

```

VERSION 5.00
Begin VB.Form frmVbDemo
    Caption           =   "CMG Demo Program"
    ClientHeight      =   2820
    ClientLeft        =   60
    ClientTop         =   345
    ClientWidth       =   4455
    LinkTopic         =   "Form1"
    ScaleHeight       =   2820
    ScaleWidth        =   4455
    StartUpPosition  =   2   'CenterScreen
    Begin VB.CommandButton cmdQuit
        Caption       =   "Quit"
        Height        =   495
        Left          =   2640
        TabIndex      =   3
        Top           =   1920
        Width         =   1215
    End
    Begin VB.CommandButton cmdProcess
        Caption       =   "Process"
        Height        =   495
        Left          =   600
        TabIndex      =   2
        Top           =   1920
        Width         =   1215
    End
    Begin VB.Label lblStatus
        Height        =   255
        Left          =   600
        TabIndex      =   4
        Top           =   1080
        Width         =   3135
    End
    Begin VB.Label lblVersion
        Height        =   255
        Left          =   2280
        TabIndex      =   1
        Top           =   480
        Width         =   1455
    End
    Begin VB.Label Label1
        Caption       =   "CMG/DLL Version"
        Height        =   255
        Left          =   600
        TabIndex      =   0
        Top           =   480
        Width         =   1455
    End
End
Attribute VB_Name = "frmVbDemo"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True

```

```
Attribute VB_Exposed = False
Option Explicit
```

```
Private Sub cmdProcess_Click()
    'VbDemo illustrates the use of the CMG DLL.
    '
    '*****
    '***** Changes for Version 2.40 of the DLL *****
    '*****
    'Note the following changes required for version 2.30 of the DLL:
    '
    ' 1. References to CMG_230 were changed to CMG_240.
    '*****
    'Program reads a single input file: CmgTest.txt which must be located
    'in the application directory. Each record of CmgTest.txt contains
    'the expected CMG group code in bytes 774-778.
    '
    'The program reads each record from the CmgTest.txt, uses the CMG
    'DLL to calculate the expected CMG group code, and writes data to
    'an output file in the application directory called VbDemo.log.
    '
    'This log file will contain one record for each input record which
    'was processed. The output records will show the expected CMG group
    'code (from the input record) and the calculated CMG group code.
    'The output file will also list variables used in the CMG
calculation.
    '
    'After the last input record is processed, a display box will report
    'the number of mismatches between the expected and computed CMG
    'group codes. There should be no mismatches.
    '
    'Before running this demo program, make sure that you have registered
    'the CMG DLL with Windows using RegSvr32.exe. In addition, you must
    'add a reference to the DLL to this Visual Basic project by doing
    'the following:
    ' 1. Select Project|References.
    ' 2. Select "CMG Version 2.30 32-bit ActiveX DLL".
    '
    'Note that the file CMG_240.DAT must be in the application
    'directory. Please refer to the CMG technical documentation for
    'details.
    '
    'Note that the expected values for the CMG group and the other
    'variables are not normally present in an IRF-PAI record. These
    'are present in the test file for demonstration purposes only.
    'Normally, to calculate CMG group codes, a programmer will need to
    'do the following:
    ' 1. Create a local instance of the CMG object.
    ' 2. Set the IrfPaiRecord property to the input record string in
    ' standard submission format.
    ' 3. Invoke the CmgCalc method to force the CMG object to perform
    ' the CMG calculations.
    ' 4. Get the CmgValu property from the CMG object -- this will
    ' contain the CMG group code.
```

```

'    5.  Get the ErrorCode property from the CMG object -- this will
'        have a value of zero if no errors were encountered.
'Additional steps in this demo program are for illustrative purposes
'only.
'
Dim Cmg As New CMG_240          'Creates a local instance of the CMG
object
Dim sCmgIn As String           'Expected CMG group code from input
record
Dim sErrorCodeIn As String     'Expected error code from input
record
Dim sMotorScoreIn As String    'Expected motor score from input
record
Dim sCognitiveScoreIn As String 'Expected cognitive score from input
rec
Dim sAgeIn As String           'Expected age from input record
Dim sImpairGroupIn As String   'Impairment group from input record
Dim sComorbidIn(10) As String  'Comorbid conditions from input
record
Dim sCmg As String             'Calculated CMG group code
Dim nMotor As Single           'Calculated motor score
Dim iCognitive As Integer      'Calculated cognitive score
Dim iAge As Integer            'Calculated age
Dim iError As Integer          'Error code from CMG object
Dim i As Integer               'General purpose index
Dim iInFile As Integer         'File number for input file
Dim iOutFile As Integer        'File number for output file
Dim lRecord As Long            'Record counter
Dim lMismatch As Long          'Mismatch counter
Dim sBuffer As String          'Buffer holds current input record
Dim sList As String            'Buffer used to build output string
Dim sString As String          'Buffer used to build output string
Dim sText As String            'Buffer used to build output string

DoEvents
On Error GoTo FileError

'Open input file: CmgTest.Txt
iInFile = FreeFile
Open App.Path & "\cmgtest.txt" For Input As iInFile
On Error GoTo EndOfFile

'Open output file: VbDemo.Log
iOutFile = FreeFile
Open App.Path & "\vbdemo.log" For Output As iOutFile

'Output column headings in output file
sText = "RecNo  CmgIn|Out--  Error  Motor      Cognt  Age----
Impair-  Comorbids"
Print #iOutFile, sText

'Initialize counters
lRecord = 0
lMismatch = 0

```

```

'Loop through records in input file
Do While True

    'Read input record, increment counter, update status display
    Line Input #iInFile, sBuffer
    lRecord = lRecord + 1
    lblStatus.Caption = "Processing record " & lRecord
    DoEvents

    'Calculate CMG group code using CMG object
    Cmg.IrfPaiRecord = sBuffer    'Set input record property
    Cmg.CmgCalc                  'Invoke calculation method

    'Get output values from CMG object
    sCmg = Cmg.CmgValue           'Get calculated CMG group
    iError = Cmg.ErrorCode        'Get error code
    nMotor = Cmg.Motor            'Get calculated motor score
    iCognitive = Cmg.Cognitive    'Get calculated cognitive score
    iAge = Cmg.Age                'Get calculated age

    'Get expected values from input record (these will not normally
    'be present in an IRF-PAI record -- they are in the test file
    'for illustrative purposes only).
    sCmgIn = Mid$(sBuffer, 774, 5)    'Expected CMG group
code
    sImpairGroupIn = Mid$(sBuffer, 339, 9)    'Expected impairment
group
    sComorbidIn(1) = Mid$(sBuffer, 372, 7)    'Comorbidity code 1
    sComorbidIn(2) = Mid$(sBuffer, 379, 7)    'Comorbidity code 2
    sComorbidIn(3) = Mid$(sBuffer, 386, 7)    'Comorbidity code 3
    sComorbidIn(4) = Mid$(sBuffer, 393, 7)    'Comorbidity code 4
    sComorbidIn(5) = Mid$(sBuffer, 400, 7)    'Comorbidity code 5
    sComorbidIn(6) = Mid$(sBuffer, 407, 7)    'Comorbidity code 6
    sComorbidIn(7) = Mid$(sBuffer, 414, 7)    'Comorbidity code 7
    sComorbidIn(8) = Mid$(sBuffer, 421, 7)    'Comorbidity code 8
    sComorbidIn(9) = Mid$(sBuffer, 428, 7)    'Comorbidity code 9
    sComorbidIn(10) = Mid$(sBuffer, 435, 7)    'Comorbidity code 10
    sErrorcodeIn = Mid$(sBuffer, 779, 2)    'Expected error code
    sMotorScoreIn = Mid$(sBuffer, 781, 4)    'Expected motor score
    sCognitiveScoreIn = Mid$(sBuffer, 785, 2) 'Expected cognitive
score
    sAgeIn = Mid$(sBuffer, 787, 3)    'Expected age

    'Check calculated CMG group code against expected code. If they
    'don't match, increment the mismatch counter.
    If sCmgIn <> sCmg Then
        lMismatch = lMismatch + 1
    End If

    'Build string for output record
    sString = Format$(lRecord, "00000") & " " & _
        sCmgIn & "|" & _
        sCmg & " " & _

```

```

        sErrorCodeIn & "|" & _
        Format$(iError, "00") & " " & _
        sMotorScoreIn & "|" & _
        Format$(nMotor, "00.0") & " " & _
        sCognitiveScoreIn & "|" & _
        Format$(iCognitive, "00") & " " & _
        sAgeIn & "|" & _
        Format$(iAge, "000") & " " & _
        sImpairGroupIn & " "

    'Build string containing comorbidity codes for output record
    sList = ""
    For i = 1 To 10
        If Len(Trim(sComorbidIn(i))) > 0 Then
            If Len(sList) > 0 Then
                sList = sList & ","
            End If
            sList = sList & sComorbidIn(i)
        End If
    Next
    sString = sString & sList

    'Write record to output file
    Print #iOutFile, sString

Loop

EndOfFile:    'End of input file

    'Update status display
    lblStatus.Caption = "Processing complete."

    'Display message box
    MsgBox "Records processed = " & lRecord & Chr(10) & _
        "CMG mismatches = " & lMismatch

    'Close files
    Close iInFile
    Close iOutFile

    'End program
    End
    Exit Sub

FileError:    'Error handler

    MsgBox "Input file not found."
    Exit Sub

End Sub

Private Sub cmdQuit_Click()
    End
End Sub

```

```
Private Sub Form_Load()  
    Dim Cmg As New CMG_240  
    lblVersion.Caption = Cmg.CmgVersion & " / " & Cmg.DllVersion  
End Sub
```