

Zatímco StatusBar je ovládací prvek představující stavový řádek, ProgressBar je narůstající sloupeček určený například pro zobrazení průběhu instalace. Microsoft je přitom na instalace expert. Nejprve doháněl uživatele k šílenství tím, že ProgressBar po dosažení 100 % začal opakovaně plnit zase od nuly, později tak, že z ProgressBaru udělal cosi jako policejní maják z amerických filmů, a dnes k tomu využívá textové žvásty (aktualizace se stahují, probíhá příprava na aktualizacích, procenta, nakonec vše je téměř dokončeno, a pak stejně ještě čekáte, až zahřmí).

Stavový řádek už v Excelu přeci jeden máme. Učebnicově je považován za nejlepší místo, kde zobrazit průběh delšího výpočtu, makra, atp. Základní ovládání ukazuje následující procedura.

```
1  Sub StavovyRadekObecne()  
2  
3      'je zobrazen stavový řádek?  
4      blnStavovyRadekZobrazen = Application.DisplayStatusBar  
5  
6      'zobrazení stavového řádku  
7      Application.DisplayStatusBar = True  
8  
9      'zobrazení vlastního textu ve stavovém řádku  
10     Application.StatusBar = "Probíhá výpočet..."  
11  
12     'nevhodný (nefunkční) reset stavového řádku  
13     'Application.StatusBar = ""  
14     'Application.StatusBar = vbNullString  
15  
16     'korektní reset stavového řádku  
17     Application.StatusBar = False  
18  
19 End Sub
```



Excel – stavový řádek

Pokud v cyklu VBA provádíme delší výpočty, je vhodné informovat o stavu detailněji.

```

1  Sub StavovyRadekPrubeznyStav()
2
3  Dim i As Long
4  Dim lngPocet As Long
5
6  'počet cyklů
7  lngPocet = 10000
8
9  For i = 1 To lngPocet
10
11     'prostor pro zpracování systémových událostí
12     DoEvents
13
14     'změna textu ve stavovém řádku
15     Application.StatusBar = "Probíhá zpracování záznamu č. " & i & _
16         " z celkového počtu " & lngPocet & "."
17
18     Next i
19
20     'reset stavového řádku
21     Application.StatusBar = False
22
23 End Sub

```



Excel – průběh ve stavovém řádku

Je možné umístit informaci o průběhu jinak? Teoreticky je možné s využitím API přiřadit bublinu s textem do vlastní ikony v oznamovací části systémové lišty. Ta ale není určena k tomu, abyste do ní několikrát za sekundu „prali“ text. Takže co s tím? Co zkusit měnit text v titulku okna? Titulek v aplikaci Excel sestává při maximalizovaném okně sešitu ze dvou částí – z názvu sešitu a názvu aplikace (např. Sešit1.xlsm – Microsoft Excel).



Excel – titulky v oknech

Pojďme tedy informaci o stavu úlohy umístit do titulku.

```
1 Sub TitulekAplikacePrubeznyStav()  
2  
3 Dim i As Long  
4  
5 For i = 1 To 1000  
6  
7 'prostor pro zpracování systémových událostí  
8 DoEvents  
9  
10 'změna textu ve stavovém řádku  
11 Application.Caption = "Probíhá zpracování záznamu č. " & i & _  
12 " z celkového počtu 1000."  
13  
14 Next i  
15  
16 'reset stavového řádku  
17 Application.Caption = vbNullString  
18  
19 End Sub
```



Excel - zobrazení průběhu v titulku okna aplikace

Zástupce sešitu na systémové liště přebírá titulek okna do svého popisku. Nesmíte ovšem mít nastavené seskupování zástupců (pravé tlačítko myši na liště a volba Vlastnosti).



Windows - seskupování zástupců na systémové liště

Není mi úplně jasný princip, podle kterého je v zástupci stanoveno pořadí dvojice název aplikace - aktivní sešit. Jednou je to tak, podruhé opačně. Ve Windows 8.1 se mi také stává, že občas jedno z oken sešitů o svého zástupce v systémové liště přijde a popisek v něm pak neodpovídá situaci. A do třetice, zástupce běžně pojme maximálně 15-18 znaků z celého titulku (velikost je tuším možné měnit v registrech). Přes to všechno jsem se pustil do experimentu, kdy aplikaci minimalizuji (v průběhu běžícího makra nám stejně není k užítku) a změnu titulku okna (aplikace) promítám právě do jejího zástupce.

```

1  Sub ZastupcePrubeznyStav()
2
3  Dim i As Long
4  Dim IngStavOknoSesit As Long
5  Dim IngStavOknoAplikace As Long
6
7  Dim wndOkno As Window
8
9  Set wndOkno = ActiveWindow
10
11 'uložení stavu okna sešitu
12 IngStavOknoSesit = wndOkno.WindowState
13
14 'maximalizace okna sešitu
15 wndOkno.WindowState = xlNormal
16
17 'uložení stavu okna aplikace
18 IngStavOknoAplikace = Application.WindowState
19
20 'minimalizace okna aplikace
21 Application.WindowState = xlMinimized
22
23 For i = 1 To 10000
24
25     'prostor pro zpracování systémových událostí
26     DoEvents
27
28     'změna titulku v okně sešitu (zástupci na liště)
29     Application.Caption = "č. " & i & " z 10000"
30
31 Next i
32
33 'reset titulku okna sešitu
34 wndOkno.Caption = False
35
36 'reset titulku okna aplikace
37 Application.Caption = vbNullString
38
39 'návrat k původnímu stavu okna sešitu
40 wndOkno.WindowState = IngStavOknoSesit
41
42 'návrat k původnímu stavu okna aplikace
43 Application.WindowState = IngStavOknoAplikace
44
45 End Sub

```



Excel - zobrazení

průběhu v zástupci na
liště

Vizuálně zajímavější možnost zobrazení průběhu představují právě ProgressBar. Nasimulujeme si první z nich ve stavovém řádku. Vystačíme si přitom s vhodným znakem Unicode sady.

```
1 Sub StavovyRadekProgressBar1()
2
3 Dim intPocetZnakuMax As Integer
4 Dim intPocetZnaku As Integer
5
6 Dim i As Long
7 Dim lngPocetCyklu As Long
8
9 Dim sngProcento As Single
10
11 'maximální počet znaků ve stavovém řádku (100 %)
12 intPocetZnakuMax = 50
13
14 'počet cyklů (záznamů ke zpracování, výpočtů, ...)
15 lngPocetCyklu = 100000
16
17 For i = 1 To lngPocetCyklu
18
19     'prostor pro zpracování systémových událostí
20     DoEvents
21
22     'procento plnění
23     sngProcento = CInt((99 * i) / lngPocetCyklu)
24
25     'počet znaků odpovídajících procentu plnění
26     intPocetZnaku = CInt(intPocetZnakuMax * sngProcento / 99)
27
28     'změna textu ve stavovém řádku
29     If intPocetZnaku > 1 Then
30
31         'zobrazení procenta a symbolů kostičky
32         'nespojité kostičky ... &H25FC
33         Application.StatusBar = Format(sngProcento, "00") & " % " & _
34             String(intPocetZnaku, ChrW("&H2587"))
35
36     End If
37
38 Next i
39
40 'reset stavového řádku
41 Application.StatusBar = False
42
43 End Sub
```



Excel – simulovaný ProgressBar ve stavovém řádku

Historicky starší ProgressBary používali oddělené čtverečky. Zde je tedy ještě jeden způsob.

```
1 Sub StavovyRadekProgressBar2()  
2  
3 Dim intPocetZnakuMax As Integer  
4 Dim intPocetZnaku As Integer  
5  
6 Dim i As Long  
7 Dim lngPocetCyklu As Long  
8  
9 Dim sngProcento As Single  
10 Dim strProcento As String  
11  
12 'maximální počet znaků ve stavovém řádku (100 %)  
13 intPocetZnakuMax = 50  
14  
15 'počet cyklů (záznamů ke zpracování, výpočtů, ...)  
16 lngPocetCyklu = 100000  
17  
18 'třímezerová maska  
19 strProcento = Space(3)  
20  
21 For i = 1 To lngPocetCyklu  
22  
23 'prostor pro zpracování systémových událostí  
24 DoEvents  
25  
26 'procento plnění  
27 sngProcento = CInt((100 * i) / lngPocetCyklu)  
28  
29 'dosazení procenta do masky (mezery zleva)  
30 RSet strProcento = CStr(sngProcento)  
31  
32 'počet znaků odpovídajících procentu plnění  
33 intPocetZnaku = CInt(intPocetZnakuMax * sngProcento / 100)  
34  
35 'změna textu ve stavovém řádku  
36 If intPocetZnaku > 1 Then  
37  
38 'zobrazení procenta a symbolů kostiček  
39 Application.StatusBar = strProcento & " %" & vbTab & _  
40 String(intPocetZnaku, ChrW("&H25FC")) & String(intPocetZnakuMax - _  
41 intPocetZnaku, ChrW("&H25FB"))  
42  
43 End If  
44  
45 Next i  
46  
47 'reset stavového řádku  
48 Application.StatusBar = False  
49  
50 End Sub
```



Excel – simulovaný ProgressBar ve stavovém řádku

Dalším tipem budiž buňka s aplikovaným podmíněným formátem...



Excel – simulace ProgressBaru buňkou a podmíněným formátem

```
1 Sub BunkaProgressBar()  
2  
3 Dim i As Long  
4 Dim lngPocetCyklu As Long  
5  
6 Dim sngProcento As Single  
7  
8 'počet cyklů (záznamů ke zpracování, výpočtů, ...)  
9 lngPocetCyklu = 100000  
10  
11 For i = 1 To lngPocetCyklu  
12  
13 'prostor pro zpracování systémových událostí  
14 DoEvents  
15  
16 'procento plnění  
17 sngProcento = CInt((100 * i) / lngPocetCyklu)  
18  
19 If i Mod 1000 = 0 Then  
20  
21 Range("rngProgressBar").Value = sngProcento  
22  
23 End If  
24  
25 Next i  
26  
27 End Sub
```



Excel – ProgressBar v buňce v akci

Na internetu lze nalézt návody, jak vytvořit skutečný ProgressBar třeba na místě stavového řádku nebo v Řádku vzorců a to s pomocí API. Zde se tím zabývat nebudeme. I tak jsme si ukázali řadu

vizuálních hraček. Nyní ovšem přijde pořádná herda do zad. Průběh stavu má smysl ukazovat tehdy, když je výpočet zdlouhavý (např. v cyklu zpracováváme tisíce záznamů). V takovém případě – jak jistě víte – používáme syntaxi `Application.ScreenUpdating = False`. Pokud ta ale bude plnit svou funkci, pak přeci k žádnému překreslování textu někde ve stavovém řádku dojít nemůže! Jenže ouha. I když ji v kódu aplikujeme, změnu ve stavovém řádku uvidíme (překreslování bude doprovázeno „flickeringem“, tj. problikáváním textu). Možná také víte, že v dlouhých cyklech je dobrým zvykem užít klauzuli `DoEvents`, kterou říkáme „dej taky prostor systému, ať si udělá to svoje“. Udělal jsem tedy řadu testů a nestačil se divit.

V tabulkách je zachycena průměrná doba běhu procedury v sekundách s různými obměnami podmínek (`ScreenUpdating`, `DoEvents`, bez výpisu, s výpisem do stavového řádku, titulku okna sešitu a titulku okna aplikace).



Excel – testy `ScreenUpdating` a `DoEvents`

Poznatky

`ScreenUpdating` neumí částečné zmrazení okna, a proto, pokud zasahujeme do stavového řádku, zapíná se (časy jsou prakticky stejné). Jinak je jeho funkčnost srovnatelná s API funkcí `LockWindowUpdate` (velmi pravděpodobně ji na pozadí používá).

Příkaz `DoEvents` sice zpomalí průběh procedury, ale bez něj je jakýkoliv pokus o výpis bezpředmětný (překreslování velmi brzy zamrzne).

Překreslování titulku (nemaximalizovaného) okna sešitu doprovází vždy flickering. Stejně tak k němu pokaždé dochází v rámci stavového řádku při `Application.ScreenUpdating = True`.

Když se podíváte na čas bez všech serepetiček a srovnáte ho s ostatními, asi dojdete ke stejnému závěru, že jakákoliv nutnost překreslování vede k řádově jiným časům vykonání kódu (v příloze se můžete vyzkoušet). Znamená to, že máme na výpisy průběhu zapomenout? Ve smyslu výše uvedených technik ano. Asi nejschůdnější řešení vede mimo Excel – logování průběhu do textového souboru.


```

1  Sub LogovaniPrubeznyStav()
2
3      Dim i As Long
4
5      Dim iFile As Integer
6      Dim strSoubor As String
7      Dim strObsah As String
8
9      'generované číslo
10     'pod kterým se později na soubor odkazujeme
11     iFile = FreeFile
12
13     'cesta a soubor
14     strSoubor = ThisWorkbook.Path & "\log.txt"
15
16     'otevření souboru pro přidávání záznamů na konec souboru
17     Open strSoubor For Append Access Write As iFile
18
19     For i = 1 To 10000
20
21         'prostor pro zpracování systémových událostí
22         DoEvents
23
24         'sestavení obsahu
25         strObsah = "Záznam č. " & i & " z 10000." & vbCrLf
26
27         'zápis do souboru
28         Print #iFile, strObsah;
29
30     Next i
31
32     'uzavření souboru
33     Close iFile
34
35 End Sub

```

Procedura v tomto případě zabrala cca 0,588 s. Do souboru je možné nahlížet i během jeho užívání programovým kódem.

Ovládací prvky StatusBar a ProgressBar na formuláři

Ve VBA nejsou tyto prvky běžně vidět. Je potřeba při návrhu formuláře klepnout pravým tlačítkem myši na Toolbox, zvolit Additional Controls a vybrat Microsoft StatusBar Control 6, resp. Microsoft ProgressBar Control 6. Následuje ukázka užití StatusBaru, ve kterém si zobrazíme aktuální datum a čas, text s informací o průběhu, a stav klávesy CAPS LOCK. A přirozeně nesmí chybět ani obyčejný

Label, který svou úlohu splní také s přehledem.



Excel a VBA – prvek StatusBar

```

1 'https://msdn.microsoft.com/en-us/library/aa733695(v=vs.60).aspx
2
3 Private Sub UserForm_Initialize()
4
5     With StatusBar1
6
7         'přidání čtyř sekcí do prvku StatusBar1
8         For i = 1 To 4
9             .Panels.Add
10            Next i
11
12            'datum
13            .Panels(1).Style = sbrDate
14            .Panels(1).Width = 50
15            .Panels(1).Bevel = sbrNoBevel
16
17            'čas
18            .Panels(2).Style = sbrTime
19            .Panels(2).Width = 25
20            .Panels(2).Bevel = sbrNoBevel
21
22            'příprava pro text
23            .Panels(3).Style = sbrText
24            .Panels(3).Alignment = sbrCenter
25            .Panels(3).Text = ""
26            .Panels(3).Width = 100
27            .Panels(3).Bevel = sbrNoBevel
28
29            'stav klávesy CAPS
30            .Panels(4).Style = sbrCaps
31            .Panels(4).AutoSize = sbrSpring
32            .Panels(4).Bevel = sbrNoBevel
33
34        End With
35
36    End Sub
37
38 Private Sub UserForm_Activate()
39
40     Dim i As Long
41     Dim lngPocetCyklu As Long
42
43     Dim strText As String
44
45     'počet cyklů (záznamů ke zpracování, výpočtů, ...)
46     lngPocetCyklu = 10000
47
48     For i = 1 To lngPocetCyklu
49
50         'text k zobrazení
51         strText = "Záznam č. " & i & " z " & lngPocetCyklu
52
53         'naplnění prvků
54         StatusBar1.Panels(3).Text = strText
55         Label1.Caption = strText
56
57         'překreslení formuláře
58         Me.Repaint
59
60     Next i
61
62 End Sub

```



Excel a VBA – prvek StatusBar v akci

Během prvních testů docházelo k flickeringu u obou prvků. Zatímco Labelu pomohlo navýšení hodnoty ve vlastnosti DrawBuffer formuláře z 32000 na 64000 až 128000, u StatusBaru jsem se jevu nezbavil. Z blíže neznámých příčin překreslování mělo tendenci i vytuhnout. StatusBar tedy není ideální prvek pro častou obměnu textu.

Pozn. Ve Windows 8.1 se mi nepodařilo v okně Properties využít položku Custom, kde bychom v dialogu nastavili vše, co je uvedeno v události Initialize formuláře výše.

Na druhém formuláři nasadíme do akce jak prvek ProgressBar, tak obyčejný prvek Label, u něhož nastavíme barvu pozadí, text skryjeme, a změníme programově pouze šířku.



Excel a VBA – prvek ProgressBar

```

1  Private intLabel2Sirka As Integer
2
3  Private Sub UserForm_Activate()
4
5      Dim i As Long
6      Dim lngPocetCyklu As Long
7
8      Dim sngProcento As Single
9
10     'počet cyklů (záznamů ke zpracování, výpočtů, ...)
11     lngPocetCyklu = 100000
12
13     For i = 1 To lngPocetCyklu
14
15         'prostor pro zpracování systémových událostí
16         DoEvents
17
18         'procento plnění
19         sngProcento = CInt((100 * i) / lngPocetCyklu)
20
21         'hodnota do popisku a prvku ProgressBar1
22         Label1.Caption = sngProcento & " %"
23         ProgressBar1.Value = sngProcento
24
25         'šířka prvku Label2
26         Label2.Width = sngProcento * intLabel2Sirka / 100
27
28     Next i
29
30 End Sub
31
32 Private Sub UserForm_Initialize()
33
34     'uložení šířky pro prvek Label2
35     intLabel2Sirka = Label2.Width
36
37     'nastvení nulové šířky
38     Label2.Width = 0
39
40 End Sub

```



Excel a VBA – prvek ProgressBar a Label v akci

Příloha:

[excel-statusbar-progressbar.zip](#)