Функции для работы с текстовым буфером обмена (API) Печать E-mail

Автор Administrator

07.12.2020 г.

Функции для извлечения и записи строк в буфер обмена Windows.

При копировании текста в буфер обмена функция временно переключает раскладку клавиатуры на русскую для того, чтобы избежать проблем, связанных с некорректным определением буфером обмена кодовой страницы русского текста в формате Unicode. При выоде из функции восстанавливается старая раскладка клавиатуры (которая была при вызове функции)

Автор: "найдено в интернете"

Option Compare Database Option Explicit

'am 030305\_14:59:26 --begin-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*

'так как проблема при копировании в буфер тектста

'в юникода при латинской раскладке

Private Declare Function GetKeyboardLayoutName Lib "user32" \_

Alias "GetKeyboardLayoutNameA" (ByVal pwszKLID As String) As Long

Private Declare Function LoadKeyboardLayout Lib "user32" \_

Alias "LoadKeyboardLayoutA" (ByVal HKL As String, \_

ByVal flags As Long) As Long

Private Const KL\_NAMELENGTH = 9

'am 030305\_14:59:26 --end-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*

Declare Function OpenClipboard Lib "user32" (ByVal hwnd As Long) As Long

Declare Function CloseClipboard Lib "user32" () As Long

Declare Function GetClipboardData Lib "user32" (ByVal wFormat As Long) As Long

Declare Function GlobalAlloc Lib "kernel32" \_

(ByVal wFlags&, ByVal dwBytes As Long) As Long

Declare Function GlobalLock Lib "kernel32" (ByVal hMem As Long) As Long

Declare Function GlobalUnlock Lib "kernel32" (ByVal hMem As Long) As Long

Declare Function GlobalSize Lib "kernel32" (ByVal hMem As Long) As Long

Declare Function lstrcpy Lib "kernel32" (ByVal lpString1 As Any, \_

ByVal lpString2 As Any) As Long

Declare Function lstrcpy2 Lib "kernel32" Alias "lstrcpy" \_

(ByVal lpString1 As Any, ByRef lpString2 As Byte) As Long

Declare Function EmptyClipboard Lib "user32" () As Long

Declare Function SetClipboardData Lib "user32" (ByVal wFormat As Long, \_

ByVal hMem As Long) As Long

Public Const GHND = &H42

Public Const MAXSIZE = 4096

Public Const CF\_TEXT = 1

Function ClipBoard\_GetData() As String

Dim hClipMemory As Long

Dim lpClipMemory As Long

Dim MyString As String

Dim RetVal As Long

Dim lSize As Long

If OpenClipboard(0&) = 0 Then

MsgBox "Cannot open Clipboard. Another app. may have it open"

Exit Function

End If

' Obtain the handle to the global memory

' block that is referencing the text.

hClipMemory = GetClipboardData(CF\_TEXT)

If hClipMemory = 0 Then

MsgBox "Could not allocate memory"

GoTo OutOfHere

End If

'am 040610 - размер нужно узнать

lSize = GlobalSize(hClipMemory)

' Lock Clipboard memory so we can reference

' the actual data string.

lpClipMemory = GlobalLock(hClipMemory)

If lpClipMemory <> 0 Then

MyString = Space$(lSize)

RetVal = lstrcpy(MyString, lpClipMemory)

RetVal = GlobalUnlock(hClipMemory)

' Peel off the null terminating character.

MyString = mID(MyString, 1, InStr(1, MyString, Chr$(0), 0) - 1)

Else

MsgBox "Could not lock memory to copy string from."

End If

OutOfHere:

RetVal = CloseClipboard()

ClipBoard\_GetData = MyString

End Function

Function ClipBoard\_SetData(MyString As String)

Dim hGlobalMemory As Long, lpGlobalMemory As Long

Dim hClipMemory As Long, x As Long

' Allocate movable global memory.

'-------------------------------------------

hGlobalMemory = GlobalAlloc(GHND, Len(MyString) + 1)

' Lock the block to get a far pointer

' to this memory.

lpGlobalMemory = GlobalLock(hGlobalMemory)

' Copy the string to this global memory.

lpGlobalMemory = lstrcpy(lpGlobalMemory, MyString)

'lpGlobalMemory = lstrcpy2(lpGlobalMemory, b(0))

' Unlock the memory.

If GlobalUnlock(hGlobalMemory) <> 0 Then

MsgBox "Could not unlock memory location. Copy aborted."

GoTo OutOfHere2

End If

' Open the Clipboard to copy data to.

If OpenClipboard(0&) = 0 Then

MsgBox "Could not open the Clipboard. Copy aborted."

Exit Function

End If

' Clear the Clipboard.

x = EmptyClipboard()

'am 030305\_15:16:05 --begin-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*

'переключаемся на русскую раскладку чтобы не иметь

'проблем с русским текстом в буфере

'(некорректно понимается кодовая страница)

Dim sOldLang As String

sOldLang = switchLang("00000419")

'am 030305\_15:16:05 --end-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Copy the data to the Clipboard.

hClipMemory = SetClipboardData(CF\_TEXT, hGlobalMemory)

OutOfHere2:

If CloseClipboard() = 0 Then

MsgBox "Could not close Clipboard."

End If

'am 030305\_15:17:26 --begin-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*

'возвращаем раскладку на место

If Len(sOldLang) > 0 Then \_

sOldLang = switchLang(sOldLang)

'am 030305\_15:17:26 --end-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*

End Function

Private Function getCurrLang() As String

Dim layoutname As String \* KL\_NAMELENGTH

Dim z As Long

z = GetKeyboardLayoutName(layoutname)

If z = 0 Then

getCurrLang = ""

Else

getCurrLang = StrZ(layoutname)

End If

End Function

'Переключает на указанную sNewLang раскладку - возвращает старую раскладку

'am 030305\_15:13:39

Private Function switchLang(sNewLang As String) As String

'"00000419" - русская '"00000409" - латинская

switchLang = getCurrLang

If StrComp(switchLang, sNewLang) <> 0 Then

LoadKeyboardLayout sNewLang, 1

End If

End Function

'v\_1.0.0 990630

Public Function StrZ(par As String) As String

Dim nSize As Long, i As Long, Rez As String

nSize = Len(par)

i = InStr(1, par, Chr(0)) - 1

If i > nSize Then i = nSize

If i < 0 Then i = nSize

StrZ = mID(par, 1, i)

End Function