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

Kurt D. Zeilenga
OpenLDAP Foundation

27 October 2003

LDAP: Internationalized String Preparation
<draft-ietf-ldapbis-strprep-02.txt>

Status of this Memo

Abstract

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

Note: a glossary of terms used in Unicode can be found in [Glossary]. Information on the Unicode character encoding model can be found in [CharModel].

1. Introduction

1.1. Background

A Lightweight Directory Access Protocol (LDAP) [Roadmap] 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 character 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 character strings for matching.

1.3. Relationship to "stringprep"

The character string preparation algorithms described in this document are based upon the "stringprep" approach [StringPrep]. 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" [StringPrep] 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 character 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 new LDAP internationalized character 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 character 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 character string matching rule evaluation.

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

Failure in any step causes the assertion to evaluate to Undefined.

This process is intended to act upon non-empty character strings. If the string to prepare is empty, this process is not applied and the assertion is evaluated to 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 [X.680][T.61] values are transcoded to Unicode as described in Appendix A.

PrintableString [X.680] value are transcoded directly to Unicode.

UniversalString, UTF8String, and bmpString [X.680] values need not be transcoded as they are Unicode-based strings (in the case of bmpString, a subset of Unicode).

The output is the transcoded 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 [StringPrep].

The output is the mapped string.

2.3. Normalize

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

2.4. Prohibit

All Unassigned code points are prohibited. Unassigned code points are listed in Table A.1 of [StringPrep].

Private Use (U+E000-F8FF, F0000-FFFFD, 100000-10FFFD) code points are prohibited.

All non-character code points (U+FDD0-FDEF, FFFE-FFFF, 1FFFE-1FFFF, 2FFFE-2FFFF, 3FFFE-3FFFF, 4FFFE-4FFFF, 5FFFE-5FFFF, 6FFFE-6FFFF, 7FFFE-7FFFF, 8FFFE-8FFFF, 9FFFE-9FFFF, AFFFE-AFFFF, BFFFE-BFFFF, CFFFE-CFFFF, DFFFE-DFFFF, EFFFE-EFFFF, FFFFE-FFFFF, 10FFFE-10FFFF) 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.

The step fails if the input string contains any prohibited code point. The output is the input string.

2.5. Check bidi

There are no bidirectional restrictions. The output 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 2.5.1 applies to case ignore and exact string matching.
Section 2.5.2 applies to numericString matching.
Section 2.5.3 applies to telephoneNumber matching

2.5.1. 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).

If the input string consists entirely of spaces or is empty, the output is a string consisting of exactly one space (e.g. " ").

Otherwise, the following spaces are 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).

For example, removal of spaces from the Form KC string:
"<SPACE><SPACE>foo<SPACE><SPACE>bar<SPACE><SPACE>"
would result in the output string:
"foo<SPACE>bar"
and the Form KC string:
"<SPACE><SPACE><SPACE>"
would result in the output string:
"<SPACE>".

2.5.2. 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. If the input string consists entirely of spaces or is empty, the output is a string consisting of exactly one space (e.g. " "). Otherwise, all spaces 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 the output string:
"<SPACE>".

2.5.3. 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 considered insignificant. If the string contains only spaces and hyphens or is empty, then the output is a string consisting of one space. Otherwise, all hyphens and spaces are removed.

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

3. Security Considerations

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

4. Contributors

Appendix A and B of this document were authored by Howard Chu <hyc@symas.com> of Symas Corporation (based upon information provided in RFC 1345).

5. Acknowledgments

The approach used in this document is based upon design principles and algorithms described in "Preparation of Internationalized Strings ('stringprep')" [StringPrep] by Paul Hoffman and Marc Blanchet. Some additional guidance was drawn from Unicode Technical Standards, Technical Reports, and Notes.

This document is a product of the IETF LDAP Revision (LDAPBIS) Working Group.

6. Author's Address

Kurt Zeilenga

E-mail: <kurt@openldap.org>

7. References

7.1. Normative References

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

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

[StringPrep] Hoffman P. and M. Blanchet, "Preparation of Internationalized Strings ('stringprep')", draft-hoffman-rfc3454bis-xx.txt, a work in progress.

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

[ISO10646] International Organization for Standardization, "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] Davis, M. and 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.

[X.680] International Telecommunication Union - Telecommunication Standardization Sector, "Abstract Syntax Notation One (ASN.1) - Specification of Basic Notation", X.680(1997) (also ISO/IEC 8824-1:1998).

[T.61] CCITT (now ITU), "Character Repertoire and Coded Character Sets for the International Teletex Service", T.61, 1988.

7.2. Informative References

[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.501] International Telecommunication Union - Telecommunication Standardization Sector, "The Directory -- Models," X.501(1993) (also ISO/IEC 9594-2:1994).

[X.520] International Telecommunication Union - Telecommunication Standardization Sector, "The Directory: Selected Attribute Types", X.520(1993) (also ISO/IEC 9594-6:1994).

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

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

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

[RFC1345] Simonsen, K., "Character Mnemonics & Character Sets", RFC 1345, June 1992.

Appendix A. Teletex (T.61) to Unicode

This appendix defines an algorithm for transcoding [T.61] characters to [Unicode] characters for use in string preparation for LDAP matching rules. This appendix is normative.

The transcoding algorithm is derived from the T.61-8bit definition provided in [RFC1345]. With a few exceptions, the T.61 character codes from x00 to x7f are equivalent to the corresponding [Unicode] code points, and their values are left unchanged by this algorithm. E.g. the T.61 code x20 is identical to (U+0020). The exceptions are for these T.61 codes that are undefined: x23, x24, x5c, x5e, x60, x7b, x7d, and x7e.

The codes from x80 to x9f are also equivalent to the corresponding Unicode code points. This is specified for completeness only, as

these codes are control characters, and will be mapped to nothing in the LDAP String Preparation Mapping step.

The remaining T.61 codes are mapped below in Table A.1. Table positions marked "??" are undefined.

Input strings containing undefined T.61 codes SHALL produce an Undefined matching result. For diagnostic purposes, this algorithm does not fail for undefined input codes. Instead, undefined codes in the input are mapped to the Unicode REPLACEMENT CHARACTER (U+FFFD). As the LDAP String Preparation Prohibit step disallows the REPLACEMENT CHARACTER from appearing in its output, this transcoding yields the desired effect.

Note: RFC 1345 listed the non-spacing accent codepoints as residing in the range starting at (U+E000). In the current Unicode standard, the (U+E000) range is reserved for Private Use, and the non-spacing accents are in the range starting at (U+0300). The tables here use the (U+0300) range for these accents.

	0	1	2	3	4	5	6	7
a0	00a0	00a1	00a2	00a3	0024	00a5	0023	00a7
a8	00a8	??	??	00ab	??	??	??	??
b0	00b0	00b1	00b2	00b3	00d7	00b5	00b6	00b7
b8	00f7	??	??	00bb	00bc	00bd	00be	00bf
c0	??	0300	0301	0302	0303	0304	0306	0307
c8	0308	??	030a	0327	0332	030b	0328	030c
d0	??	??	??	??	??	??	??	??
d8	??	??	??	??	??	??	??	??
e0	2126	00c6	00d0	00aa	??	0126	0132	013f
e8	0141	00d8	0152	00ba	00de	0166	014a	0149
f0	0138	00e6	0111	00f0	0127	0131	0133	0140
f8	0142	00f8	0153	00df	00fe	0167	014b	??

Table A.1: Mapping of 8-bit T.61 codes to Unicode

T.61 also defines a number of accented characters that are formed by combining an accent prefix followed by a base character. These prefixes are in the code range xc1 to xcf. If a prefix character appears at the end of a string, the result is undefined. Otherwise these sequences are mapped to Unicode by substituting the corresponding non-spacing accent code (as listed in Table A.1) for the accent prefix, and exchanging the order so that the base character precedes the accent.

Appendix B. Additional Teletex (T.61) to Unicode Tables

All of the accented characters in T.61 have a corresponding code point in Unicode. For the sake of completeness, the combined character codes are presented in the following tables. This is informational only; for matching purposes it is sufficient to map the non-spacing accent and exchange the order of the character pair as specified in Appendix A. This appendix is informative.

B.1. Combinations with SPACE

Accents may be combined with a <SPACE> to generate the accent by itself. For each accent code, the result of combining with <SPACE> is listed in Table B.1.

	0	1	2	3	4	5	6	7
c0	??	0060	00b4	005e	007e	00af	02d8	02d9
c8	00a8	??	02da	00b8	??	02dd	02db	02c7

Table B.1: Mapping of T.61 Accents with <SPACE> to Unicode

B.2. Combinations for xc1: (Grave accent)

T.61 has predefined characters for combinations with A, E, I, O, and U. Unicode also defines combinations for N, W, and Y. All of these combinations are present in Table B.2.

	0	1	2	3	4	5	6	7
40	??	00c0	??	??	??	00c8	??	??
48	??	00cc	??	??	??	??	01f8	00d2
50	??	??	??	??	??	00d9	??	1e80
58	??	1ef2	??	??	??	??	??	??
60	??	00e0	??	??	??	00e8	??	??
68	??	00ec	??	??	??	??	01f9	00f2
70	??	??	??	??	??	00f9	??	1e81
78	??	1ef3	??	??	??	??	??	??

Table B.2: Mapping of T.61 Grave Accent Combinations

B.3. Combinations for xc2: (Acute accent)

T.61 has predefined characters for combinations with A, E, I, O, U, Y, C, L, N, R, S, and Z. Unicode also defines G, K, M, P, and W. All of these combinations are present in Table B.3.

	0	1	2	3	4	5	6	7
40	??	00c1	??	0106	??	00c9	??	01f4
48	??	00cd	??	1e30	0139	1e3e	0143	00d3
50	1e54	??	0154	015a	??	00da	??	1e82
58	??	00dd	0179	??	??	??	??	??
60	??	00e1	??	0107	??	00e9	??	01f5
68	??	00ed	??	1e31	013a	1e3f	0144	00f3
70	1e55	??	0155	015b	??	00fa	??	1e83
78	??	00fd	017a	??	??	??	??	??

Table B.3: Mapping of T.61 Acute Accent Combinations

B.4. Combinations for xc3: (Circumflex)

T.61 has predefined characters for combinations with A, E, I, O, U, Y, C, G, H, J, S, and W. Unicode also defines the combination for Z. All of these combinations are present in Table B.4.

	0	1	2	3	4	5	6	7
40	??	00c2	??	0108	??	00ca	??	011c
48	0124	00ce	0134	??	??	??	??	00d4
50	??	??	??	015c	??	00db	??	0174
58	??	0176	1e90	??	??	??	??	??
60	??	00e2	??	0109	??	00ea	??	011d
68	0125	00ee	0135	??	??	??	??	00f4
70	??	??	??	015d	??	00fb	??	0175
78	??	0177	1e91	??	??	??	??	??

Table B.4: Mapping of T.61 Circumflex Accent Combinations

B.5. Combinations for xc4: (Tilde)

T.61 has predefined characters for combinations with A, I, O, U, and N. Unicode also defines E, V, and Y. All of these combinations are present in Table B.5.

	0	1	2	3	4	5	6	7
40	??	00c3	??	??	??	1ebc	??	??
48	??	0128	??	??	??	??	00d1	00d5
50	??	??	??	??	??	0168	1e7c	??
58	??	1ef8	??	??	??	??	??	??
60	??	00e3	??	??	??	1ebd	??	??
68	??	0129	??	??	??	??	00f1	00f5

	0	1	2	3	4	5	6	7
70	??	??	??	??	??	0169	1e7d	??
78	??	1ef9	??	??	??	??	??	??

Table B.5: Mapping of T.61 Tilde Accent Combinations

B.6. Combinations for xc5: (Macron)

T.61 has predefined characters for combinations with A, E, I, O, and U. Unicode also defines Y, G, and AE. All of these combinations are present in Table B.6.

	0	1	2	3	4	5	6	7
40	??	0100	??	??	??	0112	??	1e20
48	??	012a	??	??	??	??	??	014c
50	??	??	??	??	??	016a	??	??
58	??	0232	??	??	??	??	??	??
60	??	0101	??	??	??	0113	??	1e21
68	??	012b	??	??	??	??	??	014d
70	??	??	??	??	??	016b	??	??
78	??	0233	??	??	??	??	??	??
e0	??	01e2	??	??	??	??	??	??
f0	??	01e3	??	??	??	??	??	??

Table B.6: Mapping of T.61 Macron Accent Combinations

B.7. Combinations for xc6: (Breve)

T.61 has predefined characters for combinations with A, U, and G. Unicode also defines E, I, and O. All of these combinations are present in Table B.7.

	0	1	2	3	4	5	6	7
40	??	0102	??	??	??	0114	??	011e
48	??	012c	??	??	??	??	??	014e
50	??	??	??	??	??	016c	??	??
58	??	??	??	??	??	??	??	??
60	??	0103	??	??	??	0115	??	011f
68	??	012d	??	??	??	??	00f1	014f
70	??	??	??	??	??	016d	??	??
78	??	??	??	??	??	??	??	??

Table B.7: Mapping of T.61 Breve Accent Combinations

B.8. Combinations for xc7: (Dot Above)

T.61 has predefined characters for C, E, G, I, and Z. Unicode also defines A, O, B, D, F, H, M, N, P, R, S, T, W, X, and Y. All of these combinations are present in Table B.8.

	0	1	2	3	4	5	6	7
40	??	0226	1e02	010a	1e0a	0116	1e1e	0120
48	1e22	0130	??	??	??	1e40	1e44	022e
50	1e56	??	1e58	1e60	1e6a	??	??	1e86
58	1e8a	1e8e	017b	??	??	??	??	??
60	??	0227	1e03	010b	1e0b	0117	1e1f	0121
68	1e23	??	??	??	??	1e41	1e45	022f
70	1e57	??	1e59	1e61	1e6b	??	??	1e87
78	1e8b	1e8f	017c	??	??	??	??	??

Table B.8: Mapping of T.61 Dot Above Accent Combinations

B.9. Combinations for xc8: (Diaeresis)

T.61 has predefined characters for A, E, I, O, U, and Y. Unicode also defines H, W, X, and t. All of these combinations are present in Table B.9.

	0	1	2	3	4	5	6	7
40	??	00c4	??	??	??	00cb	??	??
48	1e26	00cf	??	??	??	??	??	00d6
50	??	??	??	??	??	00dc	??	1e84
58	1e8c	0178	??	??	??	??	??	??
60	??	00e4	??	??	??	00eb	??	??
68	1e27	00ef	??	??	??	??	??	00f6
70	??	??	??	??	1e97	00fc	??	1e85
78	1e8d	00ff	??	??	??	??	??	??

Table B.8: Mapping of T.61 Diaeresis Accent Combinations

B.10. Combinations for xca: (Ring Above)

T.61 has predefined characters for A, and U. Unicode also defines w and y. All of these combinations are present in Table B.10.

	0	1	2	3	4	5	6	7
40	??	00c5	??	??	??	??	??	??

48	??	??	??	??	??	??	??	??
50	??	??	??	??	??	016e	??	??
58	??	??	??	??	??	??	??	??
60	??	00e5	??	??	??	??	??	??
68	??	??	??	??	??	??	??	??
70	??	??	??	??	??	016f	??	1e98
78	??	1e99	??	??	??	??	??	??

Table B.10: Mapping of T.61 Ring Above Accent Combinations

B.11. Combinations for xcb: (Cedilla)

T.61 has predefined characters for C, G, K, L, N, R, S, and T. Unicode also defines E, D, and H. All of these combinations are present in Table B.11.

	0	1	2	3	4	5	6	7
40	??	??	??	00c7	1e10	0228	??	0122
48	1e28	??	??	0136	013b	??	0145	??
50	??	??	0156	015e	0162	??	??	??
58	??	??	??	??	??	??	??	??
60	??	??	??	00e7	1e11	0229	??	0123
68	1e29	??	??	0137	013c	??	0146	??
70	??	??	0157	015f	0163	??	??	??
78	??	??	??	??	??	??	??	??

Table B.11: Mapping of T.61 Cedilla Accent Combinations

B.12. Combinations for xcd: (Double Acute Accent)

T.61 has predefined characters for O, and U. These combinations are present in Table B.12.

	0	1	2	3	4	5	6	7
48	??	??	??	??	??	??	??	0150
50	??	??	??	??	??	0170	??	??
68	??	??	??	??	??	??	??	0151
70	??	??	??	??	??	0171	??	??

Table B.12: Mapping of T.61 Double Acute Accent Combinations

B.13. Combinations for xce: (Ogonek)

T.61 has predefined characters for A, E, I, and U. Unicode also defines the combination for O. All of these combinations are present in Table B.13.

	0	1	2	3	4	5	6	7
40	??	0104	??	??	??	0118	??	??
48	??	012e	??	??	??	??	??	01ea
50	??	??	??	??	??	0172	??	??
58	??	??	??	??	??	??	??	??
60	??	0105	??	??	??	0119	??	??
68	??	012f	??	??	??	??	??	01eb
70	??	??	??	??	??	0173	??	??
78	??	??	??	??	??	??	??	??

Table B.13: Mapping of T.61 Ogonek Accent Combinations

B.14. Combinations for xcf: (Caron)

T.61 has predefined characters for C, D, E, L, N, R, S, T, and Z. Unicode also defines A, I, O, U, G, H, j,and K. All of these combinations are present in Table B.14.

	0	1	2	3	4	5	6	7
40	??	01cd	??	010c	010e	011a	??	01e6
48	021e	01cf	??	01e8	013d	??	0147	01d1
50	??	??	0158	0160	0164	01d3	??	??
58	??	??	017d	??	??	??	??	??
60	??	01ce	??	010d	010f	011b	??	01e7
68	021f	01d0	01f0	01e9	013e	??	0148	01d2
70	??	??	0159	0161	0165	01d4	??	??
78	??	??	017e	??	??	??	??	??

Table B.14: Mapping of T.61 Caron Accent Combinations

Intellectual Property Rights

standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright