Internet-Draft
Intended Category: Standard Track
Expires in six months

Editor: Kurt D. Zeilenga
OpenLDAP Foundation

4 May 2003

LDAP: Internationalized String Preparation
<draft-zeilenga-ldapbis-strprep-00.txt>

Status of this Memo

Abstract

The previous Lightweight Directory Access Protocol (LDAP) technical specifications did not precisely define how string matching is to be performed. This lead to a number of usability and interoperability problems. This document defines string preparation algorithms for matching rules defined for use in LDAP.

Conventions

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].

Character names in this document use the notation for code points and names from the Unicode Standard [UNICODE] and ISO/IEC 10646-1 [ISO10646]. For example, the letter "a" may be represented as either <U+0061> or <LATIN SMALL LETTER A>. In the lists of mappings and the prohibited characters, the "U+" is left off to make the lists easier to read. The comments for character ranges are shown in square brackets (such as "[CONTROL CHARACTERS]") and do not come from the standards.

Note: a glossary of terms used in Unicode and ISO/IEC 10646 can be found in [GLOSSARY]. Information on the ISO/IEC 10646/Unicode character encoding model can be found in [UTR17].

1. Introduction

1.1. Background

An LDAP matching rule [Syntaxes] defines an algorithm for determining whether a presented value matches an attribute value in accordance with the criteria defined for the rule. The proposition may be evaluated to True, False, or Undefined.

True - the attribute contains a matching value,

False - the attribute contains no matching value,

Undefined - it cannot be determined whether the attribute contains a matching value or not.

For instance, the caseIgnoreMatch matching rule may be used to compare whether the commonName attribute contains a particular value without regard for case and insignificant spaces.

1.2. X.500 String Matching Rules

"X.520: Selected attribute types" [X.520] provides (amongst other things) value syntaxes and matching rules for comparing values commonly used in the Directory. These specifications are inadequate for strings composed of characters from the Universal Character Set (UCS) [ISO10646], a superset of Unicode [UNICODE].

The CaseIgnoreMatch matching rule [X.520], for example, is simply defined as being a case insensitive comparison where insignificant spaces are ignored. For printableString, there is only one space character and case mapping is bijective, hence this definition is sufficient. However, for UCS-based string types such as universalString, this is not sufficient. For example, a case insensitive matching implementation which folded lower case characters to upper case would yield different different results than an implementation which used upper case to lower case folding. Or one implementation may view space as referring to only SPACE (U+0020), a second implementation may view any character with the space separator (Zs) property as a space, and another implementation may view any character with the whitespace (WS) category as a space.

The lack of precise specification for string matching has led to significant interoperability problems. When used in certificate chain validation, security vulnerabilities can arise. To address these problems, this document defines precise algorithms for preparing strings for matching.

1.3. Relationship to "stringprep"

The string preparation algorithms described in this document are based upon the "stringprep" approach [RFC3454]. In "stringprep", presented and stored values are first prepared for comparison and so that a character-by-character comparison yields the "correct" result.

The approach used here is a refinement of the "stringprep" [RFC3454] approach. Each algorithm involves two additional preparation steps.

a) prior to applying the Unicode string preparation steps outlined in "stringprep", the string is transcoded to Unicode;

b) after applying the Unicode string preparation steps outlined in "stringprep", characters insignificant to the matching rules are removed.

Hence, preparation of strings for X.500 matching involves the following steps:

1) Transcode
2) Map
3) Normalize
4) Prohibit
5) Check Bidi (Bidirectional)
6) Insignificant Character Removal

These steps are described in Section 2.

1.4. Relationship to the LDAP Technical Specification

This document is a integral part of the LDAP technical specification [Roadmap] which obsoletes the previously defined LDAP technical specification [RFC3377] in its entirety.

This document details LDAP internationalized string preparation algorithms used by [Syntaxes] and possible other technical specifications defining LDAP syntaxes and/or matching rules.

1.5. Relationship to X.500

LDAP is defined [Roadmap] in X.500 terms as an X.500 access mechanism. As such, there is a strong desire for alignment between LDAP and X.500 syntax and semantics. The string preparation algorithms described in this document are based upon "Internationalized String Matching Rules for X.500" [XMATCH] proposal to ITU/ISO Joint Study Group 2.

2. String Preparation

The following six-step process SHALL be applied to each presented and attribute value in preparation for string match rule evaluation.

1) Transcode
2) Map
3) Normalize
4) Prohibit
5) Check bidi
6) Insignificant Character Removal

Failure in any step is be cause the assertion to be Undefined.

The character repertoire of this process is Unicode 3.2 [UNICODE].

2.1. Transcode

Each non-Unicode string value is transcoded to Unicode.

TeletexString values are transcoded to Unicode as described in Appendix A.

PrintableString value are transcoded directly to Unicode.

UniversalString, UTF8String, and bmpString values need not be transcoded as they are Unicode-based strings (in the case of bmpString, restricted to a subset of Unicode).

If the implementation is unable or unwilling to perform the transcoding as described above, or the transcoding fails, this step fails and the assertion is evaluated to Undefined.

The transcoded string is the output string.

2.2. Map

SOFT HYPHEN (U+00AD) and MONGOLIAN TODO SOFT HYPHEN (U+1806) code points are mapped to nothing. COMBINING GRAPHEME JOINER (U+034F) and VARIATION SELECTORs (U+180B-180D,FF00-FE0F) code points are also mapped to nothing. The OBJECT REPLACEMENT CHARACTER (U+FFFC) is mapped to nothing.

CHARACTER TABULATION (U+0009), LINE FEED (LF) (U+000A), LINE TABULATION (U+000B), FORM FEED (FF) (U+000C), CARRIAGE RETURN (CR) (U+000D), and NEXT LINE (NEL) (U+0085) are mapped to SPACE (U+0020).

All other control code points (e.g., Cc) or code points with a control function (e.g., Cf) are mapped to nothing.

ZERO WIDTH SPACE (U+200B) is mapped to nothing. All other code points with Separator (space, line, or paragraph) property (e.g, Zs, Zl, or Zp) are mapped to SPACE (U+0020).

For case ignore, numeric, and stored prefix string matching rules, characters are case folded per B.2 of [RFC3454].

2.3. Normalize

The input string is be normalized to Unicode Form KC (compatibility composed) as described in [UAX15].

2.4. Prohibit

All Unassigned, Private Use, and non-character code points are prohibited. Surrogate codes (U+D800-DFFFF) are prohibited.

The REPLACEMENT CHARACTER (U+FFFD) code point is prohibited.

The first code point of a string is prohibited from being a combining

character.

Empty strings are prohibited.

The step fails and the assertion is evaluated to Undefined if the input string contains any prohibited code point. The output string is the input string.

2.5. Check bidi

There are no bidirectional restrictions. The output string is the input string.

2.5. Insignificant Character Removal

In this step, characters insignificant to the matching rule are to be removed. The characters to be removed differ from matching rule to matching rule.

Section 2.6.1 applies to case ignore and exact string matching.
Section 2.6.2 applies to numericString matching.
Section 2.6.3 applies to telephoneNumber matching

2.6.1. Insignificant Space Removal

For the purposes of this section, a space is defined to be the SPACE (U+0020) code point followed by no combining marks.

> NOTE - The previous steps ensure that the string cannot contain any code points in the separator class, other than SPACE (U+0020).

The following spaces are regarded as not significant and are to be removed:

- leading spaces (i.e. those preceding the first character that is not a space);
- trailing spaces (i.e. those following the last character that is not a space);
- multiple consecutive spaces (these are taken as equivalent to a single space character).

(A string consisting entirely of spaces is equivalent to a string containing exactly one space.)

For example, removal of spaces from the Form KC string:

"<SPACE><SPACE>foo<SPACE><SPACE>bar<SPACE><SPACE>" would result in the output string:
"<SPACE>foo<SPACE>bar<SPACE>".

and the Form KC string:
"<SPACE><SPACE><SPACE>" would result in the output string:
"<SPACE>".

2.6.2. NumericString Insignificant Character Removal

For the purposes of this section, a space is defined to be the SPACE (U+0020) code point followed by no combining marks.

All spaces are regarded as not significant and are to be removed.

For example, removal of spaces from the Form KC string:
"<SPACE><SPACE>123<SPACE><SPACE>456<SPACE><SPACE>" would result in the output string:
"123456".

and the Form KC string:
"<SPACE><SPACE><SPACE>" would result in an empty output string.

2.6.3. TelephoneNumber Insignificant Character Removal

For the purposes of this section, a hyphen is defined to be HYPHEN-MINUS (U+002D), ARMENIAN HYPHEN (U+058A), HYPHEN (U+2010), NON-BREAKING HYPHEN (U+2011), MINUS SIGN (U+2212), SMALL HYPHEN-MINUS (U+FE63), or FULLWIDTH HYPHEN-MINUS (U+FF0D) code point followed by no combining marks and a space is defined to be the SPACE (U+0020) code point followed by no combining marks.

All hyphens and spaces are regarded as not significant and are to be removed.

3. Security Considerations

"Preparation for International Strings ('stringprep')" [RFC3454] security considerations generally apply to the algorithms described here.

4. Acknowledgments

The approach used in this document is based upon design principles and

algorithms described in "Preparation of Internationalized Strings ('stringprep')" [RFC3454] by Paul Hoffman and Marc Blanchet. Some additional guidance was drawn from Unicode Technical Standards, Technical Reports, and Notes.

5. Editor's Address

Kurt Zeilenga
E-mail: <kurt@openldap.org>

6. References

6.1. Normative References

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

[RFC3454] P. Hoffman, M. Blanchet, "Preparation of Internationalized Strings ('stringprep')", RFC 3454, December 2002.

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

[Syntaxes] S. Legg (editor), "LDAP: Syntaxes and Matching Rules", draft-ietf-ldapbis-syntaxes-xx.txt, a work in progress.

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

[UNICODE] The Unicode Consortium, "The Unicode Standard, Version 3.2.0" is defined by "The Unicode Standard, Version 3.0" (Reading, MA, Addison-Wesley, 2000. ISBN 0-201-61633-5), as amended by the "Unicode Standard Annex #27: Unicode 3.1" (http://www.unicode.org/reports/tr27/) and by the "Unicode Standard Annex #28: Unicode 3.2" (http://www.unicode.org/reports/tr28/).

[UAX15] M. Davis, M. Duerst, "Unicode Standard Annex #15: Unicode Normalization Forms, Version 3.2.0". <http://www.unicode.org/unicode/reports/tr15/tr15-22.html>, March 2002.

6.2. Informative References

[X.500] International Telephone Union, "The Directory: Overview of Concepts, Models and Service", X.500, 2000.

[X.501] International Telephone Union, "The Directory: The Models", X.501, 2000.

[X.520] International Telephone Union, "The Directory: Selected Attribute Types", X.520, 2000.

[XMATCH] K. Zeilenga, "Internationalized String Matching Rules for X.500", draft-zeilenga-ldapbis-strmatch-xx.txt a work in progress.

[GLOSSARY] The Unicode Consortium, "Unicode Glossary", <http://www.unicode.org/glossary/>.

[UTR17] K. Whistler, M. Davis, "Unicode Technical Report #17, Character Encoding Model", UTR17, <http://www.unicode.org/unicode/reports/tr17/>, August 2000.

Appendix A. Teletex (T.61) to Unicode

TBD.