Soubory typu INI představují prosté textové soubory, které slouží zpravidla aplikacím jako místo pro uložení výchozích, přednastavených hodnot. Snadno se upravují, lze je pohodlně přenášet a snižují riziko nakopnutí aplikace, což v případě Excelu znamená ruční zápis uživatelem do listu, názvu či deklarační části VBA (typicky cesty k souborům, pokud je nemůžeme deklarovat s pomocí ThisWorkbook.Path a které budou různé v době návrhu aplikace a při jejím ostrém nasazení). Ačkoliv otevírání souborů a čtení z nich patří k nejpomalejším operacím pod VBA, máme k dispozici API funkce, které nám práci zpříjemní.

Soubory INI sestávají z jednoho nebo více následujících bloků:

[sekce]
klíč=hodnota

Komentáře začínají středníkem. Jak vidíte na obrázku níže, klíč CLOSE v sekci [SK] nemá přiřazenu hodnotu.

Soubor INI – ukázka

Možná jste si na obrázku také všimli nezvyklého kódování UCS-2 Little Endian. Je to proto, že jsem chtěl narozdíl od mnoha návodů na internetu ukázat práci s texty ve formátu Unicode a uvedené kódování mi vyšlo jako jediné spolehlivé pod Windows. Můžete experimentovat i s Poznámkovým blokem, nicméně pokud do budoucna budete potřebovat překódovat textové soubory (TXT, PHP apod.) mezi ANSI, UTF-8 a UCS-2, pak doporučuji Notepad++, který je zdarma.

Jak jsem se zmínil, ke čtení a zápisu nám poslouží funkce API, konkrétně GetPrivateProfileString a WritePrivateProfileString (WritePrivateProfileSection ponecháme stranou). My budeme potřebovat jejich Unicode verzi, tedy GetPrivateProfileStringW a WritePrivateProfileStringW. Ty mají parametry typu Long, neboť nepracují se samotnými textovými řetězci, ale s tzv. „pointery“, což jsou v podstatě odkazy do paměti, kde se uložené řetězce nacházejí. Pro nás to znamená, že namísto předání prostého textu budeme uvádět StrPtr(text).

Pozn. StrPtr podle všeho neumí pracovat s řetězci o pevné délce. Ostatně řetězce pevné délky, stejně jako pointery již Visual Basic od verze 2008 neobsahuje.

```vba
'konstanta pro MessageBox
Public Const MB_OK = &H0&

Public Declare Function MessageBox Lib "user32" Alias "MessageBoxW" (ByVal hwnd _
   As Long, ByVal lpText As Long, ByVal lpCaption As Long, ByVal wType As Long) As _
   Long

Public Declare Function GetPrivateProfileString Lib "kernel32" Alias _
   "GetPrivateProfileStringW" (ByVal lpApplicationName As Long, ByVal lpKeyName As _
   Long, ByVal lpDefault As Long, ByVal lpReturnedString As Long, ByVal nSize As _
   Long, ByVal lpFileName As Long) As Long

Public Declare Function WritePrivateProfileString Lib "kernel32" Alias _
   "WritePrivateProfileStringW" (ByVal lpApplicationName As Long, ByVal lpKeyName _
   As Long, ByVal lpString As Long, ByVal lpFileName As Long) As Long

'INI soubor
Public Const sFileINI As String = "language.ini"

Sub TestCteniZapisINI()

   'cesta k souboru INI
   sPathFileINI = ThisWorkbook.Path & "" & sFileINI

   'přečtení hodnot
   'v okně Locals se nezobrazí korektně
   sText1 = ReadItemUnicodeINI(sPathFileINI, "CZ", "CLOSE")
   sText2 = ReadItemUnicodeINI(sPathFileINI, "RU", "CLOSE")
   sText3 = ReadItemUnicodeINI(sPathFileINI, "FR", "CLOSE")

   'použití textu
   Range("A1") = sText1
   Range("A2") = sText2
   Range("A3") = sText3

   'MsgBox nezvládá Unicode
   'použijeme API funkci MessageBox
   MessageBox 0&, StrPtr(sText2), StrPtr("Microsoft Excel"), MB_OK

   'zápis
   Call WriteItemUnicodeINI(sPathFileINI, "SK", "CLOSE", "zatvoriť")

End Sub

Public Function ReadItemUnicodeINI(ByVal sFile As String, ByVal sSection As _
   String, ByVal sKey As String, Optional sDefaultValue As String) As String
   Dim lsRetval As String
   Dim liLength
   Dim lsDefVal As String
   If IsMissing(sDefaultValue) Then
      lsDefVal = "(nenalezeno)"
   Else
      lsDefVal = sDefaultValue
   End If
   lsRetval = VBA.Space$(1024)
   liLength = GetPrivateProfileString(StrPtr(sSection), StrPtr(sKey), _
      StrPtr(lsDefVal), StrPtr(lsRetval), 1024, StrPtr(sFile))
   lsRetval = VBA.Left(lsRetval, liLength)
   ReadItemUnicodeINI = lsRetval
End Function

Public Function WriteItemUnicodeINI(ByVal sFile As String, ByVal sSection As _
   String, ByVal sKey As String, sHodnota As String) As String
   Dim liLength
   liLength = WritePrivateProfileString(StrPtr(sSection), StrPtr(sKey), _
      StrPtr(sHodnota), StrPtr(sFile))
   WriteItemUnicodeINI = liLength
End Function
```

Dialog MessageBox

Myslím, že nyní je už jasnější, proč se dnes věnuji Unicode. Příklad také dokumentuje další drobné potíže – MsgBox ve VBA si s texty Unicode neporadí a tak jsme se obrátili na API funkci MessageBoxW. Jinak, ačkoliv interně Visual Basic pracuje s tímto kódováním, okno Locals má podobně jako MsgBox problémy se zobrazením.

Následující výpis a obrázek ukazuje aplikaci Unicode řetězců na formuláři.

```
1   Private Sub UserForm_Activate()
2
3       'cesta k souboru INI
4       sPathFileINI = ThisWorkbook.Path & "" & sFileINI
5
6       'přečtení hodnot
7       sText1 = ReadItemUnicodeINI(sPathFileINI, "CZ", "CLOSE")
8       sText2 = ReadItemUnicodeINI(sPathFileINI, "RU", "CLOSE")
9       sText3 = ReadItemUnicodeINI(sPathFileINI, "FR", "CLOSE")
10
11      'přiřazení k tlačítkům
12      Me.CommandButton1.Caption = sText1
13      Me.CommandButton2.Caption = sText2
14      Me.CommandButton3.Caption = sText3
15
16  End Sub
```

Formulář s Unicode popisky

Odkazy:

INI File, Wikipedia

Karl E. Peterson Code Samples

Sešit ke stažení s příklady:

excel_ini_unicode.zip