

Formát CSV, jak se dočtete i na Wikipedii ([odkaz](#)) nemá oficiální specifikaci, jen určitý předpis ([odkaz](#)). Jsou aplikace, které dokáží na tomto formátu postavit datové OLAP kostky ([Jedox base](#)) a jsou takové, které vám pod tímto formátem drze předhodí luštěnky (několikařádkové hlavičky, stránkování po nestejně dlouhých blocích se sumačními řádky a čísla stránek, znakové grafické symboly aj.). Právě mám před sebou kupříkladu XLS soubor typu „local file“ vyexportovaný ze SAPu, za kterým je ve skutečnosti něco jen vzdáleně připomínajícího CSV. Ach jo, proč mi zkratka SAP přijde jako SaveAsPaskvil...

Říkáte si, co je tak těžkého na ukládání excelové tabulky do formátu CSV? Vcelku nic, Excel nabízí v dialogu volby CSV (oddělený středníkem), CSV (Macintosh) a CSV (MS-DOS). Prvně, Excel nezobrazuje možnost uložení v americkém formátu, tedy v pravém „zkratky smyslu“ CSV (Comma separated values, čárkou oddělená data), a ve světě nikoho nezajímá náš lokalizovaný formát CSV (středníkem oddělená data). Americký formát podporuje naštěstí v rámci ukládání alespoň VBA. Pojdme se podívat na kód.

```

1  Sub UlozitVyberJakoCSV(ByVal rngOblast As Range, ByVal strCestaSoubor As String, _
2  Optional ByVal boolWindowsFormat As Boolean = True, Optional ByVal _
3  boolCeskyFormat As Boolean = True)
4
5  'zákaz překreslování obrazovky
6  Application.ScreenUpdating = False
7
8  Dim wbSesitTemp As Workbook
9
10 'nový sešit
11 Set wbSesitTemp = Workbooks.Add
12
13 'kopie oblasti do nového sešitu
14 rngOblast.Copy wbSesitTemp.Worksheets(1).Cells(1)
15
16 'potlačení varovných hlášek
17 Application.DisplayAlerts = False
18
19 'uložení do souboru CSV
20 'uložení sešitu ve formátu CSV
21 'xlCSV, xlCSVMSDOS, xlCSVWindows
22 'Local:=True ... český formát CSV
23 'Local:=False ... americký formát CSV
24
25 With wbSesitTemp
26     .SaveAs strCestaSoubor, If(boolWindowsFormat, xlCSVWindows, _
27         xlCSVMSDOS), Local:=boolCeskyFormat
28     'přepis souboru bez dotazu
29     .Close SaveChanges:=False
30 End With
31
32 'povolení varovných hlášek
33 Application.DisplayAlerts = True
34
35 'vyčištění paměti
36 Set wbSesitTemp = Nothing
37
38 'povolení překreslování obrazovky
39 Application.ScreenUpdating = True
40
41 End Sub

```

Může se vám zdát divné, proč neukládám sešit přímo, ale používám nový soubor. Ukládání do CSV totiž nerespektuje vaše skryté sloupce, do výsledných dat se promítnou! A až to zjistíte, bude pravděpodobně pozdě (asi víte, proč je skrýváte, že). Proto do ukládací procedury – jak později uvidíte – posílám pouze viditelné buňky výběru. Samotný formát souboru řídí metoda SaveAs a především

parametry FileFormat a Local. No jo, jenže...

K sepsání toho článku mě přiměla potřeba převést tabulku Excelu do MySQL databáze (CSV je v tu chvíli prostředník). To je běžně možné u databázových tabulek s primárním klíčem po aplikaci výběrového (SELECT) dotazu v MySQL Workbench.



Import CSV souboru do MySQL

A v čem je jádro pudla? Běžně dnes už pracuji s databázovými tabulkami v UTF-8 a ne ANSI (Windows 1250, tj. CP1250), které generuje Excel. MySQL při importu CSV očekává americký UTF-8 a navíc bez BOM ([Byte order mark](#)). Je to volitelných pár bajtů z kraje textového souboru, které říkají, jak ho číst.

EF BB BF	UTF-8
FF FE	UTF-16, little endian
FE FF	UTF-16, big endian
FF FE 00 00	UTF-32, little endian
00 00 FE FF	UTF-32, big-endian

Jestliže tedy aplikace očekává soubor bez BOM, pak se těchto nadbytečných bajtů musíte zbavit. Na internetu převažují tipy na konvertování souboru prostřednictvím dvojího streamu (unicode a binárního) v rámci ADO pod VBA. Mně se to nepodařilo. Ručně jsem kdysi měnil kódování v Notepadu++, ale to není zrovna pohodlné, a dodnes nevím o spolehlivé aplikaci pro hromadné zpracování (zdarma). Jednou z možností je převzít [kód Martina Bertenshawa](#), využívající API funkce (v příloze je uveden). Já se vydal cestou utility [iconv.exe](#) řízené z příkazového řádku, jejíž výsledky konverze BOM neobsahují. V kódu níže se neděste množství uvozovek ve spouštěcím příkazu. Jsou nutné z důvodu uvádění cest k souborům, v nichž se vyskytují mezery.

```

1  Sub TestUlozitVyberJakoCSV()
2
3      Dim strCestaSoubor As String
4
5      'cesta a název souboru (podle listu, na kterém se nachází výběr)
6      strCestaSoubor = ThisWorkbook.Path & "" & Selection.Parent.Name & _
7          "-ANSI-CP1250.csv"
8
9      'formát souboru: Windows (ANSI, CP1250), čárkou oddělené hodnoty
10     Call UlozitVyberJakoCSV(Selection.SpecialCells(xlCellTypeVisible), _
11         strCestaSoubor, True, False)
12
13     'konverze do UTF-8 bez BOM
14     'první způsob, API, Mark Bertenshaw
15     'Call FileConvert(strCestaSoubor)
16
17     'druhý způsob
18     'http://gnuwin32.sourceforge.net/packages/libiconv.htm
19     'http://www.gnu.org/software/libiconv/
20     'cesta ... c:\Program Files (x86)\GnuWin32\bin\
21     'příkaz ... iconv -f CP1250 -t UTF-8 soubor.csv >> soubor_utf8.csv
22
23     Dim wshShell As Object
24     Dim strCestalconvExe As String
25
26     'WSH pro spuštění "commandline" aplikace
27     Set wshShell = CreateObject("WScript.Shell")
28     'cesta k iconv.exe
29     strCestalconvExe = "c:\Program Files (x86)\GnuWin32\bin"
30
31     Prikaz = "%COMSPEC% /c """" & strCestalconvExe & _
32         "iconv.exe"" -f CP1250 -t UTF-8 """" & strCestaSoubor & """" >> """" & _
33         Replace(strCestaSoubor, "ANSI-CP1250", "UTF-8-BEZ-BOM") & """"""
34
35     'běh na pozadí
36     wshShell.Run Prikaz, 0, True
37
38 End Sub

```

O čtení souborů CSV si řekneme zase něco jindy. A nebude to věřím vůbec nudné.