

**Príklad aplikácie využívajúcej SAEAUT SMS Service.
Informáciu požadovanú prostredníctvom SMS vyhledá
v databáze a zašle odpoveď v SMS.**

V SMS správe o dĺžke 160 znakov je možné odovzdať množstvo užitočných informácií. Napríklad, ak **zákazník potrebuje získať informáciu o cene a množstve položky tovaru č. 2577 na sklade** môže poslať SMS. v nasledovnom tvare:

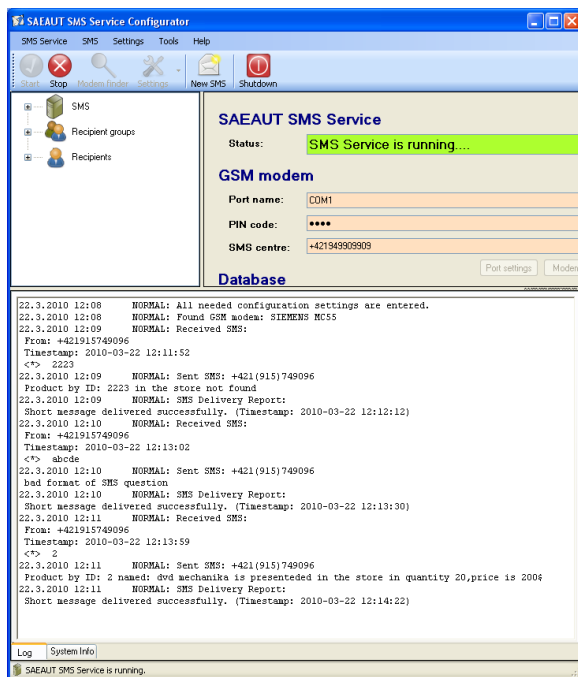
<*>2577

kde „<*>“ je označenie typu SMS.

Softvérová aplikácia nájde požadovanú informáciu v databáze, vloží ju do SMS a zašle zákazníkovi napr. v nasledovnom tvare:

Produkt ID:2577 nie je na sklade

Podobným spôsobom je možné riešiť požiadavku na rezervovanie tovaru alebo objednanie tovaru. Zmení sa len označenie typu SMS.



Obr.1 SAEAUT SMS Service – Konfigurátor

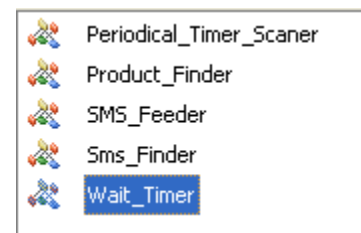
Práve pre aplikácie takéhoto typu bol navrhnutý produkt SAEAUT SMS Service. Tento produkt pozostáva z dvoch samostatných softvérových

aplikácií – konfigurátora (Obr.1.) a zo služby OS Windows (bez užívateľského rozhrania), ktorá zabezpečuje prijímanie a odosielanie SMS správ.

Túto službu môže využívať viacero klientských aplikácií, podobného typu ako je tu popísaný automatický SAEAUT SMSresponder, bežiacich na rovnakom počítači, alebo na inom počítači v sieti.

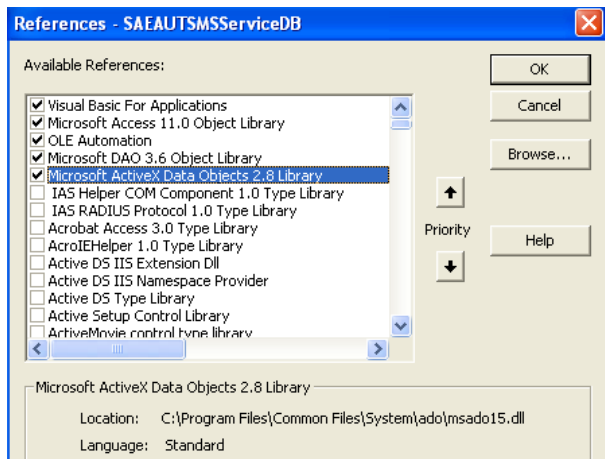
SAEAUT SMSresponder bol vytvorený v jazyku **VBA priamo v MS Acces databázovom súbore, ktorý využíva aj SAEAUT SMS service.** Uvedený súbor si môžete stiahnuť na www.saeautom.sk Samozrejme, táto aplikácia mohla byť súčasťou aj externého databázového súboru, alebo mohla byť napísaná v ľubovlnom inom jazyku. Jedinou podmienkou je zabezpečiť vkladanie a vyberanie SMS do/z databázového súboru SMS Service. Pre Vaše konkrétne využívanie aplikácie pravdepodobne bude potrebné urobiť malé zmeny. Veríme však že Vaši programátori to na základe popisu zvládnu, prípadne SAE – Automation Vám tieto úpravy implementuje na požiadanie.

Aplikácia pozostáva z modulov na Obr.2: Modul Periodical_Timer_Scaner zabezpečuje periodické spúšťanie hlavného cyklu aplikácie. Modul Product_Finder vyhledá pomocou Id produktu v indexe databázy. Modul SMS_Feeder vytvorí SMS odpoveď a uloží do tabuľky opdcchádzajúcich SMS. Modul SMS_Finder zisťuje či prišla nová SMS očakávaného typu. Wait_Timer vytvára stavový automat programu. .



Obr. 2

Pripojenie Microsoft ActiveX Data Objects 2.8 Library Pre správnu funkcionality demo aplikácie je nevyhnutne najprv referencovať knižnicu msado15.dll podľa hora priloženého Obr. 3.



Obr. 3

ID	recSenderPhoneNumber	recSendTimeStamp	recSMSText	MarketSmsOccur
1	+421915749096	5.3.2010 12:56:09	<*>3	<input checked="" type="checkbox"/>
2	+421915749096	5.3.2010 12:56:20	<*>1	<input checked="" type="checkbox"/>

Obr. 4

Rozšírenie tabuľky ReceivedSMSList Pre rozlíšenie štandardných SMS od SMS špecifických pre aplikáciu bola pôvodná tabuľka *ReceivedSMSList* na Obr.4 doplnená o stĺpec *MarketSmsOccur*.

Prepojenie alebo vloženie tabuľky store. Informácie o produktoch na sklade sú uložené v tabuľke *store*, (Obr. 5) ktorú je možné buď vytvoriť priamo v pôvodnej databáze, alebo vytvoriť prepojenie s inou databázou..

	Název pole	Datový typ
?	ProdID	číslo
	ProdName	text
	ProdPrice	číslo
	ProdQuantity	číslo

Obr. 5

iba od programátora ako si zdefinuje a nastaví kľúčové znaky v *store SMS* správe a na akej pozícii.

Výpis kompletného programu nájdete na nasledujúcich stránkach.

```
Option Compare Database
```

```
Public strCnxn As String
```

```
Function ScanTimer()
```

```
Dim telProd As String
```

```
Dim outSms As String
```

```
Dim MDBpath As String
```

```
MDBpath = _
```

```
"c:\Program Files\SAE - Automation, s.r.o\SAEAUT SMS  
Service\Database\SAEAUTSMSServiceDB.mdb"
```

```
strCnxn = _
```

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

```
'Main cycle repeating by every 10 second period
```

```
Do
```

```
telProd = FindMarketSMS
```

```
    If telProd <> "notFound" Then
```

```
        outSms = FindProduct(telProd)
```

```
        FeedSMS telProd, outSms
```

```
    End If
```

```
    wait 10
```

```
Loop
```

Tab 1: TimerScanner

Option Compare Database

```

'Function for periodically
'finding of arrived new storeSMS in tab ReceivedSMSList
Function FindMarketSMS() As String

    On Error GoTo ErrorHandler

    'recordset and connection variables
    Dim rstListSms As ADODB.Recordset
    Dim Cnxn As ADODB.Connection
    Dim strSQLListSms As String
    Dim key As String

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

    'open recordset server-side for indexing
    Set rstListSms = New ADODB.Recordset
    rstListSms.CursorLocation = adUseServer
    strSQLListSms = "ReceivedSMSList"

    rstListSms.Open strSQLListSms, strCnxn, adOpenKeyset, _
    adLockOptimistic, adCmdTableDirect

    rstListSms.MoveFirst
    maxRec = rstListSms.RecordCount
    key = Left(rstListSms!recSMSText, 3)

    'Cycle for sequential serching
    'of storeSMS in tab ReceivedSMSList
    Do While rstListSms.EOF = False
        'Logic of deciding by format key
        'whether the SMS is specific storeSMS
        If (key = "<*>") And (rstListSms!MarketSmsOccur = False) Then
            Beep
            rstListSms!MarketSmsOccur = True
            rstListSms.Update
            FindMarketSMS = rstListSms!recSenderPhoneNumber & "," _
            & rstListSms!recSMSText
            Exit Do
        End If

        If rstListSms!id <= maxRec Then
            rstListSms.MoveNext
        Else
            FindMarketSMS = "notFound"
            Exit Do
        End If
    Loop

    If rstListSms.EOF = True Then FindMarketSMS = "notFound"

    ' clean up
    rstListSms.Close

```

```

Cnxn.Close
Set rstListSms = Nothing
Set Cnxn = Nothing
Exit Function

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

```

Tab 2: SmsFinder

```

Option Compare Database

Function FindProduct (telProd As String) As String

    On Error GoTo ErrorHandler

    'recordset and connection variables
    Dim rstProdukt As ADODB.Recordset
    Dim Cnxn As ADODB.Connection
    Dim strSQLProdukt As String
    Dim tmp, strID As String
    Dim id As Integer

    'tmp = "+444222111111<*>1
    adr1 = InStr(1, telProd, ">")
    adr2 = Len(telProd)
    lenIdProd = adr2 - (adr1)

    tmp = Mid(telProd, adr1 + 1, lenIdProd)

    strID = Trim(tmp)

    'Logic of deciding whether the format of store SMS is correct
    If Not IsNumeric(strID) Then
        FindProduct = "bad format of SMS question"
        Exit Function
    End If

    'Open connection
    Set Cnxn = New ADODB.Connection

    Cnxn.Open strCnxn

    'Open recordset server-side for indexing
    Set rstProdukt = New ADODB.Recordset
    rstProdukt.CursorLocation = adUseServer

```

```

strSQLProdukt = "Store"
rstProdukt.Open strSQLProdukt, strCnxn, adOpenKeyset, _
adLockOptimistic, adCmdTableDirect
rstProdukt.Index = "PrimaryKey"

'point of deciding whether a product is presented in store or not
rstProdukt.Seek Array(strID), adSeekFirstEQ
If rstProdukt.EOF Then
    FindProduct = "Product by ID: " & strID & _
    " in the store not found"
Else
    FindProduct = "Product by ID: " & rstProdukt!prodID & _
    " named: " & rstProdukt!prodName & _
    " is presented in the store in quantity " & _
    rstProdukt!prodQuantity & _
    ", price is " & rstProdukt!prodPrice & "$"
End If

' clean up
rstProdukt.Close
Cnxn.Close
Set rstProdukt = Nothing
Set Cnxn = Nothing
Exit Function

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

Set rstProdukt = Nothing
Set Cnxn = 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 Function

```

Tab 3: ProdFinder

```

Option Compare Database

'Declaring of function important for the getting of Computer name
Private Declare Function GetComputerName Lib "kernel32" _
    Alias "GetComputerNameA" _
    (ByVal lpBuffer As String, nSize As Long) As Long

'Procedure for exact formatting and inserting a new SMS to the SimpleSMSList

```

```

table
'SimpleSMSList table serves as a outputing gate for the outgoing SMS
Sub FeedSMS(ByVal tel As String, ByVal sms As String)

    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 strSQLFeedSms

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

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

    'preformatting of inputing tel. +444222111111 number
    'to the standard world format +444(222)111111
    adr1 = InStr(1, tel, ",") - 1
    tel = Left(tel, adr1)

    lenTelPreselection = Len(tel) - 9

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

    telSum = tel1 & "(" & tel2 & ")" & tel3

    'inserting of the formatted strings
    'to the specific fields of the table SimpleSMSList
    rstFeedSms.AddNew
    rstFeedSms!usComputerName = ReturnComputerName "SAE18"
    rstFeedSms!usSenderPhone = telSum
    rstFeedSms!usSMSText = sms
    rstFeedSms!usTimeStamp = datTim
    rstFeedSms.Update
    blnRecordAdded = True

    '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 geting of Computer name

Function ReturnComputerName() As String
Dim rString As String * 255, sLen As Long, tString As String
tString = ""
On Error Resume Next
sLen = GetComputerName(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
ReturnComputerName = UCase(Trim(tString))
End Function

```

Tab 4: SMSFeeder