

Network Working Group
Request for Comments: DRAFT
Obsoletes: RFC [2254](#)
Expires: 28 August 2003

M. Smith, Editor
Netscape Communications Corp.
T. Howes
Opsware, Inc.
28 February 2003

LDAP: String Representation of Search Filters
<[draft-ietf-ldapbis-filter-04.txt](#)>

1. Status of this Memo

This document is an Internet-Draft and is subject to all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/1id-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Discussion of this document should take place on the LDAP (v3) Revision (ldapbis) Working Group mailing list <ietf-ldapbis@openldap.org>.

Copyright (C) The Internet Society (2003). All Rights Reserved.

2. Abstract

LDAP search filters are transmitted in the LDAP protocol using a binary representation that is appropriate for use on the network. This document defines a human-readable string representation of LDAP search filters that is appropriate for use in LDAP URLs and in other applications.

3. Table of Contents

1.	Status of this Memo.....	1
2.	Abstract.....	1
3.	Table of Contents.....	2
4.	Introduction.....	2
5.	LDAP Search Filter Definition.....	2
6.	String Search Filter Definition.....	4
7.	Examples.....	5
8.	Security Considerations.....	7
9.	Normative References.....	7
10.	Informative References.....	8
11.	Acknowledgments.....	8
12.	Authors' Address.....	8
13.	Full Copyright Statement.....	9
14.	Appendix A : Changes Since RFC 2254	9
14.1.	Technical Changes.....	9
14.2.	Editorial Changes.....	10
15.	Appendix B : Changes Since Previous Document Revision.....	11
15.1.	Technical Changes.....	11
15.2.	Editorial Changes.....	11

4. Introduction

The Lightweight Directory Access Protocol (LDAP) [[Protocol](#)] defines a network representation of a search filter transmitted to an LDAP server. Some applications may find it useful to have a common way of representing these search filters in a human-readable form; LDAP URLs are an example of one such application. This document defines a human-readable string format for representing the full range of possible LDAP version 3 search filters, including extended match filters.

This document is an integral part of the LDAP Technical Specification [[Roadmap](#)].

This document replaces [RFC 2254](#). Changes to [RFC 2254](#) are summarized in [Appendix A](#).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)].

5. LDAP Search Filter Definition

An LDAPv3 search filter is defined in Section 4.5.1 of [[Protocol](#)] as follows:


```

Filter ::= CHOICE {
    and                [0] SET SIZE (1..MAX) OF Filter,
    or                 [1] SET SIZE (1..MAX) OF 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,
    -- at least one must be present,
    -- initial and final can occur at most once
    substrings SEQUENCE OF 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
    -- [Models]

AttributeValue ::= OCTET STRING

MatchingRuleId ::= LDAPString

AssertionValue ::= OCTET STRING

LDAPString ::= OCTET STRING -- UTF-8 encoded,
    -- ISO 10646 characters

```

where the LDAPString above is limited to the UTF-8 encoding [[RFC2279](#)] of the ISO 10646 character set [[ISO10646](#)]. The AttributeDescription is a string representation of the attribute description and is defined in [[Protocol](#)]. The AttributeValue and AssertionValue OCTET

STRING have the form defined in [[Syntaxes](#)]. The Filter is encoded for transmission over a network using the Basic Encoding Rules defined in [[ASN.1](#)], with simplifications described in [[Protocol](#)].

6. String Search Filter Definition

The string representation of an LDAP search filter is a string of UTF-8 encoded ISO 10646-1 characters that is defined by the following grammar, following the ABNF notation defined in [[RFC2234](#)]. The productions used that are not defined here are defined in [section 1.3](#) (Common ABNF Productions) of [[Models](#)] unless otherwise noted. The filter format uses a prefix notation.

```

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 / greater / less
equal            = EQUALS
approx           = TILDE EQUALS
greater          = RANGLE EQUALS
less             = LANGLE EQUALS
extensible       = attr [dnattrs] [matchingrule] COLON EQUALS
assertionvalue   = attr [dnattrs] matchingrule COLON EQUALS assertionvalue
                  / COLON EQUALS assertionvalue
present          = attr EQUALS ASTERIX
substring        = attr EQUALS [initial] any [final]
initial          = assertionvalue
any              = ASTERIX *(assertionvalue ASTERIX)
final            = assertionvalue
attr             = attributedescription
                  ; The attributedescription rule is defined in
                  ; Section 2.5 of [Models].
dnattrs          = COLON "dn"
matchingrule     = COLON oid
assertionvalue   = valueencoding
                  ; The <valueencoding> rule is used to encode an
                  ; <AssertionValue> from Section 4.1.6 of [Protocol].
valueencoding    = 0*(normal / escaped)
normal           = UTF1SUBSET / UTFMB
escaped          = ESC HEX HEX
UTF1SUBSET       = %x01-27 / %x2B-5B / %x5D-7F
                  ; UTF1SUBSET excludes 0x00 (NUL), LPAREN,
```

; RPAREN, ASTERIX, and ESC.

EXCLAMATION	= %x21 ; exclamation mark ("!")
AMPERSAND	= %x26 ; ampersand (or AND symbol)("&")
ASTERIX	= %x2A ; asterix ("*")
COLON	= %x3A ; colon (":")
VERTBAR	= %x7C ; vertical bar (or pipe) (" ")
TILDE	= %x7E ; tilde ("~")

Note that although both the <substring> and <present> productions in the grammar above can produce the "attr=" construct, this construct is used only to denote a presence filter.

The <valueencoding> rule ensures that the entire filter string is a valid UTF-8 string and provides that the octets that represent the ASCII characters "*" (ASCII 0x2a), "(" (ASCII 0x28), ")" (ASCII 0x29), "\" (ASCII 0x5c), and NUL (ASCII 0x00) are represented as a backslash "\" (ASCII 0x5c) followed by the two hexadecimal digits representing the value of the encoded octet.

This simple escaping mechanism eliminates filter-parsing ambiguities and allows any filter that can be represented in LDAP to be represented as a NUL-terminated string. Other octets that are part of the <normal> set may be escaped using this mechanism, for example, non-printing ASCII characters.

For AssertionValues that contain UTF-8 character data, each octet of the character to be escaped is replaced by a backslash and two hex digits, which form a single octet in the code of the character.

For example, the filter checking whether the "cn" attribute contained a value with the character "*" anywhere in it would be represented as "(cn=*\2a*)".

As indicated by the valueencoding rule, implementations MUST escape all octets greater than 0x7F that are not part of a valid UTF-8 encoding sequence when they generate a string representation of a search filter. Since [RFC 2254](#) does not clearly define the term "string representation" (and in particular does mention that the string representation of an LDAP search filter is a string of UTF-8 encoded ISO 10646-1 characters) implementations SHOULD accept as input strings that include invalid UTF-8 octet sequences.

7. Examples

This section gives a few examples of search filters written using this notation.

```
(cn=Babs Jensen)
```



```
(!(cn=Tim Howes))
(&(objectClass=Person)(|(sn=Jensen)(cn=Babs J*)))
(o=univ*of*mich*)
(seeAlso=)
```

The following examples illustrate the use of extensible matching.

```
(cn:1.2.3.4.5:=Fred Flintstone)
(cn:=Betty Rubble)
(sn:dn:2.4.6.8.10:=Barney Rubble)
(o:dn:=Ace Industry)
(:1.2.3:=Wilma Flintstone)
(:dn:2.4.6.8.10:=Dino)
(::=Fred Flintstone)
```

The first example shows use of the matching rule "1.2.3.4.5".

The second example demonstrates use of a MatchingRuleAssertion form without a matchingRule.

The third example illustrates the use of the ":dn" notation to indicate that matching rule "2.4.6.8.10" should be used when making comparisons, and that the attributes of an entry's distinguished name should be considered part of the entry when evaluating the match.

The fourth example denotes an equality match, except that DN components should be considered part of the entry when doing the match.

The fifth example is a filter that should be applied to any attribute supporting the matching rule given (since the attr has been omitted).

The sixth example is also a filter that should be applied to any attribute supporting the matching rule given. Attributes supporting the matching rule contained in the DN should also be considered.

The seventh and final example is a filter that should be applied to any attribute (since both the attr and matching rule have been omitted).

The following examples illustrate the use of the escaping mechanism.

```
(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)
```


The first example shows the use of the escaping mechanism to represent parenthesis characters. The second shows how to represent a "*" in an assertion value, preventing it from being interpreted as a substring indicator. The third illustrates the escaping of the backslash character.

The fourth example shows a filter searching for the four-byte value 0x00000004, illustrating the use of the escaping mechanism to represent arbitrary data, including NUL characters.

The fifth example illustrates the use of the escaping mechanism to represent various non-ASCII UTF-8 characters.

The sixth and final example demonstrates assertion of a BER encoded value.

8. Security Considerations

This memo describes a string representation of LDAP search filters. While the representation itself has no known security implications, LDAP search filters do. They are interpreted by LDAP servers to select entries from which data is retrieved. LDAP servers should take care to protect the data they maintain from unauthorized access.

Please refer to the Security Considerations sections of [[Protocol](#)] and [[AuthMeth](#)] for more information.

9. Normative References

[ASN.1] Specification of ASN.1 encoding rules: Basic, Canonical, and Distinguished Encoding Rules, ITU-T Recommendation X.690, 1994.

[AuthMeth] Harrison, R. (editor), "LDAP: Authentication Methods and Connection Level Security Mechanisms", [draft-ietf-ldapbis-authmeth-xx.txt](#), a work in progress.

[ISO10646] Universal Multiple-Octet Coded Character Set (UCS) - Architecture and Basic Multilingual Plane, ISO/IEC 10646-1, 1993.

[Models] Zeilenga, K. (editor), "LDAP: Directory Information Models", [draft-ietf-ldapbis-models-xx.txt](#), a work in progress.

[Protocol] Serfersheim, J. (editor), "LDAP: The Protocol", [draft-ietf-ldapbis-protocol-xx.txt](#), a work in progress.

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#) (also [RFC 2119](#)), March 1997.

[RFC2234] Crocker, D., Overell, P., "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.

[RFC2279] Yergeau, F., "UTF-8, a transformation format of ISO 10646", [RFC 2279](#), January 1998.

[Roadmap] Zeilenga, K. (editor), "LDAP: Technical Specification Road Map", [draft-ietf-ldapbis-roadmap-xx.txt](#), a work in progress.

[Syntaxes] Dally, K. (editor), "LDAP: Syntaxes", [draft-ietf-ldapbis-syntaxes-xx.txt](#), a work in progress.

10. Informative References

None.

11. Acknowledgments

This document replaces [RFC 2254](#) by Tim Howes. Changes included in this revised specification are based upon discussions among the authors, discussions within the LDAP (v3) Revision Working Group (ldapbis), and discussions within other IETF Working Groups. The contributions of individuals in these working groups is gratefully acknowledged.

12. Authors' Address

Mark Smith, Editor
Netscape Communications Corp.
360 W. Caribbean Drive
Sunnyvale, CA 94089
USA
+1 650 937-3477
mcs@netscape.com

Tim Howes
Opsware, Inc.
599 N. Mathilda Ave.
Sunnyvale, CA 94085
USA
+1 408 744-7509
howes@opsware.com

13. Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

14. [Appendix A](#): Changes Since [RFC 2254](#)

14.1. Technical Changes

The following technical changes were made to the contents of the "String Search Filter Definition" section:

Added statement that the string representation is a string of UTF-8 encoded ISO 10646-1 characters.

Revised all of the ABNF to use common productions from [[Models](#)].

Replaced the "value" rule with a new "assertionvalue" rule within the "simple", "extensible", and "substring" ("initial", "any", and "final") rules. This matches a change made in [[Syntaxes](#)].

Revised the "attr", "matchingrule", and "assertionvalue" ABNF to more precisely reference productions from the [[Models](#)] and [[Protocol](#)] documents.

Introduced the "valueencoding" and associated "normal" and "escaped" rules to reduce the dependence on descriptive text. The "normal" production restricts filter strings to valid UTF-8 sequences.

Added a third option to the "extensible" production to allow creation of a MatchingRuleAssertion that only has a matchValue.

Added a statement about expected behavior in light of [RFC 2254](#)'s lack of a clear definition of "string representation."

14.2. Editorial Changes

Changed document title to include "LDAP:" prefix.

IESG Note: removed note about lack of satisfactory mandatory authentication mechanisms.

Header and "Authors' Addresses" sections: added Mark Smith as the document editor and updated affiliation and contact information.

"Table of Contents" section: added.

Copyright: updated the year.

"Abstract" section: separated from introductory material.

"Introduction" section: new section; separated from the Abstract. Updated second paragraph to indicate that [RFC 2254](#) is replaced by this document (instead of [RFC 1960](#)). Added reference to the [[Roadmap](#)] document.

"LDAP Search Filter Definition" section: made corrections to the LDAPv3 search filter ABNF so it matches that used in [[Protocol](#)].

Clarified the definition of 'value' (now 'assertionvalue') to take into account the fact that it is not precisely an AttributeAssertion from [[Protocol](#)] [section 4.1.6](#) (special handling is required for some characters). Added a note that each octet of a character to be escaped is replaced by a backslash and two hex digits, which represent a single octet.

"Examples" section: added five additional examples: (seeAlso=), (cn:=Betty Rubble), (:1.2.3:=Wilma Flintstone), (:=Fred Flintstone), and (1.3.6.1.4.1.1466.0:=\04\02\48\69). Replaced one occurrence of "a value" with "an assertion value".

"Security Considerations" section: added references to [[Protocol](#)] and

[\[AuthMeth\]](#).

"Normative References" section: renamed from "References" per new RFC guidelines. Changed from [1] style to [\[Protocol\]](#) style throughout the document. Added entries for [\[ISO10646\]](#), [\[RFC2119\]](#), [\[AuthMeth\]](#), [\[Models\]](#), and [\[Roadmap\]](#) and updated UTF-8 reference to [RFC 2279](#). Replaced [RFC 822](#) reference with a reference to [RFC 2234](#).

"Informative References" section: added for clarity.

"Acknowledgments" section: added.

"Appendix A: Changes Since [RFC 2254](#)" section: added.

"Appendix B: Changes Since Previous Document Revision" section: added.

[15. Appendix B: Changes Since Previous Document Revision](#)

This appendix lists all changes relative to the last published revision, [draft-ietf-ldapbis-filter-03.txt](#). Note that when appropriate these changes are also included in [Appendix A](#), but are also included here for the benefit of the people who have already reviewed [draft-ietf-ldapbis-filter-03.txt](#). This section will be removed before this document is published as an RFC.

[15.1. Technical Changes](#)

"String Search Filter Definition" section: Added statement that the string representation is a string of UTF-8 encoded ISO 10646-1 characters and statement about expected behavior in light of [RFC 2254](#)'s lack of a clear definition of "string representation."

"String Search Filter Definition" section: Revised all of the ABNF to use common productions from [\[Models\]](#). Revised the "normal" production to restrict filter strings to valid UTF-8 sequences.

[15.2. Editorial Changes](#)

"Status of this Memo" section: updated boilerplate to match current I-D guidelines.

"Examples" section: removed ;binary from an example.

"LDAP Search Filter Definition " section: updated section references

to match current LDAPBis drafts. Made minor changes to the ASN.1 so it exactly matches that used in the Protocol document (added comments).

"Normative References" section: added references to [[ISO10646](#)], [[RFC2119](#)] and [[Models](#)].

"Informative References" section: added for clarity.

Updated copyright year to 2003.

This Internet Draft expires on 28 August 2003.