

---

# rfc4515

## Représentation des chaînes dans les filtres de recherche

## Définition d'un filtre de recherche

```
Filter ::= CHOICE {  
  and [0] SET SIZE (1..MAX) OF filter Filter,  
  or [1] SET SIZE (1..MAX) OF filter Filter,  
  not [2] Filter,  
  equalityMatch [3] AttributeValueAssertion,  
  substrings [4] SubstringFilter,  
  greaterOrEqual [5] AttributeValueAssertion,  
  lessOrEqual [6] AttributeValueAssertion,  
  present [7] AttributeDescription,  
  approxMatch [8] AttributeValueAssertion,  
  extensibleMatch [9] MatchingRuleAssertion }
```

```
SubstringFilter ::= SEQUENCE {  
  type AttributeDescription, initial and final can occur at most once  
  substrings SEQUENCE SIZE (1..MAX) OF substring CHOICE {  
    initial [0] AssertionValue,  
    any [1] AssertionValue,  
    final [2] AssertionValue } }
```

```
AttributeValueAssertion ::= SEQUENCE {  
  attributeDesc AttributeDescription,  
  assertionValue AssertionValue }
```

```
MatchingRuleAssertion ::= SEQUENCE {  
  matchingRule [1] MatchingRuleId OPTIONAL,  
  type [2] AttributeDescription OPTIONAL,  
  matchValue [3] AssertionValue,  
  dnAttributes [4] BOOLEAN DEFAULT FALSE }
```

```
AttributeDescription ::= LDAPString Constrained to <attributedescription> [RFC4512]
```

```
AttributeValue ::= OCTET STRING
```

```
MatchingRuleId ::= LDAPString
```

```
AssertionValue ::= OCTET STRING
```

```
LDAPString ::= OCTET STRING UTF-8 encoded, [Unicode] characters
```

**AttributeDescription** est une représentation chaîne d'une description d'attribut  
**AttributeValue** et **AssertionValue** ont une forme définie dans la RFC4517

## Définition des filtres de recherche

---

Un filtre LDAP est une chaîne UTF8 définis par la grammaire suivante (notation ABNF)

```
filter = LPAREN filtercomp RPAREN
filtercomp = and / or / not / item
and = AMPERSAND filterlist
or = VERTBAR filterlist
not = EXCLAMATION filter
filterlist = 1*filter
item = simple / present / substring / extensible
simple = attr filtertype assertionvalue
filtertype = equal / approx / greaterorequal / lessorequal
equal = EQUALS
approx = TILDE EQUALS
greaterorequal = RANGLE EQUALS
lessorequal = LANGLE EQUALS
extensible = ( attr [dnattrs]
  [matchingrule] COLON EQUALS assertionvalue )
  / ( [dnattrs]
  matchingrule COLON EQUALS assertionvalue )
present = attr EQUALS ASTERISK
substring = attr EQUALS [initial] any [final]
initial = assertionvalue
any = ASTERISK *(assertionvalue ASTERISK)
final = assertionvalue
attr = attributedescription
dnattrs = COLON "dn"
matchingrule = COLON oid
assertionvalue = valueencoding
valueencoding = 0*(normal / escaped)
normal = UTF1SUBSET / UTFMB
escaped = ESC HEX HEX
UTF1SUBSET = %x01-27 / %x2B-5B / %x5D-7F
  ; UTF1SUBSET excludes 0x00 (NUL), LPAREN,
  ; RPAREN, ASTERISK, and ESC.
EXCLAMATION = %x21 ; exclamation mark ("!")
AMPERSAND = %x26 ; ampersand (or AND symbol)("&")
ASTERISK = %x2A ; asterisk ("*")
COLON = %x3A ; colon (":")
VERTBAR = %x7C ; vertical bar (or pipe) ("|")
TILDE = %x7E ; tilde ("~")
```

## Exemples

**cn=Babs Jensen)**

**(!(cn=Tim Howes))**

**(&(objectClass=Person)(!(sn=Jensen)(cn=Babs J\*)))**

**(o=univ\*of\*mich\*)**

**(seeAlso=)**

utilisation du matching rule "caseExactMatch" :

**(cn :caseExactMatch :=Fred Flintstone)**

utilisation d'un MatchingRuleAssertion sans matchingRule :

**(cn :=Betty Rubble)**

Utilisation de la notation :oid :

**(sn :dn :2.4.6.8.10 :=Barney Rubble)**

equality match, excepté que le DN devrait faire partie de l'entrée :

---

**(o :dn :=Ace Industry)**

exemples de matching rules :

**( :1.2.3 :=Wilma Flintstone)**

**( :DN :2.4.6.8.10 :=Dino)**

utilisation du mécanisme d'échappement :

**(o=Parens R Us \28for all your parenthetical needs\29)**

**(cn=\*\2A\*)**

**(filename=C :\5cMyFile)**

**(bin=\00\00\00\04)**

**(sn=Lu\c4\8di\c4\87)**

**(1.3.6.1.4.1.1466.0=\04\02\48\69)**