Urdu Component Development Project

Urdu Collation Utility v1.0
(Source Code)

October 26, 2007

CENTER FOR RESEARCH IN URDU LANGUAGE PROCESSING
NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES, LAHORE
PAKISTAN

Table of Contents

Revision History

Name	Change Date	Version	Description of Changes
Atif Gulzar	26-10-2007	1.0.0.0	Initial document

1 Introduction

Urdu collation utility provides a language sensitive comparison of two strings with respect to sorting. This document enlists Urdu Collation utility with its complete specification.

In order to test the Urdu Collation utility, a sample application has been developed. It uses Urdu Collation utility and provides example for sorting Urdu words.

2 Urdu Collation Utility

Urdu Collation utility provides language sensitive comparison for two strings with respect to sorting. There are mainly three functions in the utility:

- Initialize
- Compare
- GetCollationKey

2.1 Initialize Function

This function initializes the Urdu Collation utility. It must be called before invoking any other function of the Urdu Collation utility. This function loads the collation element table from a specified file.

Syntax

```
bool Initialize (char* collationTable);
```

Parameters

collationTable file path and name of collation table to be loaded

Return Value

If the function succeeds, it returns true, else false.

2.2 Compare Function

Initialize function should be called before calling this function. This function takes two Unicode strings and compares them.

Syntax

int Compare(wstring source, wstring target, int strength=3);

Parameters

source the source string to be compared with

Target the string that is to be compared with the source string

strength 1: to ignore secondary and tertiary differences (e.g. ignore the diacritical differences on the same base letter)
2: to ignore tertiary differences (e.g. ignore honorific signs differences)
3: to include all comparisons
The default value is 3.

Return Value

Returns the comparison result; >0: if the source string is greater than the target string, <0: if the source is less than the target. Otherwise, returns 0 (equal).

2.3 *GetCollationKey*

Initialize function should be called before calling this function. This function takes a Unicode strings and returns a collation key against that string. This function is helpful for speeding the sorting algorithm.

Syntax

string GetCollationKey(wstring str, int strength=3)

Parameters

str Unicode string

strength 1: to ignore secondary and tertiary differences (e.g. ignore the diacritical differences on the same base letter)
2: to ignore tertiary differences (e.g. ignore honorific signs differences)
3: to include all comparisons
The default value is 3.

Return Value

Returns the collation key of the given *str*.

3 Sample Application

Syntax

Collation *filename1* *filename2*

Where
filename1: source filename for sorting
filename2: target filename for sorted results

Format of source file

The first line contains the count of total number of lines in a file. The remaining lines contain words for sorting.

4 File Formats

4.1 Urdu Collation Table

This file contains the collation sequence of Urdu alphabet. Urdu sorting requires at least three levels. At the first level of sorting, only the basic Urdu characters will be sorted. Once the characters determine word sequence, aerab are used to determine the sequence of words having the same characters. Finally the third level is used to sort special symbols e.g. honorific mark.

The first line of this file contains the number (in hexadecimal) of collation elements (excluding composite collation elements). The remaining each line has format [Code][C_1,C_2,C_3] for singe character or [Code1,Code2] [C_1,C_2,C_3] for composite characters. Where *Code* is the Unicode value of the character and C_1, C_2 and C_3 define the level of collation sequence at level1, level2 and level3 respectively. The values of C_1, C_2 and C_3 must be less than 0x7F.