

VB Code: Wie kann man Kontakte aus Microsoft Outlook auslesen?

Option Explicit

```
' *** Use this declarations during development only:
' Dim olApp As Outlook.Application
' Dim olNS As Outlook.NameSpace
' Dim olFolder As Outlook.MAPIFolder
' Dim olContact As Outlook.ContactItem
' *** Reference to "Microsoft Outlook x.0 Object Library"
```

Public Type Contact

```
Categories As String           ' Used by Siemens VCard
LastName As String
FirstName As String
HomeAddressStreet As String
HomeAddressPostalCode As String
HomeAddressCity As String
HomeAddressCountry As String
CompanyName As String
HomeTelephoneNumber As String
BusinessTelephoneNumber As String
MobileTelephoneNumber As String
HomeFaxNumber As String
Email1Address As String
PersonalHomePage As String
'...   There are many more items used in Outlook
'       but the Siemens mobile VCard doesn't support them
```

End Type

```
' declared like Outlook does
```

```
Private Const olFolderContacts As Long = 10
```

```
' Log errors into this file (File path is App.Path)
```

```
Private Const errFile As String = "error.log"
```

```
' Switch to turn off the logging functionality
```

```
' Shouldn't be turned off, only unexpected errors were logged
```

```
Private Const doErrorLogging As Boolean = True
```

```
Public Function isOutlookInstalled() As Boolean
```

```
' Tries to create an instance of Outlook
```

```
' Returns true if successful, otherwise false
```

```
Dim olApp As Object
```

```
On Error Goto errHandler
```

```
Set olApp = CreateObject("Outlook.Application")
isOutlookInstalled = True
Set olApp = Nothing

Exit Function
errHandler:
If Err.Number = 429 Then
    ' Can't create Object -> Outlook is not installed
    ' Don't need to log as error.
Else
    ErrPrint "Function isOutlookInstalled() returned with error"
End If
isOutlookInstalled = False
End Function

Public Function getOutlookShortVersion() As Integer
    ' Returns the major version of installed Outlook
    ' If Outlook isn't installed or an error occurs
    ' the return value is zero

    Dim olApp As Object
    Dim splittArr() As String

    On Error Goto errHandler

    If Not isOutlookInstalled Then
        getOutlookShortVersion = 0
    Else
        Set olApp = CreateObject("Outlook.Application")
        splittArr = Split(olApp.Version, ".")

        If Not isArrayDimensioned(splittArr) Then
            getOutlookShortVersion = 0
        Else
            getOutlookShortVersion = CInt(splittArr(0))
        End If

        Set olApp = Nothing
    End If

    Exit Function
errHandler:
    ErrPrint "Function getOutlookShortVersion() returned with error"
    getOutlookShortVersion = 0
End Function

Public Function getOutlookVersionName() As String
    ' Returns the "spoken version" of installed Outlook
    ' If Outlook isn't installed or an error occurs
    ' the return value a nullstring
    On Error Goto errHandler
```

```
If Not isOutlookInstalled Then
    getOutlookVersionName = vbNullString
Else
    Select Case getOutlookShortVersion
    Case 10
        getOutlookVersionName = "Outlook® 2002"
    Case 9
        getOutlookVersionName = "Outlook® 2000"
    Case 8
        getOutlookVersionName = "Outlook® 97"
    Case 7
        getOutlookVersionName = "Outlook® 95"
    Case Is < 7
        getOutlookVersionName = "Outlook® version less than 95"
    End Select
End If
Exit Function
errHandler:
    ErrPrint "Function getOutlookVersionName() returned with error"
    getOutlookVersionName = vbNullString
End Function

Public Function getOutlookLongVersion() As String
    ' Returns the whole version of installed Outlook
    ' If Outlook isn't installed or an error occurs
    ' the return value is a nullstring

    Dim olApp As Object

    On Error Goto errHandler

    If Not isOutlookInstalled Then
        getOutlookLongVersion = vbNullString
    Else
        Set olApp = CreateObject("Outlook.Application")
        getOutlookLongVersion = olApp.Version
        Set olApp = Nothing
    End If

    Exit Function
errHandler:
    ErrPrint "Function getOutlookLongVersion() returned with error"
    getOutlookLongVersion = vbNullString
End Function

Public Function getContactFolderCount() As Integer
    ' Returns the number of contacts in the Outlook contacts folder,
    ' returns zero if Outlook is not installed

    Dim olApp As Object
    Dim olNameSpace As Object
```

```
Dim olFolder As Object

On Error Goto errHandler

If Not isOutlookInstalled Then
    getContactFolderCount = 0
Else
    Set olApp = CreateObject("Outlook.Application")
    Set olNameSpace = olApp.GetNamespace("MAPI")
    Set olFolder = olNameSpace.GetDefaultFolder(olFolderContacts)

    getContactFolderCount = olFolder.Items.Count

    Set olFolder = Nothing
    Set olNameSpace = Nothing
    Set olApp = Nothing
End If

Exit Function
errHandler:
    ErrPrint "Function getContactFolderCount() returned with error"
    getContactFolderCount = 0
End Function

Public Function getContact(index As Integer) As Contact
    ' Returns the Outlook contact information of the entry,
    ' contains only the information which siemens mobile supports.
    ' Returns Nothing if Outlook is not installed or
    ' the specified entry doesn't exist.
    Dim olApp As Object
    Dim olNameSpace As Object
    Dim olFolder As Object
    Dim olContact As Object
    Dim tmpContact As Contact
    On Error Goto errHandler

    If Not isOutlookInstalled Then
        'getContact = Nothing
    ElseIf getContactFolderCount < index Then
        'getContact = Nothing
    Else
        Set olApp = CreateObject("Outlook.Application")
        Set olNameSpace = olApp.GetNamespace("MAPI")
        Set olFolder = olNameSpace.GetDefaultFolder(olFolderContacts)
        Set olContact = olFolder.Items(index)
        With tmpContact
            .BusinessTelephoneNumber = olContact.BusinessTelephoneNumber
            .Categories = olContact.Categories
            .CompanyName = olContact.CompanyName
            .Email1Address = olContact.Email1Address
            .FirstName = olContact.FirstName
```

```
.HomeAddressCity = olContact.HomeAddressCity
.HomeAddressCountry = olContact.HomeAddressCountry
.HomeAddressPostalCode = olContact.HomeAddressPostalCode
.HomeAddressStreet = olContact.HomeAddressStreet
.HomeFaxNumber = olContact.HomeFaxNumber
.HomeTelephoneNumber = olContact.HomeTelephoneNumber
.LastName = olContact.LastName
.MobileTelephoneNumber = olContact.MobileTelephoneNumber
.PersonalHomePage = olContact.PersonalHomePage
End With
Set olContact = Nothing
Set olFolder = Nothing
Set olNameSpace = Nothing
Set olApp = Nothing
getContact = tmpContact
End If

Exit Function
errHandler:
  ErrPrint "Function getContact() returned with error"
  'getContact = Nothing
End Function

Public Function modifyContact(index As Integer, _
                             modContact As Contact) As Boolean
  ' Sets the new information to the outlook contact entry
  ' with the specified index.
  ' Returns True if successful,
  ' otherwise False (e.G. if entry doesn't exist)
  Dim olApp As Object
  Dim olNameSpace As Object
  Dim olFolder As Object
  Dim olContact As Object
  On Error Goto errHandler

  If Not isOutlookInstalled Then
    modifyContact = False
  ElseIf getContactFolderCount < index Then
    modifyContact = False
  Else
    Set olApp = CreateObject("Outlook.Application")
    Set olNameSpace = olApp.GetNamespace("MAPI")
    Set olFolder = olNameSpace.GetDefaultFolder(olFolderContacts)
    Set olContact = olFolder.Items(index)
    With olContact
      .BusinessTelephoneNumber = _
        modContact.BusinessTelephoneNumber
      .Categories = modContact.Categories
      .CompanyName = modContact.CompanyName
      .EmailAddress = modContact.EmailAddress
      .FirstName = modContact.FirstName
    End With
  End If
  modifyContact = True
End Function
```

```
.HomeAddressCity = modContact.HomeAddressCity
.HomeAddressCountry = modContact.HomeAddressCountry
.HomeAddressPostalCode = modContact.HomeAddressPostalCode
.HomeAddressStreet = modContact.HomeAddressStreet
.HomeFaxNumber = modContact.HomeFaxNumber
.HomeTelephoneNumber = modContact.HomeTelephoneNumber
.LastName = modContact.LastName
.MobileTelephoneNumber = modContact.MobileTelephoneNumber
.PersonalHomePage = modContact.PersonalHomePage
.Save
End With
Set olContact = Nothing
Set olFolder = Nothing
Set olNameSpace = Nothing
Set olApp = Nothing
modifyContact = True
End If

Exit Function
errHandler:
  ErrPrint "Function modifyContact() returned with error"
  modifyContact = False
End Function

Public Function addContact(newContact As Contact) As Integer
  ' Adds the new contact to the outlook contact folder
  ' Returns Tthe index of the new entry, Zero if creation failed
  Dim olApp As Object
  Dim olNameSpace As Object
  Dim olFolder As Object
  Dim olContact As Object
  On Error Goto errHandler

  If Not isOutlookInstalled Then
    addContact = 0
  Else
    Set olApp = CreateObject("Outlook.Application")
    Set olNameSpace = olApp.GetNamespace("MAPI")
    Set olFolder = olNameSpace.GetDefaultFolder(olFolderContacts)
    Set olContact = olFolder.Items.Add
    With olContact
      .BusinessTelephoneNumber = _
        newContact.BusinessTelephoneNumber
      .Categories = newContact.Categories
      .CompanyName = newContact.CompanyName
      .Email1Address = newContact.Email1Address
      .FirstName = newContact.FirstName
      .HomeAddressCity = newContact.HomeAddressCity
      .HomeAddressCountry = newContact.HomeAddressCountry
      .HomeAddressPostalCode = newContact.HomeAddressPostalCode
      .HomeAddressStreet = newContact.HomeAddressStreet
```

```
        .HomeFaxNumber = newContact.HomeFaxNumber
        .HomeTelephoneNumber = newContact.HomeTelephoneNumber
        .LastName = newContact.LastName
        .MobileTelephoneNumber = newContact.MobileTelephoneNumber
        .PersonalHomePage = newContact.PersonalHomePage
        .Save
    End With
    Set olContact = Nothing
    Set olFolder = Nothing
    Set olNameSpace = Nothing
    Set olApp = Nothing
    addContact = getContactFolderCount
End If

Exit Function
errHandler:
    ErrPrint "Function addContact() returned with error"
    addContact = 0
End Function

Public Function deleteContact(index As Integer) As Boolean
    ' Deletes the specified entry from the Outlook contact folder
    ' Returns True if successful,
    ' otherwise False (e.G. if entry doesn't exist)
    Dim olApp As Object
    Dim olNameSpace As Object
    Dim olFolder As Object
    Dim olContact As Object
    On Error Goto errHandler

    If Not isOutlookInstalled Then
        deleteContact = False
    ElseIf getContactFolderCount < index Then
        deleteContact = False
    Else
        Set olApp = CreateObject("Outlook.Application")
        Set olNameSpace = olApp.GetNamespace("MAPI")
        Set olFolder = olNameSpace.GetDefaultFolder(olFolderContacts)
        Set olContact = olFolder.Items(index)
        olContact.Delete
        Set olContact = Nothing
        Set olFolder = Nothing
        Set olNameSpace = Nothing
        Set olApp = Nothing
        deleteContact = True
    End If

    Exit Function
errHandler:
    ErrPrint "Function deleteContact() returned with error"
    deleteContact = False
```

End Function

```
Public Function findContact(LastName As String, _
    Optional FirstName As String = vbNullString) As Integer
    ' Searches for specified entry in the Outlook contact folder
    ' Returns the index if found, otherwise Zero
    Dim olApp As Object
    Dim olNameSpace As Object
    Dim olFolder As Outlook.MAPIFolder 'Object
    Dim olContact As Outlook.ContactItem 'Object
    Dim i As Integer
    On Error Goto errHandler

    If Not isOutlookInstalled Then
        findContact = 0
    Else
        Set olApp = CreateObject("Outlook.Application")
        Set olNameSpace = olApp.GetNamespace("MAPI")
        Set olFolder = olNameSpace.GetDefaultFolder(olFolderContacts)
        findContact = 0
        For i = 1 To olFolder.Items.Count
            Set olContact = olFolder.Items(i)
            If LCase(olContact.LastName) = LCase(LastName) Then
                If FirstName = "" Then
                    findContact = i
                    Exit For
                ElseIf olContact.FirstName = FirstName Then
                    findContact = i
                    Exit For
                End If
            End If
        Next i
        Set olContact = Nothing
        Set olFolder = Nothing
        Set olNameSpace = Nothing
        Set olApp = Nothing
    End If

    Exit Function
errHandler:
    ErrPrint "Function findContact() returned with error"
    findContact = 0
End Function
```

```
'***** Error logging *****
Public Sub ErrPrint(str As Variant)
    ' Error logging into file
    ' Adds a timestamp and the number/description of the error
    ' Should only be used if an unexpected error occurs
    Dim fnr As Long
    If doErrorLogging Then
```

```

    fnr = FreeFile
    Open checkPath(App.Path) & errFile For Append As #fnr
    Print #fnr, CStr(Time) & vbTab & CStr(str)
    Print #fnr, vbTab & "Fehler " & _
        CStr(Err.Number) & ": " & Err.Description
    Print #fnr, ""
    Close #fnr
    DoEvents
End If
End Sub

```

\$HEADER

Hier kommt die Beschreibung rein

Hier kommt die Hardcopy rein

Felder

\$FELDER

Tabelle

\$TABELLE

Menüpunkte

Datei	Informationen zum Menüpunkt "Datei"
Bearbeiten	Informationen zum Menüpunkt "Bearbeiten"
Extras	Informationen zum Menüpunkt "Extras"
Hilfe	Informationen zum Menüpunkt "Hilfe"

\$MENU

Buttons

[Informationen zu den "Buttons"](#) \$BUTTONS

[\\$TAGS](#)



From:

<http://172.30.2.91/> - **cimERP Online Hilfe**

Permanent link:

http://172.30.2.91/doku.php?id=cimerp:0200_tipps:0030_vb_code_contacts_from_outlook

Last update: **28.09.2021 15:34:05**

