

Groupe de travail Réseau  
Request for Comments: 3382  
Mise à jour de : 2910, 2911  
Catégorie : Norme  
Septembre 2002

R. deBry, Utah Valley State College  
R. Herriot, Consultant  
T. Hastings, K. Ocke, P. Zehler : Xerox Corporation

## **Protocole d'impression sur Internet (IPP) : Syntaxe de l'attribut 'collection'**

### **Statut du présent Mémo**

Le présent document spécifie un protocole de normalisation Internet pour la communauté Internet, et appelle à discussion et suggestions en vue de son amélioration. Prière de se reporter à l'édition en cours des "Internet Official Protocol Standards" (*normes officielles du protocole Internet*) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémo n'est pas soumise à restrictions.

### **Déclaration de copyright**

Copyright (C) The Internet Society (2002). Tous droits réservés

### **Résumé**

Le présent document spécifie une syntaxe d'attribut FACULTATIVE appelée 'collection' à utiliser avec le Protocole d'impression sur Internet (IPP/1.0 et IPP/1.1). Une 'collection' est une enveloppe qui conserve une ou plusieurs valeurs désignées, qu'on appelle attributs "membres". Une collection permet de grouper les données comme un dictionnaire PostScript ou un "Map" Java. Le présent document spécifie aussi les exigences de conformité pour un document de définition d'un attribut collection. Enfin, le présent document donne quelques exemples pour illustrer les définitions d'attribut collection qui ne sont pas destinés à représenter des spécifications réelles d'attribut.

## Table des matières

1	Introduction .....	2
1.1	Etat du problème .....	2
1.2	Solution .....	3
2	Terminologie .....	4
2.1	Terminologie de conformité.....	4
2.2	Autre terminologie .....	4
3	Définition d'un attribut de collection .....	4
3.1	Informations à inclure .....	4
3.2	Collections enchâssées .....	7
4	Attributs de collection comme attributs d'opérations .....	7
4.1	Règles générales.....	7
4.2	Valeurs non acceptées .....	7
5	Exemple de définition d'un attribut de collection.....	7
5.1	support-col (collection) .....	8
5.1.1	support-couleur (type3 mot clé   nom(MAX)).....	8
5.1.2	taille-de-support (collection).....	8
5.2	support-col-par-défaut (collection) .....	9
5.3	support-col-prêt (1setOf collection).....	9
5.4	support-col-accepté (1setOf type2 mot clé) .....	9
6	Second exemple de définition d'un attribut Collection.....	10
7	Codage.....	10
7.1	Etiquettes supplémentaires définies pour représenter un attribut de valeur de collection .	10
7.2	Exemple de codage : "support-col" (collection).....	11
8	Questions de succession .....	14
9	Considérations relatives à l'IANA .....	14
10	Considérations d'internationalisation.....	14
11	Considérations de sécurité.....	14
12	Références .....	14
12.1	Références normatives .....	14
12.2	Références informatives.....	15
	Appendice A : Exemple de codage d'une collection simple (pour information).....	15
	Appendice B : Exemple de codage de 1setOf Collection (pour information) .....	16
	Appendice C : Exemple de codage de Collection contenant 1setOf d'attribut XXX (pour information).....	18
	Appendice D : Description des documents IPP de base (pour information).....	20
	Adresse des auteurs .....	21
	Déclaration de copyright .....	21

## 1 Introduction

Le présent document est une extension FACULTATIVE à IPP/1.0 [RFC2565, RFC2566] et IPP/1.1 [RFC2911, RFC2910]. Voir une description des documents IPP de base à l'Appendice D.

### 1.1 Etat du problème

Le document Modèle et sémantique IPP [RFC2911] prend en charge la plupart des structures de données communes qui sont disponibles dans les langages de programmation. Il manque un

mécanisme pour grouper plusieurs attributs de différents types. Le langage Java utilise le "Map" pour résoudre ce problème et PostScript a un dictionnaire. Le nouveau mécanisme pour grouper les attributs (appelé mécanisme de 'collection') doit permettre des membres facultatifs et l'ajout ultérieur de nouveaux membres.

Le mécanisme 'collection' doit être codé de façon cohérente avec les règles d'analyse grammaticale d'IPP 1.0 et 1.1 existantes (voir la [RFC2910]). Les analyseurs actuels 1.0 et 1.1 qui n'acceptent pas le mécanisme 'collection' ne doivent pas confondre les collections ou parties d'une collection qu'ils reçoivent avec les autres attributs.

## 1.2 Solution

Le nouveau mécanisme est une nouvelle syntaxe d'attribut IPP appelée une 'collection'. Chaque valeur de collection est une valeur d'un attribut dont le type de syntaxe d'attribut est défini comme une 'collection'. Un tel attribut est appelé attribut de collection. Le nom de l'attribut de collection sert à identifier la valeur de la collection dans une demande ou réponse d'opération, comme pour toute valeur d'attribut.

La syntaxe d'attribut 'collection' est une enveloppe qui conserve une ou plusieurs valeurs désignées (c'est-à-dire, d'attributs), qu'on appelle attributs membres. Chaque document de définition d'attribut de collection fait la liste des attributs membres obligatoires et facultatifs de chaque valeur de collection. Une valeur de collection est semblable à un groupe d'attributs IPP dans une demande ou réponse, comme le groupe d'attributs d'opération. Ils consistent tous deux en un ensemble d'attributs.

Comme avec toute syntaxe d'attribut, le document qui définit un attribut de collection spécifie si l'attribut est à une seule valeur (collection) ou multi-valeurs (1setOf collection). Si l'attribut est multi-valeurs (1setOf collection), chaque valeur de collection DOIT être une instance distincte d'une seule définition d'une collection, c'est-à-dire, il DOIT avoir les mêmes attributs membres excepté pour les attributs membres FACULTATIFS. Si on considère chaque définition de collection comme un type de syntaxe distinct, cette règle continue la notion IPP/1.1 selon laquelle chaque attribut a un seul type ou schéma (par exemple, "mot clé | nom" est un schéma). Sans cette règle, les valeurs prises en charge seraient plus difficiles à décrire et le mécanisme défini au point 4 du paragraphe 3.1 ne serait pas suffisant.

Le nom de chaque attribut membre DOIT être unique pour un attribut de collection, mais PEUT être le même que le nom d'un attribut membre dans un autre attribut de collection et/ou PEUT être le même que le nom d'un attribut qui n'est pas un membre d'une collection. Les règles de dénomination des attributs membres sont données au paragraphe 3.1.

Chaque attribut membre peut avoir tout type de syntaxe d'attribut, y compris 'collection', et peut être à une seule valeur ou multi-valeurs. La longueur d'une valeur de collection n'est pas limitée. Cependant, la longueur de chaque attribut membre NE DOIT PAS excéder la limite de sa syntaxe d'attribut.

Les attributs membres d'une collection PEUVENT être dans n'importe quel ordre dans une demande ou réponse. Lorsqu'un client envoie un attribut de collection à l'imprimante, l'ordre dans lequel l'imprimante mémorise les attributs membres de la valeur de collection et l'ordre retourné dans une réponse PEUVENT être différents de l'ordre d'envoi par le client.

Une valeur de collection NE DOIT PAS contenir deux ou plus attributs membres avec le même nom d'attribut. Une telle collection est mal formée. Les clients NE DOIVENT PAS soumettre de telles demandes mal formées et les imprimantes NE DOIVENT PAS retourner de telles réponses mal formées. Si une telle demande mal formée est soumise à une imprimante, elle DOIT (selon la mise en œuvre) soit (1) rejeter la demande avec le code d'état 'erreur-client-mauvaise-demande' (voir au paragraphe 13.1.4.1 de la [RFC2911]), soit (2) accepter la demande et utiliser un seul de chaque attribut membre en double.

## 2 Terminologie

La présente section définit la terminologie utilisée dans le présent document.

### 2.1 Terminologie de conformité

Les termes en majuscules, tels que DOIT, NE DOIT PAS, EXIGE, DEVRAIT, NE DEVRAIT PAS, PEUT, PEUT NE PAS, et FACULTATIF, ont une signification particulière qui se rapporte à la conformité telle que définie dans la [RFC2119] et au paragraphe 12.1 de la [RFC2911]. Si une mise en œuvre prend en charge l'extension définie dans le présent document, ces termes s'appliquent alors ; autrement, ils ne s'appliquent pas. Ces termes définissent seulement la conformité au présent document ; ils n'affectent pas la conformité aux autres documents, sauf déclaré explicitement par ailleurs.

### 2.2 Autre terminologie

Le présent document utilise des termes tels que objet Tâche (ou Tâche), objet Imprimante IPP (ou Imprimante), "opération", "demande", réponse", "attributs", "mots clé", et "prise en charge". Ces termes ont une signification particulière et sont définis au paragraphe 12.2 de la terminologie modèle de la [RFC2911]. Les termes supplémentaires suivants sont introduits dans le présent document:

collection : syntaxe d'attribut dans laquelle chaque valeur d'attribut est un ensemble d'attributs, appelés attributs membres.

attribut membre : attribut défini comme étant utilisé comme un des attributs dans une collection.

attribut de collection : attribut dont la définition spécifie la syntaxe d'attribut 'collection' et chaque attribut membre qui PEUT survenir dans une valeur d'attribut de collection.

## 3 Définition d'un attribut de collection

La présente section décrit les exigences pour toute définition d'attribut de collection.

### 3.1 Informations à inclure

Lorsqu'un document de spécification définit un attribut de collection "xxx", c'est-à-dire, un attribut dont le type de syntaxe d'attribut est 'collection' ou '1setOf collection' ; le document de définition

DOIT inclure les aspects suivants de la sémantique d'attribut. Supposons que l'attribut de collection "xxx" contient N attributs membres nommés "aaa1", "aaa2", ..., "aaaN" ("aaaI" représente n'importe lequel de ces N attributs membres).

1. Le nom de l'attribut de collection DOIT être spécifié (par exemple, "xxx"). Le choix du nom "xxx" DOIT suivre les mêmes règles d'univocité que les attributs de tout autre type de syntaxe, (comme défini par IPP/1.1) à moins que "xxx" soit un attribut membre d'une autre collection. Le choix du nom "xxx" DOIT alors suivre les règles d'univocité définies au point 5a) de cette liste.
2. La syntaxe d'attribut collection DOIT être du type 'collection' ou 'IsetOf collection'.
3. Le contexte de l'attribut de collection DOIT être spécifié, c'est-à-dire, si l'attribut est un attribut d'opération, un attribut de gabarit de tâche, un attribut de description de tâche un a attribut de description d'imprimante, un attribut membre d'un attribut de collection particulier, etc.
4. Un attribut "xxx-accepté" DOIT être spécifié et il a une des deux formes suivantes :

a) "xxx-accepté" est un "IsetOf collection", qui énumère toutes les valeurs de collection prises en charge de "xxx". Si une collection de cette forme contient une collection enchâssée, elle DOIT être de la même forme.

Par exemple, "taille-de-support-acceptée" pourrait avoir les valeurs { {x-dimension:210, y-dimension:297}, {x-dimension:297, y-dimension:420} } pour montrer qu'elle accepte deux valeurs de "taille de support": A4 (210x297) et A3 (297x420). Elle n'accepte pas d'autres combinaisons d'attributs membres "x-dimension" et "y-dimension", comme 210x420 ou 297x297, et elle n'accepte pas les valeurs non énumérées, telles que 420x595.

b) "xxx-accepté" est un "IsetOf type2 mot clé", qui énumère les noms de tous les attributs membres de "xxx": "aaa1", "aaa2", ..., "aaaN". Si une collection de cette forme contient une collection enchâssée, elle PEUT être de l'une ou l'autre forme. Voir au point 5f) ci-dessous des précisions sur les valeurs acceptées d'attributs membres.

Par exemple, "support-couleur-accepté" pourrait avoir les valeurs de mot clé : "taille-de-support" et "support-couleur".

5. Les attributs membres DOIVENT être définis. Pour chaque attribut membre, le document de définition DOIT fournir les informations suivantes :

a) Le nom de l'attribut membre (par exemple, "aaa") DOIT être unique au sein de la collection définie et DOIT soit :

i) réutiliser le nom d'attribut d'un autre attribut (qui est unique dans la totalité de l'espace de nom d'attribut IPP) et avoir la même syntaxe et sémantique que l'attribut réutilisé (si la condition du point 4b) ci-dessus est satisfaite). Par exemple, une définition d'attribut membre pourrait réutiliser l'attribut IPP/1.1 "support".

ii) potentiellement survenir ailleurs dans la totalité de l'espace de nom d'attribut IPP. (Si la condition du point 4a) ci-dessus est satisfaite). Par exemple, un attribut membre pourrait être "x-dimension", qui pourrait survenir dans une autre collection ou comme un attribut en dehors d'une collection.

iii) être unique dans la totalité de l'espace de nom d'attribut IPP si la condition du point 4b) ci-dessus est satisfaite). Par exemple, un attribut membre pourrait être "support-couleur" qui doit être unique dans la totalité de l'espace de nom d'attribut IPP.

b) Si la prise en charge de l'attribut membre est EXIGÉE ou FACULTATIVE pour l'imprimante.

c) Si la fourniture de l'attribut membre est EXIGÉE ou FACULTATIVE pour le client dans une demande.

d) Le type de syntaxe de l'attribut membre, qui peut être toute syntaxe d'attribut, y compris '1setOf X', 'collection', et '1setOf collection'. Si ce nom d'attribut réutilise le nom d'un autre attribut (cas de d'item a1 ci-dessus), il DOIT avoir la même syntaxe d'attribut, y compris la cardinalité (qu'il soit ou non 1setOf).

e) La sémantique de l'attribut membre "aaa". La définition de la sémantique DOIT inclure une description de toute contrainte ou condition limite que l'attribut membre fait porter sur l'attribut associé, particulièrement si l'attribut réutilise le nom d'un autre attribut (cas du point a1) ci-dessus).

f) Les valeurs acceptées pour chaque attribut membre "aaaI" (des attributs membres "aaa1", "aaa2", ..., "aaaN") sont spécifiées par un des deux mécanismes suivants :

i) Si "xxx-accepté" est un "1setOf collection" (voir le point 4a) ci-dessus), la valeur pour chaque "aaaI" est spécifiée dans chaque valeur de collection de "xxx-accepté", dans le contexte d'autres attributs membres. C'est-à-dire que "xxx-accepté" énumère toutes les valeurs acceptées de "xxx".

ii) Si la valeur de "xxx-accepté" est un "1setOf type2 mot clé" (voir le point 4b) ci-dessus), les valeurs prises en charge de "aaaI" sont les valeurs spécifiées soit par i) l'attribut "aaaI-accepté" soit par ii) la définition de l'attribut membre "aaaI" dans le document qui définit l'attribut "xxx". Les valeurs de chaque attribut membre "aaaI" sont spécifiées indépendamment des autres attributs membres, bien qu'une imprimante ne soit pas obligée de prendre en charge toutes les combinaisons des valeurs prises en charge.

Par exemple, "support-couleur-accepté" pourrait avoir les valeurs de mot clé : "taille-de-support" et "support-couleur". En utilisant la première méthode pour définir les valeurs acceptées (un attribut "aaaI-accepté"), les valeurs de collection de "support-couleur" sont des combinaisons des valeurs de "taille-de-support-accepté" et "support-couleur-accepté". Si "taille-de-support-accepté" a les valeurs de '210x297' et '297x420' et "support-couleur-accepté" a les valeurs de 'blanc' et 'rose', l'imprimante pourrait ne prendre en charge que les combinaisons 'blanc-210x297', 'rose-210x297' et 'blanc-297x420', et pas 'rose-297x420'.

Si une collection contient un membre "aaaI", dont le type de syntaxe est "texte", les valeurs prises en charge devraient probablement être définies par la définition de "xxx" plutôt que par l'attribut "aaaI-accepté".

g) La valeur par défaut de chaque attribut membre "aaaI" si il est FACULTATIF pour un client de fournir l'attribut membre "aaa" dans une demande. La valeur par défaut est spécifiée par la définition de l'attribut au sein d'un document et DOIT être une des suivantes :

i) par défaut fixée,

ii) un mécanisme par lequel l'imprimante détermine la valeur par défaut,

iii) une valeur par défaut indéfinie dont le choix est laissé à la mise en œuvre,

iv) un attribut que l'imprimante utilise pour déterminer la valeur par défaut.

6. La valeur par défaut de "xxx", si un client ne la fournit pas. La valeur par défaut est spécifiée par la définition de l'attribut au sein d'un document et DOIT être des suivantes :

a) par défaut fixée,

b) un mécanisme par lequel l'imprimante détermine la valeur par défaut,

c) une valeur par défaut indéfinie dont le choix est laissé à la mise en œuvre,  
d) un attribut d'imprimante "xxx-par-défaut" qui est une collection avec les mêmes attributs membres que "xxx". Si les attributs membres facultatifs sont absents, l'imprimante utilise les règles de fourniture par défaut du point 5g) ci-dessus.

7. L'attribut "xxx-prêt (1setOf collection)", si une intervention humaine est nécessaire pour rendre disponibles un grand nombre des valeurs prises en charge. Par exemple, "support-couleur" est un attribut qui a un attribut "prêt". La plupart des attributs n'ont pas d'attribut "prêt".

### **3.2 Collections enchâssées**

Un attribut membre peut avoir un type de syntaxe de 'collection' ou '1setOf collection', auquel cas il est appelé attribut de collection enchâssé. Les règles pour un attribut de collection enchâssé sont les mêmes que spécifié au paragraphe 3.1 pour un attribut de collection.

## **4 Attributs de collection comme attributs d'opérations**

### **4.1 Règles générales**

Une valeur de collection est comme tout autre valeur IPP/1.1, sauf qu'elle est structurée. Les règles pour les attributs avec une valeur de collection sont les mêmes que pour les attributs de tout autre type de syntaxe (voir IPP/1.1), dans quelque groupe de demande ou de réponse qu'ils soient.

### **4.2 Valeurs non acceptées**

Les règles de retour d'un attribut de collection non accepté sont une extension des règles courantes :

1. Si la totalité de l'attribut de collection n'est pas acceptée, l'imprimante retourne alors simplement le nom de l'attribut de collection avec la valeur hors bande 'non-accepté' (voir au début du paragraphe 4.1 de la [RFC2911]) dans le groupe des attributs non acceptés.

2. Si une collection contient des attributs membres non reconnus, non acceptés et/ou des valeurs contradictoires, l'attribut retourné dans le groupe des attributs non acceptés est une collection contenant les attributs membres non reconnus, non acceptés et/ou les valeurs contradictoires. Les attributs membres non reconnus ont une valeur hors bande de 'non-accepté' (voir au début du paragraphe 4.1 de la [RFC2911]). Les attributs membres non acceptés et les valeurs contradictoires ont leurs valeurs non acceptées ou contradictoires.

## **5 Exemple de définition d'un attribut de collection**

Dans certains environnements d'impression, il est souhaitable de permettre au client de choisir le support d'après ses propriétés, par exemple, grammage, couleur, taille, etc., plutôt que par nom. Dans IPP/1.1 (voir la [RFC2911]), l'attribut de gabarit de tâche "support (type3 mot clé | nom)" permet le choix par nom. Il serait tentant d'étendre la syntaxe d'attribut "support" de façon à inclure "collection", mais les clients existants ne pourraient alors pas comprendre les valeurs de support par défaut ou prises en charge qui utilisent la valeur de collection. Pour préserver l'interopérabilité, un nouvel attribut DOIT être ajouté, par exemple, "support-col (collection)". Les paragraphes suivants

contiennent un échantillon de définition d'un attribut "support-col" simplifié. La définition suit les règles de la section 3.

Tous les exemples de définitions d'attribut dans le présent document sont des illustrations plutôt que des définitions réelles. Ces exemples sont destinés à illustrer comment définir les attributs de collection. D'autres documents DOIVENT définir les attributs de collection à utiliser dans un échange réel. De telles définitions peuvent être très semblables aux exemples du présent document, car on a essayé de prendre des exemples réalistes.

Note : on a pris le nom "support-col" parce que le nom "support" est déjà utilisé. Ordinairement l'attribut de collection aura un nom comme tout autre attribut et ne se terminerait pas par "col".

Les attributs membres de l'attribut "support-col" ("support-couleur (type 3 mot clé)" et "taille-de-support (collection)") suivent tous deux les règles de nommage du point 4a3 de la section 3, c'est-à-dire, les nom sont uniques à travers tout l'espace de nom d'attribut IPP. Les attributs membres de l'attribut membre "taille-de-support (collection)" ("x-dimension (entier(0:MAX))" et "y-dimension (entier(0:MAX))") suivent tous deux les règles de nommage du point 4a2 de la section 3, c'est-à-dire qu'ils peuvent survenir ailleurs dans l'espace de nom d'attribut IPP.

## 5.1 support-col (collection)

L'attribut "support-col" (collection) enrichit l'attribut IPP/1.1 "support" de la [RFC2911]. Cet attribut de collection permet à un utilisateur final client de soumettre une liste de caractéristiques de supports à l'imprimante. Lorsque le client spécifie le support en utilisant l'attribut de collection "support-col", l'objet Imprimante DOIT satisfaire exactement le support demandé. La 'collection' consiste en les attributs membres suivants :

**Tableau 1 - Attributs membres "support-col"**

Nom d'attribut	syntaxe d'attribut	demande	Pris en charge par l'imprimante
support-couleur	type3 mot clé   nom (MAX)	PEUT	DOIT
taille-de-support	collection	DOIT	DOIT

Les définitions pour les attributs membres sont données dans les paragraphes suivants.

### 5.1.1 support-couleur (type3 mot clé | nom(MAX))

Cet attribut membre identifie la couleur du support. Les valeurs valides sont 'rouge', 'blanc' et 'bleu'.

L'attribut d'imprimante "support-couleur-accepté" (1setOf (type3 mot clé | nom(MAX))) identifie les valeurs de cet attribut membre "support-couleur" que l'imprimante prend en charge, c'est-à-dire, les couleurs acceptées.

Si le client omet cet attribut membre, l'imprimante détermine la valeur d'une façon qui dépend de la mise en œuvre.

### 5.1.2 taille-de-support (collection)

Cet attribut membre identifie la taille du support. La 'collection' consiste en les attributs membres indiqués au Tableau 2 :



**Tableau 2 - Attributs membres de collection "taille-de-support"**

Nom d'attribut	syntaxe d'attribut	demande	Pris en charge par l'imprimante
x-dimension	entier (0:MAX)	DOIT	DOIT
y-dimension	entier (0:MAX)	DOIT	DOIT

Les définitions pour les attributs membres sont données dans les paragraphes suivants :

#### **5.1.2.1 x-dimension (entier(0:MAX))**

Cet attribut identifie la largeur du support en unités de pouces le long de l'axe X.

#### **5.1.2.2 y-dimension (entier(0:MAX))**

Cet attribut identifie la hauteur du support en unités de pouces le long de l'axe Y.

L'attribut d'imprimante "taille-de-support-acceptée" (1setOf collection) identifie les valeurs de cet attribut membre "taille-de-support" que l'imprimante prend en charge, c'est-à-dire, les combinaisons de taille acceptées. Les noms des attributs membres sont les mêmes que ceux des attributs membres de l'attribut de collection "taille-de-support", à savoir "x-dimension", et "y-dimension", car ils ont la même syntaxe d'attribut et la même sémantique.

### **5.2 support-col-par-défaut (collection)**

L'attribut d'imprimante "support-col-par-défaut" spécifie le support qu'utilise l'imprimante, s'il en est, si le client omet les attributs de gabarit de tâche "support-col" et "support" dans l'opération de création de tâche et que la PDL n'inclut pas de spécification du support. Les attributs membres sont définis au Tableau 1. Une imprimante DOIT prendre en charge les mêmes attributs membres pour cet attribut de collection par défaut que ceux qu'elle prend en charge pour l'attribut de gabarit de tâche "attribut-de-gabarit-de-tâche-col" correspondant.

### **5.3 support-col-prêt (1setOf collection)**

L'attribut d'imprimante "support-col-prêt" identifie les supports qui sont disponibles pour une utilisation sans intervention humaine, c'est-à-dire que les supports sont prêts à être utilisés sans intervention humaine. La valeur de collection DOIT avoir tous les attributs membres qui sont pris en charge dans le Tableau 1.

### **5.4 support-col-accepté (1setOf type2 mot clé)**

L'attribut d'imprimante "support-col-accepté" identifie les noms de mot clé des attributs membres pris en charge dans l'attribut de gabarit de tâche de collection "support-col", c'est-à-dire, les noms de mot clé des attributs membres du Tableau 1 que l'imprimante accepte.

## 6 Second exemple de définition d'un attribut Collection

Tous les exemples de définitions d'attribut dans le présent document sont des illustrations plutôt que des définitions réelles. Ces exemples sont destinés à illustrer comment définir les attributs de collection. D'autres documents DOIVENT définir les attributs de collection à utiliser dans un échange réel. De telles définitions peuvent être très semblables aux exemples du présent document, car on a essayé de prendre des exemples réalistes.

Dans certains environnements d'impression, il est souhaitable de permettre au client de choisir le support pour la feuille de début de tâche. La raison pour ne pas ajouter la syntaxe d'attribut 'collection' à l'attribut de gabarit de tâche "feuilles-de-tâche" existant est la même que pour "support". A la place est introduit un nouvel attribut de gabarit de tâche, par exemple, "feuille-de-tâche-col (collection)".

Les attributs membres de l'attribut "feuille-de-tâche-col" ("feuilles-de-tâche (type 3 mot clé)" et "support (type3 mot clé | nom)") suivent tous deux les règles de nommage du point 4a1 de la section 3, c'est-à-dire, ils réutilisent les attributs IPP existants. Conformément aux règles, leurs valeurs acceptées viennent des attributs IPP existants : "feuilles-de-tâche-acceptées" et "support-accepté". Cependant, leurs valeurs par défaut ne viennent pas respectivement de "feuilles-de-tâche-par-défaut" et "support-par-défaut". Au lieu de cela, la définition de "feuilles-de-tâche-col" dit que "feuilles-de-tâche (type 3 mot clé)" est nécessaire et si "support (type3 mot clé | nom)" est absent, l'imprimante utilise le même support que le reste de la tâche.

Si l'attribut "feuilles-de-tâche-col" était défini comme contenant l'attribut membre "support-de-feuilles-de-tâche (type3 mot clé | nom)" au lieu de "support (type3 mot clé | nom)", la définition aurait alors à spécifier un "support-de-feuilles-de-tâche-accepté (1setOf (type3 mot clé | nom))" dont les valeurs devraient être indépendantes de "support-accepté (1setOf (type3 mot clé | nom))" et devraient être réglées séparément par un administrateur de système.

Le texte réel de définition de l'attribut est laissé à titre d'exercice pour le lecteur.

## 7 Codage

La présente section définit les étiquettes de codage supplémentaires utilisées conformément à la [RFC2910] et donne un exemple de leur utilisation. Les étiquettes de codage définies dans le présent document DOIVENT être utilisées par tous les attributs de collection définis dans d'autres documents. Cependant, les exemples d'utilisation ne sont que des illustrations.

### 7.1 *Étiquettes supplémentaires définies pour représenter un attribut de valeur de collection*

La syntaxe d'attribut 'collection' utilise les étiquettes définies au Tableau 3.

**Tableau 3 - Etiquettes définies pour le codage de la syntaxe d'attribut 'collection'**

Nom d'étiquette	Valeur d'étiquette	Signification
débutCollection	0x34	Début de l'attribut de valeur de collection.
finCollection	0x37	Fin de l'attribut de valeur de collection.
nomAttrMembre	0x4A	La valeur est le nom de l'attribut membre de collection

Lors du codage d'un attribut de collection "xxx" qui contient un attribut "aaa" et n'est pas à l'intérieur d'une autre collection, le codage suit ces règles :

1. Le début de la collection est indiqué par une étiquette de valeur qui DOIT être du type de syntaxe 'débutCollection' (0x34) avec une longueur de nom et un champ Nom qui représentent le nom de l'attribut de collection ("xxx") comme avec tout attribut, suivis par une valeur. L'imprimante PEUT ignorer la valeur et sa longueur PEUT être 0. A l'avenir, cependant, ce champ PEUT contenir des informations, utiles telles que le nom de collection (cf. le nom d'une struct C).

2. Chaque attribut membre est codé comme une séquence de deux, ou plus, valeurs qui font partie d'un seul attribut multi-valeurs, c'est-à-dire, l'ensemble. La première valeur après la valeur 'débutCollection' a la syntaxe d'attribut 'nomAttrMembre' (0x4A), et sa valeur garde le nom du premier attribut membre (par exemple, "aaa"). La seconde valeur garde la valeur d'attribut du premier membre, qui peut être de n'importe quelle syntaxe d'attribut, excepté 'nomAttrMembre' ou 'finCollection'. Si la valeur d'attribut du premier membre est multi-valeurs, la troisième valeur garde la seconde valeur de la valeur du premier membre. Autrement, la troisième valeur garde le nom du second attribut membre (par exemple, "bbb"), et sa syntaxe d'attribut est 'nomAttrMembre'. Dans ce cas, la valeur du quatrième membre est la valeur de "bbb".

Noter que la technique de codage d'une 'collection' comme l'ensemble rend aisé pour une imprimante qui ne prend pas en charge un attribut de collection particulier (ou toute la syntaxe d'attribut collection) de sauter simplement la valeur de collection toute entière.

3. La fin de la collection est indiquée par une étiquette de valeur qui DOIT être du type de syntaxe 'finCollection' (par exemple, 0x37) et PEUT avoir une longueur de nom de zéro et une longueur de valeur de zéro. A l'avenir, ce champ POURRA contenir des informations utiles, telles que le nom de collection qui correspond à celui dans 'débutCollection'.

4. Il est valide d'avoir un attribut membre qui soit lui-même, un attribut de collection, c'est-à-dire que des collections peuvent être enchâssées au sein de collections. Ceci est représenté par l'occurrence d'un attribut membre qui est du type de syntaxe d'attribut 'débutCollection'. Une telle collection est terminée par un 'finCollection' correspondant. Le nom d'un tel attribut membre est dans la valeur immédiatement précédente, dont le type de syntaxe est 'nomAttrMembre'.

5. Il est valide pour un attribut de collection d'être multi-valeurs, c'est-à-dire, d'avoir plus d'une valeur de collection. Si le prochain attribut qui suit immédiatement 'finCollection' a une longueur de nom de zéro et une étiquette de 'débutCollection', l'attribut de collection est alors une collection multi-valeurs, comme avec n'importe quel attribut. Ceci s'applique aux collections au sein de collections et aux collections qui ne sont pas dans des collections.

## **7.2 Exemple de codage : "support-col" (collection)**

La collection spécifiée à la section 5 sert pour l'exemple de codage indiqué au Tableau 5. L'exemple montre aussi des collections enchâssées, car l'attribut membre "taille-de-support" est une 'collection. L'exemple de codage représente une carte d'indice 4x6 bleu et prend 216 octets. Les Appendices contiennent des exemples plus complexes.

Des exemples supplémentaires figurent dans les appendices.

La structure globale des deux valeurs de collections peut être dépeinte comme :  
 "support-col" = { "support-couleur" = 'bleu' ; "taille-de-support" = { "x-dimension" = 6 ; "y-dimension" = 4 } },

Le codage complet figure au Tableau 5. Une forme simplifiée du codage ressemble à ceci :

**Tableau 4 – Vue générale du codage de la collection "support-col"**

Valeur d'étiquette	Nom	Valeur
débutCollection	support-col	""
nomAttrMembre	""	support-couleur
mot clé	""	bleu
nomAttrMembre	""	taille-de-support
débutCollection	""	""
nomAttrMembre	""	x-dimension
entier	""	6
nomAttrMembre	""	y-dimension
entier	""	4
finCollection	""	""
finCollection	""	""

**Tableau 5 - Exemple de codage de la collection "support-col"**

Octets	Valeur symbolique	Champ de protocole	Commentaires
0x34	débutCollection	étiquette-de-valeur	début de l'attribut de collection "support-col"
0x0009		longueur-de-nom	longueur du nom d'attribut (collection)
support-col	support-col	nom	nom de (collection)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)
0x4A	nomAttrMembre	étiquette-de-valeur	commence un nouvel attribut membre : "support-couleur"
0x0000		longueur-de-nom	défini pour être 0 pour ce type, soit partie de 1setOf, pas de nom (car longueur-de-nom était 0)
0x000B		longueur de valeur	longueur de mot clé "support-couleur"
support-couleur	support-couleur	valeur	valeur est le nom du 1 <sup>er</sup> attribut membre
0x44	mot clé type	étiquette-de-valeur	type de mot clé
0x0000		longueur-de-nom	0 indique 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	
bleu	bleu	valeur	valeur du 1 <sup>er</sup> attribut membre

0x4A	nomAttrMembre	étiquette-de-valeur	commence un nouvel attribut membre : "taille-de-support"
0x0000		longueur-de-nom	défini pour être 0 pour ce type, soit partie de 1setOf, pas de nom (car longueur-de-nom était 0)
0x000A		longueur-de-valeur	longueur du mot clé "taille-de-support"
taille-de-support	taille-de-support	valeur	Nom du 2 <sup>ème</sup> attribut membre
0x34	débutCollection	étiquette-de-valeur	début de l'attribut de collection "taille-de-support" qui est une sous collection
0x0000		longueur-de-nom	0 indique 1setOf, pas de nom (car longueur-de-nom était 0)
0x0000		longueur-de-valeur	les noms d'attribut de collection n'ont pas de valeur, pas de valeur (car longueur-de-valeur était 0)
0x4A	nomAttrMembre	étiquette-de-valeur	débute un nouvel attribut membre : "x-dimension"
0x0000		longueur de nom	défini pour être 0 pour ce type, soit partie de 1setOf, pas de nom (car longueur-de-nom était 0)
0x000B		longueur-de-valeur	longueur du mot clé "x-dimension"
x-dimension	x-dimension	valeur	nom du 1 <sup>er</sup> attribut membre sous collection
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-de-nom	0 indique 1setOf, pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0006		valeur	valeur du 1 <sup>er</sup> attribut membre sous collection
0x4A	nomAttrMembre	étiquette-de-valeur	débute un nouvel attribut membre : "y-dimension"
0x0000		longueur de nom	défini pour être 0 pour ce type, soit partie de 1setOf, pas de nom (car longueur-de-nom était 0)
0x000B		longueur-de-valeur	longueur du mot clé "y-dimension"
y-dimension	y-dimension	valeur	nom du 2 <sup>ème</sup> attribut membre sous collection
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-de-nom	0 indique 1setOf, pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0004		valeur	valeur du 2 <sup>ème</sup> attribut membre sous collection
0x37	finCollection	étiquette-de-valeur	fin de la sous collection
0x0000		longueur-de-nom	défini pour être 0 pour ce type, soit partie de 1setOf, pas de nom (car longueur-de-nom était 0)
0x0000		longueur de valeur	défini pour être 0 pour ce type, pas de valeur (car longueur-de-valeur était 0)
0x37	finCollection	étiquette-de-valeur	fin de la 1 <sup>ère</sup> valeur de collection dans 1setOf
0x0000		longueur-de-nom	défini pour être 0 pour ce type, soit partie de 1setOf, pas de nom (car longueur-de-nom était 0)
0x0000		longueur de valeur	défini pour être 0 pour ce type, pas de valeur (car longueur-de-valeur était 0)

## 8 Questions de succession

Les imprimantes et clients IPP 1.x ignoreront les collections et leurs attributs membres si ils ne comprennent pas la collection. Les éléments `débutCollection` et `finCollection` ressemblent chacun à un attribut avec une syntaxe d'attribut que le receveur ne prend pas en charge et il devrait donc ignorer la totalité de l'attribut. Les attributs membres individuels et leurs valeurs ressembleront à des valeurs `1setOf` de l'attribut de collection, de sorte que l'imprimante ignore la totalité de l'attribut et toutes ses valeurs. Retourner des attributs non acceptés est aussi simple, car seul le nom de l'attribut de collection est retourné avec la valeur hors bande (voir au paragraphe 4.2).

## 9 Considérations relatives à l'IANA

Le Tableau suivant donne l'enregistrement pour la syntaxe d'attribut 'collection' définie dans le présent document. Ceci est destiné à être enregistré conformément aux procédures du paragraphe 6.3 de la [RFC2911]3.

Valeur d'étiquette :	Syntaxes d'attribut :	Référence	Section :
Collection		RFC 3382	3
0x34	<code>débutCollection</code>	RFC 3382	7.1
0x37	<code>finCollection</code>	RFC 3382	7.1
0x4A	<code>nomAttrMembre</code>	RFC 3382	7.1

L'enregistrement de syntaxe d'attribut résultant sera publié dans le registre <http://www.iana.org/assignments/ipp-registrations>.

## 10 Considérations d'internationalisation

La présente syntaxe d'attribut n'a par elle-même pas d'impact sur l'internationalisation. Cependant, les attributs membres qui sont définis ultérieurement pour être utilisés dans une collection peuvent avoir des implications internationales, comme le peuvent tous les attributs, conformément à la [RFC2911].

## 11 Considérations de sécurité

La présente syntaxe d'attribut ne cause pas plus de problèmes de sécurité qu'aucune autre syntaxe d'attribut. Ce sont seulement les attributs qui sont définis ultérieurement, pour utiliser cette syntaxe d'attribut, ou toute autre, qui peuvent poser des problèmes de sécurité, en fonction de la sémantique de l'attribut, conformément à la [RFC2911].

## 12 Références

### 12.1 Références normatives

[RFC2910] Herriot, R., Butler, S., Moore, P. et R. Turner, "Internet Printing Protocol/1.1: Encoding et Transport" (*IPPI/1.1 : codage et transport*), RFC 2910, septembre 2000.

[RFC2911] Hastings, T., Herriot, R., deBry, R., Isaacson, S. et P. Powell, "Internet Printing Protocol/1.0 : Model et Semantics" (*IPPI/1.1 : modèle et sémantique*), RFC 2911, septembre 2000.

## 12.2 Références informatives

[RFC2565] Herriot, R., Butler, S., Moore, P. et R. Tuner, "Internet Printing Protocol/1.0: Encoding et Transport" (*IPPI/1.0 : codage et transport*), RFC 2565, avril 1999.

[RFC2566] deBry, R., Hastings, T., Herriot, R., Isaacson, S. et P. Powell, "Internet Printing Protocol/1.0: Model et Semantics" (*IPPI/1.0 : modèle et sémantique*), RFC 2566, avril 1999.

[RFC2567] Wright, F.D., "Design Goals for an Internet Printing Protocol" (*Objectifs de conception pour un protocole d'impression sur Internet*), RFC 2567, avril 1999.

[RFC2568] Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" (*Fondements de la structure, du modèle et du protocole pour le protocole d'impression sur Internet*), RFC 2568, avril 1999.

[RFC2569] Herriot, R., Hastings, T., Jacobs, N. and J. Martin, "Mapping between LPD and IPP Protocols" (*Transposition entre les protocoles LPD et IPP*), RFC 2569, avril 1999.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. et T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1" (*Protocole de transfert hypertexte – HTTP/1.1*), RFC 2616, juin 1999.

[RFC3196] Hastings, T., Manros, C., Zehler, P., Kugler, C. and H. Holst, "Internet Printing Protocol/1.1: Implementor's Guide" (*Guide pour la mise en œuvre du protocole 1/1.1 d'impression sur Internet*), RFC 3196, novembre 2001.

## Appendice A : Exemple de codage d'une collection simple (pour information)

La structure globale de la valeur de collection peut être représentée comme :

"taille-de-support" = { "x-dimension" = 6 ; "y-dimension" = 4 }

Une vue simplifiée du codage devrait ressembler à :

**Tableau 6 – Vue générale du codage d'une collection simple**

Valeur d'étiquette	Nom	Valeur
débutCollection	taille-de-support	""
nomAttrMembre	""	x-dimension
entier	""	6
nomAttrMembre	""	y-dimension
entier	""	4

finCollection	""	""
---------------	----	----

Note : "" représente un nom ou valeur dont la longueur est 0.

**Tableau 7 - Exemple de codage d'une collection simple**

Octets	Valeur symbolique	Champ de protocole	Commentaires
0x34	débutCollection	étiquette-de-valeur	début de l'attribut de collection "taille-du-support"
0x000A		longueur-du-nom	longueur du nom d'attribut (collection)
taille-de-support	taille-de-support	nom	nom de l'attribut (collection)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)
0x4A	nomAttrMembre	étiquette-de-valeur	début de l'attribut membre: "x-dimension"
0x0000		longueur-du-nom	défini pour être 0 pour ce type, soit partie de 1setOf pas de nom (car longueur-de-nom était 0)
0x000B		longueur-de-valeur	longueur du mot clé "x-dimension"
x-dimension	x-dimension	valeur	nom du 1 <sup>er</sup> attribut membre collection
0x21	type-d'entier	étiquette-de-valeur	Type d'attribut
0x0000		longueur-du-nom	0 indique 1setOf pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0006		valeur	valeur du 1 <sup>er</sup> attribut membre collection
0x4A	nomAttrMembre	étiquette-de-valeur	début du nouvel attribut membre : "y-dimension"
0x0000		longueur-du-nom	défini pour être 0 pour ce type, soit partie de 1setOf pas de nom (car longueur-de-nom était 0)
0x000B		longueur valeur	longueur du mot clé "y-dimension"
y-dimension	y-dimension	valeur	nom du 2 <sup>ème</sup> attribut membre collection
0x21	type-d'entier	étiquette-de-valeur	Type d'attribut
0x0000		longueur-du-nom	0 indique 1setOf pour taille-de-support pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0004		valeur	valeur du 2 <sup>ème</sup> attribut membre collection
0x37	finCollection	étiquette-de-valeur	fin de collection
0x0000		longueur-du-nom	défini pou être 0 pour ce type, soit partie de 1setOf pas de nom (car longueur-de-nom était 0)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)

## Appendice B : Exemple de codage de 1setOf Collection (pour information)

La structure générale de la valeur de collection peut être représentée par :

"taille-de-support-accepté" = { "x-dimension" = 6 ; "y-dimension" = 4 }, { "x-dimension" = 3 ; "y-dimension" = 5 } ;



Une représentation simplifiée du codage ressemblerait à ceci :

**Tableau 8 – Vue générale du codage de 1setOf collection**

Valeur d'étiquette	Nom	Valeur
débutCollection	taille-de-support-accepté	""
nomAttrMembre	""	x-dimension
entier	""	6
nomAttrMembre	""	y-dimension
entier	""	4
finCollection	""	""
débutCollection	""	""
nomAttrMembre	""	x-dimension
entier	""	3
nomAttrMembre	""	y-dimension
entier	""	5
finCollection	""	""

**Tableau 9 - Exemple de codage de 1setOf collection**

Octets	Valeur symbolique	Champ du protocole	Commentaires
0x34	débutCollection	étiquette-de-valeur	début de l'attribut "taille-de-support-accepté (1setOf collection)"
0x00014		longueur-du-nom	longueur du nom d'attribut (collection)
taille-de-support-accepté	taille-de-support-accepté	nom	nom de l'attribut (collection)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)
0x4A	nomAttrMembre	étiquette-de-valeur	début de l'attribut membre : "x-dimension"
0x0000		longueur-du-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x000B		longueur-de-valeur	longueur du mot clé "x-dimension"
x-dimension	x-dimension	valeur	nom du 1 <sup>er</sup> attribut membre collection
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-du-nom	0 indique 1setOf pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0006		valeur	valeur du 1 <sup>er</sup> attribut membre collection
0x4A	nomAttrMembre	étiquette-de-valeur	début de l'attribut membre "y-dimension"
0x0000		longueur-du-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x000B		longueur-de-valeur	longueur du mot clé "y-dimension"
y-dimension	y-dimension	valeur	nom du 2 <sup>ème</sup> attribut membre collection
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-du-nom	0 indique 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0004		valeur	valeur du 2 <sup>ème</sup> attribut membre collection

0x37	finCollection	étiquette-de-valeur	fin de collection
0x0000		longueur-du-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)
0x34	débutCollection	étiquette-de-valeur	début du 2 <sup>ème</sup> attribut membre collection du 1setOf "tailles-disponibles"
0x0000		longueur-du-nom	nom de longueur zéro indique qu'il est membre du nom d'attribut précédent ; pas de nom (car longueur-de-nom était 0)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)
0x4A	nomAttrMembre	étiquette-de-valeur	début de l'attribut membre "x-dimension"
0x0000		longueur-du-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x000B		longueur-de-valeur	longueur du mot clé "x-dimension"
x-dimension	x-dimension	valeur	nom du 1 <sup>er</sup> attribut membre collection
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-du-nom	0 indique 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0003		valeur	valeur du 1 <sup>er</sup> attribut membre collection
0x4A	nomAttrMembre	étiquette-de-valeur	début de l'attribut membre "y-dimension"
0x0000		longueur-du-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x000B		longueur-de-valeur	longueur du mot clé "y- dimension"
y-dimension	y-dimension	valeur	nom du 2 <sup>ème</sup> attribut membre collection
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-du-nom	0 indique 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0005		valeur	valeur du 2 <sup>ème</sup> attribut membre collection
0x37	finCollection	étiquette-de-valeur	fin de 1setOf valeur de collection
0x0000		longueur-du-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)

## Appendice C : Exemple de codage de Collection contenant 1setOf d'attribut XXX (pour information)

La structure générale de la valeur de collection peut être représentée par :

"wagons" = { "couleurs" = rouge, bleu ; tailles" = 4, 6, 8 }

Une représentation simplifiée du codage ressemblerait à ceci :

**Tableau 10 – Vue générale du codage de collection avec 1setOf valeur**

Valeur d'étiquette	Nom	Valeur
débutCollection	wagons	""
nomAttrMembre	""	couleurs
mot-clé	""	rouge
mot-clé	""	bleu
nomAttrMembre	""	tailles
entier	""	4
entier	""	6
entier	""	8
finCollection	""	""

**Tableau 11 - Exemple de codage de collection avec 1setOf valeur**

Octet	Valeur symbolique	Champ de protocole	Commentaires
0x34	débutCollection	étiquette-de-valeur	début de l'attribut de collection "wagons"
0x0005		longueur-de-nom	longueur du nom d'attribut (collection)
wagons	wagons	nom	nom de l'attribut (collection)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)
0x4A	nomAttrMembre	étiquette-de-valeur	début d'un nouvel attribut membre : "couleurs"
0x0000		longueur-de-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0006		longueur-de-valeur	longueur du mot clé "couleurs"
couleurs	couleurs	valeur	valeur est le nom du 1 <sup>er</sup> attribut membre
0x44	type de mot clé	étiquette-de-valeur	type de mot clé
0x0000		longueur-de-nom	0 indique 1setOf wagons ; pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	
bleu	bleu	valeur	valeur du 1 <sup>er</sup> attribut membre
0x44	type de mot clé	étiquette-de-valeur	type de mot clé
0x0000		longueur-de-nom	0 indique 1setOf wagons ; pas de nom (car longueur-de-nom était 0)
0x0003		longueur-de-valeur	
rouge	rouge	valeur	valeur du 1 <sup>er</sup> attribut membre
0x4A	nomAttrMembre	étiquette-de-valeur	début d'un nouvel attribut membre: "tailles"
0x0000		longueur-de-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0005		longueur-de-valeur	longueur du mot clé "longueur-disponible"
tailles	tailles	valeur	nom du 2 <sup>ème</sup> attribut membre
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-de-nom	0 indique 1setOf wagons ; pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0004		valeur	1 <sup>ère</sup> valeur pour l'attribut 1setOf entier
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-de-nom	0 indique 1setOf ; pas de nom (car longueur-de-nom était 0)

0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0006		valeur	2 <sup>ème</sup> valeur pour l'attribut 1setOf entier
0x21	type d'entier	étiquette-de-valeur	type d'attribut
0x0000		longueur-de-nom	0 indique 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0004		longueur-de-valeur	longueur d'un entier = 4
0x0008		valeur	3 <sup>ème</sup> valeur pour l'attribut 1setOf entier
0x37	finCollection	étiquette-de-valeur	fin de la collection
0x0000		longueur-de-nom	défini pour être 0 pour ce type, soit partie de 1setOf ; pas de nom (car longueur-de-nom était 0)
0x0000		longueur-de-valeur	défini pour être 0 pour ce type pas de valeur (car longueur-de-valeur était 0)

## Appendice D : Description des documents IPP de base (pour information)

L'ensemble complet des documents IPP comporte :

Objectifs de conception pour un protocole d'impression sur Internet [RFC2567]

Fondements de la structure, modèle et protocole du protocole d'impression sur Internet [RFC2568]

Protocole/1.1 d'impression sur Internet : Modèle et sémantique [RFC2911]

Protocole/1.1 d'impression sur Internet : Codage et transport [RFC2910]

Protocole/1.1 d'impression sur Internet : Guide de mise en œuvre [RFC3196]

Transposition entre les protocoles LPD et IPP [RFC2569]

Le document "Objectifs de conception pour un protocole d'impression sur Internet" porte un large regard sur la fonction d'impression distribuée, et énumère des scénarios réels qui aident à préciser les caractéristiques qu'il est nécessaire d'inclure dans un protocole d'impression pour l'Internet. Il identifie les exigences pour trois types d'utilisateurs : utilisateur final, opérateur, et administrateur. Il fait intervenir un sous ensemble d'exigences d'utilisateur final qui sont satisfaites dans IPP/1.1. Quelques opérations d'opérateur FACULTATIVES ont été ajoutées à IPP/1.1.

Le document "Fondements de la structure, modèle et protocole du protocole d'impression sur Internet" décrit IPP d'un point de vue plus élevé et définit la mission des divers documents qui forment la série des documents de spécification d'IPP. Il donne les fondements des décisions majeures du groupe de travail IPP de l'IETF.

Le document "Protocole/1.1 d'impression sur Internet : Modèle et sémantique" décrit un modèle simplifié avec des objets abstraits, leurs attributs, et leurs opérations qui sont indépendantes du codage et du transport. Le modèle introduit un objet imprimante et un objet tâche. L'objet tâche prend facultativement en charge plusieurs documents par tâche. Il traite aussi des questions de sécurité, d'internationalisation, et d'annuaire.

Le document "Protocole/1.1 d'impression sur Internet : Codage et Transport" est une transposition formelle des opérations et attributs abstraits définis dans le document modèle sur HTTP/1.1 [RFC2616]. Il définit aussi les règles de codage pour un nouveau type de support Internet MIME appelé "application/ipp". Ce document définit aussi les règles de transport sur HTTP, un corps de message dont le type-de-contenu est "application/ipp". Ce document définit le schéma 'ipp' pour l'identification des imprimantes et tâches IPP.

Le document "Protocole/1.1 d'impression sur Internet : Guide de mise en œuvre" donne des perspectives et des conseils pour les mises en œuvre de clients et objets IPP. Il est destiné à aider à

comprendre IPP/1.1 et certaines de ses considérations peuvent faciliter la conception des mises en œuvre client et/ou objet IPP. Par exemple, il donne l'ordre normal de traitement des demandes, y compris la recherche d'erreurs. Il inclut aussi les motivations de certaines des décisions de la spécification.

Le document "Transposition entre protocoles LPD et IPP" donne quelques conseils à ceux qui mettent en œuvre des passerelles entre des mises en œuvre IPP et LPD (*Line Imprimante Daemon*).

## Adresse des auteurs

Roger deBry  
Utah Valley State College  
Orem, UT 84058

Phone: (801) 222-8000

Mél: debryro@uvsc.edu

Tom Hastings  
Xerox Corporation  
737 Hawaii St. ESAE 231  
El Segundo, CA 90245

Phone: 310-333-6413

Fax: 310-333-5514

Mél: hastings@cp10.es.xerox.com

Robert Herriot  
Consultant

706 Colorado Ave  
Palo Alto, CA 94303

Phone: 650-327-4466

Fax: 650-327-4466

Mél: bob@herriot.com

Kirk Ocke  
Xerox Corp.  
800 Phillips Rd  
M/S 128-30E  
Webster, NY 14580  
Phone: (585) 442-4832  
EMail: KOcke@crt.xerox.com

Peter Zehler  
Xerox Corp.  
800 Phillips Rd  
M/S 128-30E  
Webster, NY 14580  
Phone: (585) 265-8755  
EMail: PZehler@crt.xerox.com

Page Web d'IPP : <http://www.pwg.org/ipp/>

Liste de diffusion IPP : [ipp@pwg.org](mailto:ipp@pwg.org)

Pour s'abonner à la liste de diffusion ipp, envoyer le courriel suivant :

- 1) s'adresser à [majordomo@pwg.org](mailto:majordomo@pwg.org)
- 2) laisser la ligne «subject» en blanc
- 3) mettre les deux lignes suivantes dans le corps du message :  
subscribe ipp  
end

Les personnes qui mettent en œuvre le présent document de spécification sont invitées à s'inscrire sur la liste de diffusion IPP afin de participer à toutes les discussions sur les questions à préciser et les révisions de proposition d'enregistrement des attributs et valeurs supplémentaires. Dans le but de réduire les diffusions inutiles, la liste de diffusion rejette les messages des non-abonnés, aussi devez-vous vous inscrire sur la liste de diffusion pour envoyer une question ou un commentaire à la liste de diffusion.

## Déclaration de copyright

Copyright (C) The Internet Society (2002). Tous droits réservés

Le présent document et ses traductions peuvent être copiés et fournis à des tiers, et les travaux dérivés qui le commentent ou l'expliquent ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou en partie, sans restriction d'aucune sorte, pourvu que la déclaration de copyright ci-dessus et le présent paragraphe soient inclus sur de telles copies et

travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, ni en retirant la déclaration de copyright ou les références à la Société Internet ou à d'autres organisations Internet, excepté en tant que de besoin pour le développement des normes Internet auquel cas les procédures de protection des droits de propriété intellectuelle définies dans le traitement des normes de l'Internet doivent être suivies, ou autant qu'il est nécessaire pour la traduction dans d'autres langues que l'anglais.

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

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à [www.rfc-editor.org](http://www.rfc-editor.org), et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et LE CONTRIBUTEUR, L'ORGANISATION QU'IL OU ELLE REPRÉSENTE OU QUI LE/LA FINANCE (S'IL EN EST), LA INTERNET SOCIETY ET LA INTERNET ENGINEERING TASK FORCE DÉCLINENT TOUTES GARANTIES, EXPRIMÉES OU IMPLICITES, Y COMPRIS MAIS NON LIMITÉES À TOUTE GARANTIE QUE L'UTILISATION DES INFORMATIONS CI-ENCLOSES NE VIOLENT AUCUN DROIT OU AUCUNE GARANTIE IMPLICITE DE COMMERCIALISATION OU D'APTITUDE À UN OBJET PARTICULIER.

**Remerciement**

Le financement de la fonction d'édition des RFC est actuellement fourni par Internet Society.