

Groupe de travail Réseau
Request for Comments : 5234
STD 68
RFC rendue obsolète : 4234
Catégorie : Norme
janvier 2008

D. Crocker, éditeur, Brandenburg InternetWorking
P. Overell, THUS plc.

Traduction Claude Brière de L'Isle
mars 2008

BNF augmenté pour les spécifications de syntaxe : ABNF

Statut de ce mémo

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

Résumé

Les spécifications techniques de l'Internet ont souvent besoin de définir une syntaxe formelle. Au cours des années, une version modifiée du formatage Backus-Naur (BNF), appelé BNF augmenté (ABNF), a eu du succès dans de nombreuses spécifications de l'Internet. La présente spécification expose l'ABNF. Elle équilibre la compacité et la simplicité, avec un pouvoir de représentation raisonnable. Les différences entre le BNF standard et l'ABNF touchent les règles de dénomination, de répétition, de remplacement, d'indépendance à l'ordre et les gammes de valeurs. La présente spécification fournit aussi des définitions de règle supplémentaires et le codage pour un analyseur lexical central du type commun à plusieurs spécifications Internet.

Table des matières

1.	Introduction.....
2.	Définition de règle.....
2.1	Dénomination de règle.....
2.2	Forme de règle.....
2.3	Valeurs terminales.....
2.4	Codages externes.....
3.	Opérateurs.....
3.1	Concaténation : Règle1 Règle2.....
3.2	Alternatives : Règle1 / Règle2.....
3.3	Alternatives incrémentaires : Règle1 =/ Règle2.....
3.4	Alternatives de gammes de valeurs : %c##-##.....
3.5	Groupe de séquences : (Règle1 Règle2).....
3.6	Répétition variable : *Règle.....
3.7	Répétition spécifique : nRègle.....
3.8	Séquence facultative : [Règle].....
3.9	Commentaires : ; Commentaire.....
3.10	Préséance d'opérateur.....
4.	Définition ABNF de l'ABNF.....
5.	Considérations pour la sécurité.....
6.	Références.....
6.1	Références normatives.....
6.2	Références informatives.....
Appendice A.	Remerciements.....
Appendice B.	Cœur ABNF de l'ABNF.....
B.1	Cœur des règles.....
B.2	Codage commun.....

1. Introduction

Les spécifications techniques de l'Internet ont souvent besoin de définir une syntaxe formelle et elles ont toute liberté pour utiliser toute notation que leurs auteurs estiment utile. Au cours des années, une version modifiée du formalisme Backus-Naur (BNF), appelée BNF augmenté (ABNF), a été retenue par de nombreuses spécifications de l'Internet. Elle allie compacité et simplicité, avec un pouvoir de représentation raisonnable. Dans les premiers jours de l'Arpanet, chaque spécification contenait sa propre définition de l'ABNF. Cela a inclus les spécifications de messagerie électronique, [RFC733] puis [RFC822], qui en sont venues à être la référence commune pour définir l'ABNF. Le document actuel sépare ces définitions pour permettre une référence sélective. Comme vous pouvez le prévoir, il fournit aussi quelques modifications et améliorations.

Les différences entre le BNF standard et l'ABNF touchent les règles de dénomination, de répétition, de solutions de remplacement, l'indépendance à l'ordre, et les gammes de valeurs. L'appendice B fournit des définitions de règles et des codages pour un analyseur lexical central du type commun à plusieurs spécifications de l'Internet. Il est fourni comme une facilité qui est séparée du méta langage défini dans le corps du présent document, ainsi que de son statut formel.

2. Définition de règle

2.1 Dénomination de règle

Le nom d'une règle est simplement le nom lui-même, c'est-à-dire, une séquence de caractères, commençant par un caractère alphabétique, et suivi par une combinaison de caractères alphabétiques, chiffres, et traits d'union (tirets).

NOTE : Les noms de règle sont insensibles à la casse

Les noms <rulename>, <RuleName>, <RULENAME>, et <rUIENamE> se réfèrent tous à la même règle.

À la différence du BNF d'origine, les crochets ("`<`", "`>`") ne sont pas exigés. Cependant, des crochets peuvent être utilisés autour d'un nom de règle chaque fois que leur présence facilite l'identification de l'utilisation d'un nom de règle. C'est normalement réservé à la référence à un nom de règle en prose en forme libre, ou pour distinguer des règles partielles qui se combinent en une chaîne non séparée par une espace, comme le montre l'exposé sur la répétition, ci-dessous.

2.2 Forme de règle

Une règle est définie par la séquence suivante :

```
name = elements crlf
```

où <name> est le nom de la règle, <elements> est un ou plusieurs noms de règle ou de spécifications terminales, et <crlf> est l'indicateur de fin de ligne (retour chariot suivi par saut à la ligne). Le signe égal sépare le nom de la définition de la règle. Les éléments forment une séquence d'un ou plusieurs noms de règle et/ou définitions de valeur, combinés conformément aux divers opérateurs définis dans le présent document, comme la solution de remplacement et la répétition.

Pour faciliter la visualisation, les définitions de règles sont alignées à gauche. Lorsqu'une règle a besoin de plusieurs lignes, les lignes de continuation sont en retrait. L'alignement à gauche et le retrait sont par rapport à la première ligne de la règle ABNF et n'ont pas besoin de correspondre à la marge gauche du document.

2.3 Valeurs terminales

Les règles se résolvent en une chaîne de valeurs terminales, appelées parfois caractères. En ABNF, un caractère est simplement un entier non négatif. Dans certains contextes sera spécifiée une transposition spécifique (codage) des valeurs en un ensemble de caractères (tel que ASCII).

Les valeurs terminales sont spécifiées par un ou plusieurs caractères numériques, avec l'interprétation de base de ces caractères indiquée explicitement. Les bases suivantes sont actuellement définies :

```
b    = binaire  
d    = décimal
```

x = hexadécimal

Et donc :

CR = %d13
CR = %x0D

spécifient respectivement la représentation décimale et hexadécimale en [US-ASCII] du retour chariot.

Une chaîne concaténée de telles valeurs est spécifiée de façon compacte, en utilisant un point (".") pour indiquer une séparation des caractères au sein de cette valeur. Et donc :

CRLF = %d13.10

L'ABNF permet directement la spécification de chaînes de texte littérales, incluses entre guillemets. Et donc :

command = "command string"

Les chaînes de texte littéral sont interprétées comme un ensemble enchaîné de caractères imprimables.

NOTE : Les chaînes ABNF sont insensibles à la casse et l'ensemble de caractères pour ces chaînes est US-ASCII.

Et donc :

rulename = "abc"

et:

rulename = "aBc"

correspondra à "abc", "Abc", "aBc", "abC", "ABc", "aBC", "AbC", et "ABC".

Pour spécifier une règle qui est sensible à la casse, il faut spécifier individuellement les caractères.

Par exemple :

rulename = %d97 %d98 %d99

ou

rulename = %d97.98.99

ne correspondront qu'à la chaîne qui ne comporte que les caractères minuscules, abc.

2.4 Codages externes

Les représentations externes de caractères de valeur terminale vont varier selon les contraintes de la mémorisation ou de l'environnement de transmission. Et donc, la même grammaire fondée sur ABNF peut avoir plusieurs codages externes, comme un pour l'environnement US-ASCII à 7 bits, un autre pour un environnement d'octets binaires, et encore un autre si on utilise Unicode à 16 bits. Les détails du codages sont en dehors du domaine d'application de l'ABNF, bien que l'Appendice B donne des définitions pour l'environnement US-ASCII à 7 bits qui a été courant dans une grande partie de l'Internet.

En séparant le codage externe de la syntaxe, on entend que les environnements de codage de remplacement puissent être utilisés pour la même syntaxe.

3. Opérateurs

3.1 Concaténation : Règle1 Règle2

Une règle peut définir une chaîne simple, ordonnée, de valeurs (c'est-à-dire, l'enchaînement de caractères contigus) en donnant la liste d'une séquence de noms de règles. Par exemple :

```
foo      = %x61      ; a
bar      = %x62      ; b
mumble   = foo bar foo
```

De sorte que la règle <mumble> corresponde à la chaîne minuscule "aba".

Espace blanche linéaire : La concaténation est au cœur du modèle d'analyse grammaticale de l'ABNF. Une chaîne de caractères contigus (valeurs) est analysée selon les règles définies en ABNF. Pour les spécifications de l'Internet, il a toujours été permis d'insérer librement et implicitement des espaces blancs linéaires (espace et tabulation horizontale) autour des constructions majeures, telles que la délimitation de caractères spéciaux, ou des chaînes atomiques.

NOTE : La présente spécification de l'ABNF ne prévoit pas de spécification implicite d'espace blanche linéaire.

Toute grammaire qui souhaite permettre des espaces blancs linéaires autour des délimiteurs ou des segments de chaînes doit le spécifier explicitement. C'est souvent utile pour fournir de tels espaces blancs dans les règles "centrales" qui sont alors utilisées de façon diverse parmi des règles de niveau supérieur. Les règles "centrales" peuvent être formées dans un analyseur lexical ou faire simplement partie de l'ensemble principal de règles

3.2 Alternatives : Règle1 / Règle2

Les éléments séparés par une barre oblique ("/") sont des alternatives (solutions de remplacement).

Donc, foo / bar acceptera <foo> ou <bar>.

NOTE : Une chaîne entre guillemets qui contient des caractères alphabétiques est une forme particulière pour spécifier des caractères de remplacement et elles est interprétée comme non terminale représentant l'ensemble des chaînes combinatoires avec les caractères contenus, dans l'ordre spécifié mais avec tout arrangement de minuscules et majuscules.

3.3 Alternatives incrémentaires : Règle1 =/ Règle2

Il est parfois pratique de spécifier une liste d'alternatives en fragments. C'est à dire, une règle initiale peut correspondre à une ou plusieurs solutions de remplacement, des définitions de règles ultérieures s'ajoutant à l'ensemble d'alternatives. Ceci est particulièrement utile pour des spécifications par ailleurs indépendantes qui dérivent du même ensemble de règles parentes, comme il survient souvent avec des listes de paramètres. ABNF permet cette définition incrémentaire par la construction :

```
oldrule  =/ additional-alternatives
```

Ainsi, l'ensemble de règles

```
ruleset  = alt1 / alt2
ruleset  =/ alt3
ruleset  =/ alt4 / alt5
```

est la même chose que de spécifier

```
ruleset  = alt1 / alt2 / alt3 / alt4 / alt5
```

3.4 Alternatives de gammes de valeurs : %c##-##

Une gamme de valeurs numériques alternatives peut être spécifiée de façon compacte, en utilisant un trait d'union ("-") pour indiquer la gamme des valeurs de remplacement. Et donc :

```
DIGIT    = %x30-39
```

est équivalent à :

$$\text{DIGIT} = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"$$

Les valeurs numériques enchaînées et les gammes de valeurs numériques ne peuvent pas être spécifiées dans la même chaîne. Une valeur numérique peut utiliser la notation séparée par des points pour l'enchaînement ou elle peut utiliser la notation par trait d'union pour spécifier une gamme de valeurs. Et donc, pour spécifier un caractère imprimable entre des séquences de fin de ligne, la spécification pourrait être :

$$\text{char-line} = \%x0D.0A \%x20-7E \%x0D.0A$$

3.5 Groupe de séquences : (Règle1 Règle2)

Les éléments entre parenthèses sont traités comme un seul élément, dont le contenu est strictement ordonné. Et donc,

$$\text{elem (foo / bar) blat}$$

correspond à (elem foo blat) ou à (elem bar blat), et

$$\text{elem foo / bar blat}$$

correspond à (elem foo) ou à (bar blat).

NOTE : Il est fortement recommandé d'utiliser la notation de groupage, plutôt que de s'appuyer sur la lecture appropriée d'alternatives "nues", lorsque ces alternatives consistent en plusieurs noms ou caractères littéraux de règles.

Et donc, il est recommandé d'utiliser la forme suivante :

$$\text{(elem foo) / (bar blat)}$$

Cela évitera une mauvaise interprétation par les lecteurs peu attentifs.

La notation en groupe de séquences est aussi utilisée dans du texte libre pour faire ressortir une séquence d'éléments du texte.

3.6 Répétition variable : *Règle

L'opérateur "*" précédant un élément indique la répétition. La forme complète est :

$$\langle a \rangle * \langle b \rangle \text{élément}$$

où $\langle a \rangle$ et $\langle b \rangle$ sont des valeurs décimales facultatives, qui indiquent au moins $\langle a \rangle$ et au plus $\langle b \rangle$ occurrences de l'élément.

Les valeurs par défaut sont 0 et l'infini de sorte que $*\langle \text{élément} \rangle$ permet tout nombre, y compris zéro; $1*\langle \text{élément} \rangle$ exige au moins un ; $3*3\langle \text{élément} \rangle$ permet exactement 3 et $1*2\langle \text{élément} \rangle$ permet un ou deux.

3.7 Répétition spécifique : nRègle

Une règle de la forme :

$$\langle n \rangle \text{élément}$$

est équivalente à

$$\langle n \rangle * \langle n \rangle \text{élément}$$

C'est à dire, exactement $\langle n \rangle$ occurrences de $\langle \text{élément} \rangle$. Et donc, 2DIGIT est un nombre à deux chiffres, et 3ALPHA est une chaîne de trois caractères alphabétiques.

3.8 Séquence facultative : [Règle]

Les crochets enclosent une séquence d'élément facultatif :

[foo bar]

est équivalent à

*1(foo bar).

3.9 Commentaires : ; Commentaire

Un point virgule débute un commentaire qui continue jusqu'à la fin de ligne. C'est une façon simple d'inclure des notes utiles en parallèle aux spécifications.

3.10 Préséance d'opérateur

Les divers mécanismes décrits ci-dessus ont la préséance suivante, de la plus élevée (la plus forte contrainte) au sommet, à la plus basse (la plus souple) en bas :

Nom de règle, valeur en prose, valeur terminale
 Commentaire
 Gamme de valeur
 Répétition
 Groupage, facultatif
 Concaténation
 Alternative

L'utilisation de l'opérateur alternatif, librement mêlé à des concaténations, peut entraîner des confusions.

Il est à nouveau recommandé d'utiliser l'opérateur de groupage pour rendre explicites les groupes enchaînés.

4. Définition ABNF de l'ABNF

NOTES :

1. Cette syntaxe exige un formatage des règles qui est relativement strict. Et donc, la version d'un ensemble de règles qui est incluse dans une spécification peut nécessiter un prétraitement pour s'assurer qu'elle peut être interprétée par un analyseur ABNF.
2. Cette syntaxe utilise les règles fournies à l'Appendice B.

```

rulelist = 1*( rule / (*c-wsp c-nl) )

rule     = rulename defined-as elements c-nl
           ; continue si la ligne suivante commence par une espace blanche

rulename = ALPHA *(ALPHA / DIGIT / "-")

defined-as= *c-wsp ("=" / "=") *c-wsp
           ; définition des règles de base et alternatives incrémentaires

elements = alternation *c-wsp

c-wsp    = WSP / (c-nl WSP)

c-nl     = comment / CRLF
           commentaire ou nouvelle ligne
  
```

comment = ";" *(WSP / VCHAR) CRLF

alternation = concatenation *(**c-wsp* "/" **c-wsp* concatenation)

concatenation = repetition *(*1*c-wsp* repetition)

repetition = [repeat] element

repeat = *1**DIGIT / (**DIGIT* "*" **DIGIT*)

element = rulename / group / option / char-val / num-val / prose-val

group = "(" **c-wsp* alternation **c-wsp* ")"

option = "[" **c-wsp* alternation **c-wsp* "]"

char-val = DQUOTE *(%x20-21 / %x23-7E) DQUOTE
; chaîne entre guillemets de SP et VCHAR sans DQUOTE

num-val = "%" (bin-val / dec-val / hex-val)

bin-val = "b" *1**BIT [*1**("." *1**BIT) / ("-" *1**BIT)]
; séries de valeurs binaires concaténés ou d'une seule gamme ONEOF

dec-val = "d" *1**DIGIT [*1**("." *1**DIGIT) / ("-" *1**DIGIT)]

hex-val = "x" *1**HEXDIG [*1**("." *1**HEXDIG) / ("-" *1**HEXDIG)]

prose-val = "<" *(%x20-3D / %x3F-7E) ">"
; chaîne entre crochets angulaires de SP et VCHAR sans angles
; description en prose, à utiliser en dernier lieu

5. Considérations pour la sécurité

La sécurité n'est pas considérée comme une question pertinente pour le présent document.

6 Références

6.1 Références normatives

[US-ASCII] American National Standards Institute, "Ensemble de caractères codés – Norme de code américaine à 7 bits pour les échanges d'informations", ANSI X3.4, 1986.

6.2 Références informatives

[RFC733] D. Crocker, J. Vittal, K. Pogran, et D. Henderson, "Norme pour le format des messages de texte du réseau ARPA", novembre 1977.

[RFC822] D. Crocker, "Norme pour le format des messages de texte ARPA Internet", STD 11, août 1982.

Appendice A. Remerciements

La syntaxe de l'ABNF a été spécifiée à l'origine dans la RFC 733. Ken L. Harrenstien, de SRI International, était responsable du recodage du BNF en un BNF augmenté qui rend la représentation plus brève et plus facile à comprendre.

Ce projet a commencé comme un simple effort pour recueillir la portion de la RFC 822 qui est toujours citée par les auteurs de spécifications étrangères à la messagerie électronique, à savoir la description du BNF augmenté. Plutôt que de simplement convertir aveuglément le texte existant en un document distinct, le groupe de travail a choisi de prêter une considération attentive aux forces et aux faiblesses de la spécification existante et aux spécifications en rapport publiées depuis les quinze dernières années, et donc d'en poursuivre l'amélioration. Ceci a amené le projet sur des voies plus ambitieuses que ce qui était prévu au départ. Il est intéressant de constater que le résultat n'est pas très différent de l'original, bien que des décisions, telles que le retrait de la notation de liste, aient créé la surprise.

Cette version "séparée" de la spécification est l'œuvre du groupe de travail DRUMS, avec des contributions significatives de Jerome Abela, Harald Alvestrand, Robert Elz, Roger Fajman, Aviva Garrett, Tom Harsch, Dan Kohn, Bill McQuillan, Keith Moore, Chris Newman, Pete Resnick, et Henning Schulzrinne.

Julian Reschke mérite des remerciements particuliers pour la conversion du projet de norme en format source XML.

Appendice B. Cœur ABNF de l'ABNF

Cet appendice contient certaines règles de base qui sont d'usage courant. Les règles de base sont en majuscules. Noter que ces règles ne sont valides que pour l'ABNF codé en ASCII à 7 bits ou dans des jeux de caractères qui sont un surensemble de l'ASCII à 7 bits.

B.1 Cœur des règles

Certaines règles de base sont en majuscules, comme SP, HTAB, CRLF, DIGIT, ALPHA, etc.

ALPHA	= %x41-5A / %x61-7A ; A-Z / a-z
BIT	= "0" / "1"
CHAR	= %x01-7F ; tout caractère US-ASCII à 7 bits, à l'exclusion de NUL
CR	= %x0D ; retour chariot
CRLF	= CR LF ; nouvelle ligne standard de l'Internet
CTL	= %x00-1F / %x7F ; contrôles
DIGIT	= %x30-39 ; 0 à 9
DQUOTE	= %x22 ; " (double guillemets)
HEXDIG	= DIGIT / "A" / "B" / "C" / "D" / "E" / "F"
HTAB	= %x09 ; tabulation horizontale
LF	= %x0A ; saut à la ligne
LWSP	= *(WSP / CRLF WSP) ; L'utilisation de cette règle espace-blanche-linéaire permet des lignes ne contenant que des ; espaces blanches qui ne sont plus légales dans les en-têtes de messagerie et ont causé des ; problèmes d'interopérabilité dans d'autres contextes. À ne pas utiliser lors de la définition ; d'en-têtes de messagerie et à utiliser avec prudence dans les autres contextes.

OCTET	= %x00-FF ; 8 bits de données
SP	= %x20
VCHAR	= %x21-7E ; caractères visibles (imprimables)
WSP	= SP / HTAB ; espace blanche

B.2 Codage commun

En externe, les données sont représentées comme "ASCII virtuel de réseau" (à savoir, de l'US-ASCII à 7 bits dans un champ de 8 bits), avec le bit de plus fort poids (le huitième) mis à zéro. Une chaîne de valeurs est dans "l'ordre des octets du réseau", dans lequel les octets de plus fort poids sont représentés sur le côté gauche et sont envoyés en premier sur le réseau.

Adresse des auteurs

Dave Crocker (éditeur)
Brandenburg InternetWorking
675 Spruce Dr.
Sunnyvale, CA 94086
US
téléphone : +1.408.246.8253
mél : dcrocker@bbiw.net

Paul Overell
THUS plc.
1/2 Berkeley Square
99 Berkeley Street
Glasgow G3 7HR
UK
mél : paul.overell@thus.net

Déclaration de copyright

Copyright (C) The IETF Trust (2008).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, 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, le IETF TRUST 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.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.