

# SỬ DỤNG CÔNG CỤ LẬP TRÌNH MACRO VBA XÂY DỰNG CÁC TIỆN ÍCH XỬ LÝ VĂN BẢN USING THE MACRO VBA TO CREATE WORD PROCESSING UTILITIES

PHAN HUY KHÁNH

*Trường Đại học Bách khoa, Đại học Đà Nẵng*

## TÓM TẮT

Sử dụng công cụ vĩ lệnh (macro) VBA (Visual Basic for Application) trong hệ thống phần mềm MSOffice, chúng tôi đã xây dựng một bộ sưu tập các trình tiện ích giúp giải quyết nhanh chóng và hiệu quả nhiều vấn đề đặt ra trong quá trình thực hiện các đề tài nghiên cứu khoa học về xử lý tiếng Việt: sắp xếp văn bản, chuyển đổi dạng thức văn bản, chèn nghĩa tương đương trong các từ điển đa ngữ, gỡ nhanh gỡ tắt văn bản tiếng Việt, v.v... Hệ thống được xây dựng theo hướng mở, một mặt góp phần nâng cao chất lượng đào tạo, một mặt tạo ra môi trường nghiên cứu khoa học về xử lý ngôn ngữ tại khoa Công nghệ Thông tin & Điện tử Viễn thông, trường Đại học Bách khoa, Đại học Đà Nẵng.

## ABSTRACT

This paper presents the use of the macro VBA (Visual Basic for Application programming) within the preeminent features of MSOffice software, in order to create a collection of utility programs to solving quickly and effectively most of the problems occurred in the realization of our research projects: conversion of any codes systems, conversion of any texts formats, automatic insertion of the equivalent meanings in the multilingual dictionaries, increasing the speed of editing for the Vietnamese documents, etc. Orienting toward the open sources method, our solution allows not only to enhance the training for IT human resources but also to provide the R&D environment for the language processing at our Faculty of IT.

**Từ khóa:** Vĩ lệnh macro VBA, trình tiện ích, chuyển đổi mã, chuyển đổi dạng thức văn bản, xử lý văn bản, từ điển đa ngữ, mã nguồn mở.

## 1. ĐẶT VẤN ĐỀ

Ngôn ngữ BASIC (viết tắt của Beginner's All-purpose Symbolic Instruction Code) là ngôn ngữ lập trình bậc cao do J. Kemeny và Th. Kurtz đề xuất từ những năm 1960 tại trường Đại học Dartmouth, Hoa Kỳ [1][10]. Nhờ tính năng dễ học, dễ lập trình, BASIC nhanh chóng được cải biên, phát triển và cài đặt thành nhiều phiên bản trên các dòng máy tính khác nhau như TinyBASIC, MicrosoftBASIC, CBASIC, BASICA, GW-BASIC, TrueBASIC... Từ năm 1988, A. Cooper đưa ra ý tưởng phát triển BASIC thành ngôn ngữ lập trình trực quan (visual programming language) và người ta gọi A. Cooper là cha đẻ của Visual Basic (viết tắt VB) [7]. Hiện nay, VB là một trong những ngôn ngữ được sử dụng chủ yếu trong môi trường Windows. VB rất phong phú về câu lệnh và từ khóa. Thư viện VB chứa hàng trăm hàm cho phép người sử dụng (NSD) phát triển dễ dàng ứng dụng trong các giao diện đồ họa GUI (Graphical User Interface). Khi lập trình VB, NSD thiết kế các giao diện và viết lệnh xử lý các đối tượng nhìn thấy được trên các giao diện này. VB có hai dạng sử dụng khác là VBScript và VBA (Visual Basic for Application). VBScript được dùng cho Windows và các ứng dụng trên Internet. VBA được sử dụng trong MSOffice (Word, Excel, Access, Project, v.v...) [8][9] để tạo ra các vĩ lệnh tự động hóa. Có thể nói phương pháp lập trình vĩ lệnh đã nâng cao hiệu quả sử dụng của các trình MSOffice.

Nhờ những tính năng ưu việt của lập trình vĩ lệnh VBA trong môi trường Windows, từ năm 1996, chúng tôi đã sử dụng VBA để xây dựng các trình tiện ích sử dụng trong các đề tài nghiên cứu khoa học thuộc lĩnh vực xử lý ngôn ngữ và xử lý tiếng Việt [2][3][4][5][6].

Tiếp theo đây, chúng tôi sẽ trình bày một số trình tiện ích chủ yếu phục vụ nâng cao tính năng soạn thảo văn bản như chuyển mã, gõ nhanh, gõ tắt, sắp xếp, định dạng văn bản..., phục vụ sưu tập dữ liệu và chèn nghĩa tiếng Việt tương đương cho các từ điển đa ngữ và đơn ngữ, chuyển đổi dạng thức dữ liệu từ RTF (Rich Text Format) sang XML (eXtensible Markup Language) hay sang cơ sở dữ liệu quan hệ, v.v... Để dễ theo dõi, trước hết chúng tôi sẽ trình bày cách tạo ra các vĩ lệnh VBA từ một ứng dụng MSOffice.

## 2. XÂY DỰNG CÁC TRÌNH TIỆN ÍCH

### 2.1 Tạo tự động vĩ lệnh VBA trong một ứng dụng MSOffice

Trong khi soạn thảo văn bản với Winword, hoặc thao tác bảng tính với Excel, hoặc tạo ra các trang trình diễn với PowerPoint..., NSD có thể tạo ra các vĩ lệnh một cách tự động hoặc như cách lập trình thông thường. Đó là các đoạn chương trình con (sub-program) có dạng:

```
Sub macro-name(arg-list)
' Dòng chú thích (comment line)
...
Các câu lệnh VBA (macro commands/instructions)
...
End Sub 'macro-name
```

Sau từ khoá **sub** (chữ đậm) là tên của vĩ lệnh do NSD tự đặt tùy ý. Mỗi vĩ lệnh có thể có hoặc vắng mặt các tham đối *arg-list*. Phần thân của vĩ lệnh gồm các dòng lệnh VBA. Kết thúc một vĩ lệnh là cặp từ khoá **End sub**. NSD có thể đặt tùy ý các dòng chú thích trong vĩ lệnh. Mỗi dòng chú thích được bắt đầu bởi một dấu nháy đơn ' (quote). Ví dụ sau đây là một vĩ lệnh tính diện tích hình chữ nhật:

```
Sub ComputeArea(Length, Width)
    Dim Area As Double ' Khai báo biến cục bộ
    If Length = 0 Or Width = 0 Then
        Exit Sub ' Nếu một trong hai tham đối = 0 thì thoát khỏi Sub
    End If
    Area = Length * Width ' Tính diện tích hình chữ nhật
    MsgBox Area ' In ra kết quả
End Sub ' ComputeArea
```

Để gọi chạy vĩ lệnh trên đây, ta cần viết một vĩ lệnh khác sử dụng lời gọi **Call**:

```
Sub MyFirstExample()
    Call ComputeArea(20, 10) ' Tính diện tích hình chữ nhật 10x20
End Sub ' MyFirstExample
```

Cách tạo mới tự động một vĩ lệnh như sau: gọi lệnh **Tool-Macro-Record New Macro...** để làm xuất hiện hộp thoại Record Macro. Đặt lại tên vĩ lệnh trong ô **Macro Name**. Chọn chế độ lưu cất vĩ lệnh (hộp thoại **Store Macro in**) rồi OK. Từ lúc này, NSD có thể tiến hành công việc và hệ thống sẽ tự động ghi chép lại bằng các lệnh VBA tương ứng. Sau khi thực hiện xong, NSD nhấp chuột tại nút **Stop Recording** (hoặc gọi lệnh **Tool-Macro-Stop Recording**) để kết thúc ghi vĩ lệnh.

Để có thể tận dụng chức năng tự động tạo ra các lệnh VBA để viết nhanh các vĩ lệnh xử lý một công việc nào đó, ta tiến hành qua hai bước như sau:

Bước 1: Tạo tự động một vĩ lệnh thực hiện một chức năng nào đó có liên quan đến công việc. Nhờ cách “bắt chước này”, còn được gọi vui là tạo “vốn lập trình” (programming funds), NSD đỡ mất thời gian tìm hiểu kỹ thuật lập trình VBA.

Bước 2: Lập trình trực tiếp trong cửa sổ soạn thảo của VBA. Gọi lệnh **Tool-Macro- Macros...** (hoặc tổ hợp phím **Alt+F8**) để làm xuất hiện hộp thoại Macros. Chọn tên vĩ lệnh rồi gọi **Edit**, hoặc gọi **Create** để bắt đầu làm việc với cửa sổ **NewMacros (code)**.

VBA có hệ thống trợ giúp lập trình rất phong phú. NSD có thể gõ một dấu chấm sau tên một đối tượng nào đó để chọn một mục con thừa kế hoặc gọi lệnh **Help** (phím **F1**). Ví dụ để đặt độ giãn giữa các dòng văn bản là 16, ta thực hiện lần lượt như sau: gõ **Selection.**, chọn **ParagraphFormat**, gõ dấu chấm rồi chọn **LineSpacing**, gõ tiếp giá trị 16, ta nhận được:

```
Selection.ParagraphFormat.LineSpacing = 16
```

Ví dụ sau đây tạo ra một hộp đối thoại để gõ vào tên tệp trước khi mở tệp:

```
Sub EnterFileName()  
Dim FileNam As Variant  
ChangeFileOpenDirectory "C:\MyDocs"  
FileNam = InputBox("Cho biết tên tệp ?",, 20)  
If Len(FileNam) = 0 Then  
    MsgBox "Tên tệp chưa hợp lý."  
Else  
    MsgBox "Bắt đầu mở tệp " & FileNam  
    Documents.Open FileName:=FileNam, ReadOnly:=True  
End If  
End Sub
```

Cách tạo vĩ lệnh trong Excel hay PowerPoint cũng được thực hiện tương tự.

## 2.2 Các trình tiện ích xử lý văn bản

### a. Vĩ lệnh chuyển đổi văn bản chữ hoa chữ thường

Một số phiên bản Windows hiện nay chưa cho phép chuyển đổi hoàn toàn một đoạn văn bản tiếng Việt dùng phông chữ Unicode từ chữ thường (lowercase) thành chữ hoa (uppercase) hay ngược lại. Chẳng hạn nếu chuyển đoạn *Trường Đại học Bách nghệ* bởi lệnh `Format-Change Case`, thì ta sẽ chỉ nhận được kết quả *TRƯỜNG Đại Học BÁCH NGHỆ*: các chữ có dấu ở *a o* và *ê* không chuyển được thành chữ hoa. NSD có thể sử dụng công cụ chuyển đổi trong các bộ gõ như UniKey, tuy nhiên, ta xây dựng vĩ lệnh chuyển đổi hoàn toàn một đoạn văn bản tiếng Việt Unicode từ chữ thường thành chữ hoa như sau:

```
Sub Lower2Upper()  
Dim i, j As Integer  
' Khai báo mảng 45 chữ thường và mảng 45 chữ hoa  
LC = Array(ChrW(7843), ChrW(7841), ChrW(7867), ChrW(7869), ChrW(7865), _  
    ChrW(7881), ChrW(7883), ChrW(7887), ChrW(7885), ChrW(7911), ChrW(7909), _  
    ChrW(7923), ChrW(7927), ChrW(7929), ChrW(7925), ChrW(7847), ChrW(7849), _  
    ChrW(7851), ChrW(7845), ChrW(7853), ChrW(7857), ChrW(7859), ChrW(7861), _  
    ChrW(7855), ChrW(7863), ChrW(7873), ChrW(7875), ChrW(7877), ChrW(7871), _  
    ChrW(7879), ChrW(7891), ChrW(7893), ChrW(7895), ChrW(7889), ChrW(7897), _  
    ChrW(7901), ChrW(7903), ChrW(7905), ChrW(7899), ChrW(7907), ChrW(7915), _  
    ChrW(7917), ChrW(7919), ChrW(7913), ChrW(7921))  
UC = Array(ChrW(7842), ChrW(7840), ChrW(7866), ChrW(7864), _  
    ChrW(7880), ChrW(7882), ChrW(7886), ChrW(7884), ChrW(7910), ChrW(7908), _  
    ChrW(7922), ChrW(7926), ChrW(7928), ChrW(7924), ChrW(7846), ChrW(7848), _  
    ChrW(7850), ChrW(7844), ChrW(7852), ChrW(7856), ChrW(7858), ChrW(7860), _  
    ChrW(7854), ChrW(7862), ChrW(7872), ChrW(7874), ChrW(7876), ChrW(7870), _  
    ChrW(7878), ChrW(7890), ChrW(7892), ChrW(7894), ChrW(7888), ChrW(7896), _  
    ChrW(7900), ChrW(7902), ChrW(7904), ChrW(7898), ChrW(7906), ChrW(7914), _  
    ChrW(7916), ChrW(7918), ChrW(7912), ChrW(7920))  
Selection.Range.Case = wdLowerCase ' Chuyển tự động chữ thường thành chữ hoa  
N = Selection.Characters.Count ' Nhận biết đoạn văn bản  
L = Selection.Text  
' Chuyển đổi những chữ chưa được đổi tìm thấy trong mảng chữ thường  
For i = 1 To N  
    For j = 0 To 44  
        If Mid(L, i, 1) = LC(j) Then  
            L = Left(L, i - 1) & UC(j) & Right(L, N - i)  
        Next j  
    Next i  
If (Asc(Right(L, 1))) = 13 Then ' Xử lý ký tự hết đoạn  
    Selection.TypeText Text:=Left(L, N - 1)  
Else  
    Selection.TypeText Text:=L  
End If  
End Sub ' Lower2Upper
```

Trình tự thực hiện như sau: đầu tiên, tạo tự động vĩ lệnh tìm kiếm thay thế toàn bộ các chữ cái có dấu và có dấu thanh để nhận được các mã Unicode trong dạng hàm ChrW của VBA. Sau đó gọi lệnh `Tool-Macro- Macros...`, chọn Create để soạn thảo chương trình vĩ lệnh.

Ngược lại, để tạo vĩ lệnh chuyển đổi văn bản tiếng Việt Unicode từ chữ hoa ra chữ thường, sử dụng lại hai mảng chữ thường và chữ hoa và tiến hành theo cách tương tự với chú ý hoán đổi vai trò xử lý các chữ cái.

**b. Vĩ lệnh sắp xếp văn bản**

Hầu hết các trình gõ tiếng Việt thông dụng hiện nay như UniKey, Vietkey... đều chưa có công cụ trợ giúp sắp xếp văn bản mã Unicode. Nếu NSD sắp xếp văn bản sử dụng lệnh sắp xếp của MSOffice, chẳng hạn gọi lệnh `Table-Sort` để sắp xếp các văn bản tiếng Việt thì kết quả nhận được (xem bảng 1) không tuân theo thứ tự chữ cái như quy ước trong các từ điển tiếng Việt (Hoàng Phê, Nguyễn Như Ý...).

<i>Đinh</i>	<i>Đính</i>	<i>Đình</i>	<i>Đĩnh</i>	<i>Đinh</i>	<i>Định</i>	<i>Dùng</i>
-------------	-------------	-------------	-------------	-------------	-------------	-------------

*Bảng 1. Kết quả sắp xếp văn bản tiếng Việt mã Unicode.*

Nhìn vào bảng ta thấy thứ tự chữ cái tham đối như sau: *Đ* đứng trước *D*, thứ tự dấu thanh là *không dấu, sắc, huyền, ngã, hỏi, nặng*. Theo các từ điển tiếng Việt thông dụng hiện nay, đối với các chữ cái mang dấu, thứ tự chữ cái quy ước là *a ă â d đ e ê o ô ơ u ư* và thứ tự dấu thanh là *không dấu, huyền, hỏi, ngã, sắc, nặng*.

Để xây dựng trình sắp xếp, ta xây dựng một ánh xạ sắp xếp *I* cho mỗi chữ cái như sau:

$$\begin{aligned}
 I(\alpha) &= \alpha \text{ nếu } \alpha \text{ là một chữ cái không mang dấu,} \\
 &= \beta \text{ nếu } \alpha \text{ là một chữ cái mang dấu,} \\
 &= \gamma \text{ nếu } \alpha \text{ là một chữ cái có dấu thanh quy ước đặt sau } \alpha \text{ hoặc } \beta.
 \end{aligned}$$

Các giá trị mã sắp xếp  $\beta$  và  $\gamma$  được cho trong bảng 2. Khi đó, mỗi từ đơn, hay âm tiết tiếng Việt (chẳng hạn từ "*may mắn*" có hai âm tiết), có mã sắp xếp là kết quả của phép ghép liên tiếp (biểu diễn bởi phép +) các giá trị mã  $\alpha$ ,  $\beta$  và  $\gamma$  của từng chữ cái tương ứng.

Chữ cái mang dấu	Mã sắp xếp $\beta$	Dấu thanh	Mã sắp xếp $\gamma$
<i>ă</i>	az	<i>Huyền</i>	1
<i>â</i>	azz	<i>Hỏi</i>	2
<i>đ</i>	dz	<i>Ngã</i>	3
<i>ê</i>	ez	<i>Sắc</i>	4
<i>ô</i>	oz	<i>Nặng</i>	5
<i>ơ</i>	ozz		
<i>ư</i>	uz		

*Bảng 2. Bảng mã quy ước sắp xếp chữ cái và dấu thanh tiếng Việt.*

Ví dụ: *huong* có mã sắp xếp  $I(huong) = I(h)+I(u)+I(o)+I(n)+I(g) = huzozzng$ ,  
*hường* có mã sắp xếp  $I(hường) = I(h)+I(u)+I(ơ)+I(n)+I(g) = huzozz2ng$ , v.v...

Trong trình vĩ lệnh, quá trình sắp xếp văn bản được diễn ra thành 3 bước: bước 1 chuyển đổi xuôi văn bản sử dụng ánh xạ *I* quy ước trên đây, bước 2 tiến hành sắp xếp theo chế độ hiện có của MSOffice, cuối cùng bước 3 chuyển đổi ngược văn bản về lại mã Unicode.

Đối với các văn bản trong bảng (table), hiện tại Word chưa cho phép sắp xếp văn bản theo hàng (row) mà chỉ cho phép sắp xếp theo cột (column). Do đó trước khi sắp xếp, vĩ lệnh sắp xếp văn bản tiếng Việt Unicode cần nhận biết dữ liệu văn bản cần sắp xếp có phải nằm trong bảng không? Nếu có thì chép tất cả các phần tử, hay các ô (cell), của hàng hay cột sang một mảng trung gian để tiến hành sắp xếp theo thuật toán trên đây, sau đó chép kết quả vào lại hàng hay cột tương ứng của bảng.

Để tự động hoá thủ tục sắp xếp, ta có thể viết một vĩ lệnh dạng vị từ đoán nhận phần tử được lựa (selected) thường được thể hiện bằng cách đảo ngược màu nền (inverse) là gì? Chẳng hạn phần tử được lựa có thể là ký tự, từ đơn hay từ ghép, cụm từ, dòng (line), đoạn (paragraph), nhiều đoạn, ảnh, đồ hoạ, phần tử bảng, hàng, cột, v.v...

### c. *Vĩ lệnh quản lý từ điển gõ tắt văn bản*

Lâu nay, khi soạn thảo văn bản, NSD thường sử dụng các phông chữ (font) tiếng Việt theo thói quen hoặc theo cấu hình cài đặt sẵn của hệ thống. Các phông chữ hay gặp đó là ABC, TCVN, VNI, VietWare, BK HCM... Khi cần trao đổi văn bản, NSD có thể sử dụng công cụ trợ giúp chuyển mã văn bản (text conversion) có trong các trình gõ tiếng Việt như UniKey, Vietkey... để chuyển đổi văn bản sang phông chữ yêu cầu.

Tuy nhiên, nếu NSD muốn chuyển hoàn toàn hệ thống của mình sang một phông chữ mới, Unicode chẳng hạn, thì ngoài vấn đề chuyển đổi văn bản, NSD cần giải quyết vấn đề làm sao có thể sử dụng lại từ điển gõ tắt văn bản theo chế độ AutoText, AutoCorrect trong Word?

Các vĩ lệnh sau đây chuyển đổi phông chữ của từ điển phục vụ chế độ AutoText:

```
' Sao chép từ điển AutoText lên cửa sổ văn bản hiện hành
Sub CopyAutoText ()
Set myTemplate = ActiveDocument.AttachedTemplate
N = NormalTemplate.AutoTextEntries.Count
For I = 1 To N
    With Selection
        .TypeText Text:=myTemplate.AutoTextEntries(I).Name
        .TypeText Text:=vbTab
        .TypeText Text:=myTemplate.AutoTextEntries(I).Value
        .TypeParagraph
    End With
Next I
MsgBox "N=" & Str(N) ' Cho biết giá trị N
End Sub ' CopyAutoText
' Chuyển mã nội dung từ điển AutoText sang phông chữ mới
' Tạo mới từ điển AutoText
Sub CreateAutotext ()
I = 0: N = 1400 ' Lấy giá trị N trên đây
While I < N
    I = I + 1: Selection.HomeKey Unit:=wdLine
    TxtName = Selection.Paragraphs.First
    TxtName = Left(TxtName, Len(TxtName) - 1)
    With Selection
        .MoveDown:.HomeKey Unit:=wdLine
        .MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
        .MoveLeft Unit:=wdCharacter, Count:=1, Extend:=wdExtend
        ActiveDocument.AttachedTemplate.AutoTextEntries.Add _
        Range:=Selection.Range, Name:=TxtName
        .MoveDown
    End With
Wend
End Sub
```

Một cách tương tự, NSD tạo mới từ điển AutoCorrect trong phông chữ mới. Với phương pháp tạo vĩ lệnh như trên, NSD có tạo các tiện ích xử lý văn bản khác như định dạng trên các đoạn văn bản (thay đổi vị trí trước hay sau một đoạn, co giãn chữ, chuyển đổi bảng, v.v...).

### 2.3 *Quản lý cơ sở dữ liệu (CSDL) từ vựng*

Trong quá trình thực hiện đề tài xây dựng từ điển đa ngữ Pháp-Anh-Việt (dự án FEV [2], [3], [6]), chúng tôi cần sưu tập dữ liệu trong các từ điển khác nhau (Pháp-Việt-Pháp, Anh-Việt-Anh) để xây dựng các CSDL từ vựng trung gian. Dữ liệu sau khi xử lý phục vụ chèn nghĩa tiếng Việt tương đương cho các từ điển đa ngữ.

Từ điển đa ngữ gồm các khối dữ liệu là các tệp văn bản Winword được định nghĩa nhất quán theo một mẫu văn bản xác định. Đó là một tập hợp các dạng thức (style) thể hiện cách định dạng một đoạn văn bản được định nghĩa nhờ các lệnh đơn (Format) như: phông chữ (Format-Font)..., kiểu trình bày đoạn (Format-Paragraph), v.v... Mỗi dạng thức biểu diễn một thành phần của từ điển đa ngữ. Chính sự khác nhau về cách trình bày đoạn cho phép phân biệt

các thành phần của từ điển như mục từ và nội dung của mục từ, phần hướng dẫn phát âm, hình loại ngữ pháp, thứ tự chỉ dẫn nghĩa, các nghĩa tương đương, cụm từ, câu ví dụ, v.v...

Vĩ lệnh sau đây tạo ra tệp chứa các cụm từ Pháp-Việt từ CSDL từ vựng trung gian. Hoạt động của vĩ lệnh như sau: khi gặp ký tự @ hoặc nghĩa tiếng Việt thì vượt qua, dừng lại để sao chép các cụm từ "french\_phrase". Sau khi sao chép hết các cụm từ, các vĩ lệnh khác sẽ chèn tự động vào các tệp từ điển đa ngữ FEV.

```

Sub LV_XuLyPara()
Dim i, k As Integer
' Nhận biết các tệp CSDL từ vựng trung gian nguồn
Fdoc = Array("FV-a-d", "FV-d-h", "FV-h-m", "FV-m-s", "FV-s-z")
For k = 0 To 4 ' xử lý từng tệp nguồn
' Chuẩn bị tệp đích
Windows(SrcFileName).Activate: Selection.HomeKey Unit:=wdStory
' Đếm số đoạn văn bản trong tệp nguồn
N = ActiveDocument.Paragraphs.Count: i = 1
Do Until i >= N
Selection.HomeKey Unit:=wdLine
If Selection.Style = "entry" Then
Selection.MoveDown: i = i + 1
If Selection.Style = "french_cat" Then
Selection.MoveDown: i = i + 1
ElseIf Selection.Characters(1) <> " " Then
' Chuẩn bị dữ liệu trong tệp nguồn
' Kiểm tra tệp đích để chép lên tệp đích
End If
Else
' Chép dữ liệu tệp nguồn
If Head=" S" Or Head=" s" Or Selection.Characters(1)<>" " Then
' Chép lên tệp đích
' Tiếp tục quay lại tệp nguồn để kiểm tra đã chép hết chưa ?
' Chưa hết thì tiếp tục chép lên tệp đích
ElseIf Selection.Characters(1) = " " Then
' Vượt qua phần nghĩa tiếng Việt tương đương
End If' Xử lý hết một kiểu đoạn "entry"
End If
End If
Loop ' Xử lý hết một đoạn văn bản
Next k ' Xử lý hết một tệp nguồn
' Dọn dẹp kết thúc
End Sub ' LV_XuLyPara

```

## 2.4 Chuyển dạng thức dữ liệu RTF/XML

Vĩ lệnh sau đây cho phép chuyển các tệp văn bản Word RTF sang XML.

```

Sub CreateXMLFeV()
Dim i, k As Integer
' Khai báo các tệp làm việc:
Stl = Array("viet_equ", "viet_phrase", "viet_sentence")
TxtU = Array("A", "B", "C1", "C2", "D", "E", "F", "G", "H", "I", "JK", "L", "M", "N", "O", "P1", "P2", "Q", "R", "S", "T", "U", "V", "WXYZ")
' Tạo các tiêu đề:
' <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
' <?xml:stylesheet type="text/css" href="fev.css"?>
' <!DOCTYPE dictionary SYSTEM "fev.dtd">
' <dictionary name="FeV" source-language="fr" target-language="en,vn">
For k = 0 To 23
' Mở các tệp nguồn và tệp đích, chuẩn bị chuyển đổi
Selection.HomeKey Unit:=wdStory
S = LCase(Selection.ParagraphFormat.Style)
Do While S <> "endfev"
Do While S <> "entry"

```

```

Selection.MoveDown Unit:=wdParagraph, Count:=1
Loop
' Xử lý một đoạn "entry"
  Do While S <> "entry" And S <> "endfev"
    ' Tạo một khoản mục XML từ tệp nguồn
    Windows(TxtU(k)).Activate
    ' Chép khoản mục XML lên tệp đích
    Selection.MoveDown Unit:=wdParagraph, Count:=1
    S = LCase(Selection.ParagraphFormat.Style)
    If S = "endfev" Then
      Windows(XMLFName).Activate
      Selection.TypeText Text:=DB2 & S_old & FN1
      Selection.TypeText Text:=H5
    End If
    Txt = Selection.Paragraphs.First: L1 = Len(Txt)
  Loop
' Kết thúc một đoạn "entry"
  Loop
' Dọn dẹp
Next k
End Sub ' CreateXMLFeV

```

### 3. KẾT LUẬN

Hệ thống các trình tiện ích viết bằng vĩ lệnh VBA được lưu giữ trong các mẫu văn bản Normal.dot cho phép NSD tùy ý cải biên phát triển, nhằm tạo ra một môi trường làm việc cho sinh viên ngành Tin học.

Những ý tưởng xây dựng các trình vĩ lệnh trên đây đều có thể vận dụng được cho các trình ứng dụng Excel, PowerPoint của MSOffice tùy theo nhu cầu.

### TÀI LIỆU THAM KHẢO

- [1] Nguyễn Tiến và các tác giả khác, *Kỹ năng lập trình Visual Basic 6.0*, Nhà XB Giáo dục, 2002.
- [2] Phan Huy Khánh, *Thiết kế cơ sở dữ liệu từ vựng tiếng Việt có cấu trúc phục vụ xử lý ngôn ngữ*, Đề tài NCKH trọng điểm cấp Bộ 1996-1998, mã số B96-15-TĐ-03, đã nghiệm thu.
- [3] Phan Huy Khánh, *Thiết kế hệ thống khai thác CSDL từ vựng đa ngữ Pháp-Anh-Việt*, Đề tài NCKH cấp Bộ 1998-2000, mã số B2001-15-04, đã nghiệm thu.
- [4] Phan Huy Khánh, *Good Spelling of Vietnamese Texts, one aspect of computational linguistics in Vietnam*, Proceeding of ACL-2000, 38<sup>th</sup> Conference, HồngKông, 2000, t.593-594.
- [5] Phan Huy Khánh, Võ Trung Hùng, *Thiết kế cơ sở dữ liệu đa ngữ ngữ pháp tiếng Việt*, Tạp chí Khoa học Công nghệ các trường Đại học Bách khoa, No 36+37, 2002, t19-t24.
- [6] Phan Huy Khánh, *Xây dựng từ điển đa ngữ sử dụng dạng thức văn bản RTF Winword*, Kỷ yếu Hội thảo Khoa học Quốc gia ICT.rda'2003, trang 323-332.
- [7] A. Cooper, *Why I am called "the Father of Visual Basic"*, Cooper Interaction Design, 1996.
- [8] Microsoft Knowledge Base Article – 212682, *VBA macro examples to insert text into a document in Word 2000*, Trang web: <http://support.microsoft.com/>
- [9] Microsoft Knowledge Base Article – 212710, *WD2000: Retrieving Settings from Word Dialog Boxes Using OLE*, Trang web: <http://support.microsoft.com/>
- [10] Microsoft Knowledge Base Article – 226118, *List of resources that are available to help you learn Visual Basic for Applications programming*, Trang web: <http://support.microsoft.com/>