Network Working Group
Internet-Draft
Intended status: Proposed Standard
Expires: November 2, 2007
Document: internet-drafts/draft-crispin-collation-unicasemap-04.txt

M. Crispin
University of Washington

May 2, 2007

i;unicode-casemap - Simple Unicode Collation Algorithm

Status of this Memo

Abstract

This document describes "i;unicode-casemap", a simple case-insensitive collation for Unicode strings. It provides equality, substring and ordering operations.

Introduction

The "i;ascii-casemap" collation described in [COMPARATOR] is quite simple to implement and provides case-independent comparisons for the 26 Latin alphabetics. It is specified as the default and/or baseline comparator in some application protocols, e.g., [IMAP-SORT].

It is possible, with a modest extension, to provide a more sophisticated collation with greater multilingual applicability than "i;ascii-casemap".

This collation, "i;unicode-casemap", is intended to be an alternative to, and preferred over, "i;ascii-casemap". It does not replace the "i;basic" collation described in [BASIC].

1. Unicode Casemap Collation Description

The "i;unicode-casemap" collation is a simple collation which operates on [UNICODE] strings and is case-insensitive in its treatment of characters. It provides equality, substring and ordering operations. All input is valid.

The algorithm that describes the behavior of this collation is specified for Unicode input encoded in [UTF-8]. This is for ease of description only. An implementation is free to use another internal storage format for Unicode strings, as long as it produces the same result as produced by the algorithm specified in this document for any set of Unicode strings.

As this collation algorithm is specified for UTF-8 strings, strings in other character sets and/or encodings can not be used with this collation unless they are first converted to UTF-8.

Any input that is already in UTF-8 must be checked for invalid UTF-8 sequences, such as overlong sequences. A UTF-8 string that is generated from a sequence of Unicode characters according to the rules in [UTF-8] will not contain such invalid sequences.

For the equality and ordering operations, each input UTF-8 string is prepared by converting it to "titlecased canonicalized UTF-8", using UnicodeData.txt distributed by [UNICODE], as follows on a per-character basis:

(1) If the codepoint has a titlecase property in UnicodeData.txt (this is normally the same as the uppercase property) the codepoint is converted to the titlecased codepoint.
(2) If the codepoint has a decomposition property of any type in UnicodeData.txt the codepoint is converted to the decomposed codepoints (effectively Normalization Form KD).
(3) The resulting codepoint(s) is/are appended to the titlecased canonicalized UTF-8 string.

The resulting two titlecased canonicalized UTF-8 strings are then

treated as in i;octet for equality and ordering.

Care should be taken when using OS-supplied functions to implement this collation as it is not locale sensitive. Functions such as strcasecmp and toupper are sometimes locale sensitive and may inconsistently casemap letters.

The i;unicode-casemap collation is well suited to use with many Internet protocols and computer languages. Use with natural language is often inappropriate; even though the collation apparently supports languages such as Swahili and English, in real-world use it tends to mis-sort a number of types of string:

- people and place names containing scripts that are not collated according to "alphabetical order".
- words with characters that have diacriticals. However, i;unicode-casemap generally does a better job than i;ascii-casemap for most (but not all) languages. For example, German umlaut letters will sort correctly, but some Scandinavian letters will not.
- names such as "Lloyd" (which in Welsh sorts after "Lyon", unlike in English),
- strings containing other non-letter symbols; e.g., euro and pound sterling symbols, quotation marks other than '"', dashes/hyphens, etc.

2. Unicode Casemap Collation Registration

```
<?xml version='1.0'?>
<!DOCTYPE collation SYSTEM 'collationreg.dtd'>
<collation rfc="XXXX" scope="local" intendedUse="common">
  <identifier>i;unicode-casemap</identifier>
  <title>Unicode Casemap</title>
  <operations>equality order substring</operations>
  <specification>RFC XXXX</specification>
  <owner>IETF</owner>
  <submitter>mrc@cac.washington.edu</submitter>
</collation>
```

3. Security Considerations

Collations will normally be used with UTF-8 strings. Thus the security considerations for [UTF-8], [STRINGPREP] and [UNICODE-SECURITY] also apply and are normative to this specification.

4. IANA Considerations

The i;unicode-casemap collation defined in section 2 should be added to the registry of collations defined in [COMPARATOR].

5. Normative References

The following documents are normative to this document:

[COMPARATOR] Newman, C., "Internet Appplication Protocol Collation Registry", RFC 4790, February 2007.

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

[UTF-8] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[UNICODE] <http://www.unicode.org>, UnicodeData.txt

Although the UnicodeData.txt file referenced here is part of the Unicode standard, it is subject to change as new characters are added to Unicode and errors are corrected in Unicode revisions. As a result, it may be less stable than might otherwise be implied by the standards status of this specification.

[UNICODE-SECURITY] Davis, M. and M. Suignard, "Unicode Security Considerations", February 2006, <http://www.unicode.org/reports/tr36/>.

6. Informative References:

[BASIC] Newman, C., Duerst, M., and Gulbrandsen, A., "i;basic - the Unicode Collation Algorithm", draft-gulbrandsen-collation-basic, Work in Progress.

[IMAP-SORT] Crispin, M. "Internet Message Access Protocol - SORT and THREAD Extensions", draft-ietf-imapext-sort, Work in Progress (in RFC Editor queue).

Appendices

Author's Address

Mark R. Crispin
Networks and Distributed Computing
University of Washington
4545 15th Avenue NE
Seattle, WA 98105-4527

Phone: +1 (206) 543-5762

EMail: MRC@CAC.Washington.EDU

Full Copyright Statement

Intellectual Property

Acknowledgement

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