

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.

```

1 'konstanta pro MessageBox
2 Public Const MB_OK = &H0&
3
4 Public Declare Function MessageBox Lib "user32" Alias "MessageBoxW" (ByVal hwnd _
5     As Long, ByVal lpText As Long, ByVal lpCaption As Long, ByVal wType As Long) As _
6     Long
7
8 Public Declare Function GetPrivateProfileString Lib "kernel32" Alias _
9     "GetPrivateProfileStringW" (ByVal lpApplicationName As Long, ByVal lpKeyName As _
10    Long, ByVal lpDefault As Long, ByVal lpReturnedString As Long, ByVal nSize As _
11    Long, ByVal lpFileName As Long) As Long
12
13 Public Declare Function WritePrivateProfileString Lib "kernel32" Alias _
14    "WritePrivateProfileStringW" (ByVal lpApplicationName As Long, ByVal lpKeyName _
15    As Long, ByVal lpString As Long, ByVal lpFileName As Long) As Long
16
17 'INI soubor
18 Public Const sFileINI As String = "language.ini"
19
20 Sub TestCteniZapisINI()
21
22     'cesta k souboru INI
23     sPathFileINI = ThisWorkbook.Path & "" & sFileINI
24
25     'přečtení hodnot
26     'v okně Locals se nezobrazí korektně
27     sText1 = ReadItemUnicodeINI(sPathFileINI, "CZ", "CLOSE")
28     sText2 = ReadItemUnicodeINI(sPathFileINI, "RU", "CLOSE")
29     sText3 = ReadItemUnicodeINI(sPathFileINI, "FR", "CLOSE")
30
31     'použití textu
32     Range("A1") = sText1
33     Range("A2") = sText2
34     Range("A3") = sText3
35
36     'MsgBox nezvládá Unicode
37     'použijeme API funkci MessageBox
38     MsgBox 0&, StrPtr(sText2), StrPtr("Microsoft Excel"), MB_OK
39
40     'zápis
41     Call WriteItemUnicodeINI(sPathFileINI, "SK", "CLOSE", "zatvorit")
42
43 End Sub
44
45 Public Function ReadItemUnicodeINI(ByVal sFile As String, ByVal sSection As _
46    String, ByVal sKey As String, Optional sDefaultValue As String) As String _
47    Dim lsRetVal As String
48    Dim liLength
49    Dim lsDefVal As String
50    If IsMissing(sDefaultValue) Then
51        lsDefVal = "(nenalezeno)"
52    Else
53        lsDefVal = sDefaultValue
54    End If
55    lsRetVal = VBA.Space$(1024)
56    liLength = GetPrivateProfileString(StrPtr(sSection), StrPtr(sKey), _
57        StrPtr(lsDefVal), StrPtr(lsRetVal), 1024, StrPtr(sFile))
58    lsRetVal = VBA.Left(lsRetVal, liLength)
59    ReadItemUnicodeINI = lsRetVal
60 End Function
61
62 Public Function WriteItemUnicodeINI(ByVal sFile As String, ByVal sSection As _
63    String, ByVal sKey As String, sHodnota As String) As String _
64    Dim liLength
65    liLength = WritePrivateProfileString(StrPtr(sSection), StrPtr(sKey), _
66        StrPtr(sHodnota), StrPtr(sFile))
67    WriteItemUnicodeINI = liLength
68 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](#)