# Find and Replace Utility for Khmer

## Abstract

*This paper discusses the research on the Find and Replace for Khmer Unicode fonts. The document is divided into three main sections. Section 2 discusses the analysis of the condition for Finding procedure and the Find and Replace technique. Section 3 presents the result of the studies. Section 4 discusses the findings and implications of the work that will present all the cases that have not been handled yet.*

## 1. Introduction

According to the Unicode Standard Version 4.0 from the Unicode Consortium Khmer script, called *aksaa khmae* ("Khmer Letters"), is the official script in Cambodia. It is descended from the Brahmi script of South India, as are Thai, Lao, Myanmar, Old Mon, and others. The exact sources have not been determined, but there is a great similarity between the earliest inscriptions in the region and the Pallawa script of the Coromandel Coast of India. Khmer has been a unique and independent script for more than 1,400 years.

After creating Khmer Unicode, It will be accessible to all members of the Cambodian Society. More Khmer applications will be developed, including Conversion, Smart Typing, Spell Checker, Khmer Dictionary, Lexicon Development, which will help Cambodians to understand better their own language.

Moreover, people, especially in rural areas where foreign language is not commonly used, they are able to use these applications with their work. Khmer Smart Typing is always used mostly for typing Khmer document. Nowadays, It has difficulty when Cambodian people want to find or replace Khmer language because this application don't exist.

## 2. Method

### 2.1. Architecture of the application

To ensure the user-friendliness, the embedded applications for MS Office are proposed. The system will be developed by dividing its functionalities into three main modules: MS Word Suite, Automation Application, and find and replace Assembly. Figure A.1 of Appendix A illustrates the system.

- **Automation application:** All the applications in this module act as the intermediate between the find and replace assembly and MS Word suite. Its tasks are to get the content of the text box when user input to be compared to the MS Word suite, to find and replace the text in MS Word suite, and select text in document when it found.
- **Conversion Assembly** is the core module of the system. Its main function is to find or replace text, return index and select text when it found.

### 2.2. Find and Replace algorithm

The find and replace process:

- For finding, input sentence to find, finding function gets this finding text to compare with content in MS Word document. If it found, it can return index and select text on that word.
- Replacing, according to find function it can get index and select text when it found, enter the text that we want to replace then replace function get this text to replace in MS Word document.

### 2.3 Option and Condition Find and Replace

**Whole words only** (តែពាក្យទាំងមូល)

Searches for whole words or cells that are identical to the search text

**Match Case** (ករណី ដំណូច)

Distinguishes between uppercase and lowercase characters

### Regular expressions (កន្សោមនិយ័ត)

Allows you to use wildcards in your search

### List of Regular Expressions

. Match any single character (e.g., m.d matches *mad*, *mod*, *m3d*, etc.)

[] *Bracket expression*: Match any one of the enclosed characters (e.g., [a-z] matches a lowercase ASCII letter, a digit)

^ *Start-of-string anchor*: Match only at the start of a string (e.g., ^hi matches *hi* and *his* but not *this*)

$ *End-of-string anchor*: Match only at the end of a string (e.g., hi$ matches *hi* and *chi* but not *this*)

* *Zero-or-more quantifier*: makes the previous part of the RE match zero or more times (e.g., M.*D matches *MD*, *MAD*, *MooD*, *M.D*, etc.)

? *Zero-or-one quantifier*: makes the previous part of the RE match zero or one time (e.g., hi!? matches *hi* or *hi!*)

+ *One-or-more quantifier*: makes the previous part of the RE match one or more times (e.g., hi!+ matches *hi!* or *hi!!* or *hi!!!* or ...)

| *Alternation* (vertical bar): Match just one alternative (e.g., this|that matches *this* or *that*)

[^a-s] Represents any character that is not between a and s.

{2} Defines the number of times that the character in front of the opening bracket occurs. For example, "tre{2}" finds "tree".

## 3. Experiment result

In our experiments, we tested the find and replace application of the approach. The approach has achieved a good rate, but it is not perfect as we are expected.

## 4. Discussion

### 4.1. Subscript of TA "◌្ត" and DA "◌្ដ"

In Khmer script, the subscript of TA and DA are visibly the same. It is very difficult for the find and replaces assembly to identify which one the user means to write. The find function does not handle the case. It makes user confuse between Subscript of TA and DA when user enter text.

### 4.2. Reordering Characters

All Khmer syllables begin with a consonant, independent vowel, or number. The following should be considered as canonical ordering for Khmer Unicode input. The ordering is in the same order that the Khmer syllable is formed and produces the correct sort/search order. Any device using Khmer Unicode should use this input sequence order to correctly handle Khmer text.

It is important for the user inputting the text to remember that although it is possible to input some of the formed sequences by using individual glyphs, the Unicode characters that are input must be in a correctly defined and consistent order for sorting and searching mechanisms to work.

For example, a person might try to enter a syllable with U+179F, U+179A and U+178A.
A user might think that they look the same as inserting U+179F, U+178A and U+179A. However, the meaning is the same, but find and replace assembly not handle the case.

## 5. Conclusion

This paper presents an approach to find and replace Khmer Unicode. The approach has achieved a satisfactory result. The utility is the main bridge to increase the use of Khmer Unicode. It enhances the standardization and localization of Khmer scripts in the computer. Moreover, the project is able to user find and replace Khmer text document. However, the find and replace application is not yet as high as expected, but acceptable.

### Future works

For the project, some work need to be done to improve the performance of the system. There is a need to:

- Find solution for some unhandled problems as discussed in the discussion part such as the subscript of TA and DA and Reordering Characters.
- Find and Replace data in the database and other applications.
- Find and Replace procedure depends on Khmer Grammar

## 6. References

- http://ftp.unicode.org/charts/PDF/Unicode-4.0/U40-1780.pdf

- http://www.tcl.tk/doc/howto/regexp81.tml