

[White Paper]

**T-Engine Forum**  
Ubiquitous ID Center  
Specification  
**DRAFT**

910-S004-0.00.05/UID-CO00003-0.00.05

2006-10-12

---

uBNF: ユビキタス ID アーキテクチャ仕様のための  
BNF 記法  
uBNF: BNF for ubiquitous ID architecture



---

Number: 910-S004-0.00.05/UID-CO00003-0.00.05  
Title: uBNF: ユビキタス ID アーキテクチャ仕様のための BNF 記法  
uBNF: BNF for ubiquitous ID architecture  
Status:  Working Draft,  Final Draft for Voting,  Standard  
Date: 2006/10/12

Copyright (C) 2006, Ubiquitous ID Center, T-Engine Forum, all rights reserved.

---

---

## 目次 (Table of Contents)

---

はじめに.....	5
規定範囲.....	5
本書の位置付け.....	5
参照規定.....	5
用語定義.....	5
1. 規則定義.....	6
1.1. 命名規則.....	6
1.2. 規則形式.....	6
1.3. 文字.....	6
2. 演算子定義.....	8
2.1. 名前生成.....	8
2.2. 結合.....	8
2.3. 選択.....	8
2.4. 逐次選択.....	9
2.5. 値範囲選択.....	9
2.6. 順序グループ.....	9
2.7. 可変回の繰り返し.....	10
2.8. 特定繰り返し.....	10
2.9. オプション.....	10
2.10. コメント.....	11
2.11. 演算子の優先度.....	11
3. uBNF による uBNF 定義と予約規則名.....	12
3.1. 核となる規則.....	12
3.2. uBNF の uBNF による定義.....	12

# uBNF: ユビキタス ID アーキテクチャ仕様のための

## BNF 記法

uBNF: BNF for ubiquitous ID architecture

---

### はじめに

---

#### 規定範囲

本仕様書は、ユビキタス ID アーキテクチャの各仕様書でプロトコルやフォーマットを記述する際の記法を規定するものである。

#### 本書の位置付け

本仕様書は、ユビキタス ID アーキテクチャの各仕様書がプロトコルやフォーマットを記述する際の形式を規定するものである。プロトコルやフォーマットを記述する際に、BNF (Backus-Naur Form) 記法が一般的に用いられるが、その記述方式は種々に拡張されている。本仕様書はその BNF 拡張方式のうちの Augmented BNF[1]を基本とする。ただし、その一部の記法を拡張する。

#### 参照規定

[1] The Internet Engineering Task Force (IETF), “Augmented BNF for Syntax Specifications: ABNF”, RFC4284, 2005.

---

### 用語定義

---

本文中で定義する。

---

## 1. 規則定義

---

本章では、ユビキタス ID アーキテクチャ仕様が使用する BNF 形式 (uBNF) の規則を規定する。本規則は基本的に BNF[1] と同一であるが、規則名として使用できる文字にアンダーバー “\_” を追加した。

### 1.1. 命名規則

規則の名前を**規則名** (Rule Name) という。規則名はその名前自身である。すなわち、アルファベット文字から始まり、アルファベットと数字文字とハイフン (ダッシュ) とアンダーバーが続く列である。

オリジナルの BNF では規則名に山括弧 (“<”, “>”) を要するが、本規則では必ずしも必要としない。ただし、それが規則名であることを明示するために、規則名を山括弧 (“<”, “>”) でくくって表現することがある。

規則名は大文字・小文字を区別しない。つまり、<name>, <Name>, <nAmE> はすべて同じ規則を表す。

### 1.2. 規則形式

規則は次の列で定義される。

$$\langle \text{name} \rangle = \langle \text{elements} \rangle \langle \text{crlf} \rangle$$

ここで、<name> は規則名であり、<elements> は 1 つ以上の規則名または文字、<crLf> は「行末」表示 (キャリッジリターンに続くラインフィード) である。等号は規則名と規則の定義を分離するものである。<elements> は本仕様が規定する種々の演算子で結合された、1 つ以上の規則名や値定義の列であり、これを要素と呼ぶ。<name> と等号、ならびに等号と <elements> の間に 0 つ以上の空白スペース (空白または水平タブ) を含めてもよい。

規則は左詰めで標記する。規則が 2 行以上を要するときは、継続行は 1 つ以上の空白スペースを補って字下げする。左詰めと字下げは、規則定義の最初の行と関係がある。すなわち、規則の左詰めとは、文書の左端から記述することとは限らない。

### 1.3. 文字

規則は、最終的に文字と呼ばれる列に変換される。文字とは非負整数をいう。文字は以下のいずれかの方法で記述される。

- (1) リテラルテキスト文字列を用いた表現

引用符で囲われたリテラルテキスト文字列は大文字・小文字を区別しない。

い文字である。たとえば“abc”は、“abc”、“Abc”、“aBc”、“abC”、“ABc”、“aBC”、“AbC”、“ABC”のいずれかに一致する文字列である。大文字・小文字の区別を要する場合は、次の「値範囲選択」演算子%を用いた表現を用いること。

(2) 「値範囲選択」演算子%を用いた表現

「値範囲選択」演算子%を用いて文字を表現する。この形式で表現する文字のベースは以下の3通りである。これをベース指定子と呼ぶ。

“b”	2進数標記
“d”	10進数標記
“x”	16進数標記

たとえば、アスキーコード(US-ASCII)におけるキャリッジリターン文字は、%d13または%x0Dと表現する。

---

## 2. 演算子定義

---

本章では、規則を定義する際に使用する演算子(Operator)を定義する。演算子は基本的に BNF[1]と同一であるが、選択演算子に縦棒("|", %0x2F)を追加してある。

### 2.1. 名前生成

<name>という名前が<elements>という要素の集合であると規定する規則を**名前生成**(Names formation)と呼ぶ。名前生成は、名前と要素の集合を等号で結んで表現する。すなわち、以下の通りである。

$$\langle \text{name} \rangle = \langle \text{elements} \rangle$$

### 2.2. 結合

<rule1>と<rule2>をこの順に並べることを<rule1>と<rule2>の**結合**(Concatenation)という。<rule1>と<rule2>の結合は、その2つの規則の間に1つ以上の空白スペース(空白または水平タブ)を補った列で表現する。すなわち、以下の通りである。

$$\langle \text{rule1} \rangle \langle \text{rule2} \rangle$$

たとえば、以下のように定義される<aba>は、小文字文字列“aba”に一致する。

```
<rule1> = %x61           ; a
<rule2> = %x62           ; b
<aba>   = <rule1> <rule2> <rule1>
```

### 2.3. 選択

<rule1>と<rule2>のいずれか一方をとることを<rule1>と<rule2>の**選択**(Alternatives)という。<rule1>と<rule2>の選択は、その2つの規則の間に縦棒("|", %x7c)またはフォワードスラッシュ("/", %x2F)を補った列で表現する。すなわち、以下の通りである。

$$\langle \text{rule1} \rangle | \langle \text{rule2} \rangle$$
$$\langle \text{rule1} \rangle / \langle \text{rule2} \rangle$$

たとえば、以下のように定義される<aorb>は、小文字文字列“a”または小文字文字列“b”に一致する。

```
<rule1> = %x61           ; a
<rule2> = %x62           ; b
<aorb>  = <rule1> | <rule2>
```

ただし、1 規則に使用する選択演算子は、“|”または“/”いずれかであること。両方を混在させてはならない。

## 2.4. 逐次選択

最初の規則がいくつかの選択肢に一致し、後の規則が選択を付け加える場合、後の規則を**逐次選択**(Incremental Alternatives)と呼ぶ。逐次選択は規則名と規則列を分割する等号“=”と、選択演算子である“|”または“/”をこの順に並べて表現する。すなわち、以下の通りである。

$\langle \text{oldrule} \rangle = | \langle \text{additional-alternatives} \rangle$

$\langle \text{oldrule} \rangle = / \langle \text{additional-alternatives} \rangle$

たとえば、規則集合

$\langle \text{ruleset} \rangle = \langle \text{rule1} \rangle | \langle \text{rule2} \rangle$

$\langle \text{ruleset} \rangle = | \langle \text{rule3} \rangle$

は、次の規則と等価である。

$\langle \text{ruleset} \rangle = \langle \text{rule1} \rangle | \langle \text{rule2} \rangle | \langle \text{rule3} \rangle$

ただし、1 規則に使用する選択演算子は、“|”または“/”いずれかであること。両方を混在させてはならない。

## 2.5. 値範囲選択

ある有限数のいずれかの文字を表すことを**値範囲選択**(Value Range)と呼ぶ。値範囲選択は、パーセント記号“%”，ベース指定子，下限値，ドット(“-”，%x2D)，上限値をこの順に並べた列で表現する。すなわち、以下の通りである。

$\%c\#\#-\#\#$

上限値と下限値が等しい場合、ハイフン“-”，上限値を省略できる。また、値範囲が連続値でない場合は、それぞれの値をドット(“.”，%x2E)で区切って標記する。

例えば、以下の3規則は等価である。

$\langle \text{DIGIT} \rangle = \%x30-39$

$\langle \text{DIGIT} \rangle = \%x30.31.32.33.34.35.36.37.38.39$

$\langle \text{DIGIT} \rangle = \text{“0”} | \text{“1”} | \text{“2”} | \text{“3”} | \text{“4”} |$   
 $\text{“5”} | \text{“6”} | \text{“7”} | \text{“8”} | \text{“9”}$

## 2.6. 順序グループ

厳密に順序づけられた規則の固まりを**順序グループ**(Sequence Group)と呼ぶ。順序グループはその固まりを括弧で囲って表現する。すなわち、以下の通りである。

$(\langle \text{rule1} \rangle \langle \text{rule2} \rangle)$

たとえば、以下のように定義される<rule0>は、<rule1> <rule2> <rule4> または <rule1> <rule3> <rule4>に一致する。

$$\langle \text{rule0} \rangle = \langle \text{rule1} \rangle (\langle \text{rule2} \rangle | \langle \text{rule3} \rangle) \langle \text{rule4} \rangle$$

## 2.7. 可変回の繰り返し

同じ要素がある指定された範囲の回数繰り返されることを**可変回の繰り返し** (Variable Repetition)とよぶ。可変回の繰り返しは、繰り返される要素の前に繰り返し回数の下限の 10 進数表現、アスタリスク(“\*”, %x2A), 繰り返し回数の上限の 10 進数表現をこの順に並べて表現する。すなわち<a>, <b>を繰り返し回数を示す 10 進数表現として以下の通りである。

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

ただし、下限値が 0 であるときは下限値を、上限値が無限大であるときは上限値を省略する。

例えば、以下のように定義される<zeros>は、“0”, “00”, “000”のいずれかに一致する。

$$\langle \text{zeros} \rangle = 1 * 3 "0"$$

## 2.8. 特定繰り返し

同じ要素がある指定された回数繰り返されることを**特定繰り返し** (Specific Repetition)とよぶ。特定繰り返しは、繰り返される要素の前に繰り返し回数の 10 進数表現を補って表現する。すなわち<n>を繰り返し回数を示す 10 進数表現として以下の通りである。

$$\langle n \rangle \langle \text{element} \rangle$$

これは、<n>\*<n><element>と等価である。

## 2.9. オプション

任意に含められる要素を**オプション** (Optional Sequence)と呼ぶ。オプションは、その要素または要素の組を角括弧で囲って表現する。すなわち、以下の通りである。

$$[\langle \text{rule1} \rangle \langle \text{rule2} \rangle]$$

これは、\*1(<rule1> <rule2>)と等価である。

## 2.10. コメント

人間が読むために、規則を自然言語で補完したものを**コメント** (Comment)と呼ぶ。コメントはセミコロン(“;”, %x3B)から行末までの文字である。

## 2.11. 演算子の優先度

本章で定義した演算子は、次に示す優先度をもつ。上にあるものほど優先度が高い、すなわち結合が強い。

- 文字列・名前生成
- コメント
- 値範囲
- 繰り返し
- 順序グループ・オプション
- 結合
- 選択

### 3. uBNF による uBNF 定義と予約規則名

本章では、本仕様で規定した uBNF に基づいて uBNF を規定する。なお、本章で規定する規則名は uBNF 予約とする。

#### 3.1. 核となる規則

アルファベット、10進数標記などの uBNF の核となる規則を定義する。これは、[1] に記述されている核規則(Core Rules)と同一である。

ALPHA	=	%x41-5A   %x61-7A	; A-Z a-z
BIT	=	“0”   “1”	
CHAR	=	%x01-7F	; NULL を除く ASCII 文字
CR	=	%x0D	
CRLF	=	CR LF	
CTL	=	%x00-1F   %x7F	; 制御文字
DIGIT	=	%x30-39	; 0-9
DQUOTE	=	%x22	; “(ダブルクォート)
HEXDIG	=	DIGIT   “A”   “B”   “C”   “D”   “E”   “F”	
HTAB	=	%x09	; 水平タブ
LF	=	%x0A	; ラインフィード
LWSP	=	*(WSP   CRLF WSP)	; 連続空白スペース
OCTET	=	%x00-FF	; 8 ビットデータ
SP	=	%x20	
VCHAR	=	%x21-7E	; 印刷可能文字
WSP	=	SP   HTAB	; 空白スペース

#### 3.2. uBNF の uBNF による定義

uBNF の文法定義を uBNF によって記述する。これは、基本的に[1]の「ABNF による ABNF の定義」と同一だが、本仕様による拡張分を追記してある。

<rulelist>	=	1*(<rule>   (*<c-wsp> <c-nl>))
<rule>	=	<rulename> <defined-as> <elements> <c-nl>
<rulename>	=	ALPHA *(ALPHA   DIGIT   “-”   “_”)
<defined-as>	=	*<c-wsp> (“=”   “=”   “=/”) *<c-wsp>
<elements>	=	<alternation> *<c-wsp>

<c-wsp> = WSP | (<c-nl> WSP)  
 <c-nl> = <comment> | CRLF  
 <comment> = “;” \*(WSP | VCHAR) CRLF  
 <alternation> = <concatenation> \*(\*<c-wsp> <concatenation>)  
 <concatenation> = <repetition> \*(1\*<c-wsp> <repetition>)  
 <repetition> = [<repeat>] <repeat>  
 <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  
                   ; ダブルクォートを除く VCHAR または  
                   ; スペースからなる文字列  
 <num-val> = “%” (<bin-val> | <dec-val> | <hex-val>)  
 <bin-val> = “b” 1\*BIT [1\*(“.” 1\*BIT) | (“-” 1\*BIT)]  
 <dec-val> = “d” 1\*DIGIT [1\*(“.” 1\*DIGIT) | (“-” 1\*DIGIT)]  
 <hex-val> = “x” 1\*HEXDIG  
                   [1\*(“.” 1\*HEXDIG) | (“-” 1\*HEXDIG)]  
 <prose-val>= “<” \*(%20-3D | %x3F-7E) “>”

---

## 索引

---

<b>あ</b>	<b>こ</b>
値範囲選択 .....10	コメント .....11
<b>お</b>	<b>し</b>
オプション .....11	順序グループ .....10
<b>か</b>	<b>せ</b>
可変回の繰り返し .....11	選択 .....9
<b>き</b>	<b>ち</b>
規則名 .....7	逐次選択 .....10
<b>く</b>	<b>と</b>
繰り返し	特定繰り返し .....11
可変回の繰り返し .....11	
特定繰り返し .....11	
<b>け</b>	<b>な</b>
結合 .....9	名前生成 .....9
	<b>へ</b>
	ベース指定子 .....8

