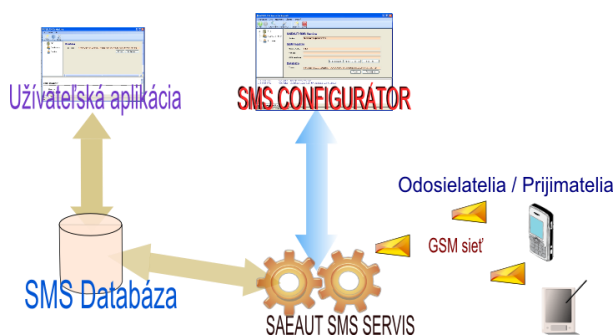


XlsSmsSender

Príklad použitia SAEAUT SMS Service s užívateľským rozhraním v xls - súbore pre posielanie a prijímanie SMS.

Hlavným zámerom pri návrhu produktu SAEAUT SMS Service bolo **vytvorenie takej aplikácie, ktorú by mohli jednoducho využívať iné klientské aplikácie (v počítačovej sieti alebo cez Internet) pre posielanie a prijímanie SMS.** Existuje viacero možností pre splnenie tohto zámeru - napr. vytvorenie dynamicky linkovanej knižnice. Takéto riešenie SAE - Automation, s.r.o. ponúka tiež.



Obr. 1. Spolupráca užívateľskej aplikácie s SMS softvérom s využitím zdieľanej databázy

Snažili sme sa však vytvoriť ešte jednoduchšie riešenie (Obr. 1) - umožniť spoluprácu ľubovoľnej aplikácie (vytvorenej pomocou skriptu alebo programovacieho jazyka) prostredníctvom zdieľanej databázy alebo databázového súboru.



Obr. 2. Klientské aplikácie pre SAEAUT SMS Service

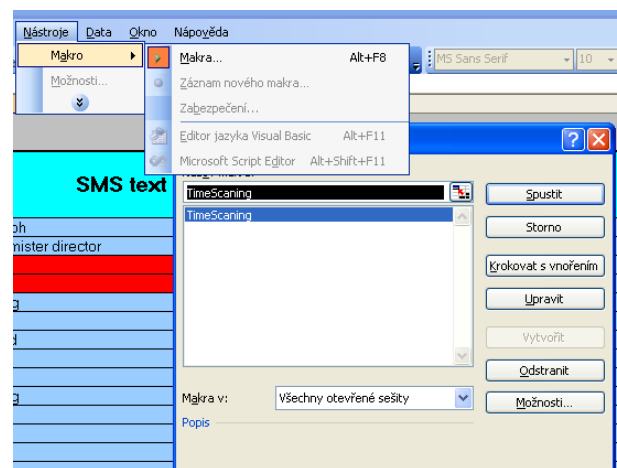
Ponúkame (1) vlastnú klientsku „desktopovú“ aplikáciu umožňujúcu využívať SAEAUT SMS

Service z ľubovoľného počítača v sieti, (2) webovú aplikáciu umožňujúcu jeho využitie z Internetu, (3) klienta zabezpečujúceho prepojenie takmer ľubovoľnej databázy s databázou SAEAUT SMS Service, (4) našu aplikáciu **OpcDbGateway** ktorá dokáže využívať databázu aplikácie ako vlastnú procesnú databázu, (5) konfigurátor SAEAUT SMS Service samotný predstavuje tiež klientsku aplikáciu.

Predpokladáme, že SAEAUT SMS Service využijú aj pokročilí užívatelia kancelárskych aplikácií (ako je napr. MS Excel™), ktorí majú záujem implementovať posielanie a prijímanie SMS podľa vlastných predstáv priamo v nich.

Pre túto kategóriu užívateľov sme pripravili príklad aplikácie implementovanej pomocou VBA (Visual Basic for Applications). Súbor s touto aplikáciou [XlsSMSList.xls](http://www.saeautom.sk/download/XlsSMSList.xls) si môžete stiahnuť na www.saeautom.sk/download/XlsSMSList.xls. Experimentovanie so vzhľadom a funkcionalitou aplikácie je pomerne jednoduché aj pre tých, ktorí sa programovaním nezaoberajú profesionálne.

Po otvorení súboru je potrebné povoliť vykonávanie makier a spustiť makro *TimeScaning* podľa Obr.3.



Obr. 3. Štart makra TimeScaning

Po jeho spustení budete vyzvaní k zadaniu cesty na databázový súbor aplikácie SAEAUT SMS SERVICE, ktorý (ak ste jeho umiestnenie nezmenili po inštalácii) sa nachádza v adresári *c:\Program Files\SAE - Automation, s.r.o\SAEAUT*

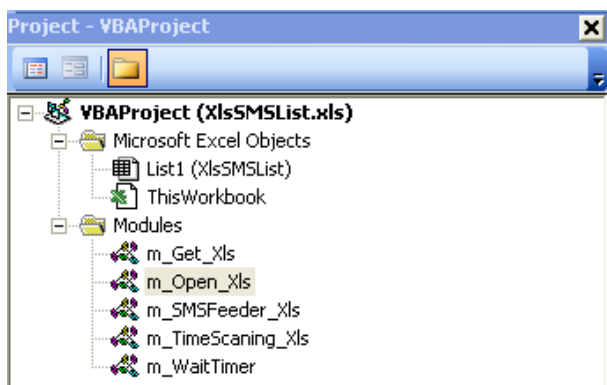
SMS Service\Datasek1. Teraz už môžete začať zapisovať texty SMS správ do stĺpca s názvom SMS Text (Obr.4) a čísla na ktoré chcete SMS poslať do stĺpca Tel. number. Po zapísaní telefónneho čísla a SMS správy program ešte očakáva vloženie príznaku potvrdenia „*“ v stĺpci Confirm Sign. Program vyhodnocuje správnosť vyplnenia údajov zoznamu a hlási prípadné chyby v stĺpci Result.

Tel. number	SMS text	Confirm Sign	Result
+421915749096	Video o SAEAUT SMS service: www.saeautom.sk/sk/products/smservice/download.htm	*	OK
+421915749096	Číslo 410-ELER	*	OK
+421915	SAE-ROBOT, nečím potvrdiť 2006, 10:00	*	RS: Tel is too short CS is empty!
+421915		*	RS: Tel is too short CS is empty!

Obr. 4.XLS-tabuľka pre odosielanie SMS

Ak chceme už raz poslanú SMS správu poslať znovu, potom stačí zmazať znak potvrdenia „*“ a opätovne ho vložiť do stĺpca Confirm Sign. Aplikácia umožňuje aj vloženie celého zoznamu SMS správ prostredníctvom Windows™-schránky z rôznych iných textových súborov.

Naším zámerom bolo predovšetkým poskytnúť námet pre tvorbu podobných aplikácií a preto si pozrite prosíme popis implementácie aj zdrojového kódu VBA makra a upravte si ich podľa vlastných predstáv a potrieb. Prehľad použitých VBA modulov je na obr. 5.



Obr. 5. Prehľad použitých modulov

Hlavný cyklus aplikácie umiestnený v module **m_TimeScanning_Xls** iniciuje periodické zisťovanie nových položiek v tabuľke podľa obr. 4.

m_Open_Xls je modul ktorý otvára predefinovaný excelovsky formulár pred spustením hlavného cyklu.

m_Get_Xls je modul využívajúci funkcionality OLE Automation nevyhnutnú prenos údajov medzi prostrediami MS Excel a MS Access. Zabezpečuje identifikáciu prírastku nových SMS správ v xls formulári, vstupnú kontrolu správnosti ich formátu a transformáciu prírastku nových SMS správ do objektu dvojrozmerného poľa, ktorý je návratovou hodnotou hlavnej funkcie **Get_Xls** tohto modulu.

Úlohou modulu **m_SMSFeder_Xls** je potom vloženie objektu dvojrozmerného poľa s nových SMS správ do výstupnej tabuľky SimpleSMSList a správne ho naformátovať.

Modul **Wait_Timer** obsahuje funkciu zabezpečujúcu spustenie príkazov v presne stanovenom čase.

Výpis kompletného programu nájdete na nasledujúcich stránkach.

```

Public strXls As String
Public oldRec As Long
Public allRec As Long
Public newRec As Long

Public objSht As Excel.Worksheet
Public MDBpath As String

Sub TimeScaning()

Dim telProd As String
Dim outSms As String
Dim RetRec As Variant
Dim IsChosen As Boolean

    IsChosen = IsChosenFile()
    If Not IsChosen Then
        MsgBox "Mdb file was not found"
        & Chr(13) & Chr(10) & "Macro will be stopped"
        Exit Sub
    End If

strXls = "Provider='Microsoft.Jet.OLEDB.4.0';" _
& "Data Source='" & MDBpath & "';"

allRec = 100
oldRec = 1
newRec = 0

OpenXls

'Main cycle repeating by period of every 15 second
Do

RetRec = GetXls
If newRec > 0 Then
    FeedXlsSMS RetRec
End If
wait 15

Loop

End Sub

Function IsChosenFile() As Boolean

MDBpath = Application.GetOpenFilename("MS access databse (*.mdb),*.mdb")

If MDBpath <> "False" Then
    IsChosenFile = True
Else
    IsChosenFile = False
End If

End Function

```

Tab 1: Modul m_TimeScaning_Xls

```

Public Function OpenXls()
On Error GoTo ErrorHandler

Set objSht = Excel.ActiveWorkbook.ActiveSheet
Exit Function

ErrorHandler:

    If Err <> 0 Then
        MsgBox Err.Source & "-->" & Err.Description, , "Error"
    End If
End Function

```

Tab 2: Modul m_Open_Xls

```

Public Function GetXls() As Variant
    On Error GoTo ErrorHandler

    'Create an array with 3 columns and 100 rows

    Dim TmpArr(1 To 100, 1 To 4) As Variant
    Dim DatArr(1 To 100, 1 To 4) As Variant

    Dim r, i, StartPointX, StartPointY As Integer
    Dim buff As String
    Dim col1, col2, col3, col4 As Boolean
    Dim ActiveRow As Long

    ActiveRow = RowNumber

    i = 0

    'address of start point for the xls data table

    StartPointX = 1
    StartPointY = 3

    For r = 1 To ActiveRow
        col1 = True
        col2 = True
        col3 = True
        col4 = True
        buff = ""
        TmpArr(r, 1) = objSht.Cells(r + StartPointY, 1 + StartPointX).Value
        TmpArr(r, 2) = objSht.Cells(r + StartPointY, 2 + StartPointX).Value
        TmpArr(r, 3) = objSht.Cells(r + StartPointY, 3 + StartPointX).Value
        TmpArr(r, 4) = objSht.Cells(r + StartPointY, 4 + StartPointX).Value

    'End of table
    If TmpArr(r, 1) = "" And TmpArr(r, 2) = "" And TmpArr(r, 3) = "" Then
        Exit For
    End If

    'Is format right?

```

```

'col1*****
If TmpArr(r, 1) = "" Then
    buff = buff & "B" & r + StartPointY & " Is empty! "
    col1 = False
End If
If Val(TmpArr(r, 1)) = 0 Then
    buff = buff & "B" & r + StartPointY & " Is not digit! "
    col1 = False
End If
If Len(TmpArr(r, 1)) <= 9 Then
    buff = buff & "B" & r + StartPointY & " Tel.is too short! "
    col1 = False
End If
If Len(TmpArr(r, 1)) >= 14 Then
    buff = buff & "B" & r + StartPointY & " Tel.is too long! "
    col1 = False
End If

'Finsal evaluation for col1
If Not col1 Then
    objSht.Cells(r + StartPointY, 1 + StartPointX).Interior.ColorIndex = 3
Else
    objSht.Cells(r + StartPointY, 1 + StartPointX).Interior.ColorIndex = 37
End If

'col2*****
If TmpArr(r, 2) = "" Then
    buff = buff & "C" & r + StartPointY & " Is empty! "
    col2 = False
End If

'Finsal evaluation for col2
If Not col2 Then
    objSht.Cells(r + StartPointY, 2 + StartPointX).Interior.ColorIndex = 3
Else
    objSht.Cells(r + StartPointY, 2 + StartPointX).Interior.ColorIndex = 37
End If

'col3*****
If TmpArr(r, 3) <> "*" Then
    buff = buff & "D" & r + StartPointY & " Confirm sign is not right! "
    col3 = False
End If

'Finsal evaluation for col3
If Not col3 Then
    objSht.Cells(r + StartPointY, 3 + StartPointX).Interior.ColorIndex = 3
Else
    objSht.Cells(r + StartPointY, 3 + StartPointX).Interior.ColorIndex = 37
End If

'sum col1 col2 col3*****

If col1 And col2 And col3 Then
    objSht.Cells(r + StartPointY, 4 + StartPointX).Value = "ok"
    objSht.Cells(r + StartPointY, 4 + StartPointX).Font.ColorIndex = 10
Else
    objSht.Cells(r + StartPointY, 4 + StartPointX).Value = buff

```

```

objSht.Cells(r + StartPointY, 4 + StartPointX).Font.ColorIndex = 9
End If

'*****

'Inset SMS record to the final output table
If col1 And col2 And col3 And TmpArr(r, 4) <> "ok" Then

    i = i + 1

    DatArr(i, 1) = TmpArr(r, 1)
    DatArr(i, 2) = TmpArr(r, 2)
    DatArr(i, 3) = TmpArr(r, 3)
    DatArr(i, 4) = TmpArr(r, 4)

End If

Next

GetXls = DatArr
newRec = i

Exit Function

ErrorHandler:

If Err <> 0 Then
    MsgBox Err.Source & "-->" & Err.Description, , "Error"
End If
End Function

Function RowNumber() As Long 'Compute the number of row
    RowNumber = ActiveSheet.Cells.Find(What:="*", _
        SearchDirection:=xlPrevious, _
        SearchOrder:=xlByRows).Row
End Function

```

Tab 3: Modul m_Get_Xls

```

'Declaring of function important for the getting of Computer name

Private Declare Function GetComputerNameXls Lib "kernel32" _
    Alias "GetComputerNameA" _
    (ByVal lpBuffer As String, nSize As Long) As Long

'Procedure for exact formating and inserting a new SMS to the SimpleSMSList
table
'SimpleSMSList table servs as a outputing gate for the outgoing SMS
Sub FeedXlsSMS(ByVal xlsArr As Variant)

    On Error GoTo ErrorHandler
    'recordset and connection variables
    Dim rstFeedSms As ADODB.Recordset
    Dim Cnxn As ADODB.Connection
    Dim strSQLFeedSms As String

    'data and time format variables
    Dim time As Date
    Dim datTim As Date

    datTim = DateValue(Now) & " " & TimeValue(Now)

```

```

'open connection
Set Cnxn = New ADODB.Connection
Cnxn.Open strXls

'open recordset server-side for indexing
Set rstFeedSms = New ADODB.Recordset
rstFeedSms.CursorLocation = adUseServer
strSQLFeedSms = "SimpleSMSList"

rstFeedSms.Open strSQLFeedSms, strXls, adOpenKeyset, _
adLockOptimistic, adCmdTableDirect

'inserting of the formatted strings
'to the specific fields of the table SimpleSMSList

For r = 1 To newRec
    rstFeedSms.AddNew
    rstFeedSms!usComputerName = ReturnComputerNameXls           '"SAE18"
    rstFeedSms!usSenderPhone = formatTel(xlsArr(r, 1))         'telSum
    rstFeedSms!usSMSText = xlsArr(r, 2)                       'smsText
    rstFeedSms!usTimeStamp = datTim
    rstFeedSms.Update
    blnRecordAdded = True
Next r

'clean up
rstFeedSms.Close
Cnxn.Close
Set rstFeedSms = Nothing
Set Cnxn = Nothing
Exit Sub

ErrorHandler:
' clean up
If Not rstFeedSms Is Nothing Then
    If rstFeedSms.State = adStateOpen Then rstFeedSms.Close
End If
Set rstFeedSms = Nothing

If Not Cnxn Is Nothing Then
    If Cnxn.State = adStateOpen Then Cnxn.Close
End If
Set Cnxn = Nothing

If Err <> 0 Then
    MsgBox Err.Source & "-->" & Err.Description, , "Error"
End If
End Sub

'Functionality important for the getting of Computer name
Function ReturnComputerNameXls() As String
Dim rString As String * 255, sLen As Long, tString As String
tString = ""
On Error Resume Next
sLen = GetComputerNameXls(rString, 255)

```

```

sLen = InStr(1, rString, Chr(0))
If sLen > 0 Then
    tString = Left(rString, sLen - 1)
Else
    tString = rString
End If
On Error GoTo 0
ReturnComputerNameXls = UCase(Trim(tString))
End Function

Function formatTel(tel)
    'preformatting of inputing tel. +444222111111 number
    'to the standard world format +444(222)111111

    lenTelPreselection = Len(tel) - 9

    tel1 = Left(tel, lenTelPreselection)
    tel2 = Left(Right(tel, 9), 3)
    tel3 = Right(tel, 6)

    formatTel = tel1 & "(" & tel2 & ")" & tel3
End Function

```

Tab 4: Modul m_SMSFeeder_Xls