Urdu Component Development Project

Application Programming Interface
(Urdu Normalization Utility)

September 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	026-09-2007	1.0.0.0	Initial document

1 Introduction

Normalization is a process to convert multiple equivalent representations of data to consistent underlying normal forms. Normalized data may have two forms: composed or decomposed. Composition is a process to combine the characters wherever possible, for example ا+◌ٓ (0627+0653) will assume آ (0622). Decomposition is an opposite process, breaking pre-composed characters back into their constituents.

The Unicode Normalization standard [1] defines two equivalences between characters: canonical equivalence and compatibility equivalence. And four normalization forms have been defined by Unicode standard, that are:

1. Normalization form D (NFD) or canonical decomposition
2. Normalization form C (NFC) or canonical decomposition followed by canonical composition
3. Normalization form KD (NFKD) or compatibility decomposition
4. Normalization form KC (NFKC) or compatibility decomposition followed by canonical composition

Urdu Normalization utility provides support for three normalization forms: Normalization form D (NFD) Normalization form C (NFC) and Normalization form KD (NFKD). The normalization form KD (NFKD) provided by a utility is a non-reversal process (the result may not be converted back to its original form).

This document enlists Urdu Normalization API with its complete specification. In order to test the Urdu Normalization API, a sample application has been developed. It uses Urdu Normalization API (DLL) and provides examples for Urdu normalization. .

2 Urdu Normalization API

Urdu Normalization API is a dynamic link library that provides normalized form of given string according to the given mode. There are mainly two functions in the API:

- Initialize
- Normalize

2.1 Initialize Function

This function initializes the Urdu Normalization API. It must be called before invoking any other function of the Urdu Normalization API. This function loads the normalization composition and decomposition rules from specified files. The file paths are given in a config.ini file.

Syntax

```
bool Initialize ();
```

Return Value

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

2.2 Normalize Function

Initialize function should be called before calling this function. This function normalizes the source Unicode string according to specified normalization mode.

Syntax

```
bool Normalize(unsigned char * source, NormMode mode, unsigned char * result);
```

Parameters

source	the input string to be normalized
mode	the normalization mode (NFD, NFC, NFKD)
result	the normalized string

Return Value

True if string is successfully normalized otherwise false

3 File Formats

3.1 Config File

This file contains the file paths to be used in the API. This file should exist at the root path of application.

3.2 NormalizeNFC file

This file contains normalization rules for composition (NFC). Each line of file contains one rule. The format of a rule is: *replace:pattern* (right to left), where *replace* may be empty.

3.3 NormalizeNFD file

This file contains normalization rules for decomposition (NFD). Each line of file contains one rule. The format of a rule is*: replace:pattern* (right to left), where *replace* may be empty.

3.4 NormalizeNFKD file

This file contains normalization rules for decomposition (NFD). Each line of file contains one rule. The format of a rule is*: replace:pattern* (right to left), where *replace* may be empty. The rules given in this file are non-reversal (the result may not be converted back to its original form).

References

[1]. *Unicode Normalization Forms*, Unicode Standard Annex # 15 (http://www.unicode.org/reports/tr15/).