

Groupe de travail Réseau
Request for Comments : 2579
STD : 58
RFC rendue obsolète : 1902
Catégorie : Norme
avril 1999
Traduction Claude Brière de L'Isle

Editeurs de la présente version : K. McCloghrie, Cisco Systems
D. Perkins, SNMPinfo
J. Schoenwaelder, TU Braunschweig
Auteurs de la version précédente : J. Case, SNMP Research
K. McCloghrie, Cisco Systems
M. Rose, First Virtual Holdings
S. Waldbusser, International Network Services

Conventions textuelles pour SMIv2

Statut du présent mémoire

Le présent document spécifie un protocole sur la voie de la normalisation de l'Internet pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Notice de Copyright

Copyright (C) The Internet Society (1999). Tous droits réservés.

Table des matières

1.	Introduction.....	1
1.1.	Note sur la terminologie.....	2
2.	Définitions.....	2
3.	Transposition de la macro TEXTUAL-CONVENTION.....	10
3.1	Transposition de la clause DISPLAY-HINT.....	11
3.2	Transposition de la clause STATUS.....	12
3.3	Transposition de la clause DESCRIPTION.....	12
3.4	Transposition de la clause REFERENCE.....	12
3.5	Transposition de la clause SYNTAX.....	12
4.	Sous-types de conventions textuelles.....	12
5.	Révision d'une définition de convention textuelle.....	12
6.	Considérations pour la sécurité.....	13
7.	Adresse des éditeurs.....	13
8.	Références.....	13
9.	Déclaration complète de droits de reproduction.....	14

1. Introduction

Les informations de gestion sont vues comme une collection d'objets gérés, résidant dans un magasin d'informations virtuel, appelé la base de données d'informations de gestion (MIB, *Management Information Base*). Les collections des objets qui s'y rapportent sont définies dans des modules de MIB. Ces modules sont écrits en utilisant un sous-ensemble adapté de la notation de syntaxe abstraite n° 1 (ASN.1, *Abstract Syntax Notation One*) de l'OSI, (1988) [1], appelé la Structure d'informations de gestion (SMI) [2].

Lors de la conception d'un module de MIB, il est souvent utile de définir de nouveaux types similaires à ceux définis dans le SMI. Par comparaison avec un type défini dans le SMI, chacun de ces nouveaux types a un nom différent, une syntaxe similaire, mais une sémantique plus précise. Ces types nouvellement définis sont appelés conventions textuelles, et sont utilisés pour l'agrément des lecteurs humains du module de MIB. L'objet du présent document est de définir l'ensemble initial des conventions textuelles disponibles pour tous les modules de MIB.

Les objets définis à l'aide d'une convention textuelle sont toujours codés au moyen des règles qui définissent leur type de primitive. Cependant, les conventions textuelles ont souvent une sémantique particulière associée. Comme telle, une

macro ASN.1, TEXTUAL-CONVENTION, est utilisée pour porter de façon concise la syntaxe et la sémantique d'une convention textuelle.

1.1. Note sur la terminologie

Pour les besoins de l'exposé, la structure originelle des informations de gestion, telle que décrite dans les RFC 1155 (STD 16), 1212 (STD 16) et RFC 1215, est appelée SMI version 1 (SMIV1). La version actuelle de la structure des informations de gestion est appelée SMI version 2 (SMIV2).

2. Définitions

DÉFINITIONS SNMPv2-TC ::= DÉBUT

IMPORTE

TimeTicks FROM SNMPv2-SMI;

-- définition des conventions textuelles

MACRO TEXTUAL-CONVENTION ::=

BEGIN

TYPE NOTATION ::=

DisplayPart
"STATUS" Status
"DESCRIPTION" Text
ReferPart
"SYNTAX" Syntax

VALUE NOTATION ::= value(VALUE Syntax) -- ASN.1 adapté

DisplayPart ::=

"DISPLAY-HINT" Text
| vide

Status ::=

"courant"
| "déconseillé"
| "obsolète"

ReferPart ::=

"REFERENCE" Text
| vide

-- chaîne de caractères telle que définie en [2]

Text ::= value(IA5String)

Syntax ::= -- Doit être une des suivantes : un type de base (ou son raffinement), ou un pseudo-type BITS

type
| "BITS" "{" NamedBits "}"

NamedBits ::= NamedBit | NamedBits "," NamedBit

NamedBit ::= identifiant "(" number ")" -- number est non négatif

FIN

DisplayString ::= TEXTUAL-CONVENTION

DISPLAY-HINT "255a"

STATUS courant

DESCRIPTION

"Représente les informations textuelles tirées de l'ensemble de caractères NVT ASCII, comme défini aux pages 4, 10-11 de la RFC 854.

Pour résumer la RFC 854, le répertoire NVT ASCII spécifie :

- l'utilisation des codes de caractères 0-127 (décimal)
- les caractères graphiques (32-126) sont interprétés comme de l'US ASCII
- les caractères NUL, LF, CR, BEL, BS, HT, VT et FF ont une signification spéciale spécifiée dans la RFC 854
- les 25 autres codes n'ont pas d'interprétation standard
- la séquence 'CR LF' signifie une nouvelle ligne
- la séquence 'CR NUL' signifie un retour chariot
- un 'LF' non précédé d'un 'CR' signifie de passer à la même colonne sur la ligne suivante.
- la séquence 'CR x' pour tout x autre que LF ou NUL est illégale. (Noter que cela signifie aussi qu'une chaîne peut se terminer par 'CR LF' ou 'CR NUL', mais pas par CR.)

Tout objet défini en utilisant cette syntaxe ne doit pas excéder la longueur de 255 caractères."

SYNTAX OCTET STRING (SIZE (0..255))

PhysAddress ::= TEXTUAL-CONVENTION

DISPLAY-HINT "1x:"

STATUS courant

DESCRIPTION

"Représente les adresses de niveau support ou physique."

SYNTAX OCTET STRING

MacAddress ::= TEXTUAL-CONVENTION

DISPLAY-HINT "1x:"

STATUS courant

DESCRIPTION

"Représente une adresse MAC 802 représentée dans l'ordre "canonique" défini par IEEE 802.1a, c'est-à-dire, comme si elle était transmise avec le bit de moindre poids en premier, bien que 802.5 (au contraire des autres protocoles 802.x) exige que les adresses MAC soient transmises avec le bit de plus fort poids en premier."

SYNTAX OCTET STRING (SIZE (6))

TruthValue ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"Représente une valeur booléenne."

SYNTAX INTEGER { vrai(1), faux(2) }

TestAndIncr ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"Représente des informations en valeurs d'entiers utilisées pour des opérations atomiques. Lorsque le protocole de gestion est utilisé pour spécifier qu'une instance d'objet qui a cette syntaxe est à modifier, la nouvelle valeur fournie via le protocole de gestion doit correspondre précisément à la valeur actuellement détenue par cette instance. Sinon, l'opération d'ensemble du protocole de gestion échoue avec une erreur "inconsistentValue". Autrement, si la valeur actuelle est la valeur maximale de $2^{31}-1$ (2147483647 en décimal), la valeur alors détenue par l'instance est ramenée à zéro ; autrement, la valeur détenue par l'instance est incrémentée de un. (Noter que sans considération du fait que l'opération d'ensemble du protocole de gestion réussisse ou non, la liaison de variable dans les PDU de la demande et de la réponse est identique.)

La valeur de la clause ACCESS pour les objets qui ont cette syntaxe est 'lecture-écriture' ou 'lecture-crétation'. Lorsque une instance d'un objet d'une colonne qui a cette syntaxe est créé, toute valeur peut être fournie via le protocole de gestion.

Lorsque la portion gestion de réseau du système est réinitialisée, la valeur de chaque instance d'objet qui a cette syntaxe doit être incrémentée à partir de la valeur antérieure à la réinitialisation, ou (si la valeur antérieure à la réinitialisation est inconnue) être établie à une valeur générée de façon pseudo aléatoire."

SYNTAX INTEGER (0..2147483647)

AutonomousType ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"Représente une valeur d'identification de type indépendamment extensible. Il peut, par exemple, indiquer un sous-arbre particulier avec d'autres définitions de MIB, ou définir un type particulier de protocole ou de matériel."

SYNTAX OBJECT IDENTIFIER

InstancePointer ::= TEXTUAL-CONVENTION

STATUS obsolète

DESCRIPTION

"Pointeur sur une instance spécifique d'objet de MIB ou rangée conceptuelle d'un tableau de MIB dans l'objet géré. Dans ce dernier cas, par convention, c'est le nom de l'instance particulière du premier objet colonnaire de la rangée conceptuelle.

Les deux utilisations de cette convention textuelle sont remplacées respectivement par VariablePointer et RowPointer."

SYNTAX OBJECT IDENTIFIER

VariablePointer ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"Pointeur sur une instance d'objet spécifique. Par exemple, sysContact.0 ou ifInOctets.3."

SYNTAX OBJECT IDENTIFIER

RowPointer ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"Représente un pointeur sur une rangée conceptuelle. La valeur est le nom de l'instance du premier objet colonnaire accessible dans la rangée conceptuelle.

Par exemple, ifIndex.3 pointerait sur la troisième rangée dans le ifTable (noter que si ifIndex est non accessible, ifDescr.3 serait alors utilisé à la place)."

SYNTAX OBJECT IDENTIFIER

RowStatus ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"La convention textuelle RowStatus est utilisée pour gérer la création et la suppression de rangées conceptuelles, et comme valeur de la clause SYNTAX pour la colonne État d'une rangée conceptuelle (comme décrit au paragraphe 7.7.1 de [2].)

La colonne état a six valeurs définies :

- "active", qui indique que la rangée conceptuelle est disponible pour être utilisée par l'appareil géré ;
- "notInService", qui indique que la rangée conceptuelle existe chez l'agent, mais est indisponible pour l'appareil géré (voir la note ci-dessous) ; 'notInService' n'a pas d'implication en ce qui concerne la cohérence interne de la rangée, la disponibilité des ressources, ou la cohérence avec l'état en cours de l'appareil géré ;
- "notReady", qui indique que la rangée conceptuelle existe chez l'agent, mais manque des informations nécessaires afin d'être disponible pour l'appareil géré (c'est-à-dire, une ou plusieurs des colonnes exigées dans la rangée conceptuelle n'ont pas pu être instanciées) ;
- "createAndGo", qui est fournie par une station de gestion qui souhaite créer une nouvelle instance de rangée conceptuelle et avoir son état réglé automatiquement à active, la rendant disponible pour l'appareil géré ;
- "createAndWait", qui est fournie par une station de gestion qui souhaite créer une nouvelle instance de rangée conceptuelle (mais pas la rendre disponible pour l'appareil géré) ; et,

- "destroy", qui est fournie par une station de gestion qui souhaite supprimer toutes les instances associées à une rangée conceptuelle existante.

Bien que cinq des six valeurs (toutes sauf "notReady") puissent être spécifiées dans une opération d'établissement de protocole de gestion, seules trois valeurs seront retournées en réponse à une opération de restitution de protocole de gestion : "notReady", "notInService" ou "active". C'est à dire que quand elle est interrogée, une rangée conceptuelle existante a seulement trois états : ou bien elle est disponible pour l'appareil géré (la colonne état a la valeur "active") ; elle n'est pas disponible pour l'appareil géré, bien que l'agent ait des informations suffisantes pour tenter de le faire (la colonne état a la valeur "notInService") ; ou, elle n'est pas disponible pour l'appareil géré, et une tentative de la rendre disponible échouera parce que l'agent a des informations insuffisantes (la colonne état a la valeur "notReady").

Note : Cette convention textuelle peut être utilisée pour un tableau de MIB, sans considération du fait que les valeurs de ces rangées conceptuelles du tableau sont capables d'être modifiées alors qu'elle est active, ou du fait que ses rangées conceptuelles doivent être mises hors service afin d'être modifiées. C'est à dire qu'il est de la responsabilité de la clause DESCRIPTION de la colonne État de spécifier si la colonne État ne doit pas être "active" afin que la valeur d'une autre colonne de la même rangée conceptuelle soit modifiée. Si une telle spécification est faite, les colonnes affectées peuvent être changées par une PDU d'établissement SNMP si le RowStatus ne serait pas de toute façon égal à "active" soit immédiatement avant, soit après le traitement de la PDU. En d'autres termes, si la PDU contenait aussi un varbind qui changerait la valeur de RowStatus, la colonne en question peut être changée si le RowStatus n'était pas égal à "active" lorsque la PDU a été reçue, ou si le varbind établit l'état à une valeur autre que "active".

Noter aussi que chaque fois qu'un élément d'une rangée existe, la colonne RowStatus doit aussi exister.

Pour résumer les effets d'une rangée conceptuelle dans une colonne État qui a une valeur de clause SYNTAX de RowStatus, considérons le diagramme d'état suivant :

ÉTAT				
ACTION	A la colonne état n'existe pas	B la colonne état est notReady	C la colonne état est notInService	D la colonne état est active
Régler la colonne d'état à createAndGo	noError → D ou Valeur incohérente	Valeur incohérente	Valeur incohérente	Valeur incohérente
Régler la colonne d'état à createAndWait	noError voir 1 ou wrongValue	Valeur incohérente	Valeur incohérente	Valeur incohérente
Régler la colonne d'état à active	Valeur incohérente	Valeur incohérente ou voir 2 → D	noError ou voir 8 → D	noError ou voir 8 → D
Régler la colonne d'état à notInService	Valeur incohérente	Valeur incohérente ou voir 3 → C	noError → C	noError → C ou voir 6
Régler la colonne d'état à destroy	noError → A	noError → A	noError → A	noError → A ou voir 7
Régler toute autre colonne à une valeur quelconque	voir 4	noError voir 1	noError → C	voir 5 → D

- (1) aller à B ou C, selon les informations disponible pour l'agent.
- (2) si d'autres liens de variable inclus dans la même PDU fournissent pour toutes les colonnes des valeurs qui manquent mais sont exigées, et si toutes les colonnes ont des valeurs acceptables, retourner noError et aller à D ;
- (3) si d'autres liens de variable inclus dans la même PDU fournissent pour toutes les colonnes des valeurs légales qui manquent mais sont exigées, retourner alors noError et aller à C ;
- (4) à la discrétion de l'agent, la valeur retournée peut être :
 - inconsistentName : parce que l'agent ne choisit pas de créer une telle instance lorsque l'instance de RowStatus correspondante n'existe pas, ou
 - inconsistentValue : si la valeur fournie est incohérente avec l'état de la valeur de quelque autre objet du MIB, ou
 - noError : parce que l'agent choisit de créer l'instance.

Si noError est retourné, l'instance de la colonne état doit alors aussi être créée, et le nouvel état est B ou C, selon les informations disponibles pour l'agent. Si inconsistentName ou inconsistentValue est retourné, la rangée reste dans l'état A.

- (5) selon la définition de la MIB pour la colonne/tableau, noError ou inconsistentValue peut être retourné.
- (6) la valeur retournée peut indiquer une des erreurs suivantes :
 - wrongValue : parce que l'agent ne prend pas en charge
 - notInService (par exemple, un agent qui ne prend pas en charge createAndWait), ou
 - inconsistentValue : parce que l'agent est incapable de mettre à ce moment la rangée hors service, peut-être parce qu'elle est utilisée et ne peut être désactivée.
- (7) la valeur retournée peut indiquer l'erreur suivante :
 - inconsistentValue : parce que l'agent est incapable de retirer la rangée à ce moment, peut-être parce qu'elle est utilisée et ne peut être désactivée.
- (8) la transition à D peut échouer, par exemple, si les valeurs de la rangée conceptuelle sont incohérentes, le code d'erreur sera alors inconsistentValue.

Note : D'autres traitements de la demande d'établissement (de ce varbinds et d'autres) peuvent résulter en le retour d'une réponse autre que noError, par exemple, wrongValue, noCreation, etc.

Création de rangée conceptuelle

Il y a quatre interactions potentielles lors de la création d'une rangée conceptuelle : choisir un identifiant d'instance non utilisé ; créer la rangée conceptuelle ; initialiser tous objets pour lesquels l'agent ne fournit pas une valeur par défaut ; et rendre la rangée conceptuelle disponible pour l'usage de l'appareil géré.

Interaction 1 : Choix d'un identifiant d'instance

L'algorithme utilisé pour choisir un identifiant d'instance varie pour chaque rangée conceptuelle. Dans certains cas, l'identifiant d'instance est sémantiquement signifiant, par exemple, l'adresse de destination d'un chemin, et une station de gestion choisit l'identifiant d'instance conformément à la sémantique.

Dans d'autres cas, l'identifiant d'instance est utilisé seulement pour distinguer les rangées conceptuelles, et une station de gestion sans connaissance spécifique de la rangée conceptuelle peut examiner l'instance présente afin de déterminer un identifiant d'instance non utilisé. (Cette approche peut être utilisée, mais est souvent sous-optimale ; cependant, tenter la création d'une rangée conceptuelle est aussi une pratique douteuse pour une simple station de gestion.)

Autrement, le module de MIB qui définit la rangée conceptuelle peut fournir un ou plusieurs objets qui apportent une assistance à la détermination d'un identifiant d'instance non utilisé. Par exemple, si la rangée conceptuelle est indexée par une valeur d'entier, un objet qui a une clause SYNTAX d'une valeur d'entier pourrait être définie à cette fin, permettant à une station de gestion de produire une opération de restitution de protocole de gestion. Afin d'éviter des collisions inutiles entre des stations de gestion en compétition, les restitutions "adjacentes" de cet objet devraient être différentes.

Finalement, la station de gestion peut choisir d'utiliser un nombre pseudo aléatoire comme index. Dans l'éventualité où cet index serait déjà utilisé et où une inconsistentValue serait retournée en réponse à l'opération d'établissement de protocole de gestion, la station de gestion devrait simplement choisir un nouveau nombre pseudo aléatoire et réessayer l'opération.

Un concepteur de MIB devrait choisir entre les deux derniers algorithmes en fonction de la taille du tableau (et donc de l'efficacité de chaque algorithme). Pour les tableaux dans lesquels un grand nombre d'entrées est attendu, il est recommandé qu'un objet de MIB soit défini comme retournant un indice acceptable pour la création. Pour les tableaux qui ont un petit nombre d'entrées, il est recommandé d'utiliser le dernier mécanisme d'indice pseudo aléatoire.

Interaction 2 : Création de la rangée conceptuelle

Une fois choisi un identifiant d'instance non utilisé, la station de gestion détermine si elle souhaite créer et activer la rangée conceptuelle dans une transaction ou dans un ensemble négocié d'interactions.

Interaction 2a : Création et activation de la rangée conceptuelle

La station de gestion doit d'abord déterminer les exigences de la colonne, c'est-à-dire, elle doit déterminer les colonnes pour lesquelles elle doit fournir ou non des valeurs. Selon la complexité du tableau et la connaissance de la station de gestion des capacités de l'agent, cette détermination peut être faite localement par la station de gestion. Autrement, la station de gestion produit une opération d'établissement de protocole de gestion pour examiner toutes les colonnes qu'elle veut créer dans la rangée conceptuelle. En réponse, pour chaque colonne, il y a trois résultats possibles :

- une valeur est retournée, indiquant qu'une autre station de gestion a déjà créé cette rangée conceptuelle. On retourne à l'interaction 1.
- l'exception "noSuchInstance" est retournée, indiquant que l'agent met en œuvre le type d'objet associé à cette colonne, et que cette colonne dans au moins une rangée conceptuelle sera accessible dans la vue de MIB utilisée par une restitution éventuelle. Pour les colonnes auxquelles l'agent fournit un accès en lecture-crédation, l'exception "noSuchInstance" dit à la station de gestion qu'elle devrait fournir une valeur pour cette colonne lors de la création de la rangée conceptuelle.
- l'exception "noSuchObject" est retournée, indiquant que l'agent ne met pas en œuvre le type d'objet associé à cette colonne ou qu'il n'y a pas de rangée conceptuelle pour laquelle cette colonne sera accessible dans la vue de MIB utilisée par la restitution. Comme telle, la station de gestion ne peut produire aucune opération d'établissement de protocole de gestion pour créer une instance de cette colonne.

Une fois que les exigences de la colonne ont été déterminées, une opération d'établissement de protocole de gestion est produite en conséquence. Cette opération établit aussi l'état de la nouvelle instance de la colonne à "createAndGo".

Lorsque l'agent traite l'opération d'établissement, il vérifie qu'il a des informations suffisantes pour rendre la rangée conceptuelle disponible à l'usage de l'appareil géré. Les informations disponibles à l'agent sont fournies par deux sources : l'opération d'établissement de protocole de gestion qui crée la rangée conceptuelle, et les valeurs par défaut spécifiques de la mise en œuvre fournies par l'agent (noter qu'un agent doit fournir des valeurs par défaut spécifiques de la mise en œuvre au moins pour les objets qu'il met en œuvre en lecture-seule). Si il y a des informations disponibles suffisantes, la rangée conceptuelle est alors créée, une réponse "noError" est retournée, la colonne État est réglée à "active", et aucune autre interaction n'est nécessaire (c'est-à-dire que les interactions 3 et 4 sont sautées). Si les informations sont insuffisantes, la rangée conceptuelle n'est alors pas créée, et l'opération d'établissement échoue avec une erreur de "inconsistentValue". À réception de cette erreur, la station de gestion peut produire une opération de restitution de protocole de gestion pour déterminer si c'est parce qu'il a manqué à spécifier une valeur pour une colonne exigée, ou parce que l'instance choisie de colonne État existait déjà. Dans ce dernier cas, on retourne à l'interaction 1. Dans le premier cas, la station de gestion peut recommencer l'opération d'établissement avec les informations supplémentaires, ou recommencer l'interaction 2 en utilisant "createAndWait" afin de négocier la création de la rangée conceptuelle.

Note : Sans considération de la méthode utilisée pour déterminer les exigences de la colonne, il est possible que la station de gestion estime une colonne nécessaire alors qu'en fait l'agent ne va pas permettre que cette instance de colonne particulière soit créée ou écrite. Dans ce cas, l'opération d'établissement de protocole de gestion va échouer avec une erreur "noCreation" ou "notWritable". Dans ce cas, la station de gestion décide si elle a besoin d'être capable d'établir une valeur pour cette instance de colonne particulière. Sinon, la station de gestion produit à nouveau l'opération d'établissement de protocole de gestion, mais sans établir de valeur pour cette instance de colonne particulière ; autrement, la station de gestion interrompt l'algorithme de création de rangée.

Interaction 2b : Négociation de création de rangée conceptuelle

La station de gestion produit une opération d'établissement de protocole de gestion qui règle l'instance désirée de la colonne état à "createAndWait". Si l'agent ne veut pas traiter une demande de cette sorte, l'opération d'établissement échoue avec l'erreur "wrongValue". (Par conséquent, un tel agent doit être prêt à accepter une seule opération d'établissement de protocole de gestion, c'est-à-dire, l'interaction 2a ci-dessus, contenant toutes les colonnes indiquées par ses exigences de colonne.) Autrement, la rangée conceptuelle est créée, une réponse "noError" est retournée, et la colonne état est immédiatement réglée à "notInService" ou "notReady", selon qu'il a des informations suffisantes pour (tenter de) rendre la rangée conceptuelle disponible pour utilisation par appareil géré. Si il y a des informations suffisantes disponibles, la colonne État est alors réglée à "notInService" ; autrement, s'il n'y a pas d'informations suffisantes, la colonne État est alors réglée à "notReady". De toutes façons, on passe à l'interaction 3.

Interaction 3 : Initialisation des objets qui ne sont pas par défaut

La station de gestion doit maintenant déterminer les exigences de la colonne. Elle produit une opération d'établissement de protocole de gestion pour examiner toutes les colonnes dans la rangée conceptuelle créée. Dans la réponse, pour chaque colonne, il y a trois résultats possibles :

- une valeur est retournée, indiquant que l'agent met en œuvre le type d'objet associé à cette colonne et a des informations suffisantes pour fournir une valeur. Pour les colonnes auxquelles l'agent fournit un accès en lecture-création (et pour lesquelles l'agent permet que leurs valeurs soient changées après leur création) un retour de valeur dit à la station de gestion qu'elle peut produire des opérations d'établissement de protocole de gestion supplémentaires, afin de changer la valeur associée à cette colonne.
- l'exception "noSuchInstance" est retournée, indiquant que l'agent met en œuvre le type d'objet associé à cette colonne, et que cette colonne dans au moins une rangée conceptuelle sera accessible dans la vue de MIB utilisée par la restitution si elle existe. Cependant, l'agent n'a pas des informations suffisantes pour fournir une valeur, et jusqu'à ce qu'une valeur soit fournie, la rangée conceptuelle peut n'être pas rendue disponible pour l'utilisation par l'appareil géré. Pour ces colonnes auxquelles l'agent procure l'accès en lecture-création, l'exception "noSuchInstance" dit à la station de gestion qu'elle doit fournir des opérations d'établissement de protocole de gestion supplémentaires, afin de fournir une valeur associée à cette colonne.
- l'exception "noSuchObject" est retournée, indiquant que l'agent ne met pas en œuvre le type d'objet associé à cette colonne ou qu'il n'y a pas de rangée conceptuelle pour laquelle cette colonne serait accessible dans la vue de MIB utilisée par la restitution. Comme telle, la station de gestion ne peut pas produire une des opérations d'établissement de protocole de gestion pour créer une instance de cette colonne.

Si la valeur associée à la colonne état est "notReady", la station de gestion doit alors d'abord traiter toutes les colonnes "noSuchInstance", s'il en est. Cela fait, la valeur de la colonne état devient "notInService", et on passe à l'interaction 4.

Interaction 4 : Rendre disponible la rangée conceptuelle

Une fois que la station de gestion est satisfaite des valeurs associées aux colonnes de la rangée conceptuelle, elle produit une opération d'établissement de protocole de gestion pour régler la colonne État à "active". Si l'agent a des informations suffisantes pour rendre la rangée conceptuelle disponible pour l'utilisation par l'appareil géré, l'opération d'établissement de protocole de gestion réussit (une réponse "noError" est retournée). Autrement, l'opération d'établissement de protocole de gestion échoue avec une erreur "inconsistentValue".

Note : Une rangée conceptuelle qui a une colonne état avec une valeur de "notInService" ou "notReady" est indisponible pour l'appareil géré. À ce titre, il est possible à l'appareil géré de créer ses propres instances durant le temps qui s'écoule entre l'opération d'établissement de protocole de gestion qui règle la colonne État à "createAndWait" et l'opération d'établissement de protocole de gestion qui règle la colonne État à "active". Dans ce cas, lorsque l'opération d'établissement de protocole de gestion est produite pour régler la colonne État à "active", les valeurs détenues chez l'agent supplantent celles utilisées par l'appareil géré.

Si la station de gestion est empêchée de régler la colonne état à "active" (par exemple, du fait de la station de gestion ou d'une défaillance du réseau) la rangée conceptuelle sera laissée dans l'état "notInService" ou "notReady", consommant indéfiniment des ressources. L'agent doit détecter les rangées conceptuelles qui ont été dans l'un ou l'autre état pendant une période de temps anormalement longue et les retirer. Il est de la responsabilité de la clause DESCRIPTION de la colonne État d'indiquer ce que devrait être une période anormalement longue. Cette période de temps devrait être assez longue pour permettre une réponse humaine (y compris le "temps d'y réfléchir") entre la création de la rangée conceptuelle et le réglage de l'état à "active". En l'absence de telles informations dans la clause DESCRIPTION, il est suggéré que cette période soit d'environ 5 minutes. Cette action de suppression ne s'applique pas qu'aux rangées nouvellement créées, mais aussi aux rangées précédemment actives qui sont réglées à l'état notInService, et qui y sont laissées, pendant une période prolongée excédant ce qui est considéré comme normal pour une telle rangée conceptuelle.

Suspension de rangée conceptuelle

Lorsque une rangée conceptuelle est "active", la station de gestion peut produire une opération d'établissement de protocole de gestion qui règle l'instance de la colonne état à "notInService". Si l'agent ne veut pas faire ainsi, l'opération d'établissement va échouer avec une erreur de "wrongValue" ou "inconsistentValue". Autrement, la rangée conceptuelle est mise hors service, et une réponse "noError" est retournée. Il est de la responsabilité de la clause DESCRIPTION de la colonne état d'indiquer dans quelles circonstances la colonne état devrait être mise hors service (par exemple, afin que la valeur de quelque autre colonne de la même rangée conceptuelle soit modifiée).

Suppression de rangée conceptuelle

Pour la suppression de rangées conceptuelles est produite une opération d'établissement de protocole de gestion qui règle l'instance de la colonne état à "destroy". Cette requête peut être faite sans considération de la valeur actuelle de la colonne État (par exemple, il est possible de supprimer des rangées conceptuelles qui sont "notReady", "notInService" ou "active"). Si l'opération réussit, toutes les instances associées à la rangée conceptuelle sont alors immédiatement retirées."

```
SYNTAX      INTEGER {
-- les deux valeurs suivantes sont des états :
-- ces valeurs peuvent être active(1) en écriture ou lecture,
notInService(2),
-- la valeur suivante est un état :
-- cette valeur peut être lue, mais non écrite
notReady(3),
-- les trois valeurs suivantes sont des actions : ces valeurs peuvent être écrites, mais ne sont jamais lues
createAndGo(4),
createAndWait(5),
destroy(6)
}
```

```
TimeStamp    ::= TEXTUAL-CONVENTION
STATUS      courant
DESCRIPTION
```

"La valeur de l'objet sysUpTime à laquelle une occurrence spécifique est survenue. L'occurrence spécifique définie dans la description de tout objet définie qui utilise ce type.

Si sysUpTime est remis à zéro par suite d'une réinitialisation du (sous-) système de gestion de réseau, les valeurs de tout objet TimeStamp sont alors aussi réinitialisées. Cependant, après approximativement 497 jours sans réinitialisation, l'objet sysUpTime va atteindre $2^{32}-1$ et sera alors incrémenté à zéro ; dans ce cas, les valeurs existantes des objets TimeStamp ne changent pas. Cela peut conduire à des ambiguïtés dans la valeur des objets TimeStamp."

```
SYNTAX      TimeTicks
```

```
TimeInterval ::= TEXTUAL-CONVENTION
STATUS      courant
DESCRIPTION
```

"Période de temps, mesurée en unités de 0,01 secondes."

```
SYNTAX      INTEGER (0..2147483647)
```

```
DateAndTime  ::= TEXTUAL-CONVENTION
DISPLAY-HINT "2d-1d-1d,1d:1d:1d,1a1d:1d"
```

```
STATUS      courant
```

```
DESCRIPTION
```

"Spécification de la date et de l'heure.

champ	octets	contenu	gamme
1	1-2	année*	0 à 65536
2	3	mois	1 à 12
3	4	jour	1 à 31
4	5	heure	0 à 23
5	6	minutes	0 à 59
6	7	secondes	0 à 60 (on utilise 60 pour la seconde sautée)
7	8	deci-secondes	0 à 9
8	9	direction par rapport à UTC	"+" / "-"
9	10	heures par rapport à UTC*	0 à 13
10	11	minutes par rapport à UTC	0 à 59

* Notes :

- La valeur de l'année est dans l'ordre des octets du réseau.
- l'heure d'été en Nouvelle Zélande est +13

Par exemple, mardi 26 mai 1992 à 13:30:15 EDT serait affiché : 1992-5-26,13:30:15.0,-4:0

Noter que si seule l'heure locale est connue, les informations de fuseau horaire (champs 8 à 10) ne sont pas présentes."

SYNTAX OCTET STRING (SIZE (8 | 11))

StorageType ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"Décrit la réalisation de la mémoire d'une rangée conceptuelle. Une rangée qui est volatile(2) est perdue au réamorçage. Une rangée qui est nonVolatile(3), permanent(4) ou readOnly(5), est sauvegardée par une mémorisation stable. Une rangée qui est permanent(4) peut être changée mais pas supprimée. Une rangée qui est readOnly(5) ne peut être ni changée ni supprimée.

Si la valeur d'un objet ayant cette syntaxe est permanent(4) ou readOnly(5), il ne peut être écrit. À l'inverse, si la valeur est other(1), volatile(2) ou nonVolatile(3), il ne peut pas être modifié pour devenir permanent(4) ou readOnly(5). (Toutes les modifications illégales résultent en une erreur "wrongValue".)

Chaque utilisation de cette convention textuelle doit obligatoirement spécifier les objets qu'une rangée permanent(4) doit au minimum permettre d'être écrivables."

SYNTAX INTEGER {
 other(1), -- eh ?
 volatile(2), -- par exemple, dans la RAM
 nonVolatile(3), -- par exemple, dans la NVRAM
 permanent(4), -- par exemple, partiellement dans la ROM
 readOnly(5) -- par exemple, entièrement dans la ROM
}

TDomain ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"Note une sorte de service de transport.

Certaines valeurs possibles, telles que snmpUDPDomain, sont définies dans le module de MIB SNMPv2-TM. D'autres valeurs possibles sont définies dans d'autres modules de MIB."

REFERENCE "Le module de MIB SNMPv2-TM est défini dans la RFC 1906."

SYNTAX OBJECT IDENTIFIER

TAddress ::= TEXTUAL-CONVENTION

STATUS courant

DESCRIPTION

"Note une adresse de service de transport.

Une valeur TAddress est toujours interprétée dans le contexte d'une valeur de TDomain. Et donc, chaque définition d'une valeur de TDomain doit être accompagnée d'une définition d'une convention textuelle à utiliser avec ce TDomain. Certaines conventions textuelles possibles, telles que SnmpUDPAddress pour snmpUDPDomain, sont définies dans le module de MIB SNMPv2-TM. D'autres conventions textuelles possibles sont définies dans d'autres modules de MIB."

REFERENCE "Le module de MIB SNMPv2-TM est défini dans la RFC 1906."

SYNTAX OCTET STRING (SIZE (1..255))

FIN

3. Transposition de la macro TEXTUAL-CONVENTION

La macro TEXTUAL-CONVENTION est utilisée pour porter la syntaxe et la sémantique associées à une convention textuelle. On devrait noter que l'expansion de la macro TEXTUAL-CONVENTION est quelque chose qui arrive conceptuellement durant la mise en œuvre et non durant le lancement.

Le nom d'une convention textuelle doit comporter une ou plusieurs lettres ou chiffres, les caractères initiaux étant en majuscules. Le nom ne doit pas être en conflit avec un des mots réservés dont la liste figure au paragraphe 3.7 de [2], ne devrait pas être constitué uniquement de majuscules, et ne doit pas excéder 64 caractères. (Cependant, les noms plus longs que 32 caractères ne sont pas recommandés.) Le trait d'union n'est pas admis dans le nom d'une convention textuelle (excepté pour l'utilisation dans les modules d'information convertis de SMIV1 qui admettaient les traits d'union dans les allocations de type ASN.1). De plus, tous les noms utilisés pour les conventions textuelles définies dans tous les modules d'information "standard" doivent être uniques.

3.1 Transposition de la clause DISPLAY-HINT

La clause DISPLAY-HINT, qui n'a pas besoin d'être présente, donne un conseil sur la façon d'afficher la valeur d'une instance d'un objet avec la syntaxe définie en utilisant cette convention textuelle. La clause DISPLAY-HINT ne doit pas être présente si la convention textuelle est définie avec une syntaxe de : OBJECT IDENTIFIER, IpAddress, Counter32, Counter64, ou toute syntaxe énumérée (BITS ou INTEGER). La détermination du sens que cela peut avoir pour d'autres types de syntaxes dépend de la définition spécifique de la convention textuelle.

Lorsque la syntaxe a un type de primitive sous-jacente de INTEGER, le conseil consiste en une spécification de format d'entier, contenant deux parties. La première partie est un seul caractère suggérant un format d'affichage, soit : "x" pour l'hexadécimal, soit "d" pour le décimal, ou "o" pour octal, ou "b" pour binaire. Pour tous les types, lorsque la valeur est donnée, les zéros en tête sont omis, et pour les valeurs négatives, un signe moins est rendu immédiatement avant les chiffres. La seconde partie est toujours omise pour "x", "o" et "b", et n'a pas besoin d'être présente pour "d". Si elle est présente, la seconde partie commence par un trait d'union et est suivie par un chiffre décimal, qui définit la virgule des décimales du rendu de la valeur. Par exemple :

```
Hundredths ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d-2"
...
SYNTAX     INTEGER (0..10000)
```

suggère qu'une valeur de centièmes de 1234 soit rendue par "12,34"

Lorsque la syntaxe a un type de primitive sous-jacente de OCTET STRING, le conseil consiste en une ou plusieurs spécifications de format d'octet. Chaque spécification comporte cinq parties, chaque partie utilisant et supprimant zéro, un ou plusieurs des prochains octets de la valeur et produisant les prochains, zéro, un, ou plus, caractères à afficher. Les octets au sein de la valeur sont traités dans l'ordre de poids, le plus fort poids en premier.

Les cinq parties d'une spécification de format d'octet sont :

- (1) l'indicateur de répétition (facultatif) ; s'il est présent, cette partie est une "*", qui indique que l'octet actuel de la valeur est à utiliser comme compte de répétition. Le compte de répétition est un entier non signé (qui peut être zéro) qui spécifie combien de fois le reste de cette spécification de format d'octet devrait être appliqué successivement. Si l'indicateur de répétition n'est pas présent, le compte de répétition est de un.
- (2) la longueur d'octet : un ou plusieurs chiffres décimaux qui spécifient le nombre d'octets de la valeur à utiliser et formater par cette spécification d'octets. Noter que la longueur d'octet peut être zéro. S'il reste dans la valeur moins d'octets que ce nombre, le plus petit nombre d'octets est alors utilisé.
- (3) le format d'affichage, soit : "x" pour l'hexadécimal, "d" pour le décimal, "o" pour l'octal, "a" pour l'ascii, soit "t" pour l'UTF-8. Si la partie longueur d'octet est supérieure à un, et si la partie format d'affichage se réfère à un format numérique, l'ordre des octets du réseau (codage gros boutien) est alors utilisé pour interpréter les octets de

la valeur. Les octets traités par le format d'affichage "t" ne forment pas nécessairement un nombre entier de caractères UTF-8. Les octets d'en-queue qui ne forment pas un caractère codé en UTF-8 valide sont éliminés.

- (4) le caractère de séparateur d'affichage (facultatif) ; s'il est présent, cette partie est un seul caractère qui est produit pour l'affichage après chaque application de cette spécification d'octet ; cependant, ce caractère n'est pas produit pour l'affichage s'il serait immédiatement suivi par l'affichage du caractère de terminaison de répétition pour cette spécification d'octet. Ce caractère peut être n'importe quel caractère autre qu'un chiffre décimal et une "*".
- (5) le caractère de terminaison de répétition (facultatif), qui ne peut être présent que si le caractère séparateur d'affichage est présent et si cette spécification d'octets commence par un indicateur de répétition ; s'il est présent, cette partie est un seul caractère qui est produit après toutes les, zéro, une, ou plusieurs applications répétées (comme indiqué par le compteur de répétition) de cette spécification d'octets. Ce caractère peut être n'importe quel caractère autre qu'un chiffre décimal et une "*".

La production d'un caractère de séparateur d'affichage ou d'un caractère de terminaison de répétition est supprimée si cela devait survenir comme dernier caractère de l'affichage.

Si les octets de la valeur sont épuisés avant que toute la spécification de format d'octets ait été utilisée, les spécifications excédentaires sont alors ignorées. Si des octets supplémentaires restent dans la valeur après l'interprétation de toutes les spécifications de format d'octets, la dernière spécification de format d'octets est alors réinterprétée pour traiter les octets additionnels, jusqu'à ce qu'aucun octet ne reste dans la valeur.

3.2 Transposition de la clause STATUS

La clause STATUS, qui doit être présente, indique si cette définition est en cours ou historique.

La valeur "courant" signifie que la définition est en cours et valide.

La valeur "obsolète" signifie que la définition est obsolète et ne devrait pas être mise en œuvre et/ou peut être retirée si elle a été antérieurement mise en œuvre. Alors que la valeur "déconseillée" indique aussi une définition obsolète, elle permet des mises en œuvre nouvelles/poursuivies afin de ménager l'interopérabilité avec les mises en œuvre anciennes/existantes.

3.3 Transposition de la clause DESCRIPTION

La clause DESCRIPTION, qui doit être présente, contient une définition textuelle de la convention textuelle, qui fournit toutes les définitions sémantiques nécessaires à la mise en œuvre, et devrait comporter toutes les informations qui seraient autrement communiquées dans les annotations de commentaires ASN.1 associées à l'objet.

3.4 Transposition de la clause REFERENCE

La clause REFERENCE, qui n'a pas besoin d'être présente, contient une référence croisée textuelle avec certains autres documents, ou un autre module d'informations qui définit une allocation en rapport avec l'objet, ou quelque autre document qui fournit des informations supplémentaires pertinentes pour cette définition.

3.5 Transposition de la clause SYNTAX

La clause SYNTAX, qui doit être présente, définit la structure de données abstraite correspondant à la convention textuelle. La structure de données doit être une des alternatives définies dans le CHOICE ObjectSyntax ou la construction BITS (voir au paragraphe 7.1 de [2]). Noter que cela signifie que la clause SYNTAX d'une convention textuelle ne peut pas se référer à une convention textuelle définie précédemment.

Un sous-ensemble étendu des pleines capacités du sous-type de l'ASN.1 (1988) est permis, comme approprié au type ASN.1 sous jacent. Toute restriction de cette sorte sur la taille, gamme ou énumération spécifiée dans cette clause représente le niveau maximal de prise en charge qui a du sens pour le protocole. Les restrictions sur le sous-type sont spécifiées en détail à la Section 9 et l'Appendice A de [2].

4. Sous-types de conventions textuelles

La clause SYNTAX d'une macro TEXTUAL CONVENTION peut avoir des sous-types de la même façon que la clause SYNTAX d'une macro OBJECT-TYPE (voir la Section 11 de [2]).

5. Révision d'une définition de convention textuelle

Il peut être souhaitable de réviser la définition d'une convention textuelle après l'avoir expérimentée. Cependant, les changements ne sont pas admis si ils sont porteurs de tout problème potentiel d'interopérabilité "sur le réseau" entre une mise en œuvre utilisant une spécification d'origine et une mise en œuvre qui utilise une ou des spécifications mises à jour). De tels changements ne peuvent être traités qu'en définissant une nouvelle convention textuelle (c'est-à-dire, un nouveau nom).

Les révisions suivantes sont permises :

- (1) Une clause SYNTAX contenant un ENTIER énuméré peut avoir des nouvelles énumérations ajoutées ou des étiquettes existantes changées. De même, des bits désignés peuvent être ajoutés ou des étiquettes existantes changées pour la construction de BITS.
- (2) Une valeur de clause STATUS de "courante" peut être révisée en "déconseillée" ou "obsolète". De même, une valeur de clause STATUS de "déconseillée" peut être révisée en "obsolète". Lorsque de tels changements sont faits, la clause DESCRIPTION devrait être mise à jour pour en expliquer la raison.
- (3) Une clause REFERENCE peut être ajoutée ou mise à jour.
- (4) Une clause DISPLAY-HINTS peut être ajoutée ou mise à jour.
- (5) Précisions et informations supplémentaires peuvent être incluses dans la clause DESCRIPTION.
- (6) Tout changement rédactionnel.

Noter qu'avec l'introduction de la macro TEXTUAL-CONVENTION, il n'est plus du tout besoin de définir des types de la manière suivante :

```
DisplayString ::= OCTET STRING (SIZE (0..255))
```

Lors de la révision d'un module d'information contenant une définition telle que celle là, cette définition devrait être remplacée par une macro TEXTUAL-CONVENTION.

6. Considérations pour la sécurité

Le présent document définit les moyens de définir de nouveaux types de données pour le langage utilisé pour écrire et lire les descriptions des informations de gestion. Ces types de données n'ont pas d'impact sur la sécurité de l'Internet.

7. Adresse des éditeurs

Keith McCloghrie

David Perkins

Juergen Schoenwaelder

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
téléphone : +1 408 526 5260
mél : kzm@cisco.com

SNMPinfo
3763 Benton Street
Santa Clara, CA 95051
USA
téléphone : +1 408 221-8702
mél : dperkins@snmpinfo.com

TU Braunschweig
Bueltenweg 74/75
38106 Braunschweig
Germany
téléphone : +49 531 391-3283
mél : schoenw@ibr.cs.tu-bs.de

8. Références

- [1] Systèmes de traitement de l'information – Interconnexion des systèmes ouverts - Spécification de la Notation de syntaxe abstraite numéro un (ASN.1), Organisation Internationale de normalisation. Norme internationale 8824, (décembre, 1987).
- [2] K. McCloghrie, D. Perkins, J. Schoenwaelder, J. Case, M. Rose et S. Waldbusser, "Structure des informations de gestion, version 2 (SMIV2)", STD 58, RFC 2578, avril 1999.
- [3] Groupe de travail SNMPv2, J. Case, K. McCloghrie, M. Rose et S. Waldbusser, "Transpositions de transport pour la version 2 du protocole simple de gestion de réseau (SNMPv2)", RFC 1906, janvier 1996.

9. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (1999). Tous droits réservés.

Ce document et les traductions de celui-ci peuvent être copiés et diffusés, et les travaux dérivés qui commentent ou expliquent autrement ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, partiellement ou en totalité, sans restriction d'aucune sorte, à condition que l'avis de copyright ci-dessus et ce paragraphe soit inclus sur toutes ces copies et œuvres dérivées. Toutefois, ce document lui-même ne peut être modifié en aucune façon, par exemple en supprimant le droit d'auteur ou les références à l'Internet Society ou d'autres organisations Internet, sauf si c'est nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les copyrights définies dans les processus de normes pour Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet ou ses successeurs ou ayants droit.

Ce document et les renseignements qu'il contient sont fournis "TELS QUELS" et L'INTERNET SOCIETY et L'INTERNET ENGINEERING TASK FORCE déclinent toute garantie, expresse ou implicite, y compris mais sans s'y limiter, toute garantie que l'utilisation de l'information ici présente n'enfreindra aucun droit ou aucune garantie implicite de commercialisation ou d'adaptation à un objet particulier.