WorkWindow (Window)

Properties

```
f2Transform As FolderItem
AmpYOut(-1) As Double
YIn(-1) As Double
nPts As Integer
XIn(-1) As Double
AmpXOut(-1) As Double
```

WorkWindow.doProcessY:

```
Sub doProcessY()
  Dim I, J, iAng  As Integer
  Dim nPtsP1, nPtsM1, nPts2  As Integer
  Dim E(-1), C(-1), S(-1) As Double
  Dim fAng, PIoverNpts  As Double

  Const gPI=3.14159265358979323846264338327950

  //getDataFromTextFile() : only need to get the data once
  SutFileLbl.Text = SutFileLbl.Text +    " - doProcessY"
  SutFileLbl.Refresh

  nPtsP1 = nPts +  1
  nPtsM1 = nPts - 1
  nPts2 =  2*nPts

  PIoverNpts = gPI/nPts
  Redim C(nPts2)
  FOR I = 0 TO nPts2
    fAng = I * PIoverNpts
    C(I) = COS(fAng)
    // S(I) = SIN(fAng)
  NEXT //I

  Redim E(nPts)
  E(nPts) = Yin(nPtsP1)
  FOR I = 1 TO nPtsM1
    E(I) =  2*Yin(I+1)
  NEXT //I

  Redim AmpYOut(nPts)
  AmpYOut(0) = Yin(1) + E(nPts)
  FOR I = 1 TO nPtsM1
    AmpYOut(0) = AmpYOut(0) + E(I)
  NEXT //I
  AmpYOut(0) = 0.5*AmpYOut(0)/nPts

  FOR J = 1 TO nPtsM1
```

```
    iAng = (J*nPts)  MOD nPts2
    AmpYOut(J)  = Yin( 1) + E(nPts)*C(iAng)
    FOR I = 1 TO nPtsM1
      iAng = (J*I)  MOD nPts2
      AmpYOut(J) = AmpYOut(J) + E(I)*C(iAng)
    NEXT // I
    AmpYOut(J) = AmpYOut(J)/nPts
  NEXT //J
End Sub
```

WorkWindow.getDataFromTextFile:

```
Sub getDataFromTextFile()
  Dim Y As Integer
  Dim doStream  as TextInputStream
  Dim CharTemp, LineIn, LineTemp   as String
  Dim I, J, K, LineLen  As Integer
  'Dim AllIn(12000) As Double
  Dim AllIn(- 1) As Double
  Dim moreToDo  As Boolean

  SutFileLbl.Text =  "Reading File"
  SutFileLbl.Refresh
  //open the input file
  doStream = f2Transform.OpenAsTextFile

  //parse one line at a time
  J = 1
  while not doStream.EOF
    // get a line
    LineIn = doStream.ReadLine()
    LineIn = LTrim(RTrim(LineIn))
    if len(LineIn) <  5 then
      moreToDo =  False
      // the shortest line is "0,0,0"
    else
      moreToDo =  True

      //another line to parse
      LineTemp =  ""
      for I = 1 to len(LineIn)
        //step through the line
        CharTemp = Mid(LineIn, I,   1)
        if ((CharTemp<> ",") and (CharTemp<> " ")) then
          // then we must be in the midst of an entry
          LineTemp = LineTemp + CharTemp
        else
          // We just finished reading a number
          if (len(LineTemp)> 0) then
            //    values are separated by commas and/or spaces
```

```
          //    if you reach one of these, process the number
          Redim AllIn(J)
          AllIn(J) = CDbl(LineTemp)
          // increment the index count and reset the buffer
          J = J + 1
          LineTemp =  ""
        end if
      end if
    next //I
    if (len(LineTemp)> 0) then
      // catch the last entry on the line
      Redim AllIn(J)
      AllIn(J) = CDbl(LineTemp)
      J = J + 1
      LineTemp =  ""
    end if
  end if
WEND //end parsing
doStream.Close

// put the data into XIn and YIn
For I = 1 To J-1 Step 3
  K = AllIn(I)
  if K>UBound(XIn)  then
    Redim XIn(K)
  end if
  XIn(K) = AllIn(I +  1)
  if (K+1)>UBound(YIn)  then
    Redim YIn(K +  1)
  end if
  YIn(K +  1) = AllIn(I +  2)
  if K>nPts  then
    nPts = K
  end if
next
SutFileLbl.Text =  "File read"
SutFileLbl.Refresh
End Sub
```

WorkWindow.doWriteBin:

```
Sub doWriteBin()
  Dim f as FolderItem
  Dim fileName  As String
  Dim i as Integer
  Dim stream  as BinaryStream

  fileName = Left(f2Transform.Name,   4) + ".amps.bin"
  //Open the output file
  f=GetSaveFolderItem(  "plain/text" ,fileName)
```

```
  If f<> Nil Then
    stream=f.CreateBinaryFile(  "bin" )
    For i=0 to (nPts-1)
      stream.WriteShort(i)
      stream.WriteDouble(AmpXOut(i))
      stream.WriteDouble(AmpYOut(i))
    Next
    stream.Close
  End if
End Sub
```

WorkWindow.doWriteText:

```
Sub doWriteText()
  Dim i, j As Integer
  Dim f As FolderItem
  Dim fileStream  As TextOutputStream
  Dim fileName  As String

  '// ========================================
  '// --- FOR FUTURE ---
  '// --- Writes output to single file ---
  'fileName = Left(f2Transform.Name,4) + ".amps.txt"
  '//Open the output file - type is left blanki for text files
  'file=GetSaveFolderItem("",fileName)
  'If file<> Nil Then
  'fileStream=f.CreateTextFile
  'for I=0 to (nPts-1)
  'fileStream.Write Str(I)
  'fileStream.Write ", "
  'fileStream.Write Str(AmpXOut(I))
  'fileStream.Write ", "
  'fileStream.Write Str(AmpYOut(I))
  'fileStream.WriteLine
  'next
  'fileStream.Close
  'End if
  '// ========================================
  // --- temprary to create files that the old programs can use
  fileName = Left(f2Transform.Name,   4) + "." + Str(UBound(AmpXOut))+  ".ampx"
  //Open the output file - type is left blanki for text files
  f = GetSaveFolderItem(  "",fileName)
  If f<> Nil Then
    fileStream=f.CreateTextFile
    j = min( 2000, UBound(AmpXOut))
    for I=1 to j
      fileStream.Write Str(I)
      fileStream.Write  ", "
      fileStream.Write Str(AmpXOut(I))
      fileStream.Write  ", "
```

```
      fileStream.Write Str(i)   'old programs ignore this entry
      fileStream.WriteLine
    next
    fileStream.Close
  End if
  fileName = Left(f.Name,  4) + "." + Str(UBound(AmpYOut)) +   ".ampy"
  //Open the output file - type is left blanki for text files
  f=GetSaveFolderItem(  "",fileName)
  If f<> Nil Then
    fileStream=f.CreateTextFile
    j = min( 2000, UBound(AmpYOut))
    for I=0 to j
      fileStream.Write Str(I)
      fileStream.Write  ", "
      fileStream.Write Str(AmpYOut(I))
      fileStream.Write  ", "
      fileStream.Write Str(i)
      fileStream.WriteLine
    next
    fileStream.Close
  End if
End Sub
```

WorkWindow.doProcessX:

```
Sub doProcessX()
  Dim I, J As Integer
  Dim nPts2, nPtsM1  As Integer
  Dim iAng As Integer
  Dim PIoverNpts, fAng  As Double
  Dim C(-1), S(-1) As Double
  Dim F(-1), G(-1) As Double

  Const gPI=3.14159265358979323846264338327950

  getDataFromTextFile()
  SutFileLbl.Text = SutFileLbl.Text +    " - doProcessX"
  SutFileLbl.Refresh

  Redim F(nPts)
  For I = 0 To nPts
    F(I) = (I*gPI)/nPts - XIn(I)
  Next //I

  nPts2 = 2*nPts
  nPtsM1 = nPts -  1

  PIoverNpts = gPI/nPts
  Redim C(nPts2)
  Redim S(nPts2)
```

```
  For I = 0 TO nPts2
    fAng = I * PIoverNpts
    // C(I) = COS(fAng)
    S(I) = SIN(fAng)
  Next //I

  Redim G(nPtsM1)
  For I = 1 TO nPtsM1
    G(I) = 2*F(I)
  Next //I

  Redim AmpXOut(nPts)
  For J = 1 TO nPtsM1
    AmpXOut(J) = 0.
    For I = 1 TO nPtsM1
      iAng = (I*J)  MOD nPts2
      AmpXOut(J) = AmpXOut(J) + G(I)*S(iAng)
    Next //I
    AmpXOut(J) = AmpXOut(J)/nPts
  Next //J
End Sub
```

WorkWindow.Close:

```
Sub Close()
  Quit
End Sub
```

WorkWindow.TransformBtn.Action:

```
Sub Action()
  doProcessX()
  doProcessY()

  if saveBinBtn.Value= True then
    doWriteBin()
  else
    doWriteText()
  end if
End Sub
```

WorkWindow.getFileBtn.Action:

```
Sub Action()
  Dim f As folderItem
  Dim dlg As openDialog

  dlg = new openDialog
  dlg.Filter =  "application/text"
  dlg.title =  "Find suture data file to rotate"
  f = dlg.showModal()
  f2Transform = f
```

```
  if f<>Nil and f.exists  then // valid file was selected
    TransformBtn.Enabled =   True
    sutFileLbl.text = dlg.result.Name
    me.Default =  False
    TransformBtn.Default =   True
  else    // no file selected
    // there's nothing to do
  end if
End Sub
```

WorkWindow.quitBtn.Action:

```
Sub Action()
  quit
End Sub
```

Menu (Item of Unknown Type)