

Lightweight Directory Access Protocol (LDAP):
Additional Matching Rules

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This document provides a collection of matching rules for use with the Lightweight Directory Access Protocol (LDAP). As these matching rules are simple adaptations of matching rules specified for use with the X.500 Directory, most are already in wide use.

Table of Contents

1.	Background and Intended Use.	2
2.	Matching Rules	2
2.1.	booleanMatch	2
2.2.	caseExactMatch	2
2.3.	caseExactOrderingMatch	3
2.4.	caseExactSubstringsMatch	3
2.5.	caseIgnoreListSubstringsMatch.	3
2.6.	directoryStringFirstComponentMatch	4
2.7.	integerOrderingMatch	4
2.8.	keywordMatch	4
2.9.	numericStringOrderingMatch	5
2.10.	octetStringOrderingMatch	5
2.11.	storedPrefixMatch.	5
2.12.	wordMatch.	6
3.	Security Considerations.	6
4.	IANA Considerations.	6
5.	Acknowledgments.	7
6.	References	7

6.1.	Normative References	7
6.2.	Informative References	7
7.	Author's Address	8
8.	Full Copyright Statement	9

1. Background and Intended Use

This document adapts additional X.500 Directory [X.500] matching rules [X.520] for use with the Lightweight Directory Access Protocol (LDAP) [RFC3377]. Most of these rules are widely used today on the Internet, such as in support of the inetOrgPerson [RFC2798] and Policy Core Information Model [RFC3703] LDAP schemas. The rules are applicable to many other applications.

This document supersedes the informational matching rules descriptions provided in RFC 2798 that are now provided in this document. Specifically, section 2 of this document replaces section 9.3.3 of RFC 2798.

Schema definitions are provided using LDAP description formats [RFC2252]. Definitions provided here are formatted (line wrapped) for readability.

2. Matching Rules

2.1. booleanMatch

The booleanMatch rule compares for equality a asserted Boolean value with an attribute value of BOOLEAN syntax. The rule returns TRUE if and only if the values are the same, i.e., both are TRUE or both are FALSE. (Source: X.520)

```
( 2.5.13.13 NAME 'booleanMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 )
```

The BOOLEAN (1.3.6.1.4.1.1466.115.121.1.7) syntax is described in [RFC2252].

2.2. caseExactMatch

The caseExactMatch rule compares for equality the asserted value with an attribute value of DirectoryString syntax. The rule is identical to the caseIgnoreMatch [RFC2252] rule except that case is not ignored. (Source: X.520)

```
( 2.5.13.5 NAME 'caseExactMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

The DirectoryString (1.3.6.1.4.1.1466.115.121.1.15) syntax is described in [RFC2252].

2.3. caseExactOrderingMatch

The caseExactOrderingMatch rule compares the collation order of the asserted string with an attribute value of DirectoryString syntax. The rule is identical to the caseIgnoreOrderingMatch [RFC2252] rule except that letters are not folded. (Source: X.520)

```
( 2.5.13.6 NAME 'caseExactOrderingMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

The DirectoryString (1.3.6.1.4.1.1466.115.121.1.15) syntax is described in [RFC2252].

2.4. caseExactSubstringsMatch

The caseExactSubstringsMatch rule determines whether the asserted value(s) are substrings of an attribute value of DirectoryString syntax. The rule is identical to the caseIgnoreSubstringsMatch [RFC2252] rule except that case is not ignored. (Source: X.520)

```
( 2.5.13.7 NAME 'caseExactSubstringsMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )
```

The SubstringsAssertion (1.3.6.1.4.1.1466.115.121.1.58) syntax is described in [RFC2252].

2.5. caseIgnoreListSubstringsMatch

The caseIgnoreListSubstringMatch rule compares the asserted substring with an attribute value which is a sequence of DirectoryStrings, but where the case (upper or lower) is not significant for comparison purposes. The asserted value matches a stored value if and only if the asserted value matches the string formed by concatenating the strings of the stored value. This matching is done according to the caseIgnoreSubstringsMatch [RFC2252] rule; however, none of the initial, any, or final values of the asserted value are considered to match a substring of the concatenated string which spans more than one of the strings of the stored value. (Source: X.520)

```
( 2.5.13.12 NAME 'caseIgnoreListSubstringsMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )
```

The SubstringsAssertion (1.3.6.1.4.1.1466.115.121.1.58) syntax is described in [RFC2252].

2.6. directoryStringFirstComponentMatch

The directoryStringFirstComponentMatch rule compares for equality the asserted DirectoryString value with an attribute value of type SEQUENCE whose first component is mandatory and of type DirectoryString. The rule returns TRUE if and only if the attribute value has a first component whose value matches the asserted DirectoryString using the rules of caseIgnoreMatch [RFC2252]. A value of the assertion syntax is derived from a value of the attribute syntax by using the value of the first component of the SEQUENCE. (Source: X.520)

```
( 2.5.13.31 NAME 'directoryStringFirstComponentMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

The DirectoryString (1.3.6.1.4.1.1466.115.121.1.15) syntax is described in [RFC2252].

2.7. integerOrderingMatch

The integerOrderingMatch rule compares the ordering of the asserted integer with an attribute value of INTEGER syntax. The rule returns True if the attribute value is less than the asserted value. (Source: X.520)

```
( 2.5.13.15 NAME 'integerOrderingMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )
```

The INTEGER (1.3.6.1.4.1.1466.115.121.1.27) syntax is described in [RFC2252].

2.8. keywordMatch

The keywordMatch rule compares the asserted string with keywords in an attribute value of DirectoryString syntax. The rule returns TRUE if and only if the asserted value matches any keyword in the attribute value. The identification of keywords in an attribute value and of the exactness of match are both implementation specific. (Source: X.520)

```
( 2.5.13.33 NAME 'keywordMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

The DirectoryString (1.3.6.1.4.1.1466.115.121.1.15) syntax is described in [RFC2252].

2.9. numericStringOrderingMatch

The numericStringOrderingMatch rule compares the collation order of the asserted string with an attribute value of NumericString syntax. The rule is identical to the caseIgnoreOrderingMatch [RFC2252] rule except that all space characters are skipped during comparison (case is irrelevant as characters are numeric). (Source: X.520)

```
( 2.5.13.9 NAME 'numericStringOrderingMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.36 )
```

The NumericString (1.3.6.1.4.1.1466.115.121.1.36) syntax is described in [RFC2252].

2.10. octetStringOrderingMatch

The octetStringOrderingMatch rule compares the collation order of the asserted octet string with an attribute value of OCTET STRING syntax. The rule compares octet strings from first octet to last octet, and from the most significant bit to the least significant bit within the octet. The first occurrence of a different bit determines the ordering of the strings. A zero bit precedes a one bit. If the strings are identical but contain different numbers of octets, the shorter string precedes the longer string. (Source: X.520)

```
( 2.5.13.18 NAME 'octetStringOrderingMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

The OCTET STRING (1.3.6.1.4.1.1466.115.121.1.40) syntax is described in [RFC2252].

2.11. storedPrefixMatch

The storedPrefixMatch rule determines whether an attribute value, whose syntax is DirectoryString is a prefix (i.e., initial substring) of the asserted value, without regard to the case (upper or lower) of the strings. The rule returns TRUE if and only if the attribute value is an initial substring of the asserted value with corresponding characters identical except possibly with regard to case. (Source: X.520)

```
( 2.5.13.41 NAME 'storedPrefixMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

Note: This rule can be used, for example, to compare values in the Directory which are telephone area codes with a purported value which is a telephone number.

The DirectoryString (1.3.6.1.4.1.1466.115.121.1.15) syntax is described in [RFC2252].

2.12. wordMatch

The wordMatch rule compares the asserted string with words in an attribute value of DirectoryString syntax. The rule returns TRUE if and only if the asserted word matches any word in the attribute value. Individual word matching is as for the caseIgnoreMatch [RFC2252] matching rule. The precise definition of a "word" is implementation specific. (Source: X.520)

```
( 2.5.13.32 NAME 'wordMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

The DirectoryString (1.3.6.1.4.1.1466.115.121.1.15) syntax is described in [RFC2252].

3. Security Considerations

General LDAP security considerations [RFC3377] is applicable to the use of this schema. Additional considerations are noted above where appropriate.

4. IANA Considerations

The Internet Assigned Numbers Authority (IANA) has updated the LDAP descriptors registry [RFC3383] as indicated in the following template:

```
Subject: Request for LDAP Descriptor Registration Update
Descriptor (short name): see comment
Object Identifier: see comments
Person & email address to contact for further information:
    Kurt Zeilenga <kurt@OpenLDAP.org>
Usage: see comments
Specification: RFC 3698
Author/Change Controller: IESG
Comments:
```

The following descriptors have been added:

NAME	Type	OID
-----	----	-----
booleanMatch	M	2.5.13.13
caseExactMatch	M	2.5.13.5
caseExactOrderingMatch	M	2.5.13.6
caseExactSubstringsMatch	M	2.5.13.7
caseIgnoreListSubstringsMatch	M	2.5.13.12
directoryStringFirstComponentMatch	M	2.5.13.31
integerOrderingMatch	M	2.5.13.15
keywordMatch	M	2.5.13.33
numericStringOrderingMatch	M	2.5.13.9
octetStringOrderingMatch	M	2.5.13.18
storedPrefixMatch	M	2.5.13.41
wordMatch	M	2.5.13.32

where Type M is Matching Rule.

This document makes no new OID assignments. It only associates LDAP matching rule descriptions with existing X.500 matching rules.

5. Acknowledgments

This document borrows from [X.520], an ITU-T Recommendation.

6. References

6.1. Normative References

- [RFC2252] Wahl, M., Coulbeck, A., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997.
- [RFC3377] Hodges, J. and R. Morgan, "Lightweight Directory Access Protocol (v3): Technical Specification", RFC 3377, September 2002.

6.2. Informative References

- [RFC2798] Smith, M., "The LDAP inetOrgPerson Object Class", RFC 2798, April 2000.
- [RFC3383] Zeilenga, K., "IANA Considerations for LDAP", BCP 64 RFC 3383, September 2002.
- [RFC3703] Strassner, J., Moore, B., Moats, R. and E. Ellessen, "Policy Core LDAP Schema", RFC 3703, February 2004.

- [X.500] International Telecommunication Union -
Telecommunication Standardization Sector, "The
Directory -- Overview of concepts, models and
services," X.500(1993) (also ISO/IEC 9594-1:1994).
- [X.520] International Telecommunication Union -
Telecommunication Standardization Sector, "The
Directory: Selected Attribute Types", X.520(1997).

7. Author's Address

Kurt D. Zeilenga
OpenLDAP Foundation

EMail: Kurt@OpenLDAP.org

8. Full Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78 and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

