Network Working Group
Request for Comments: 3866
Obsoletes: 2596
Category: Standards Track

K. Zeilenga, Ed.
OpenLDAP Foundation

July 2004

Language Tags and Ranges in the Lightweight Directory Access Protocol (LDAP)

Status of this Memo

Copyright Notice

Abstract

It is often desirable to be able to indicate the natural language associated with values held in a directory and to be able to query the directory for values which fulfill the user's language needs. This document details the use of Language Tags and Ranges in the Lightweight Directory Access Protocol (LDAP).

1. Background and Intended Use

The Lightweight Directory Access Protocol (LDAP) [RFC3377] provides a means for clients to interrogate and modify information stored in a distributed directory system. The information in the directory is maintained as attributes of entries. Most of these attributes have syntaxes which are human-readable strings, and it is desirable to be able to indicate the natural language associated with attribute values.

This document describes how language tags and ranges [RFC3066] are carried in LDAP and are to be interpreted by LDAP implementations. All LDAP implementations MUST be prepared to accept language tags and ranges.

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].

This document replaces RFC 2596. Appendix A summaries changes made since RFC 2596.

Appendix B discusses differences from X.500(1997) "contexts" mechanism.

Appendix A and B are provided for informational purposes only.

The remainder of this section provides a summary of Language Tags, Language Ranges, and Attribute Descriptions.

1.1. Language Tags

Section 2 of BCP 47 [RFC3066] describes the language tag format which is used in LDAP. Briefly, it is a string of [ASCII] letters and hyphens. Examples include "fr", "en-US" and "ja-JP". Language tags are case insensitive. That is, the language tag "en-us" is the same as "EN-US".

Section 2 of this document details use of language tags in LDAP.

1.2. Language Ranges

Section 2.5 of BCP 47 [RFC3066] describes the language ranges. Language ranges are used to specify sets of language tags.

A language range matches a language tag if it is exactly equal to the tag, or if it is exactly equal to a prefix of the tag such that the first character following the prefix is "-". That is, the language range "de" matches the language tags "de" and "de-CH" but not "den". The special language range "*" matches all language tags.

Due to attribute description option naming restrictions in LDAP, this document defines a different language range syntax. However, the semantics of language ranges in LDAP are consistent with BCP 47.

Section 3 of this document details use of language ranges in LDAP.

1.3. Attribute Descriptions

This section provides an overview of attribute descriptions in LDAP. LDAP attributes and attribute descriptions are defined in [RFC2251].

An attribute consists of a type, a set of zero or more associated tagging options, and a set of one or more values. The type and the options are combined into the AttributeDescription.

AttributeDescriptions can also contain options which are not part of the attribute, but indicate some other function (such as range assertion or transfer encoding).

An AttributeDescription with one or more tagging options is a direct subtype of each AttributeDescription of the same type with all but one of the tagging options. If the AttributeDescription's type is a direct subtype of some other type, then the AttributeDescription is also a direct subtype of the AttributeDescription which consists of the supertype and all of the tagging options. That is, "CN;x-bar;x-foo" is a direct subtype of "CN;x-bar", "CN;x-foo", and "name;x-bar;x-foo". Note that "CN" is a subtype of "name".

2. Use of Language Tags in LDAP

This section describes how LDAP implementations MUST interpret language tags in performing operations.

Servers which support storing attributes with language tag options in the Directory Information Tree (DIT) SHOULD allow any attribute type it recognizes that has the Directory String, IA5 String, or other textual string syntaxes to have language tag options associated with it. Servers MAY allow language options to be associated with other attributes types.

Clients SHOULD NOT assume servers are capable of storing attributes with language tags in the directory.

Implementations MUST NOT otherwise interpret the structure of the tag when comparing two tags, and MUST treat them simply as strings of characters. Implementations MUST allow any arbitrary string which conforms to the syntax defined in BCP 47 [RFC3066] to be used as a language tag.

2.1. Language Tag Options

A language tag option associates a natural language with values of an attribute. An attribute description may contain multiple language tag options. An entry may contain multiple attributes with same attribute type but different combinations of language tag (and other) options.

A language tag option conforms to the following ABNF [RFC2234]:

```
language-tag-option = "lang-" Language-Tag
```

where the Language-Tag production is as defined in BCP 47 [RFC3066]. This production and those it imports from [RFC2234] are provided here for convenience:

```
Language-Tag = Primary-subtag *( "-" Subtag )

Primary-subtag = 1*8ALPHA

Subtag = 1*8(ALPHA / DIGIT)

ALPHA = %x41-5A / %x61-7A   ; A-Z / a-z

DIGIT = %x30-39             ; 0-9
```

A language tag option is a tagging option. A language tag option has no effect on the syntax of the attribute's values nor their transfer encoding.

Examples of valid AttributeDescription:

```
givenName;lang-en-US
CN;lang-ja
SN;lang-de;lang-gem-PFL
O;lang-i-klingon;x-foobar
description;x-foobar
CN
```

Notes: The last two have no language tag options. The x-foobar option is fictious and used for example purposes.

2.2. Search Filter

If language tag options are present in an AttributeDescription in an assertion, then for each entry within scope, the values of each attribute whose AttributeDescription consists of the same attribute type or its subtypes and contains each of the presented (and possibly other) options is to be matched.

Thus, for example, a filter of an equality match of type "name;lang-en-US" and assertion value "Billy Ray", against the following directory entry:

```
dn: SN=Ray,DC=example,DC=com
objectClass: person                 DOES NOT MATCH (wrong type)
objectClass: extensibleObject       DOES NOT MATCH (wrong type)
name;lang-en-US: Billy Ray          MATCHES
name;lang-en-US: Billy Bob          DOES NOT MATCH (wrong value)
CN;lang-en-US: Billy Ray            MATCHES
```

```
CN;lang-en-US;x-foobar: Billy Ray   MATCHES
CN;lang-en;x-foobar: Billy Ray      DOES NOT MATCH (differing lang-)
CN;x-foobar: Billy Ray              DOES NOT MATCH (no lang-)
name: Billy Ray                     DOES NOT MATCH (no lang-)
SN;lang-en-GB;lang-en-US: Billy Ray MATCHES
SN: Ray                             DOES NOT MATCH (no lang-,
                                        wrong value)
```

Note that "CN" and "SN" are subtypes of "name".

It is noted that providing a language tag option in a search filter AttributeDescription will filter out desirable values where the tag does not match exactly. For example, the filter (name;lang-en=Billy Ray) does NOT match the attribute "name;lang-en-US: Billy Ray".

If the server does not support storing attributes with language tag options in the DIT, then any assertion which includes a language tag option will not match as such it is an unrecognized attribute type. No error would be returned because of this; a presence assertion would evaluate to FALSE and all other assertions to Undefined.

If no options are specified in the assertion, then only the base attribute type and the assertion value need match the value in the directory.

Thus, for example, a filter of an equality match of type "name" and assertion value "Billy Ray", against the following directory entry:

```
dn: SN=Ray,DC=example,DC=com
objectClass: person                 DOES NOT MATCH (wrong type)
objectClass: extensibleObject       DOES NOT MATCH (wrong type)
name;lang-en-US: Billy Ray          MATCHES
name;lang-en-US: Billy Bob          DOES NOT MATCH (wrong value)
CN;lang-en-US;x-foobar: Billy Ray   MATCHES
CN;lang-en;x-foobar: Billy Ray      MATCHES
CN;x-foobar: Billy Ray              MATCHES
name: Billy Ray                     MATCHES
SN;lang-en-GB;lang-en-US: Billy Ray MATCHES
SN: Ray                             DOES NOT MATCH (wrong value)
```

2.3. Requested Attributes in Search

Clients can provide language tag options in each AttributeDescription in the requested attribute list in a search request.

If language tag options are provided in an attribute description, then only attributes in a directory entry whose attribute descriptions have the same attribute type or its subtype and contains

each of the presented (and possibly other) language tag options are to be returned. Thus if a client requests just the attribute "name;lang-en", the server would return "name;lang-en" and "CN;lang-en;lang-ja" but not "SN" nor "name;lang-fr".

Clients can provide in the attribute list multiple AttributeDescriptions which have the same base attribute type but different options. For example, a client could provide both "name;lang-en" and "name;lang-fr", and this would permit an attribute with either language tag option to be returned. Note there would be no need to provide both "name" and "name;lang-en" since all subtypes of name would match "name".

If a server does not support storing attributes with language tag options in the DIT, then any attribute descriptions in the list which include language tag options are to be ignored, just as if they were unknown attribute types.

If a request is made specifying all attributes or an attribute is requested without providing a language tag option, then all attribute values regardless of their language tag option are returned.

For example, if the client requests a "description" attribute, and a matching entry contains the following attributes:

```
objectClass: top
objectClass: organization
O: Software GmbH
description: software products
description;lang-en: software products
description;lang-de: Softwareprodukte
```

The server would return:

```
description: software products
description;lang-en: software products
description;lang-de: Softwareprodukte
```

2.4. Compare

Language tag options can be present in an AttributeDescription used in a compare request AttributeValueAssertion. This is to be treated by servers the same as the use of language tag options in a search filter with an equality match, as described in Section 2.2. If there is no attribute in the entry with the same attribute type or its subtype and contains each of the presented (or possibly other) language tag options, the noSuchAttributeType error will be returned.

Thus, for example, a compare request of type "name" and assertion value "Johann", against an entry containing the following attributes:

```
objectClass: top
objectClass: person
givenName;lang-de-DE: Johann
CN: Johann Sibelius
SN: Sibelius
```

would cause the server to return compareTrue.

However, if the client issued a compare request of type "name;lang-de" and assertion value "Johann" against the above entry, the request would fail with the noSuchAttributeType error.

If the server does not support storing attributes with language tag options in the DIT, then any comparison which includes a language tag option will always fail to locate an attribute, and noSuchAttributeType will be returned.

2.5. Add Operation

Clients can provide language options in AttributeDescription in attributes of a new entry to be created.

A client can provide multiple attributes with the same attribute type and value, so long as each attribute has a different set of language tag options.

For example, the following is a valid request:

```
dn: CN=John Smith,DC=example,DC=com
objectClass: residentialPerson
CN: John Smith
CN;lang-en: John Smith
SN: Smith
SN;lang-en: Smith
streetAddress: 1 University Street
streetAddress;lang-en-US: 1 University Street
streetAddress;lang-fr: 1 rue Universite
houseIdentifier;lang-fr: 9e etage
```

If a server does not support storing language tag options with attribute values in the DIT, then it MUST treat an AttributeDescription with a language tag option as an unrecognized attribute. If the server forbids the addition of unrecognized attributes then it MUST fail the add request with an appropriate result code.

2.6. Modify Operation

A client can provide language tag options in an AttributeDescription as part of a modification element in the modify operation.

Attribute types and language tag options MUST match exactly against values stored in the directory. For example, if the modification is a "delete", then if the stored values to be deleted have language tag options, then those language tag options MUST be provided in the modify operation, and if the stored values to be deleted do not have any language tag option, then no language tag option is to be provided.

If the server does not support storing language tag options with attribute values in the DIT, then it MUST treat an AttributeDescription with a language tag option as an unrecognized attribute, and MUST fail the request with an appropriate result code.

3. Use of Language Ranges in LDAP

Since the publication of RFC 2596, it has become apparent that there is a need to provide a mechanism for a client to request attributes based upon set of language tag options whose tags all begin with the same sequence of language sub-tags.

AttributeDescriptions containing language range options are intended to be used in attribute value assertions, search attribute lists, and other places where the client desires to provide an attribute description matching of a range of language tags associated with attributes.

A language range option conforms to the following ABNF [RFC2234]:

```
language-range-option = "lang-" [ Language-Tag "-" ]
```

where the Language-Tag production is as defined in BCP 47 [RFC3066]. This production and those it imports from [RFC2234] are provided in Section 2.1 for convenience.

A language range option matches a language tag option if the language range option less the trailing "-" matches exactly the language tag or if the language range option (including the trailing "-") matches a prefix of the language tag option. Note that the language range option "lang-" matches all language tag options.

Examples of valid AttributeDescription containing language range options:

```
givenName;lang-en-
CN;lang-
SN;lang-de-;lang-gem-
O;lang-x-;x-foobar
```

A language range option is not a tagging option. Attributes cannot be stored with language range options. Any attempt to add or update an attribute description with a language range option SHALL be treated as an undefined attribute type and result in an error.

A language range option has no effect on the transfer encoding nor on the syntax of the attribute values.

Servers SHOULD support assertion of language ranges for any attribute type which they allow to be stored with language tags.

3.1. Search Filter

If a language range option is present in an AttributeDescription in an assertion, then for each entry within scope, the values of each attribute whose AttributeDescription consists of the same attribute type or its subtypes and contains a language tag option matching the language range option are to be returned.

Thus, for example, a filter of an equality match of type "name;lang-en-" and assertion value "Billy Ray", against the following directory entry:

```
dn: SN=Ray,DC=example,DC=com
objectClass: person                 DOES NOT MATCH (wrong type)
objectClass: extensibleObject       DOES NOT MATCH (wrong type)
name;lang-en-US: Billy Ray          MATCHES
name;lang-en-US: Billy Bob          DOES NOT MATCH (wrong value)
CN;lang-en-US: Billy Ray            MATCHES
CN;lang-en-US;x-foobar: Billy Ray   MATCHES
CN;lang-en;x-foobar: Billy Ray      MATCHES
CN;x-foobar: Billy Ray              DOES NOT MATCH (no lang-)
name: Billy Ray                     DOES NOT MATCH (no lang-)
SN;lang-en-GB;lang-en-US: Billy Ray MATCHES
SN: Ray                             DOES NOT MATCH (no lang-,
                                      wrong value)
```

Note that "CN" and "SN" are subtypes of "name".

If the server does not support storing attributes with language tag options in the DIT, then any assertion which includes a language range option will not match as it is an unrecognized attribute type. No error would be returned because of this; a presence filter would evaluate to FALSE and all other assertions to Undefined.

3.2. Requested Attributes in Search

Clients can provide language range options in each AttributeDescription in the requested attribute list in a search request.

If a language range option is provided in an attribute description, then only attributes in a directory entry whose attribute descriptions have the same attribute type or its subtype and a language tag option matching the provided language range option are to be returned. Thus if a client requests just the attribute "name;lang-en-", the server would return "name;lang-en-US" and "CN;lang-en;lang-ja" but not "SN" nor "name;lang-fr".

Clients can provide in the attribute list multiple AttributeDescriptions which have the same base attribute type but different options. For example a client could provide both "name;lang-en-" and "name;lang-fr-", and this would permit an attribute whose type was name or subtype of name and with a language tag option matching either language range option to be returned.

If a server does not support storing attributes with language tag options in the DIT, then any attribute descriptions in the list which include language range options are to be ignored, just as if they were unknown attribute types.

3.3. Compare

Language range options can be present in an AttributeDescription used in a compare request AttributeValueAssertion. This is to be treated by servers the same as the use of language range options in a search filter with an equality match, as described in Section 3.1. If there is no attribute in the entry with the same subtype and a matching language tag option, the noSuchAttributeType error will be returned.

Thus, for example, a compare request of type "name;lang-" and assertion value "Johann", against the entry with the following attributes:

```
objectClass: top
objectClass: person
givenName;lang-de-DE: Johann
CN: Johann Sibelius
SN: Sibelius
```

will cause the server to return compareTrue. (Note that the language range option "lang-" matches any language tag option.)

However, if the client issued a compare request of type "name;lang-de" and assertion value "Sibelius" against the above entry, the request would fail with the noSuchAttributeType error.

If the server does not support storing attributes with language tag options in the DIT, then any comparison which includes a language range option will always fail to locate an attribute, and noSuchAttributeType will be returned.

4. Discovering Language Option Support

A server SHOULD indicate that it supports storing attributes with language tag options in the DIT by publishing 1.3.6.1.4.1.4203.1.5.4 as a value of the root DSE.

A server SHOULD indicate that it supports language range matching of attributes with language tag options stored in the DIT by publishing 1.3.6.1.4.1.4203.1.5.5 as a value of the "supportedFeatures" [RFC3674] attribute in the root DSE.

A server MAY restrict use of language tag options to a subset of the attribute types it recognizes. This document does not define a mechanism for determining which subset of attribute types can be used with language tag options.

5. Interoperability with Non-supporting Implementations

Implementators of this specification should take care that their use of language tag options does not impede proper function of implementations which do not support language tags.

Per RFC 2251, "an AttributeDescription with one or more options is treated as a subtype of the attribute type without any options." A non-supporting server will treat an AttributeDescription with any language tag options as an unrecognized attribute type. A non-supporting client will either do the same, or will treat the AttributeDescription as it would any other unknown subtype. Typically, non-supporting clients simply ignore unrecognized subtypes (and unrecognized attribute types) of attributes they request.

To ensure proper function of non-supporting clients, supporting clients SHOULD ensure that entries they populate with tagged values are also populated with non-tagged values.

Additionally, supporting clients SHOULD be prepared to handle entries which are not populated with tagged values.

6. Security Considerations

Language tags and range options are used solely to indicate the native language of values and in querying the directory for values which fulfill the user's language needed. These options are not known to raise specific security considerations. However, the reader should consider general directory security issues detailed in the LDAP technical specification [RFC3377].

7. IANA Considerations

Registration of these protocol mechanisms [RFC3383] has been completed by the IANA.

Subject: Request for LDAP Protocol Mechanism Registration
Object Identifier: 1.3.6.1.4.1.4203.1.5.4
Description: Language Tag Options
Object Identifier: 1.3.6.1.4.1.4203.1.5.5
Description: Language Range Options
Person & email address to contact for further information:
Kurt Zeilenga <kurt@openldap.org>
Usage: Feature
Specification: RFC 3866
Author/Change Controller: IESG
Comments: none

These OIDs were assigned [ASSIGN] by OpenLDAP Foundation, under its IANA-assigned private enterprise allocation [PRIVATE], for use in this specification.

8. Acknowledgments

This document is a revision of RFC 2596 by Mark Wahl and Tim Howes. RFC 2596 was a product of the IETF ASID and LDAPEXT working groups. This document also borrows from a number of IETF documents including BCP 47 by H. Alvestrand.

9. References

9.1. Normative References

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

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

[RFC2251] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", BCP 47, RFC 3066, January 2001.

[RFC3377] Hodges, J. and R. Morgan, "Lightweight Directory Access Protocol (v3): Technical Specification", RFC 3377, September 2002.

[RFC3674] Zeilenga, K., "Feature Discovery in Lightweight Directory Access Protocol (LDAP)", RFC 3674, December 2003.

[ASCII] Coded Character Set--7-bit American Standard Code for Information Interchange, ANSI X3.4-1986.

9.2. Informative References

[X.501] International Telecommunication Union - Telecommunication Standardization Sector, "The Directory -- Models," X.501(1997).

[RFC3383] Zeilenga, K., "Internet Assigned Numbers Authority (IANA) Considerations for Lightweight Directory Access Protocol (LDAP)", BCP 64, RFC 3383, September 2002.

[ASSIGN] OpenLDAP Foundation, "OpenLDAP OID Delegations", http://www.openldap.org/foundation/oid-delegate.txt.

[PRIVATE] IANA, "Private Enterprise Numbers", http://www.iana.org/assignments/enterprise-numbers.

Appendix A. Differences from RFC 2596

This document adds support for language ranges, provides a mechanism that a client can use to discover whether a server supports language tags and ranges, and clarifies how attributes with multiple language tags are to be treated. This document is a significant rewrite of RFC 2596.

Appendix B. Differences from X.500(1997)

X.500(1997) [X.501] defines a different mechanism, contexts, as the means of representing language tags (codes). This section summarizes the major differences in approach.

a) An X.500 operation which has specified a language code on a value matches a value in the directory without a language code.

b) LDAP references BCP 47 [RFC3066], which allows for IANA registration of new tags as well as unregistered tags.

c) LDAP supports language ranges (new in this revision).

d) LDAP does not allow language tags (and ranges) in distinguished names.

e) X.500 describes subschema administration procedures to allow language codes to be associated with particular attributes types.

Editor’s Address

Kurt D. Zeilenga
OpenLDAP Foundation

EMail: Kurt@OpenLDAP.org

Full Copyright Statement

Intellectual Property

Acknowledgement

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