```
'************************************************************************
' This digital signature is based on the Aloaha PDF SDK
'
' it can be downloaded from:
' http://www.aloaha.com/download/aloaha_pdf.zip
' or
' http://www.aloaha.com/download/suite_sdk.zip
'
' A valid license key is required
' Please contact aloaha@wrocklage.de for an evaluation key
'************************************************************************

Dim pdf
Dim inputpdf
Dim outputpdf
Dim signed
Dim container
Dim ntype
Dim Left
Dim top
Dim Width
Dim height
Dim ndate       'use current date/time
Dim opened      'render note opened or closed
Dim mediabox    'mediabox string
Dim page        'page to render note on
Dim x1          'x start mediabox
Dim y1          'y start mediabox
Dim x2          'x end mediabox
Dim y2          'y end mediabox
Dim ph          'pageheight - 0 for auto
Dim pw          'pagewidth - 0 for auto
Dim cert        'cert to be used


signed=false

inputpdf = CStr("d:\1.pdf")
outputpdf = CStr("d:\output.pdf")

Set pdf = CreateObject("aloahapdf.edit")

'check if document is signed already
signed=is_signed(inputpdf)

'load PDF to memory
If pdf.load_pdf_to_mem(CStr(inputpdf)) = true Then
        page = 1            'PDF Page No. to be used

        'mediabox etc only needed if page size is not known
      mediabox = pdf.get_pagesize_s(CStr(inputpdf),CLng(page))
        If mediabox<>"" Then
            x1=Trim(split(mediabox)(0))
            y1=Trim(split(mediabox)(1))
            x2=Trim(split(mediabox)(2))
            y2=Trim(split(mediabox)(3))
            ph=Abs(CLng(y2)-CLng(y1))
            pw=Abs(CLng(x2)-CLng(x1))
      End If


        Left = pw/100*10            '10% from left
        top = ph/100*25             '25% from bottom

        ndate = true
        opened = 1
        Width = pw/5
        height = ph/7

        If ndate = True Then ntype = 1 Else ntype = 101
        If signed = false Then
                'use traditional note
                If pdf.make_note(CDbl(Left), CDbl(top), CLng(ntype), CDbl(Left), CDbl(top),CDbl(Width), CDbl(height), CStr("Title"), CStr("contents"), CDbl(0), CDbl(0), CDbl(0), CLng(opened)) = true Then
                        If pdf.save_pdf(CStr(outputpdf)) = true Then
                                inputpdf=outputpdf
                        End If
                End If
        Else
                'remove document from memory
                Call pdf.clean_docu

                'use incremental note
                If pdf.add_inc_note(CStr(inputpdf), CStr(outputpdf), CLng(1), CBool(ndate), CStr("nbody"), CStr("nsubject"), CBool(opened), CLng(Left), CLng(top), CLng(Width), CLng(height)) = true Then
                        inputpdf=outputpdf
                End If
        End If
End If
'remove PDF from memory
Call pdf.clean_docu

px1 = 84            'left corner of signature in % of paper width
py1 = 8             'upper corner of signature in % of paper height
px2 = 99            'right corner of signature in % of paper width
py2 = 2             'lower corner of signature in % of paper height

cert = "0"          'use qualified certificate of first card reader
If sign(CStr(cert),CLng(page),inputpdf,outputpdf, px1, py1, px2, py2)=true Then
        signed=true
        Call pdf.clean_docu

        Left = pw/100*10            '10% from left
        top = ph/100*75             '75% from bottom
        opened = false
        If pdf.add_inc_note(CStr(outputpdf), CStr(outputpdf), CLng(1), CBool(ndate), CStr("nbody"), CStr("nsubject"), CBool(opened), CLng(Left), CLng(top), CLng(Width), CLng(height)) = true Then
                inputpdf=outputpdf
                px1 = 84            'left corner of signature in % of paper width
                py1 = 15            'upper corner of signature in % of paper height
                px2 = 99            'right corner of signature in % of paper width
                py2 = 9             'lower corner of signature in % of paper height
                cert = "1"          'use qualified certificate of second card reader
                If sign(CStr(cert),CLng(page),inputpdf,outputpdf, px1, py1, px2, py2)=true Then
                        MsgBox "Ready"
                Else
                        MsgBox "Error while signing"
                End If
        End If
Else
        MsgBox "Error while signing"
End If

Call pdf.clean_docu
Set pdf = nothing


Function is_signed(checkpdf)
        is_signed=pdf.is_pdf_signed(CStr(checkpdf))
End Function

Function sign(cert, page, infile, outfile, px1, py1, px2, py2)
        Dim reason     'signature reason
        Dim location   'signature physical location
        Dim image      'image to be used for signature - empty string allowed

        Dim xoff       'x-offset if page does not start at 0
        Dim yoff       'y-offset if page does not start at 0

        xoff = 0
        yoff = 0

        reason = "I created this document"
        location = "I am in the office"

        'if cert was not set use first card reader
        If cert ="" Then cert ="0"

        'set page where signature will be shown
        If page<1 Then page = 1

      pdf.currentpage = CLng(page)

      'set signature imagepath. Must be a color jpg file.
      'Empty string also possible
      image = pdf.aloahapath+"jpg\george1.jpg"

      If mediabox="" Then
          ph=0
          pw=0
      End If
      'set pageheight and pagewidth to 0 for autodetect
      'ph=0
      'pw=0

      'calculate offsets for signature position
      'in most cases they can be set simply to 0
      If ph >0 Then
                yoff = ((y1/ph)*100)\1
      Else
                yoff = 0
      End If

      If pw >0 Then
                xoff = ((x1/pw)*100)\1
      Else
                xoff = 0
      End if


    sign = pdf.sign_pdf_file(CStr(infile), CStr(outfile), CLng(pw), CLng(ph), CLng(px1+xoff), CLng(py1+yoff), CLng(px2+xoff), CLng(py2+yoff), CStr(reason), CStr(location), CStr(cert), CStr(image))

        If sign = false Then
                If pdf.lasterror <> "" Then MsgBox lasterror
                If pdf.signerror <> "" Then MsgBox signerror
        End If

End Function
```