forme d'identification mondialement univoque fondée sur une structure hiérarchique arborescente et sur des organismes d'enregistrement hiérarchiquement indépendants. L'arbre d'identificateurs OID possède un nœud radical, des arcs au-dessous de ce nœud radical, des arcs au-dessous de chacun de ces arcs et ainsi de suite, à toute profondeur. Les arcs sont identifiés par des valeurs d'entier non négatives (voir § 16.1.5) qui offrent une identification univoque d'un arc dans le nœud supérieur. Les arcs peuvent également recevoir des noms (consistant en un ou plusieurs caractères de lettres minuscules, de lettres majuscules, de chiffres et de tirets, avec une lettre minuscule initiale, aucun tiret adjacent à un autre et aucun tiret final). Mais ces noms sont complémentaires aux valeurs numériques et ne sont pas requis. Un objet est identifié par la séquence de valeurs d'arc (sous forme de valeurs numériques, ou également, pour les arcs initiaux, sous forme de noms de ces arcs) à partir du nœud radical de cet objet.

NOTE – On trouvera une description plus complète de l'arbre d'identificateurs OID dans la Rec. UIT-T X.660 | ISO/CEI 9834-1 et dans la Rec. UIT-T X.680 | ISO/CEI 8824-1.

16.1.3 (NOTE DIDACTIQUE) Il importe de noter qu'un identificateur UUID non enregistré peut être utilisé sous le même arc qu'un identificateur UUID enregistré (voir § 16.1.1). Des valeurs identiques d'un identificateur UUID enregistré et d'un identificateur UUID non enregistré (ou de deux identificateurs UUID non enregistrés) pourraient donc être utilisées, bien que la probabilité de cet événement soit très petite. Cette probabilité est augmentée si des identificateurs UUID sont générés à partir de valeurs de hachage MD5 ou de nombres pseudo-aléatoires, plutôt qu'à partir de valeurs de hachage SHA-1 et de nombres aléatoires de qualité cryptographique. Cela peut causer une confusion pour les utilisateurs de l'identificateur OID et pourrait déclencher une utilisation malveillante, comme l'usurpation. L'organisme d'enregistrement d'identificateurs UUID est responsable d'une collision entre identificateurs UUID enregistrés, mais n'est pas responsable d'une collision entre un identificateur UUID enregistré et un identificateur UUID non enregistré étant donné qu'il ne gère pas les identificateurs UUID non enregistrés. Si de telles collisions se produisent, la sémantique associée à l'identificateur UUID enregistré devrait avoir priorité; la sémantique associée à la valeur d'identificateur UUID non enregistré ne devrait pas être utilisée. L'enregistrement d'un identificateur OID avec un identificateur UUID ne garantit donc pas que l'identificateur OID possède une unicité supérieure à celle de son identificateur UUID. L'objectif de l'enregistrement devrait être considéré principalement comme un moyen de publier l'identificateur UUID et sa sémantique.

16.1.4 Il est possible, lors d'une représentation mécanique d'un identificateur OID, d'indiquer (par le contexte de cette représentation mécanique) l'identification d'une partie du trajet à partir de la racine d'un objet dans l'arbre d'identificateurs OID. Ainsi, si l'on sait qu'un identificateur OID est formé comme spécifié au § 7, une représentation mécanique pourrait consister seulement de la valeur de l'identificateur UUID.

16.1.5 Les composants des identificateurs OID sont des entiers non négatifs de grandeur illimitée.

16.2 Désignation des organismes d'enregistrement

16.2.1 C'est à l'UIT-T | ISO/CEI qu'il appartient d'organiser l'enregistrement comme spécifié dans la présente Recommandation | Norme internationale. A cette fin, l'UIT-T | ISO/CEI charge, conformément à ses exigences et règles internes, une organisation d'agir comme organisme d'enregistrement selon la présente Recommandation | Norme internationale.

NOTE – Aux fins de l'enregistrement, il convient d'utiliser l'adresse URL <http://www.itu.int/ITU-T/asn1/uuid.html> (qui est le site du projet de notation ASN.1 de l'UIT-T).

16.2.2 L'organisme d'enregistrement ne doit pas être tenu responsable d'un quelconque défaut de fonctionnement dans le cadre de ces procédures, ni d'éventuelles actions associées à ses devoirs en rapport avec la présente Recommandation | Norme internationale, sauf qu'il peut être déchargé de ses devoirs par la Commission d'études compétente de l'UIT-T | par le Sous-Comité compétent de l'ISO/CEI JTC 1, sans pénalité. L'organisme d'enregistrement ne doit pas être tenu responsable d'une quelconque utilisation d'une valeur d'identificateur OID non enregistré qui serait identique à la valeur d'un identificateur OID enregistré, étant donné qu'il n'a aucun contrôle sur l'utilisation de telles valeurs (voir § 16.1.3).

NOTE – Au cas où la Commission d'études compétente de l'UIT-T | le Sous-Comité compétent de l'ISO/CEI JTC 1 déterminerait que l'organisme d'enregistrement doit être déchargé de ses devoirs, pour cette raison ou pour toute autre raison, l'on s'attend à ce que les informations d'enregistrement conservées par cet organisme d'enregistrement seront mises à la disposition de tout organisme d'enregistrement éventuellement désigné par la suite.

16.3 Honoraires

16.3.1 L'organisation offrant ces services d'organisme d'enregistrement doit le faire sur une base rémunérée. La structure de rémunération doit être conçue de façon à couvrir les dépenses de fonctionnement de l'organisme d'enregistrement, à couvrir la publication des enregistrements en protocole IP, à prendre en charge les demandes de renseignements et à décourager les requêtes frivoles et multiples.

16.3.2 Les montants des honoraires doivent être déterminés par l'organisme d'enregistrement, sous réserve de l'approbation de la Commission d'études compétente de l'UIT-T | du Sous-Comité ISO/CEI JTC 1 compétent. Ces honoraires peuvent s'appliquer aux opérations suivantes:

- a) enregistrement;
- b) demande de renseignements;
- c) publication en protocole IP;
- d) demande de mise à jour.

16.3.3 Les honoraires doivent être indépendants, sous réserve des fluctuations du taux de change monétaire dans le pays à partir duquel la demande d'enregistrement est effectuée.

16.4 Procédures d'enregistrement

Ce paragraphe spécifie les procédures à suivre lors de l'enregistrement des identificateurs UUID. Ces procédures sont conçues de façon à garantir l'évolutivité et la rigueur du fonctionnement de l'organisme d'enregistrement.

16.4.1 Demande d'enregistrement d'un identificateur UUID

16.4.1.1 Une organisation soumet une demande d'enregistrement d'identificateur UUID directement à l'organisme d'enregistrement, en remplissant le formulaire sur son site IP. Le contenu de la demande est spécifié dans le § 16.4.3.

16.4.1.2 Dès achèvement correct des procédures d'enregistrement, la valeur de 128 bits de l'identificateur UUID doit être enregistrée comme ayant été assignée à (ou par) l'organisation requérante et doit être publiée.

16.4.2 Processus de confirmation

Un enregistrement réussi est confirmé par la réponse du site IP et par la publication en protocole IP.

16.4.3 Contenu de la demande

16.4.3.1 Ce paragraphe spécifie les informations requises par l'organisme d'enregistrement afin de conduire le processus d'enregistrement.

NOTE – Au moment de la publication de la présente Recommandation | Norme internationale, ces informations peuvent être soumises par courriel, par téléphone, par envoi postal ou par enregistrement sur site IP.

16.4.3.2 L'enregistrement comprend les informations suivantes:

- a) le pays dans lequel l'organisme d'enregistrement possède son siège principal;
- b) le nom de l'organisation, avec des renseignements sur son enregistrement national s'il s'agit d'une compagnie ou organisation caritative enregistrée, etc. ou son affiliation à une organisation internationale connue;
- c) le nom et le titre, l'adresse postale, l'adresse électronique, le numéro de téléphone et le numéro de télécopie du point de contact dans l'organisme d'enregistrement;
- d) des informations sous forme libre établissant la bonne foi de l'organisme d'enregistrement en tant que moyen de contrôle et d'élimination des enregistrements parasites;
- e) (facultativement) une adresse URL pouvant être consultée afin d'offrir de plus amples renseignements sur l'utilisation de l'identificateur UUID.

16.4.3.3 Le contenu d'une demande générale d'identificateur OID est spécifié dans la Rec. UIT-T X.660 | ISO/CEI 9834-1, § 8.

16.5 Maintenance d'un registre accessible par protocole IP

16.5.1 L'organisme d'enregistrement doit conserver, sur un site IP de son choix, un registre de tous les enregistrements.

16.5.2 Les informations relatives aux organisations impliquées dans un enregistrement doivent être mises à jour gratuitement par l'organisme d'enregistrement si celui-ci reçoit les détails des changements de nom de compagnie, ou des informations analogues, accompagnés de l'autorisation de changement correspondante. Le mécanisme de mise à jour doit être déterminé par l'organisme d'enregistrement et doit être annoncé sur son site IP.

Annexe A

Algorithmes pour la génération efficace d'identificateurs UUID fondés sur le temps

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

La présente annexe décrit un algorithme qui peut servir à générer, de façon répétitive dans un système informatique, des identificateurs UUID fondés sur le temps.

A.1 Algorithme de base

A.1.1 L'algorithme suivant est simple, correct, mais inefficace:

- obtenir un blocage mondial à l'échelle du système;
- à partir d'une mémoire stable partagée à l'échelle du système (par exemple un fichier), lire l'état du générateur d'identificateurs UUID, soit les valeurs d'heure, de séquence d'horloge et de nœud ayant servi à générer le dernier identificateur UUID;
- obtenir l'heure actuelle sous la forme d'un décompte des intervalles de 100 ns écoulés depuis 00:00:00.00 le 15 octobre 1582 en tant que valeur temporelle de 60 bits;
- obtenir la valeur nodale actuelle;
- si l'état était indisponible (par exemple inexistant ou corrompu), ou si la valeur nodale sauvegardée est différente de la valeur nodale actuelle, générer une valeur aléatoire de la séquence d'horloge;
- si l'état était disponible, mais que la valeur temporelle sauvegardée soit ultérieure à la valeur temporelle actuelle, augmenter la valeur de la séquence d'horloge;
- sauvegarder l'état (valeurs actuelles d'heure, de séquence d'horloge et de nœud) de façon à revenir à la mémoire stable;
- libérer le blocage mondial;
- formater un identificateur UUID à partir des valeurs actuelles d'heure, de séquence d'horloge et de nœud conformément aux étapes du § 13.

A.1.2 Si les identificateurs UUID n'ont pas besoin d'être fréquemment générés, l'algorithme ci-dessus peut être tout à fait adéquat. En présence d'exigences de performance supérieures, les problèmes posés par l'algorithme de base sont toutefois les suivants:

- la lecture de l'état dans une mémoire stable à chaque instant est inefficace;
- la résolution du système d'horloge peut ne pas être de 100 ns;
- l'écriture de l'état dans une mémoire stable à chaque instant est inefficace;
- le partage de l'état de part et d'autre des limites de processus peut être inefficace.

A.1.3 Chacun de ces problèmes peut être résolu de façon modulaire par des améliorations locales des fonctions de lecture et d'écriture de l'état et de lecture de l'horloge. Chacune de ces améliorations est décrite dans les paragraphes ci-après.

A.2 Lecture en mémoire stable

A.2.1 S'il est lu dans une mémoire volatile partagée à l'échelle du système, l'état n'a besoin d'être lu à partir de la mémoire stable qu'une seule fois pendant l'amorçage (et il est mis à jour en même temps que cette mémoire stable).

A.2.2 Si une implémentation ne possède aucune mémoire stable disponible, alors il est toujours possible d'affirmer que les valeurs étaient indisponibles. C'est l'implémentation la moins souhaitable parce qu'elle va augmenter la fréquence de création de nouveaux numéros de séquence d'horloge, ce qui augmentera la probabilité de doublons.

A.2.3 Si la valeur nodale ne peut jamais changer (par exemple si la carte réseau est indissociable du système) ou si un éventuel changement réinitialise également la séquence d'horloge à une valeur aléatoire, alors, au lieu de conserver cette valeur en mémoire stable, la valeur nodale actuelle peut être renvoyée.

A.3 Résolution de l'horloge-système

A.3.1 La valeur temporelle est générée à partir de l'heure du système, dont la résolution peut être inférieure à la résolution requise pour le temps.

A.3.2 Si les identificateurs UUID n'ont pas besoin d'être fréquemment générés, l'heure peut simplement être celle du système, multipliée par le nombre d'intervalles de 100 ns contenus dans un intervalle temporel du système.

A.3.3 Si un système dépasse la capacité du générateur en demandant un trop grand nombre d'identificateurs UUID dans un même intervalle temporel du système, le service d'identificateurs UUID devrait soit renvoyer une erreur, ou arrêter le générateur d'identificateurs UUID jusqu'à ce que le système d'horloge ait rattrapé le retard.

A.3.4 On peut simuler une valeur temporelle de haute résolution en conservant un décompte du nombre d'identificateurs UUID qui ont été générés avec la même valeur d'heure du système et en l'utilisant afin de construire les bits de poids faible de la valeur temporelle. Ce décompte sera compris entre zéro et le nombre d'intervalles de 100 ns contenus dans un intervalle temporel du système.

NOTE – Si les processeurs dépassent fréquemment la capacité de génération d'identificateurs UUID, des adresses de commande MAC supplémentaires peuvent être allouées au système, ce qui permettra une plus grande vitesse d'affectation en rendant de multiples identificateurs UUID éventuellement disponibles pour chaque valeur temporelle.

A.4 Ecriture en mémoire stable

L'état n'a pas toujours besoin d'être écrit en mémoire stable chaque fois qu'un identificateur UUID est généré. La valeur temporelle inscrite en la mémoire stable peut être périodiquement réglée à une valeur plus grande que toute valeur déjà utilisée dans un identificateur UUID. Aussi longtemps que les identificateurs UUID générés ont une valeur temporelle inférieure à cette valeur et que la séquence d'horloge et la valeur nodale restent sans changement, seule la copie volatile et partagée de l'état a besoin d'être mise à jour. Par ailleurs, si la valeur temporelle inscrite en mémoire stable devient ultérieurement inférieure à la durée normale de réamorçage du système, une panne totale ne provoquera pas de réinitialisation de la séquence d'horloge.

A.5 Partage d'état entre processus

S'il est trop onéreux d'accéder à un état partagé chaque fois qu'un identificateur UUID est généré, alors le générateur à l'échelle du système peut être implémenté de façon à attribuer un bloc de valeurs de temps chaque fois qu'il est sollicité et un générateur attribué à un processus peut attribuer des valeurs à partir de ce bloc jusqu'à ce que celui-ci soit épuisé.

Annexe B

Propriétés des identificateurs UUID fondés sur le nom

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

B.1 L'identificateur UUID fondé sur le nom vise à générer un identificateur UUID à partir d'un nom qui est extrait d'un certain espace nominatif et qui y est unique. Le concept de nom ou d'espace nominatif devrait être interprété au sens large et ne devrait pas être limité à des noms textuels. Les mécanismes ou conventions d'attribution des noms à partir de leur espace nominatif, ainsi que la vérification de leur unicité dans cet espace, sont hors du domaine d'application de la présente spécification.

NOTE – Afin d'éviter des problèmes de récurrence, les identificateurs UUID fondés sur le nom ne devraient pas être générés à partir d'un identificateur OID se terminant par un identificateur UUID fondé sur le nom.

B.2 Les propriétés des identificateurs UUID fondés sur le nom qui ont été générés conformément au § 14 et avec un espace nominatif convenablement choisi seront les suivantes:

- les identificateurs UUID générés à différents instants à partir du même nom dans le même espace nominatif seront égaux;
- les identificateurs UUID générés à partir de deux noms différents dans le même espace nominatif seront différents avec une très haute probabilité;
- les identificateurs UUID générés à partir du même nom dans deux espaces nominatifs différents seront différents avec une très haute probabilité;
- si deux identificateurs UUID fondés sur le nom sont égaux, alors ils ont été générés à partir du même nom dans le même espace nominatif avec une très haute probabilité.

Annexe C

Génération de nombres aléatoires dans un système

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

C.1 Si un système ne possède pas la capacité de générer des nombres aléatoires de qualité cryptographique, alors, dans la plupart des systèmes, il y a habituellement un assez grand nombre de sources de hasard disponibles à partir desquelles un nombre aléatoire peut être produit. De telles sources sont propres au système, mais comprennent souvent:

- le pourcentage de mémoire en usage;
- la taille de la mémoire principale, en octets;
- la quantité de mémoire principale libre, en octets;
- la taille du fichier de basculement sur disque dur ou sur mémoire à accès direct, en octets;
- les octets libres du fichier de basculement sur disque dur ou sur mémoire à accès direct;
- la taille totale de l'espace d'adresses virtuelles de l'utilisateur, en octets;
- le nombre total d'octets disponibles dans l'espace d'adresses de l'utilisateur;
- la taille du lecteur d'amorçage sur disque, en octets;
- l'espace libre du disque sur le lecteur d'amorçage, en octets;
- l'heure actuelle;
- la durée écoulée depuis l'amorçage du système;
- la taille des fichiers individuels dans divers répertoires du système;
- les heures de création, de dernière lecture et de modification de fichiers dans divers répertoires du système;
- les facteurs d'utilisation de diverses ressources du système (segments de mémoire, etc.);
- la position actuelle du curseur de la souris;
- la position actuelle du curseur complémentaire;
- le nombre actuel de processus ou de tâches en cours;
- les pointeurs ou identificateurs de la fenêtre du bureau et de la fenêtre active;
- la valeur du pointeur de pile du programme appelant;
- l'identificateur de processus et de tâche du programme appelant;
- les divers compteurs de performance propres à l'architecture du processeur (instructions exécutées, taux d'insuccès en antémémoire, tampon du répertoire de pages actives (TLB, *translation lookaside buffer*) ou taux d'insuccès du tampon TLB).

C.2 Par ailleurs, des éléments comme le nom de l'ordinateur et le nom du système d'exploitation, bien que non aléatoires au sens propre, contribueront à différencier les résultats de ceux qui ont été obtenus par d'autres systèmes.

C.3 L'algorithme exact permettant de générer une valeur nodale au moyen de ces données est propre au système, parce qu'aussi bien les données disponibles que les fonctions permettant de les obtenir sont souvent très spécifiques du système. Une approche générique consiste cependant à accumuler d'aussi nombreuses sources que possible dans un tampon et à utiliser un condensé de message comme l'algorithme SHA-1, à prendre une valeur arbitraire de six octets à partir de la valeur de hachage et à activer le bit de communication multidiffusée comme décrit ci-dessus.

C.4 D'autres fonctions de hachage, comme les fonctions MD5 et les fonctions de hachage spécifiées dans l'ISO/CEI 10118, peuvent également être utilisées. La seule exigence est que le résultat soit suffisamment aléatoire dans le sens que les sorties d'un ensemble d'entrées uniformément réparties sont elles-mêmes uniformément réparties et que le changement d'un seul bit dans l'entrée peut être censé provoquer le changement de la moitié des bits de sortie. (L'utilisation de l'algorithme MD5, cependant, n'est pas recommandée pour les nouveaux identificateurs UUID parce que des recherches récentes ont montré que ses valeurs de sortie n'étaient pas uniformément réparties.)

Annexe D

Exemple d'implémentation

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

D.1 Fichiers fournis

Cette implémentation se compose de 6 fichiers: `copyrt.h`, `uuid.h`, `uuid.c`, `sysdep.h`, `sysdep.c` et `utest.c`. Les fichiers `uuid.*` constituent l'implémentation indépendante des algorithmes de génération d'identificateurs UUID décrits dans les § 13, 14 et 15, avec inclusion de toutes les optimisations décrites dans l'Annexe A (sauf le partage efficace des états entre processus). Le code suppose la prise en charge des entiers de 64 bits, ce qui le rend beaucoup plus clair.

NOTE – Le code a été vérifié sur Linux (Red Hat 4.0) avec GCC (2.7.2) et sur Windows NT 4.0 avec VC++ 5.0.

D.2 Le fichier `copyrt.h`

Tous les fichiers source suivants devraient être considérés comme comportant l'avertissement de droits d'auteur suivant:

```
/*
** Copyright (c) 1990- 1993, 1996 Open Software Foundation, Inc.
** Copyright (c) 1989 by Hewlett-Packard Company, Palo Alto, Ca. &
** Digital Equipment Corporation, Maynard, Mass.
** Copyright (c) 1998 Microsoft.
** La permission d'utiliser, de copier, de modifier et de distribuer ce fichier à
** toutes fins utiles est ici
** accordée gratuitement à quiconque reconnaît que ce fichier est fourni "EN L'ETAT"
** sans aucune garantie expresse ou implicite,
** à condition que les avertissements de droits d'auteur ci-dessus
** et que le présent avertissement apparaissent dans toutes les copies de code
** source et qu'aucun
** des noms suivants: "Open Software Foundation, Inc.", "Hewlett-Packard
** Company", ou "Digital Equipment Corporation" ne soit utilisé dans la publicité
** ou dans les annonces relatives à la distribution de ce logiciel sans autorisation
** préalable,
** écrite et expresse.
** Aucune des compagnies "Open Software Foundation, Inc., Hewlett-Packard Company,
** Microsoft ou Digital
** Equipment Corporation" ne formule d'observations quant à la convenance du
** présent logiciel
** à une fin quelconque.
*/
```

D.3 Le fichier `uuid.h`

```
#include "copyrt.h"
#undef uuid_t
typedef struct {
    unsigned32 time_low;
    unsigned16 time_mid;
    unsigned16 time_hi_and_version;
    unsigned8 clock_seq_hi_and_reserved;
    unsigned8 clock_seq_low;
    byte node[6];
} uuid_t;

/* uuid_create -- génération d'un identificateur UUID */
int uuid_create(uuid_t * uuid);

/* uuid_create_from_name -- création d'un identificateur UUID au moyen d'un "nom"
   extrait d'un "espace nominatif" */
void uuid_create_from_name(
    uuid_t * uuid,          /* UUID résultant */
    uuid_t nsid,           /* UUID de l'espace nominatif */
    void * name,           /* nom à partir duquel un identificateur UUID est généré */
    int namelen            /* longueur du nom */
);

/* uuid_compare -- Comparer deux UUID "lexicographiquement" et renvoyer la réponse:
   -1 u1 est lexicographiquement avant u2
   0 u1 est égal à u2
   1 u1 est lexicographiquement après u2
```

```

    Noter que l'ordonnancement lexicographique n'est pas un ordonnancement temporel!
*/
int uuid_compare(uuid_t *u1, uuid_t *u2);

```

D.4 Le fichier uuid.c

```

#include "copyrt.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "sysdep.h"
#ifdef _WINDOWS_
    #include <arpa/inet.h>
#endif
#include "uuid.h"

/* diverses déclarations préalables */
static int read_state(unsigned16 *clockseq, uuid_time_t *timestamp,
    uuid_node_t *node);
static void write_state(unsigned16 clockseq, uuid_time_t timestamp,
    uuid_node_t node);
static void format_uuid_v1(uuid_t *uuid, unsigned16 clockseq,
    uuid_time_t timestamp, uuid_node_t node);
static void format_uuid_v3(uuid_t *uuid, unsigned char hash[16]);
static void get_current_time(uuid_time_t *timestamp);
static unsigned16 true_random(void);

/* uuid_create -- génération d'un identificateur UUID */
int uuid_create(uuid_t *uuid)
{
    uuid_time_t timestamp, last_time;
    unsigned16 clockseq;
    uuid_node_t node;
    uuid_node_t last_node;
    int f;

    /* acquisition d'un blocage à l'échelle du système de façon à garantir
l'unicité */
    LOCK;

    /* obtention de l'heure, de l'identificateur de noeud et de l'état sauvegardé à
partir d'une mémoire non volatile (NV) */
    get_current_time(&timestamp);
    get_ieee_node_identifieur(&node);
    f = read_state(&clockseq, &last_time, &last_node);

    /* si aucun état n'existe en mémoire NV, ou si l'horloge a été retardée, ou si
l'identificateur de noeud a été modifié (par exemple, nouvelle carte
réseau), changer la séquence d'horloge */
    if (!f || memcmp(&node, &last_node, sizeof node))
        clockseq = true_random();
    else if (timestamp < last_time)
        clockseq++;

    /* sauvegarder l'état pour la prochaine fois */
    write_state(clockseq, timestamp, node);

    UNLOCK;

    /* bourrer les champs contenus dans l'identificateur UUID */
    format_uuid_v1(uuid, clockseq, timestamp, node);
    return 1;
}

/* format_uuid_v1 -- construire un identificateur UUID à partir du marqueur
temporel, de la séquence d'horloge et de l'identificateur
de noeud */
void format_uuid_v1(uuid_t *uuid, unsigned16 clock_seq,
    uuid_time_t timestamp, uuid_node_t node)
{
    /* construire un identificateur UUID en version 1 avec les informations déjà
obtenues, plus quelques constantes. */
    uuid->time_low = (unsigned long)(timestamp & 0xFFFFFFFF);
    uuid->time_mid = (unsigned short)((timestamp >> 32) & 0xFFFF);
    uuid->time_hi_and_version =
        (unsigned short)((timestamp >> 48) & 0x0FFF);
    uuid->time_hi_and_version |= (1 << 12);

```

```

        uuid->clock_seq_low = clock_seq & 0xFF;
        uuid->clock_seq_hi_and_reserved = (clock_seq & 0x3F00) >> 8;
        uuid->clock_seq_hi_and_reserved |= 0x80;
        memcpy(&uuid->node, &node, sizeof uuid->node);
    }

    /* type de données dans l'état persistant du générateur d'identificateurs UUID */
    typedef struct {
        uuid_time_t  ts;          /* saved timestamp */
        uuid_node_t  node;        /* saved node identifier */
        unsigned16   cs;          /* saved Clock Sequence */
    } uuid_state;

    static uuid_state st;

    /* read_state -- lire l'état du générateur UUID à partir de la mémoire non volatile */
    /*
    int read_state(unsigned16 *clockseq, uuid_time_t *timestamp,
                   uuid_node_t *node)
    {
        static int initied = 0;
        FILE *fp;

        /* nécessité de ne lire l'état qu'une seule fois par amorçage */
        if (!initied) {
            fp = fopen("state", "rb");
            if (fp == NULL)
                return 0;
            fread(&st, sizeof st, 1, fp);
            fclose(fp);
            initied = 1;
        }
        *clockseq = st.cs;
        *timestamp = st.ts;
        *node = st.node;
        return 1;
    }

    /* write_state -- nouvelle sauvegarde de l'état du générateur UUID en mémoire non
    volatile */
    void write_state(unsigned16 clockseq, uuid_time_t timestamp,
                    uuid_node_t node)
    {
        static int initied = 0;
        static uuid_time_t next_save;
        FILE* fp;

        if (!initied) {
            next_save = timestamp;
            initied = 1;
        }

        /* toujours sauvegarder l'état en mémoire partagée volatile */
        st.cs = clockseq;
        st.ts = timestamp;
        st.node = node;
        if (timestamp >= next_save) {
            fp = fopen("state", "wb");
            fwrite(&st, sizeof st, 1, fp);
            fclose(fp);
            /* programmer la prochaine sauvegarde à 10 s à partir de l'actuelle */
            next_save = timestamp + (10 * 10 * 1000 * 1000);
        }
    }

    /* get_current_time -- obtenir l'heure sous la forme d'intervalles de 100 ns écoulés
    depuis le début de l'époque des identificateurs UUID, sur 60 bits.
    Compenser le fait que la résolution réelle de l'horloge est inférieure à 100 ns. */
    /*
    void get_current_time(uuid_time_t *timestamp)
    {
        static int initied = 0;
        static uuid_time_t time_last;
        static unsigned16 uuids_this_tick;
        uuid_time_t time_now;

        if (!initied) {
            get_system_time(&time_now);
            uuids_this_tick = UUIDS_PER_TICK;
            initied = 1;
        }
    }

```

```

    }

    for ( ; ; ) {
        get_system_time(&time_now);

        /* si la lecture de l'horloge a été modifiée depuis le dernier UUID généré,
*/
        if (time_last != time_now) {
            /* réinitialiser le décompte des identificateurs UUID générés avec cette
               lecture d'horloge */
            uuids_this_tick = 0;
            time_last = time_now;
            break;
        }
        if (uuids_this_tick < UUIDS_PER_TICK) {
            uuids_this_tick++;
            break;
        }
        /* allure trop rapide pour l'horloge actuelle; rattrapage */
    }
    /* ajouter le décompte des identificateurs UUID aux bits de poids faible de la
       lecture d'horloge */
    *timestamp = time_now + uuids_this_tick;
}

/* true_random -- génération d'un nombre aléatoire de qualité cryptographique.
   ** Cet échantillon n'a pas cette capacité** */
static unsigned int true_random(void)
{
    static int init = 0;
    uuid_time_t time_now;

    if (!init) {
        get_system_time(&time_now);
        time_now = time_now / UUIDS_PER_TICK;
        srand((unsigned int)((time_now >> 32) ^ time_now) & 0xffffffff);
        init = 1;
    }

    return rand();
}

/* uuid_create_from_name -- création d'un identificateur UUID au moyen d'un "nom"
   extrait d'un "espace nominatif" */
void uuid_create_from_name(uuid_t *uuid, uuid_t nsid, void *name,
                           int namelen)
{
    MD5_CTX c;

    unsigned char hash[16];
    uuid_t net_nsid;

    /* mettre l'identificateur d'espace nominatif dans l'ordre des octets du réseau,
       de façon qu'il soit haché de la même façon, quel que soit l'automate de
       calcul des poids relatifs de ces octets */
    net_nsid = nsid;
    htonl(net_nsid.time_low);
    htons(net_nsid.time_mid);
    htons(net_nsid.time_hi_and_version);

    MD5Init(&c);
    MD5Update(&c, &net_nsid, sizeof net_nsid);
    MD5Update(&c, name, namelen);
    MD5Final(hash, &c);

    /* le hachage s'effectue désormais dans l'ordre des octets du réseau */
    format_uuid_v3(uuid, hash);
}

/* format_uuid_v3 -- construire un identificateur UUID à partir d'un nombre
   (pseudo-) aléatoire de 128-bit */
void format_uuid_v3(uuid_t *uuid, unsigned char hash[16])
{
    /* convertir les UUID dans l'ordre local des octets */
    memcpy(uuid, hash, sizeof *uuid);
    ntohl(uuid->time_low);
    ntohs(uuid->time_mid);
    ntohs(uuid->time_hi_and_version);

    /* insérer les bits de variante et de version */

```

```

    uuid->time_hi_and_version &= 0x0FFF;
    uuid->time_hi_and_version |= (3 << 12);
    uuid->clock_seq_hi_and_reserved &= 0x3F;
    uuid->clock_seq_hi_and_reserved |= 0x80;
}

/* uuid_compare -- Comparer deux UUID "lexicographique" et renvoyer la réponse */
#define CHECK(f1, f2) if (f1 != f2) return f1 < f2 ? -1 : 1;
int uuid_compare(uuid_t *u1, uuid_t *u2)
{
    int i;

    CHECK(u1->time_low, u2->time_low);
    CHECK(u1->time_mid, u2->time_mid);
    CHECK(u1->time_hi_and_version, u2->time_hi_and_version);
    CHECK(u1->clock_seq_hi_and_reserved, u2->clock_seq_hi_and_reserved);
    CHECK(u1->clock_seq_low, u2->clock_seq_low)
    for (i = 0; i < 6; i++) {
        if (u1->node[i] < u2->node[i])
            return -1;
        if (u1->node[i] > u2->node[i])
            return 1;
    }
    return 0;
}
#undef CHECK

```

D.5 Le fichier sysdep.h

```

#include "copyrt.h"
/* supprimer la définition qui suit si l'on est pas en Windows 32 */
#define WININC 0

#ifdef WININC
#include <windows.h>
#else
#include <time.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/time.h>
#endif

#include "global.h"
/* passer au pointage sur le point où MD5. h est actif; le document IETF RFC 1321
   donne un exemple d'implémentation */
#include "md5.h"

/* régler ce qui suit au nombre d'intervalles de 100 ns de la résolution de
   l'horloge du système utilisé */
#define UUIDS_PER_TICK 1024

/* régler ce qui suit à un appel de façon à obtenir et à libérer un blocage mondial
*/
#define LOCK
#define UNLOCK

typedef unsigned long    unsigned32;
typedef unsigned short   unsigned16;
typedef unsigned char     unsigned8;
typedef unsigned char     byte;

/* régler ce qui suit à la structure que le compilateur actuel utilise pour les
   données de type 64 bits */
#ifdef WININC
#define unsigned64_t unsigned __int64
#define I64(C) C
#else
#define unsigned64_t unsigned long long
#define I64(C) C##LL
#endif

typedef unsigned64_t uuid_time_t;
typedef struct {
    char nodeID[6];
} uuid_node_t;

void get_ieee_node_identifieur(uuid_node_t *node);
void get_system_time(uuid_time_t *uuid_time);
void get_random_info(unsigned char seed[16]);

```

D.6 Le fichier sysdep.c

```

#include "copyrt.h"
#include <stdio.h>
#include <string.h>
#include "sysdep.h"

/* appel de procédure dépendant du système de façon à obtenir
   un identificateur de noeud de commande MAC. Cet exemple d'implémentation
   génère un identificateur de nœud aléatoire. */
void get_ieee_node_identifieur(uuid_node_t *node)
{
    static int initied = 0;
    static uuid_node_t saved_node;
    unsigned char seed[16];
    FILE *fp;

    if (!initied) {
        fp = fopen("nodeid", "rb");
        if (fp) {
            fread(&saved_node, sizeof saved_node, 1, fp);
            fclose(fp);
        }
        else {
            get_random_info(seed);
            seed[0] |= 0x80;
            memcpy(&saved_node, seed, sizeof saved_node);
            fp = fopen("nodeid", "wb");
            if (fp) {
                fwrite(&saved_node, sizeof saved_node, 1, fp);
                fclose(fp);
            }
        }
        initied = 1;
    }

    *node = saved_node;
}

/* appel de procédure dépendant du système, visant à obtenir l'heure actuelle du
   système. Réponse renvoyée sous forme d'intervalles de 100 ns écoulés depuis
   l'époque des UUID; mais la résolution d'horloge peut être inférieure à 100 ns. */
#ifdef _WINDOWS_

void get_system_time(uuid_time_t *uuid_time)
{
    ULARGE_INTEGER time;

    /* Windows NT sauvegarde l'heure en format FILETIME qui est exprimé en
       intervalles de 100 ns écoulés depuis le 1er janvier 1601. Les identificateurs
       UUID expriment le temps en intervalles de 100 ns depuis le 15 octobre 1582.
       La différence est de 17 (jours en octobre) + 30 (jours en novembre) + 31
       (jours en décembre) + 18 ans + 5 jours intercalaires.*/
    GetSystemTimeAsFileTime((FILETIME *)&time);
    time.QuadPart +=
        (unsigned __int64) (1000*1000*10) // seconds
        * (unsigned __int64) (60 * 60 * 24) // days
        * (unsigned __int64) (17+30+31+365*18+5); // # of days
    *uuid_time = time.QuadPart;
}

void get_random_info(unsigned char seed[16])
{
    MD5_CTX c;
    struct {
        MEMORYSTATUS m;
        SYSTEM_INFO s;
        FILETIME t;
        LARGE_INTEGER pc;
        DWORD tc;
        DWORD l;
        char hostname[MAX_COMPUTERNAME_LENGTH + 1];
    } r;

    MD5Init(&c);
    GlobalMemoryStatus(&r.m);
    GetSystemInfo(&r.s);
    GetSystemTimeAsFileTime(&r.t);
    QueryPerformanceCounter(&r.pc);
    r.tc = GetTickCount();

```

```

    r.l = MAX_COMPUTERNAME_LENGTH + 1;
    GetComputerName(r.hostname, &r.l);
    MD5Update(&c, &r, sizeof r);
    MD5Final(seed, &c);
}

#else

void get_system_time(uuid_time_t *uuid_time)
{
    struct timeval tp;

    gettimeofday(&tp, (struct timezone *)0);

    /* Décalage entre heures en format UUID et heures en format Unix.
       La date de base en temps UTC des identificateurs UUID est le 1er octobre 1582.
       La date de base d'Unix est le 1er janvier 1970. */
    *uuid_time = (tp.tv_sec * 100000000) + (tp.tv_usec * 10)
        + 164(0x01B21DD213814000);
}

void get_random_info(unsigned char seed[16])
{
    MD5_CTX c;
    struct {
        struct timeval t;
        char hostname[257];
    } r;

    MD5Init(&c);
    gettimeofday(&r.t, (struct timezone *)0);
    gethostname(r.hostname, 256);
    MD5Update(&c, &r, sizeof r);
    MD5Final(seed, &c);
}

#endif

```

D.7 Le fichier utest.c

```

#include "copyrt.h"
#include "sysdep.h"
#include <stdio.h>
#include "uuid.h"

uuid_t NameSpace_DNS = { /* 6ba7b810-9dad-11d1-80b4-00c04fd430c8 */
    0x6ba7b810,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

/* puid -- imprimer un identificateur UUID */
void puid(uuid_t u)
{
    int i;
    printf("%8.8x-%4.4x-%4.4x-%2.2x%2.2x-", u.time_low, u.time_mid,
        u.time_hi_and_version, u.clock_seq_hi_and_reserved,
        u.clock_seq_low);
    for (i = 0; i < 6; i++)
        printf("%2.2x", u.node[i]);
    printf("\n");
}

/* pilote simple pour générateur d'identificateurs UUID */
int main(int argc, char **argv)
{
    uuid_t u;
    int f;

    uuid_create(&u);
    printf("uuid_create(): "); puid(u);

    f = uuid_compare(&u, &u);
    printf("uuid_compare(u,u): %d\n", f); /* devrait être 0 */
    f = uuid_compare(&u, &NameSpace_DNS);
    printf("uuid_compare(u, NameSpace_DNS): %d\n", f); /* devrait être 1 */
    f = uuid_compare(&NameSpace_DNS, &u);
}

```



```

printf("uuid_compare(NameSpace_DNS, u): %d\n", f); /* devrait être -1 */
uuid_create_from_name(&u, NameSpace_DNS, "www.widgets.com", 15);
printf("uuid_create_from_name(): "); puid(u);
}

```

D.8 Exemple de sortie du fichier utable

```

uuid_create(): 7d444840-9dc0-11d1-b245-5ffdce74fad2
uuid_compare(u,u): 0
uuid_compare(u, NameSpace_DNS): 1
uuid_compare(NameSpace_DNS, u): -1
uuid_create_from_name(): e902893a-9d22-3c7e-a7b8-d6e313b71d9f

```

D.9 Quelques identificateurs d'espace nominatif

Le présent paragraphe énumère les identificateurs de quelques espaces nominatifs pouvant présenter un intérêt, en tant que structures initialisées en langage C conformément à la représentation concaténée définie ci-dessus.

```

/* une chaîne nominative est un nom de domaine entièrement qualifié */
uuid_t NameSpace_DNS = { /* 6ba7b810-9dad-11d1-80b4-00c04fd430c8 */
    0x6ba7b810,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

/* la chaîne nominative est une adresse URL */
uuid_t NameSpace_URL = { /* 6ba7b811-9dad-11d1-80b4-00c04fd430c8 */
    0x6ba7b811,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

/* la chaîne nominative est un identificateur OID */
uuid_t NameSpace_OID = { /* 6ba7b812-9dad-11d1-80b4-00c04fd430c8 */
    0x6ba7b812,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

/* la chaîne nominative est un nom distinctif d'Annuaire (en notation DER ou en
   format de sortie textuelle) */
uuid_t NameSpace_X500 = { /* 6ba7b814-9dad-11d1-80b4-00c04fd430c8 */
    0x6ba7b814,
    0x9dad,
    0x11d1,
    0x80, 0xb4, 0x00, 0xc0, 0x4f, 0xd4, 0x30, 0xc8
};

```

BIBLIOGRAPHIE

- [1] ZAHN (L.), DINEEN (T.), LEACH (P.): Network Computing Architecture (Architecture de calcul de réseau), ISBN 0-13-611674-4, janvier 1990.
- [2] Open Group CAE: DCE: Remote Procedure Call (Appel de procédure à distance), *Spécification C309*, ISBN 1-85912-041-5, août 1994.
- [3] ISO/CEI 11578:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Appel de procédures à distance (RPC)*.
- [4] IEEE, Formulaire de demande de bloc (également appelé "bloc d'adresses Ethernet") de 4096 adresses MAC individuelles: <http://standards.ieee.org/regauth/oui/pilot-ind.html>.
- [5] Recommandation UIT-T X.500 (2001) | ISO/CEI 9594-1:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: aperçu général des concepts, modèles et services*.
- [6] IETF RFC 3061, *Un espace nominatif de noms URN pour identificateurs d'objet*.
- [7] Projet Internet: draft-mealling-uuid-urn-00 – "*A UUID URN namespace*" (Un espace nominatif uniforme de ressource (URN) pour identificateurs universellement uniques (UUID)), M. Mealling, P. Leach, R. Salz, octobre 2002.

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	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 nouvelle génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

