

Jedním z věčných témat je potřeba odesílání informací přímo z Excelu prostřednictvím e-mailu. Opětovně jej uvádím na svých stránkách i já. Ukážeme si, jak poslat část tabulky, sešit jako přílohu i e-mail úplně nezávislý na Microsoft Office. Nástrojem nám budiž VBA.

Metoda FollowHyperlink

Jmenovaná metoda využívá výchozího poštovního klienta. Programově umí naplnit dialog nové zprávy. SendKeys pak může simulovat stisk klávesy Odeslat (Alt+A, dříve Alt+S). Její užití není podmíněno posláním sešitu jako celku v příloze.

```
1 Sub ExcelFollowHyperlink()
2
3 Dim rngOblast As Range
4 Dim rngBunka As Range
5 Dim strAdresat As String
6 Dim strPredmet As String
7 Dim strObsah As String
8 Dim strRet As String
9
10 'náhrada vblf
11 Const cstrLf As String = "%0A"
12
13 'adresát
14 strAdresat = "nekdo@nekde.cz"
15
16 'předmět
17 strPredmet = "Výpis z listu"
18
19 'zdroj obsahu
20 Set rngOblast = Range("rngObsah")
21
22 'hlavička obsahu
23 strObsah = rngOblast.Parent.Name & cstrLf
24
25 'načtení adres a obsahů jednotlivých buněk oblasti
26 For Each rngBunka In rngOblast
27     strObsah = strObsah & cstrLf & rngBunka.Address(0, 0) & ": " & _
28         rngBunka.Text
29 Next rngBunka
30
31 'sestavení řetězce pro metodu FollowHyperlink
32 strRet = "mailto:" & strAdresat & "?"
33 'předmět
34 strRet = strRet & "subject=" & strPredmet & "&"
35 'obsah
36 strRet = strRet & "body=" & strObsah
37
38 'odeslání e-mailu
39 ActiveWorkbook.FollowHyperlink (strRet)
40
41 'simulované potvrzení dialogu (Odeslat, ALT+A)
42 'Microsoft Outlook 2010 CZ
43 Application.Wait (Now + TimeValue("0:00:05"))
44 SendKeys "%a", True
45
46 End Sub
```



Metoda FollowHyperlink

Metoda SendMail

Tato metoda patří asi k nejnámějším, ale také činí největší potíže. Sešit je v ní posílán jako příloha a veškerá činnost podléhá vcelku přísným bezpečnostním opatřením, díky čemuž nelze úlohu plně zautomatizovat.

1	Sub ExcelSendMail()
2	
3	'aktivní sešit jako příloha
4	
5	Dim aKomu()
6	
7	'adresáti
8	aKomu = Array("nekdo@nekde.cz", "info@firma.org")
9	
10	'odeslání s uvedením předmětu zprávy
11	ActiveWorkbook.SendMail aKomu, "Výpis listu"
12	
13	End Sub

První z níže uvedených obrázků ukazuje systémový dialog při odesílání zprávy přes Microsoft Outlook. V průběhu let se měnil, tlačítka přišla o klávesovou zkratku, tlačítko Povolit není výchozí a navíc je zpřístupněno po uplynutí několika sekund. Řadu let se programátoři snaží tento dialog obejít. Pokud vím, ze strany Microsoftu je cesta hodně trnitá a svého času byla podmíněna používáním Microsoft Exchange. Druhý obrázek ukazuje výsledek klepnutí na tlačítko Odepřít či uzavření dialogu – chybovou zprávu.



Metoda SendMail – zabezpečení



Metoda SendMail

Dialog SendMail

Následující příklad využívá vestavěného dialogu pro odesílání pošty. Bohužel, v tomto případě se mi

nepodařilo zprovoznit automatické potvrzení dialogu přes SendKeys. Nezkoušel jsem cestu odeslání klávesové zkratky přes API.

```
1 Sub ExcelDialogSendMail()  
2  
3 'aktivní sešit jako příloha  
4  
5 Dim aKomu()  
6  
7 aKomu = Array("nekdo@nekde.cz", "info@firma.org")  
8  
9 'simulované potvrzení dialogu (Odeslat, ALT+A)  
10 'Microsoft Outlook 2010 CZ  
11 'neproběhne  
12 SendKeys "%a"  
13  
14 'předvyplnění a zobrazení okna se zprávou  
15 Application.Dialogs(xlDialogSendMail).Show aKomu, "Výpis listu"  
16  
17 End Sub
```



Dialog SendMail

API funkce

Při posílání e-mailu můžete sáhnout i po API funkci, konkrétně ShellExecute (popravdě nejsem si úplně jistý, proč v poznámkách nemám uveden příklad na VBA funkci Shell, ale pravděpodobně jsem narazil na nějaký problém při jejím užití).

```

1 Private Const SW_SHOWNORMAL As Long = 1
2
3 Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteW" _
4     (ByVal hWnd As Long, ByVal lpOperation As Long, ByVal lpFile As Long, ByVal _
5     lpParameters As Long, ByVal lpDirectory As Long, ByVal nShowCmd As Long) As Long
6
7 Sub ExcelAPI()
8
9     Dim strObsah As String
10    Dim strURL As String
11    Dim strAdresat As String
12    Dim strPredmet As String
13    Dim strAdresatCC As String
14    Dim strAdresatBCC As String
15    Dim rngOblast As Range
16    Dim rngBunka As Range
17
18    'náhrada vbLf
19    Const cstrLf As String = "%0A"
20
21    'adresát
22    strAdresat = "nekdo@nekde.cz"
23
24    'kopie
25    strAdresatCC = "schranka@email.com"
26
27    'skrytá kopie
28    strAdresatBCC = "info@firma.org"
29
30    'zdroj pro obsah zprávy
31    Set rngOblast = Range("rngObsah")
32
33    'předmět
34    strPredmet = "Výpis listu"
35
36    'zpracování obsahu
37    strObsah = rngOblast.Parent.Name & cstrLf
38
39    For Each rngBunka In rngOblast
40        strObsah = strObsah & cstrLf & rngBunka.Address(0, 0) & ": " & vbTab & _
41            rngBunka.Text
42    Next rngBunka
43
44    'sestavení řetězce pro funkci ShellExecute
45    strURL = "mailto:" & strAdresat & "?cc=" & strAdresatCC & "&bcc=" & _
46        strAdresatBCC & "&subject=" & strPredmet & "&body=" & strObsah
47
48    'nasazení API funkce
49    ShellExecute 0&, 0&, StrPtr(strURL), 0&, 0&, SW_SHOWNORMAL
50
51    'simulované potvrzení dialogu (Odeslat, ALT+A)
52    'Microsoft Outlook 2010 CZ
53    Application.Wait (Now + TimeValue("0:00:05"))
54    SendKeys "%a", True
55
56 End Sub

```



API funkce ShellExecute

Pozn. V původní verzi tohoto článku byla užita ANSI verze funkce ShellExecute, nyní již pracujeme s Unicode verzí (viz alias ShellExecuteW v deklaraci, parametry Long, StrPtr a správný obsah buňky B5 v těle e-mailu)

Panel Obálka (Envelope)

Dialog nové zprávy umí Excel zobrazovat i v rámci svého hlavního okna. Jedná se o panel reprezentující jakousi hlavičku formuláře. Kromě toho je také důkazem, že starší panely nástrojů lze zobrazovat v prostředí Excelu 2007 a novějším.

1	Sub ExcelPanelObalka()
2	
3	'aktivní sešit jako příloha
4	
5	'odesílaný z podokna Excelu
6	ActiveWorkbook.EnvelopeVisible = True
7	
8	End Sub
9	
10	Sub ExcelZavritPanelObalka()
11	
12	ActiveWorkbook.EnvelopeVisible = False
13	
14	End Sub



Panel Obálka (Envelope)

Pozn. Teoreticky je k dispozici přístup k panelu přes CommandBars(„Envelope“). Tento postup je ale nespolehlivý.

Objektový model Microsoft Outlook

Komfortní práci s odesíláním pošty zajistí pochopitelně přímé napojení na objektový model Outlooku. První příklad ukazuje obecný postup posílání elektronické zprávy včetně příloh.

```

1  Sub ExcelOutlookPriloha()
2
3  'Tools / References / Microsoft Outlook x.x Object Library
4
5  Dim OutApp As Outlook.Application
6  Dim OutMail As Outlook.MailItem
7
8  Set OutApp = CreateObject("Outlook.Application")
9  Set OutMail = OutApp.CreateItem(olMailItem)
10
11 With OutMail
12
13     'adresát
14     .To = "nekdo@nekde.cz"
15
16     'kopie pro
17     .CC = "schranka@email.com"
18
19     'skrytá kopie pro
20     .BCC = "info@firma.org"
21
22     'předmět zprávy
23     .Subject = "Předmět zprávy"
24
25     'text zprávy
26     .Body = "1. řádek zprávy" & Chr(13) & "2. druhý řádek zprávy"
27
28     'aktivní (uložený) sešit jako příloha
29     .Attachments.Add ActiveWorkbook.FullName
30
31     'jiná příloha
32     .Attachments.Add ActiveWorkbook.Path & "\soubor.txt"
33
34     'zobrazení okna se zprávou (není nutné)
35     .Display
36
37     'odeslání zprávy
38     '.Send
39
40 End With
41
42     'uvolnění z paměti
43     Set OutMail = Nothing
44     Set OutApp = Nothing
45
46 End Sub

```



Microsoft Outlook – příloha

Jak jistě víte, v e-mailu se může objevit i obsah v HTML formátu. Tuto možnost využívá následující procedura, která odesílá aktivní list přímo v těle zprávy.

```
1 Sub ExcelOutlookHTML()  
2  
3 'Tools / References / Microsoft Outlook x.x Object Library  
4  
5 Dim OutApp As Outlook.Application  
6 Dim OutMail As Outlook.MailItem  
7  
8 Dim strCestaSoubor As String  
9 Dim strObsahHTML As String  
10  
11 Set OutApp = CreateObject("Outlook.Application")  
12 Set OutMail = OutApp.CreateItem(olMailItem)  
13  
14 'uložení listu do HTML podoby  
15 strCestaSoubor = ActiveWorkbook.Path & "\temp.htm"  
16 ActiveWorkbook.PublishObjects.Add(xlSourceSheet, _  
17 strCestaSoubor, ActiveSheet.Name).Publish (True)  
18  
19 'načtení HTML kódu uloženého listu  
20 Set fso = CreateObject("Scripting.FileSystemObject")  
21 Set txt = fso.GetFile(strCestaSoubor).OpenAsTextStream(1, -2)  
22 strObsahHTML = txt.ReadAll  
23 txt.Close  
24  
25 With OutMail  
26  
27 'adresát  
28 .To = "nekdo@nekde.cz"  
29  
30 'kopie pro  
31 .CC = "schranka@email.com"  
32  
33 'skrytá kopie pro  
34 .BCC = "info@firma.org"  
35  
36 'předmět zprávy  
37 .Subject = "Předmět zprávy"  
38  
39 'HTML obsah zprávy  
40 .HTMLBody = strObsahHTML  
41  
42 'zobrazení okna se zprávou (není nutné)  
43 .Display  
44  
45 'odeslání zprávy  
46 .Send  
47  
48 End With  
49  
50 'uvolnění z paměti  
51 Set OutMail = Nothing  
52 Set OutApp = Nothing  
53  
54 End Sub
```



Microsoft Outlook – HTML obsah

Objektový model Outlooku je pochopitelně možné využít v daleko větším měřítku – práce s kontakty, složkami, kalendářem atd. Makra směřující k událostem Outlooku (nová příchozí zpráva, navázání pravidel, ...) je už zpravidla nutné směřovat přímo do Outlooku, kde si můžete také vytvořit formuláře coby šablony zpráv.

CDO

Ve Windows již dlouho existuje jedna cesta, jak odeslat tichý e-mail a dokonce s přílohou bez vazby na poštovního klienta. Využijeme přitom systémovou knihovnu cdosys.dll (CDO je zkratkou Collaboration Data Objects). CDO je řešením pro klientské aplikace, které v určitém bodu pracovního procesu odešlou informaci zaměstnanci, jenž má v procesu pokračovat. Může se jednat kupříkladu o proces schvalování. Bezpodmínečně nutný je SMTP server a existující poštovní účet. Dříve šlo blafovat ve vlastnosti .From, v níž se mohl objevit jiný odesílatel. Kupříkladu Seznam.cz toto již přímo zakazuje a Gmail.com ignoruje.


```

1  Sub ExcelCDO()
2
3  Dim iMsg As Object
4  Dim iConf As Object
5  Dim strBody As String
6  Dim Flds As Object
7
8  'Windows 2000 a novější
9
10 'objekty CDO
11 Set iMsg = CreateObject("CDO.Message")
12 Set iConf = CreateObject("CDO.Configuration")
13
14 'nastavení konfigurace
15 iConf.Load -1
16 Set Flds = iConf.Fields
17
18 strConf = "http://schemas.microsoft.com/cdo/configuration/"
19
20 'příklad pro Seznam.cz
21 With Flds
22
23     .Item(strConf & "sendusing") = 2
24
25     'SMTP server
26     .Item(strConf & "smtpserver") = "smtp.seznam.cz"
27
28     'port
29     .Item(strConf & "smtpserverport") = 25
30
31     .Item(strConf & "smtpauthenticate") = 1
32
33     'pro e-mail ucet@seznam.cz
34     .Item(strConf & "sendusername") = "ucet"
35     .Item(strConf & "sendpassword") = "heslo"
36
37     .Update
38
39 End With
40
41 'text v těle zprávy
42 strBody = "1. řádek zprávy" & Chr(13) & Chr(10) & "2. druhý řádek zprávy"
43
44 With iMsg
45
46     'konfigurace
47     Set .configuration = iConf
48
49     'adresát
50     .To = "nekdo@nekde.cz"
51
52     'kopie
53     .CC = ""
54
55     'skrytá kopie
56     .BCC = ""
57
58     'odesílatel
59     .From = "ucet@seznam.cz"
60
61     'předmět
62     .Subject = "Text v předmětu zprávy"
63
64     'HTML obsah zprávy
65     .HTMLBody = ...
66
67     'HTML stránka na internetu
68     .CreateMHTMLBody "http://www.excelplus.net/data/cnb-denni-kurz.php"
69
70     'lokální HTML soubor
71     .CreateMHTMLBody "file:///C:/test.htm"
72     'textový obsah zprávy
73     .TextBody = strBody
74
75     'příloha (mezeru v názvu nahraďte "%20")
76     .AddAttachment ActiveWorkbook.Path & "\soubor.txt"
77
78     'odeslání
79     .Send
80
81 End With
82
83 'odstranění spojení
84 Set iMsg = Nothing
85 Set iConf = Nothing
86
87 End Sub

```



CDO – příchozí e-mail

Uvědomte si prosím, že heslo uvedené ve VBA není nijak chráněno a heslo projektu je snadno prolomitelné.

Pro Gmail.com je nastavení následující:

1	'příklad pro Gmail.com
2	With Flds
3	
4	.Item(strConf & "sendusing") = 2
5	.Item(strConf & "smtpserver") = "smtp.gmail.com"
6	.Item(strConf & "smtpserverport") = 25 '465, 587
7	.Item(strConf & "smtpauthenticate") = 1
8	.Item(strConf & "smtpusessl") = 1
9	.Item(strConf & "smtpconnectiontimeout") = 60
10	
11	'pro e-mail ucet@gmail.com
12	.Item(strConf & "sendusername") = "ucet@gmail.com"
13	.Item(strConf & "sendpassword") = "heslo"
14	
15	.Update
16	
17	End With

Pozn. Gmail ve výchozím stavu odesílání z nedůvěryhodných aplikací nedovoluje. Naopak, na daný účet dorazí varování o využití schránky (spolu s návodem, jak lze nastavení změnit).



Gmail – změna nastavení

Častokrát jsem v nejen v rámci CDO (viz užití CreateMHTMLBody) musel diskutovat otázku špatně zobrazeného HTML obsahu s kódováním UTF-8. Pravdou je, že ať už jsem použil SMTP Seznamu nebo Googlu, tak na jejich straně je kódování v pořádku. Nicméně bez pardonů pitomý Outlook má problém s jeho zobrazením. Zatímco u odchozí pošty si můžete pohrát s nastavením, pro příchozí maily nejspíš neexistuje způsob, jak ho UTF-8 naučit (snad by to zvládl Exchange). Přitom pokud si obsah otevřete v Internet Exploreru (na kterém je podle všeho závislý), dopadne vše dobře. Tády-dády-dá.

Příloha

[excel_mail.zip](#)