

Source Code
LMRA.FRM
LMRA version 1.0

```

VERSION 5.00
Object = "{3B008041-905A-11D1-B4AE-444553540000}#1.0#0"; "Vsocx6.ocx"
Object = "{8099FCC2-0A81-11D2-BAA4-04F205C10000}#1.0#0"; "Vsflex6.ocx"
Object = "{827E9F53-96A4-11CF-823E-000021570103}#1.0#0"; "Graphs32.ocx"
Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0"; "Comdlg32.ocx"
Begin VB.Form frmMain
    Caption           = "Listeria monocytogenes risk assessment"
    ClientHeight     = 7500
    ClientLeft      = 165
    ClientTop       = 735
    ClientWidth     = 11400
    LinkTopic       = "Form1"
    ScaleHeight     = 7500
    ScaleWidth     = 11400
    StartupPosition = 3 'Windows Default
    Begin vsOcx6LibCtl.vsIndexTab tabLMRAMain
        Height       = 7095
        Left         = 0
        TabIndex     = 0
        Top         = 240
        Width       = 11295
        _ExtentX    = 19923
        _ExtentY    = 12515
        BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
            Name      = "MS Sans Serif"
            Size     = 8.25
            Charset  = 0
            Weight   = 400
            Underline = 0 'False
            Italic   = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Enabled      = -1 'True
        Appearance  = 1
        MousePointer = 0
        _ConvInfo   = 1
        Version     = 600
        BackColor   = -2147483633
        ForeColor   = -2147483630
        FrontTabColor = -2147483633
        BackTabColor = -2147483633
        TabOutlineColor = 0
        FrontTabForeColor = -2147483630
        Caption     = "&Project|P&lant Data|&Contamination
Data|P&ost-Processing Data|&Advanced Data|&Simulation|&Graphs|&Output
Stats"
        Align      = 0
        Appearance = 1
        CurrTab    = 4
        FirstTab   = 0
        Style      = 3
        Position   = 0
        AutoSwitch = -1 'True
        AutoScroll = -1 'True
        TabPreview = -1 'True
        ShowFocusRect = -1 'True
        TabsPerPage = 0
    End
End

```

```

BorderWidth      = 0
BoldCurrent      = 0  'False
DogEars          = -1  'True
MultiRow         = 0  'False
MultiRowOffset  = 200
CaptionStyle     = 0
TabHeight        = 0
Begin vsOcx6LibCtl.vsElastic elaPostProcessTab
  Height         = 6720
  Left           = -11850
  TabIndex       = 110
  TabStop        = 0  'False
  Top            = 330
  Width          = 11205
  _ExtentX      = 19764
  _ExtentY      = 11853
  BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name         = "MS Sans Serif"
    Size         = 8.25
    Charset      = 0
    Weight       = 400
    Underline    = 0  'False
    Italic       = 0  'False
    Strikethrough = 0  'False
  EndProperty
  Enabled        = -1  'True
  Appearance     = 1
  MousePointer   = 0
  _ConvInfo     = 1
  Version       = 600
  BackColor     = -2147483633
  ForeColor     = -2147483630
  FloodColor    = 192
  ForeColorDisabled = -2147483631
  Caption       = ""
  Align         = 0
  Appearance    = 1
  AutoSizeChildren = 0
  BorderWidth  = 6
  ChildSpacing = 4
  Splitter     = 0  'False
  FloodDirection = 0
  FloodPercent = 0
  CaptionPos   = 1
  WordWrap     = -1  'True
  MaxChildSize = 0
  MinChildSize = 0
  TagWidth     = 0
  TagPosition  = 0
  Style        = 0
  TagSplit     = 0  'False
  PicturePos   = 4
  CaptionStyle = 0
  ResizeFonts = 0  'False
  GridRows     = 0
  GridCols     = 0
  _GridInfo    = ""

```

```

Begin VB.Frame fraPPPackage
  Caption      =   "Growth Inhibiting Packaging"
  BeginProperty Font
    Name        =   "MS Sans Serif"
    Size        =   8.25
    Charset     =   0
    Weight      =   700
    Underline   =   0   'False
    Italic      =   0   'False
    Strikethrough = 0   'False
  EndProperty
  ForeColor    =   &H00C00000&
  Height       =   2655
  Left         =   2400
  TabIndex     =   128
  Top          =   3360
  Width        =   6015
  Begin VB.TextBox txtGIPEfficMax
    Height      =   285
    Left        =   4560
    TabIndex    =   140
    Top         =   1560
    Width       =   735
  End
  Begin VB.TextBox txtGIPEfficMin
    Height      =   285
    Left        =   4560
    TabIndex    =   139
    Top         =   1080
    Width       =   735
  End
  Begin VB.TextBox txtGIPFracVerySmall
    Height      =   285
    Left        =   1680
    TabIndex    =   135
    Top         =   1920
    Width       =   735
  End
  Begin VB.TextBox txtGIPFracSmall
    Height      =   285
    Left        =   1680
    TabIndex    =   134
    Top         =   1440
    Width       =   735
  End
  Begin VB.TextBox txtGIPFracLarge
    Height      =   285
    Left        =   1680
    TabIndex    =   133
    Top         =   960
    Width       =   735
  End
  Begin VB.Line Line7
    X1          =   3360
    X2          =   5640
    Y1          =   840
    Y2          =   840
  End

```

```

End
Begin VB.Label lblGIPEffectiveMax
    Caption      =   "Maximum:"
    Height       =   255
    Left        =   3480
    TabIndex    =   138
    Top         =   1560
    Width       =   855
End
Begin VB.Label lblGIPEffic
    Caption      =   "Fraction Efficiency (uniform)"
    Height       =   255
    Left        =   3480
    TabIndex    =   137
    Top         =   480
    Width       =   2295
End
Begin VB.Line Line6
    X1          =   1680
    X2          =   2760
    Y1          =   840
    Y2          =   840
End
Begin VB.Label lblGIPEffectiveMin
    Caption      =   "Minimum:"
    Height       =   255
    Left        =   3480
    TabIndex    =   136
    Top         =   1080
    Width       =   855
End
Begin VB.Label lblGIPFrac
    Caption      =   "Fraction of Plants Applying"
    Height       =   495
    Left        =   1680
    TabIndex    =   132
    Top         =   360
    Width       =   1095
End
Begin VB.Label lblGIPVerySmall
    Caption      =   "Very Small:"
    Height       =   255
    Left        =   360
    TabIndex    =   131
    Top         =   1920
    Width       =   975
End
Begin VB.Label lblGIPSmall
    Caption      =   "Small:"
    Height       =   255
    Left        =   360
    TabIndex    =   130
    Top         =   1440
    Width       =   735
End
Begin VB.Label lblGIPLarge
    Caption      =   "Large:"

```

```

        Height      = 255
        Left        = 360
        TabIndex    = 129
        Top         = 960
        Width       = 615
    End
End
Begin VB.Frame fraPostProcessing
    Caption      = "Post Processing Treatment"
    BeginProperty Font
        Name      = "MS Sans Serif"
        Size      = 8.25
        Charset   = 0
        Weight    = 700
        Underline = 0   'False
        Italic    = 0   'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor    = &H00C00000&
    Height       = 2895
    Left        = 2400
    TabIndex    = 111
    Top         = 240
    Width       = 6015
    Begin VB.TextBox txtPPMaxReductVerySmall
        Height      = 285
        Left        = 4680
        TabIndex    = 127
        Top         = 2280
        Width       = 735
    End
    Begin VB.TextBox txtPPMinReductVerySmall
        Height      = 285
        Left        = 3600
        TabIndex    = 126
        Top         = 2280
        Width       = 735
    End
    End
    Begin VB.TextBox txtPPMaxReductSmall
        Height      = 285
        Left        = 4680
        TabIndex    = 125
        Top         = 1800
        Width       = 735
    End
    End
    Begin VB.TextBox txtPPMinReductSmall
        Height      = 285
        Left        = 3600
        TabIndex    = 124
        Top         = 1800
        Width       = 735
    End
    End
    Begin VB.TextBox txtPPMaxReductLarge
        Height      = 285
        Left        = 4680
        TabIndex    = 123
        Top         = 1320
    End

```

```

        Width          = 735
    End
Begin VB.TextBox txtPPMinReductLarge
    Height          = 285
    Left           = 3600
    TabIndex       = 122
    Top            = 1320
    Width          = 735
End
Begin VB.TextBox txtPPFracVerySmall
    Height          = 285
    Left           = 1920
    TabIndex       = 121
    Top            = 2280
    Width          = 735
End
Begin VB.TextBox txtPPFracSmall
    Height          = 285
    Left           = 1920
    TabIndex       = 120
    Top            = 1800
    Width          = 735
End
Begin VB.TextBox txtPPFracLarge
    Height          = 285
    Left           = 1920
    TabIndex       = 119
    Top            = 1320
    Width          = 735
End
Begin VB.Line Line5
    X1              = 3480
    X2              = 5640
    Y1              = 600
    Y2              = 600
End
Begin VB.Line Line4
    X1              = 1560
    X2              = 5640
    Y1              = 1080
    Y2              = 1080
End
Begin VB.Label lblPPMax
    Caption         = "Maximum"
    Height          = 255
    Left           = 4680
    TabIndex       = 118
    Top            = 720
    Width          = 855
End
Begin VB.Label lblPPMin
    Caption         = "Minimum"
    Height          = 255
    Left           = 3600
    TabIndex       = 117
    Top            = 720
    Width          = 855

```

```

End
Begin VB.Label lblPPUniform
    Caption      = "Reduction in LM (Uniform)"
    Height       = 255
    Left         = 3600
    TabIndex     = 116
    Top          = 240
    Width        = 1935
End
Begin VB.Label lblPPFrac
    Caption      = "Fraction of Plants Applying"
    Height       = 495
    Left         = 1800
    TabIndex     = 115
    Top          = 360
    Width        = 1095
End
Begin VB.Label lblPPVSmall
    Caption      = "Very Small:"
    Height       = 255
    Left         = 360
    TabIndex     = 114
    Top          = 2280
    Width        = 975
End
Begin VB.Label lblPPSmall
    Caption      = "Small:"
    Height       = 255
    Left         = 360
    TabIndex     = 113
    Top          = 1800
    Width        = 735
End
Begin VB.Label lblPPLarge
    Caption      = "Large:"
    Height       = 255
    Left         = 360
    TabIndex     = 112
    Top          = 1320
    Width        = 615
End
End
End
Begin vsOcx6LibCtl.vsElastic elaTab6
    Height       = 6720
    Left         = 12540
    TabIndex     = 32
    TabStop      = 0    'False
    Top          = 330
    Width        = 11205
    _ExtentX     = 19764
    _ExtentY     = 11853
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
        Name      = "MS Sans Serif"
        Size      = 8.25
        Charset   = 0
        Weight    = 400
    EndProperty

```



```

        Underline      = 0   'False
        Italic         = 0   'False
        Strikethrough  = 0   'False
    EndProperty
    Enabled           = -1   'True
    Appearance        = 1
    MousePointer      = 0
    _ConvInfo         = 1
    Version           = 600
    BackColor         = -2147483633
    ForeColor         = -2147483630
    FloodColor        = 192
    ForeColorDisabled= -2147483631
    Caption           = ""
    Align             = 0
    Appearance        = 1
    AutoSizeChildren = 0
    BorderWidth       = 6
    ChildSpacing      = 4
    Splitter          = 0   'False
    FloodDirection    = 0
    FloodPercent       = 0
    CaptionPos        = 1
    WordWrap          = -1   'True
    MaxChildSize      = 0
    MinChildSize      = 0
    TagWidth          = 0
    TagPosition       = 0
    Style             = 0
    TagSplit          = 0   'False
    PicturePos        = 4
    CaptionStyle      = 0
    ResizeFonts       = 0   'False
    GridRows          = 0
    GridCols          = 0
    _GridInfo         = ""
Begin VSFlex6Ctl.vsFlexGrid flxContingency
    Height            = 1095
    Left              = 2760
    TabIndex          = 152
    Top               = 3720
    Width             = 6255
    _cx               = 1190681
    _cy               = 1181579
    _ConvInfo         = 1
    Appearance        = 1
    BorderStyle       = 1
    Enabled           = -1   'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name              = "MS Sans Serif"
    Size              = 8.25
    Charset           = 0
    Weight            = 400
    Underline         = 0   'False
    Italic            = 0   'False
    Strikethrough     = 0   'False
EndProperty

```

```

MousePointer      = 0
BackColor         = -2147483643
ForeColor        = -2147483640
BackColorFixed   = -2147483633
ForeColorFixed   = -2147483630
BackColorSel     = -2147483635
ForeColorSel     = -2147483634
BackColorBkg     = -2147483636
BackColorAlternate= -2147483643
GridColor        = -2147483633
GridColorFixed   = -2147483632
TreeColor        = -2147483632
FloodColor       = 192
SheetBorder      = -2147483642
FocusRect        = 1
HighLight        = 1
AllowSelection   = -1 'True
AllowBigSelection= -1 'True
AllowUserResizing= 0
SelectionMode    = 0
GridLines        = 1
GridLinesFixed   = 2
GridLineWidth    = 1
Rows             = 4
Cols             = 4
FixedRows        = 1
FixedCols        = 1
RowHeightMin     = 0
RowHeightMax     = 0
ColWidthMin      = 0
ColWidthMax      = 0
ExtendLastCol    = 0 'False
FormatString     = ""
ScrollTrack      = 0 'False
ScrollBars       = 3
ScrollTips       = 0 'False
MergeCells       = 0
MergeCompare     = 0
AutoResize       = -1 'True
AutoSizeMode     = 0
AutoSearch       = 0
MultiTotals      = -1 'True
SubtotalPosition= 1
OutlineBar       = 0
OutlineCol       = 0
Ellipsis        = 0
ExplorerBar      = 0
PicturesOver     = 0 'False
FillStyle        = 0
RightToLeft      = 0 'False
PictureType      = 0
TabBehavior      = 0
OwnerDraw        = 0
Editable         = 0 'False
ShowComboButton = -1 'True
WordWrap         = 0 'False
TextStyle        = 0

```

```

        TextStyleFixed = 0
        OleDragMode = 0
        OleDropMode = 0
        DataMode = 0
        VirtualData = -1 'True
        DataMember = ""
End
Begin VSFlex6Ctl.vsFlexGrid flxOutputStats
    Height = 3255
    Left = 2760
    TabIndex = 102
    Top = 360
    Width = 5895
    _cx = 1190046
    _cy = 1185389
    _ConvInfo = 1
    Appearance = 1
    BorderStyle = 1
    Enabled = -1 'True
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
        Name = "MS Sans Serif"
        Size = 8.25
        Charset = 0
        Weight = 400
        Underline = 0 'False
        Italic = 0 'False
        Strikethrough = 0 'False
    EndProperty
    MousePointer = 0
    BackColor = -2147483643
    ForeColor = -2147483640
    BackColorFixed = -2147483633
    ForeColorFixed = -2147483630
    BackColorSel = -2147483635
    ForeColorSel = -2147483634
    BackColorBkg = -2147483636
    BackColorAlternate= -2147483643
    GridColor = -2147483633
    GridColorFixed = -2147483632
    TreeColor = -2147483632
    FloodColor = 192
    SheetBorder = -2147483642
    FocusRect = 1
    HighLight = 1
    AllowSelection = -1 'True
    AllowBigSelection= -1 'True
    AllowUserResizing= 0
    SelectionMode = 0
    GridLines = 1
    GridLinesFixed = 2
    GridLineWidth = 1
    Rows = 13
    Cols = 5
    FixedRows = 1
    FixedCols = 1
    RowHeightMin = 0
    RowHeightMax = 0

```

```

ColWidthMin      = 0
ColWidthMax      = 0
ExtendLastCol    = 0   'False
FormatString     = ""
ScrollTrack      = 0   'False
ScrollBars       = 3
ScrollTips       = 0   'False
MergeCells       = 0
MergeCompare     = 0
AutoResize       = -1  'True
AutoSizeMode     = 0
AutoSearch       = 0
MultiTotals      = -1  'True
SubtotalPosition= 1
OutlineBar       = 0
OutlineCol       = 0
Ellipsis         = 0
ExplorerBar      = 0
PicturesOver     = 0   'False
FillStyle        = 0
RightToLeft      = 0   'False
PictureType      = 0
TabBehavior      = 0
OwnerDraw        = 0
Editable         = 0   'False
ShowComboButton = -1  'True
WordWrap         = 0   'False
TextStyle        = 0
TextStyleFixed   = 0
OleDragMode      = 0
OleDropMode      = 0
DataMode         = 0
VirtualData      = -1  'True
DataMember       = ""
End
Begin VSFlex6Ctl.vsFlexGrid flxContingencyLag
Height           = 1095
Left             = 2760
TabIndex        = 183
Top             = 4920
Width           = 6255
_cx             = 1190681
_cy             = 1181579
_ConvInfo       = 1
Appearance      = 1
BorderStyle     = 1
Enabled         = -1  'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
  Name           = "MS Sans Serif"
  Size           = 8.25
  Charset        = 0
  Weight         = 400
  Underline      = 0   'False
  Italic         = 0   'False
  Strikethrough  = 0   'False
EndProperty
MousePointer     = 0

```

BackColor = -2147483643
ForeColor = -2147483640
BackColorFixed = -2147483633
ForeColorFixed = -2147483630
BackColorSel = -2147483635
ForeColorSel = -2147483634
BackColorBkg = -2147483636
BackColorAlternate= -2147483643
GridColor = -2147483633
GridColorFixed = -2147483632
TreeColor = -2147483632
FloodColor = 192
SheetBorder = -2147483642
FocusRect = 1
HighLight = 1
AllowSelection = -1 'True
AllowBigSelection= -1 'True
AllowUserResizing= 0
SelectionMode = 0
GridLines = 1
GridLinesFixed = 2
GridLineWidth = 1
Rows = 4
Cols = 4
FixedRows = 1
FixedCols = 1
RowHeightMin = 0
RowHeightMax = 0
ColWidthMin = 0
ColWidthMax = 0
ExtendLastCol = 0 'False
FormatString = ""
ScrollTrack = 0 'False
ScrollBars = 3
ScrollTips = 0 'False
MergeCells = 0
MergeCompare = 0
AutoResize = -1 'True
AutoSizeMode = 0
AutoSearch = 0
MultiTotals = -1 'True
SubtotalPosition= 1
OutlineBar = 0
OutlineCol = 0
Ellipsis = 0
ExplorerBar = 0
PicturesOver = 0 'False
FillStyle = 0
RightToLeft = 0 'False
PictureType = 0
TabBehavior = 0
OwnerDraw = 0
Editable = 0 'False
ShowComboButton = -1 'True
WordWrap = 0 'False
TextStyle = 0
TextStyleFixed = 0

```

        OleDragMode      = 0
        OleDropMode      = 0
        DataMode         = 0
        VirtualData      = -1 'True
        DataMember       = ""
    End
    Begin VB.Label lblContingencyLag
        Caption          = "Large Plant Results. Lagged tests."
        Height           = 495
        Left             = 1200
        TabIndex        = 184
        Top              = 4920
        Width            = 1455
    End
    Begin VB.Label lblContingencyTable
        Caption          = "Contingency Table"
        Height           = 255
        Left             = 1200
        TabIndex        = 154
        Top              = 3720
        Width            = 1455
    End
    Begin VB.Label lblContingency
        Caption          = "Large Plant Results. Simulataneous
tests."
        Height           = 495
        Left             = 1200
        TabIndex        = 153
        Top              = 4080
        Width            = 1575
    End
End
Begin vsOcx6LibCtl.vsElastic ela5Graphs
    Height             = 6720
    Left              = 12240
    TabIndex          = 31
    TabStop           = 0 'False
    Top               = 330
    Width             = 11205
    _ExtentX          = 19764
    _ExtentY          = 11853
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
        Name           = "MS Sans Serif"
        Size           = 8.25
        Charset        = 0
        Weight         = 400
        Underline      = 0 'False
        Italic         = 0 'False
        Strikethrough  = 0 'False
    EndProperty
    Enabled           = -1 'True
    Appearance        = 1
    MousePointer      = 0
    _ConvInfo         = 1
    Version           = 600
    BackColor         = -2147483633
    ForeColor         = -2147483630

```

```

FloodColor      = 192
ForeColorDisabled= -2147483631
Caption         = ""
Align           = 0
Appearance      = 1
AutoSizeChildren= 0
BorderWidth     = 6
ChildSpacing    = 4
Splitter        = 0 'False
FloodDirection = 0
FloodPercent    = 0
CaptionPos      = 1
WordWrap        = -1 'True
MaxChildSize    = 0
MinChildSize    = 0
TagWidth        = 0
TagPosition     = 0
Style           = 0
TagSplit        = 0 'False
PicturePos      = 4
CaptionStyle    = 0
ResizeFonts     = 0 'False
GridRows        = 0
GridCols        = 0
_GridInfo       = ""
Begin VB.Frame fraGraphSelect
    Caption      = "Graph Selection"
    BeginProperty Font
        Name      = "MS Sans Serif"
        Size      = 8.25
        Charset   = 0
        Weight    = 700
        Underline = 0 'False
        Italic    = 0 'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor    = &H00C00000&
    Height       = 1560
    Left         = 360
    TabIndex     = 85
    Top          = 1680
    Width        = 2325
    Begin VB.OptionButton optGraphLsppFCS
        Caption      = "Lspp concentration on FCS (large)"
        Height       = 375
        Left         = 240
        TabIndex     = 87
        Top          = 960
        Width        = 2055
    End
    Begin VB.OptionButton optGraphLMRetail
        Caption      = "LM concentration in RTE product."
        Height       = 375
        Left         = 240
        TabIndex     = 86
        Top          = 360
        Width        = 2055
    End
End

```

```

    End
End
Begin GraphsLib.Graph grfOutput
    Height      = 5295
    Left        = 3360
    TabIndex    = 88
    Top         = 360
    Width       = 6495
    _Version    = 327680
    _ExtentX    = 11456
    _ExtentY    = 9340
    _StockProps = 96
    BorderStyle = 1
    Background  = "15~-1~-1~-1~-1~-1~-1"
    GraphType   = 3
End
End
Begin vsOcx6LibCtl.vsElastic elaSimulation
    Height      = 6720
    Left        = 11940
    TabIndex    = 30
    TabStop     = 0    'False
    Top         = 330
    Width       = 11205
    _ExtentX    = 19764
    _ExtentY    = 11853
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
        Name      = "MS Sans Serif"
        Size      = 8.25
        Charset    = 0
        Weight    = 400
        Underline  = 0    'False
        Italic     = 0    'False
        Strikethrough = 0    'False
    EndProperty
    Enabled     = -1    'True
    Appearance  = 1
    MousePointer = 0
    _ConvInfo   = 1
    Version     = 600
    BackColor   = -2147483633
    ForeColor   = -2147483630
    FloodColor  = 192
    ForeColorDisabled = -2147483631
    Caption     = ""
    Align       = 0
    Appearance  = 1
    AutoSizeChildren = 0
    BorderWidth = 6
    ChildSpacing = 4
    Splitter    = 0    'False
    FloodDirection = 0
    FloodPercent = 0
    CaptionPos  = 1
    WordWrap    = -1    'True
    MaxChildSize = 0
    MinChildSize = 0

```



```

TagWidth      = 0
TagPosition   = 0
Style         = 0
TagSplit      = 0 'False
PicturePos    = 4
CaptionStyle  = 0
ResizeFonts   = 0 'False
GridRows     = 0
GridCols     = 0
_GridInfo    = ""
Begin VB.Frame fraCalibrationChoice
    Caption    = "Calibration Choice"
    Height     = 1335
    Left       = 960
    TabIndex   = 166
    Top        = 2760
    Width      = 2775
    Begin VB.OptionButton optCalibrateFrankfurters
        Caption    = "Frankfurters"
        Height     = 255
        Left       = 360
        TabIndex   = 168
        Top        = 840
        Width      = 1575
    End
    Begin VB.OptionButton optCalibrateDeliMeats
        Caption    = "RTE Deli Meats"
        Height     = 255
        Left       = 360
        TabIndex   = 167
        Top        = 360
        Width      = 1695
    End
End
Begin VB.Frame fraCalibration
    Caption    = "Calibration Basics - Retail LM/g"
    Height     = 3015
    Left       = 3960
    TabIndex   = 148
    Top        = 2760
    Width      = 5295
    Begin VSFlex6Ctl.vsFlexGrid flxCalibration
        Height     = 3015
        Left       = 0
        TabIndex   = 149
        Top        = 360
        Width      = 5415
        _cx        = 1189199
        _cy        = 1184966
        _ConvInfo  = 1
        Appearance = 1
        BorderStyle = 1
        Enabled    = -1 'True
        BeginProperty Font {0BE35203-8F91-11CE-9DE3-
00AA004BB851}
            Name      = "MS Sans Serif"
            Size     = 8.25

```

```

        Charset          = 0
        Weight           = 400
        Underline        = 0   'False
        Italic           = 0   'False
        Strikethrough    = 0   'False
EndProperty
MousePointer          = 0
BackColor             = -2147483643
ForeColor             = -2147483640
BackColorFixed       = -2147483633
ForeColorFixed       = -2147483630
BackColorSel          = -2147483635
ForeColorSel          = -2147483634
BackColorBkg         = -2147483636
BackColorAlternate= -2147483643
GridColor             = -2147483633
GridColorFixed       = -2147483632
TreeColor             = -2147483632
FloodColor           = 192
SheetBorder           = -2147483642
FocusRect             = 1
HighLight             = 1
AllowSelection        = -1   'True
AllowBigSelection= -1   'True
AllowUserResizing= 0
SelectionMode         = 0
GridLines             = 1
GridLinesFixed       = 2
GridLineWidth        = 1
Rows                  = 10
Cols                  = 4
FixedRows             = 1
FixedCols             = 1
RowHeightMin         = 0
RowHeightMax         = 0
ColWidthMin          = 0
ColWidthMax          = 0
ExtendLastCol        = 0   'False
FormatString          = ""
ScrollTrack           = 0   'False
ScrollBars            = 3
ScrollTips            = 0   'False
MergeCells           = 0
MergeCompare          = 0
AutoResize            = -1   'True
AutoSizeMode          = 0
AutoSearch            = 0
MultiTotals           = -1   'True
SubtotalPosition= 1
OutlineBar            = 0
OutlineCol            = 0
Ellipsis              = 0
ExplorerBar           = 0
PicturesOver          = 0   'False
FillStyle             = 0
RightToLeft           = 0   'False
PictureType           = 0

```

```

        TabBehavior      = 0
        OwnerDraw        = 0
        Editable         = 0   'False
        ShowComboButton = -1  'True
        WordWrap         = 0   'False
        TextStyle        = 0
        TextStyleFixed   = 0
        OleDragMode      = 0
        OleDropMode      = 0
        DataMode         = 0
        VirtualData      = -1  'True
        DataMember       = ""
    End
End
Begin VB.Frame frmSimulationData
    Caption           = "Simulation Data"
    BeginProperty Font
        Name           = "MS Sans Serif"
        Size           = 8.25
        Charset        = 0
        Weight         = 700
        Underline      = 0   'False
        Italic         = 0   'False
        Strikethrough  = 0   'False
    EndProperty
    ForeColor         = &H00C00000&
    Height            = 1335
    Left              = 480
    TabIndex          = 89
    Top               = 840
    Width             = 4695
    Begin VB.TextBox txtNLots
        Height         = 285
        Left           = 1920
        TabIndex       = 91
        Top            = 480
        Width          = 975
    End
    Begin VB.TextBox txtOutputFile
        Height         = 285
        Left           = 1920
        TabIndex       = 90
        Top            = 840
        Width          = 2295
    End
    End
    Begin VB.Label lblNLots
        Caption        = "No. Lots to Simulate:"
        Height         = 255
        Left           = 240
        TabIndex       = 93
        Top            = 480
        Width          = 1575
    End
    End
    Begin VB.Label lblOutputFile
        Caption        = "Output file: "
        Height         = 255
        Left           = 240

```

```

        TabIndex      = 92
        Top           = 840
        Width        = 975
    End
End
Begin VB.CommandButton cmdRunModel
    Caption          = "&Run Model"
    Height          = 615
    Left            = 5520
    TabIndex        = 29
    ToolTipText     = "Click button to start the model."
    Top            = 1200
    Width          = 4695
End
Begin VB.Label lblLogSSR
    BorderStyle     = 1 'Fixed Single
    Height          = 255
    Left           = 2160
    TabIndex       = 170
    Top           = 4560
    Width         = 1455
End
Begin VB.Label lblLogSSRcaption
    Caption          = "Log SSR:"
    Height          = 375
    Left           = 960
    TabIndex       = 169
    Top           = 4560
    Width         = 855
End
End
End
Begin vsOcx6LibCtl.vsElastic ela4AdvancedData
    Height          = 6720
    Left           = 45
    TabIndex       = 28
    TabStop        = 0 'False
    Top           = 330
    Width         = 11205
    _ExtentX       = 19764
    _ExtentY       = 11853
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
        Name          = "MS Sans Serif"
        Size          = 8.25
        Charset       = 0
        Weight        = 400
        Underline     = 0 'False
        Italic        = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Enabled        = -1 'True
    Appearance     = 1
    MousePointer   = 0
    _ConvInfo      = 1
    Version        = 600
    BackColor      = -2147483633
    ForeColor      = -2147483630
    FloodColor     = 192

```

```

ForeColorDisabled= -2147483631
Caption           = ""
Align            = 0
Appearance       = 1
AutoSizeChildren= 0
BorderWidth      = 6
ChildSpacing     = 4
Splitter         = 0 'False
FloodDirection   = 0
FloodPercent     = 0
CaptionPos       = 1
WordWrap         = -1 'True
MaxChildSize    = 0
MinChildSize    = 0
TagWidth        = 0
TagPosition     = 0
Style           = 0
TagSplit        = 0 'False
PicturePos      = 4
CaptionStyle    = 0
ResizeFonts     = 0 'False
GridRows       = 0
GridCols       = 0
_GridInfo      = ""
Begin VB.Frame fraPostRetailGrowth
    Caption      = "Post Processing Growth"
    BeginProperty Font
        Name      = "MS Sans Serif"
        Size      = 8.25
        Charset   = 0
        Weight    = 700
        Underline = 0 'False
        Italic    = 0 'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor    = &H00C00000&
    Height       = 1215
    Left         = 5520
    TabIndex     = 141
    Top          = 3600
    Width        = 4455
    Begin VB.TextBox txtGrowth
        Height     = 285
        Left       = 3120
        TabIndex   = 172
        Top        = 480
        Width      = 975
    End
    Begin VB.Label lblGrowth
        Caption    = "Growth factor (log scale)"
        Height     = 255
        Left       = 480
        TabIndex   = 171
        Top        = 480
        Width      = 2055
    End
End
End

```

```

Begin VB.Frame fraRatio
  Caption           =   "Lm to Lspp Ratio (Normal)"
  BeginProperty Font
    Name             =   "MS Sans Serif"
    Size             =   8.25
    Charset          =   0
    Weight           =   700
    Underline        =   0   'False
    Italic           =   0   'False
    Strikethrough    =   0   'False
  EndProperty
  ForeColor         =   &H00C00000&
  Height            =   1335
  Left              =   240
  TabIndex          =   60
  Top               =   3600
  Width             =   4335
  Begin VB.TextBox txtRatioMean
    Height           =   285
    Left             =   3120
    TabIndex         =   62
    Top              =   360
    Width            =   975
  End
  Begin VB.TextBox txtRatioStdDev
    Height           =   285
    Left             =   3120
    TabIndex         =   61
    Top              =   840
    Width            =   975
  End
  End
  Begin VB.Label lblRatioMean
    Caption          =   "Mean Ratio:"
    Height           =   255
    Left             =   600
    TabIndex         =   64
    Top              =   360
    Width            =   1455
  End
  End
  Begin VB.Label Label4
    Caption          =   "Std Dev Ratio:"
    Height           =   375
    Left             =   600
    TabIndex         =   63
    Top              =   840
    Width            =   1455
  End
  End
End
Begin VB.Frame fraFCSArea
  Caption           =   "Food Contact Surface Area (Uniform)"
  BeginProperty Font
    Name             =   "MS Sans Serif"
    Size             =   8.25
    Charset          =   0
    Weight           =   700
    Underline        =   0   'False
    Italic           =   0   'False

```

```

        Strikethrough = 0 'False
    EndProperty
    ForeColor = &H00C00000&
    Height = 2295
    Left = 5520
    TabIndex = 55
    Top = 960
    Width = 4335
Begin VB.TextBox txtFCSAreaLargeMin
    Height = 285
    Left = 3120
    TabIndex = 57
    Top = 720
    Width = 975
End
Begin VB.TextBox txtFCSAreaLargeMax
    Height = 285
    Left = 3120
    TabIndex = 56
    Top = 1200
    Width = 975
End
Begin VB.Label lblFCSAreaLarge
    Caption = "Large Plants"
    Height = 255
    Left = 3120
    TabIndex = 182
    Top = 360
    Width = 975
End
Begin VB.Label lblFCSAreaNote
    Caption = "Area for small and very small
plants assumed proportional based on lbs/lot."
    Height = 375
    Left = 480
    TabIndex = 181
    Top = 1680
    Width = 3135
End
Begin VB.Label lblFCSAreaMin
    Caption = "Min FCS Area (cm^2): "
    Height = 255
    Left = 600
    TabIndex = 59
    Top = 720
    Width = 1935
End
Begin VB.Label lblFCSAreaMax
    Caption = "Max FCS Area (cm^2):"
    Height = 255
    Left = 600
    TabIndex = 58
    Top = 1200
    Width = 1815
End
End
End
Begin VB.Frame fraDetectionLimits

```

```

Caption          = "Testing and Detection Limits"
BeginProperty Font
  Name           = "MS Sans Serif"
  Size           = 8.25
  Charset        = 0
  Weight         = 700
  Underline      = 0 'False
  Italic         = 0 'False
  Strikethrough  = 0 'False
EndProperty
ForeColor        = &H00C00000&
Height           = 2415
Left             = 240
TabIndex        = 14
Top              = 840
Width            = 4335
Begin VB.TextBox txtLagReportLot
  Height         = 285
  Left           = 3120
  TabIndex       = 34
  Top            = 1800
  Width          = 975
End
Begin VB.TextBox txtLagReportFCS
  Height         = 285
  Left           = 3120
  TabIndex       = 26
  Top            = 1320
  Width          = 975
End
Begin VB.TextBox txtPDetectOneLot
  Height         = 285
  Left           = 3120
  TabIndex       = 16
  Top            = 840
  Width          = 975
End
Begin VB.TextBox txtPDetectOneFCS
  Height         = 285
  Left           = 3120
  TabIndex       = 15
  Top            = 360
  Width          = 975
End
Begin VB.Label lblLotReportLag
  Caption        = "Product Testing Report Lag (d):"
  Height         = 255
  Left           = 600
  TabIndex       = 33
  Top            = 1800
  Width          = 2415
End
Begin VB.Label lblReportLag
  Caption        = "FCS Testing Report Lag (d):"
  Height         = 255
  Left           = 600
  TabIndex       = 25

```



```

        Top          = 1320
        Width        = 2055
    End
    Begin VB.Label lblPDetectOneFCS
        Caption       = "Probability of detecting 1 Lspp cfu
in FCS test:"
        Height        = 495
        Left          = 600
        TabIndex      = 18
        Top           = 360
        Width         = 1695
    End
    Begin VB.Label lblPDetectOneLot
        Caption       = "Probability of detecting 1 LM cfu
in product test:"
        Height        = 375
        Left          = 600
        TabIndex      = 17
        Top           = 840
        Width         = 1695
    End
    End
    Begin VB.Label lblCaution
        Caption       = "Caution - These parameters should
generally not be changed."
        BeginProperty Font
            Name       = "MS Sans Serif"
            Size       = 13.5
            Charset    = 0
            Weight     = 400
            Underline  = 0 'False
            Italic     = 0 'False
            Strikethrough = 0 'False
        EndProperty
        ForeColor     = &H000000C0&
        Height        = 495
        Left          = 1200
        TabIndex      = 142
        Top           = 240
        Width         = 7935
    End
    End
    Begin vsOcx6LibCtl.vsElastic ela3ContaminationData
        Height        = 6720
        Left          = -12150
        TabIndex      = 13
        TabStop       = 0 'False
        Top           = 330
        Width         = 11205
        _ExtentX      = 19764
        _ExtentY      = 11853
        BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
            Name       = "MS Sans Serif"
            Size       = 8.25
            Charset    = 0
            Weight     = 400
            Underline  = 0 'False
        EndProperty
    End

```

```

        Italic          = 0   'False
        Strikethrough   = 0   'False
EndProperty
Enabled              = -1   'True
Appearance          = 1
MousePointer        = 0
_ConvInfo           = 1
Version             = 600
BackColor           = -2147483633
ForeColor           = -2147483630
FloodColor          = 192
ForeColorDisabled= -2147483631
Caption             = ""
Align               = 0
Appearance          = 1
AutoSizeChildren= 0
BorderWidth         = 6
ChildSpacing        = 4
Splitter            = 0   'False
FloodDirection      = 0
FloodPercent        = 0
CaptionPos          = 1
WordWrap            = -1   'True
MaxChildSize        = 0
MinChildSize        = 0
TagWidth            = 0
TagPosition         = 0
Style               = 0
TagSplit            = 0   'False
PicturePos          = 4
CaptionStyle        = 0
ResizeFonts         = 0   'False
GridRows            = 0
GridCols            = 0
_GridInfo           = ""
Begin VB.Frame fraRTESampled
    Caption           = "RTE Sampled Mass (Uniform)"
    BeginProperty Font
        Name          = "MS Sans Serif"
        Size          = 8.25
        Charset       = 0
        Weight        = 700
        Underline     = 0   'False
        Italic        = 0   'False
        Strikethrough = 0   'False
    EndProperty
    ForeColor         = &H00C00000&
    Height            = 1455
    Left              = 5520
    TabIndex          = 143
    Top               = 4800
    Width             = 4695
    Begin VB.TextBox txtRTESampledMax
        Height        = 285
        Left          = 2880
        TabIndex      = 147
        Top           = 840

```

```

        Width          = 975
    End
    Begin VB.TextBox txtRTESampledMin
        Height          = 285
        Left            = 2880
        TabIndex       = 146
        Top            = 360
        Width          = 975
    End
    Begin VB.Label lblMaxRTESampled
        Caption         = "Max RTE Mass Sampled (g)"
        Height          = 255
        Left            = 360
        TabIndex       = 145
        Top            = 840
        Width          = 2055
    End
    Begin VB.Label lblminRTESampled
        Caption         = "Min RTE Mass Sampled (g)"
        Height          = 255
        Left            = 360
        TabIndex       = 144
        Top            = 360
        Width          = 2055
    End
End
Begin VB.Frame fraFCSSampled
    Caption           = "FCS Tested Area (Uniform)"
    BeginProperty Font
        Name           = "MS Sans Serif"
        Size           = 8.25
        Charset        = 0
        Weight         = 700
        Underline      = 0   'False
        Italic         = 0   'False
        Strikethrough  = 0   'False
    EndProperty
    ForeColor         = &H00C00000&
    Height            = 2055
    Left              = 5520
    TabIndex         = 103
    Top               = 2400
    Width             = 4695
    Begin VB.TextBox txtFCSSwabsNComposited
        Height          = 285
        Left            = 2880
        TabIndex       = 109
        Top            = 1680
        Width          = 975
    End
    Begin VB.TextBox txtFCSSwabMax
        Height          = 285
        Left            = 2880
        TabIndex       = 108
        Top            = 1080
        Width          = 975
    End
End

```

```

Begin VB.TextBox txtFCSSwabMin
    Height      = 285
    Left        = 2880
    TabIndex    = 107
    Top         = 480
    Width       = 975
End
Begin VB.Label nSwabsComposited
    Caption     = "Number of Swabs composited per
sample:"
    Height      = 375
    Left        = 360
    TabIndex    = 106
    Top         = 1560
    Width       = 1815
End
Begin VB.Label maxFCSSwab
    Caption     = "Max FCS swabbed per test (cm^2):"
    Height      = 375
    Left        = 360
    TabIndex    = 105
    Top         = 960
    Width       = 2055
End
Begin VB.Label minFCSSwab
    Caption     = "Min FCS swabbed per test (cm^2):"
    Height      = 375
    Left        = 360
    TabIndex    = 104
    Top         = 360
    Width       = 2055
End
End
Begin VB.Frame fraDuration
    Caption     = "Contamination Event Duration (Normal
log scale)"
    BeginProperty Font
        Name      = "MS Sans Serif"
        Size      = 8.25
        Charset   = 0
        Weight    = 700
        Underline = 0 'False
        Italic    = 0 'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor   = &H00C00000&
    Height      = 2055
    Left        = 480
    TabIndex    = 50
    Top         = 2400
    Width       = 4815
    Begin VB.TextBox txtDurationMean
        Height      = 285
        Left        = 3480
        TabIndex    = 52
        Top         = 360
        Width       = 975
    End

```

```

End
Begin VB.TextBox txtDurationStdDev
    Height      = 285
    Left        = 3480
    TabIndex    = 51
    Top         = 1200
    Width       = 975
End
Begin VB.Label lblDurationMean
    Caption     = "Mean Contamintion Event Duration
(log10 d):"
    Height      = 495
    Left        = 120
    TabIndex    = 54
    Top         = 360
    Width       = 3135
End
Begin VB.Label lblDurationStdDev
    Caption     = "Std Dev Contamination Event
Duration (log10 d):"
    Height      = 495
    Left        = 120
    TabIndex    = 53
    Top         = 1200
    Width       = 3015
End
End
Begin VB.Frame fraTC
    Caption     = "Transfer Coefficients (Normal Log
scale)"
    BeginProperty Font
        Name      = "MS Sans Serif"
        Size      = 8.25
        Charset    = 0
        Weight     = 700
        Underline  = 0   'False
        Italic     = 0   'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor   = &H00C00000&
    Height      = 1575
    Left        = 5520
    TabIndex    = 45
    Top         = 480
    Width       = 4695
    Begin VB.TextBox txtTCStdDev
        Height      = 285
        Left        = 2880
        TabIndex    = 48
        Top         = 960
        Width       = 975
    End
    Begin VB.TextBox txtTCMean
        Height      = 285
        Left        = 2880
        TabIndex    = 46
        Top         = 360
    End

```

```

        Width          = 975
    End
    Begin VB.Label lblTCStdDev
        Caption          = "Std Dev Transfer Coef (log10
fraction/lot): "
        Height           = 375
        Left              = 240
        TabIndex         = 49
        Top               = 960
        Width            = 2415
    End
    Begin VB.Label lblTCMean
        Caption          = "Mean Transfer Coef (log10
fraction/lot): "
        Height           = 495
        Left              = 240
        TabIndex         = 47
        Top               = 360
        Width            = 2415
    End
End
Begin VB.Frame fraCELevels
Caption          = "Contamination Event Levels (Normal log
scale)"
    BeginProperty Font
        Name             = "MS Sans Serif"
        Size             = 8.25
        Charset          = 0
        Weight           = 700
        Underline        = 0 'False
        Italic           = 0 'False
        Strikethrough    = 0 'False
    EndProperty
    ForeColor         = &H00C00000&
    Height            = 1455
    Left              = 480
    TabIndex         = 40
    Top               = 4800
    Width            = 4815
    Begin VB.TextBox txtCEAddStdDev
        Height           = 285
        Left              = 3480
        TabIndex         = 42
        Top               = 840
        Width            = 975
    End
    Begin VB.TextBox txtCEAddMean
        Height           = 285
        Left              = 3480
        TabIndex         = 41
        Top               = 360
        Width            = 975
    End
    End
    Begin VB.Label lblLogStdDev
        Caption          = "Std Dev for Levels (log10
cfu/cm^2):"
        Height           = 255

```

```

        Left          = 240
        TabIndex     = 44
        Top          = 840
        Width        = 3255
    End
    Begin VB.Label lblLogMean
        Caption       = "Mean Levels (log10 cfu/cm^2): "
        Height        = 255
        Left          = 240
        TabIndex     = 43
        Top          = 360
        Width        = 3135
    End
End
Begin VB.Frame fraCE
Caption       = "Contamination Event Timing (Normal
log scale)"
    BeginProperty Font
        Name          = "MS Sans Serif"
        Size          = 8.25
        Charset       = 0
        Weight        = 700
        Underline     = 0 'False
        Italic        = 0 'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor       = &H00C00000&
    Height          = 1575
    Left           = 480
    TabIndex       = 35
    Top            = 480
    Width          = 4815
    Begin VB.TextBox txtCEStdDev
        Height        = 285
        Left          = 3480
        TabIndex     = 39
        Top          = 960
        Width        = 975
    End
    Begin VB.TextBox txtCEMean
        Height        = 285
        Left          = 3480
        TabIndex     = 37
        Top          = 360
        Width        = 975
    End
    End
    Begin VB.Label lblCEStdDev
Caption       = "Std Dev forTime btw Contamination
Events (log10 d):"
        Height        = 495
        Left          = 120
        TabIndex     = 38
        Top          = 960
        Width        = 3015
    End
    End
    Begin VB.Label lblLambda

```

```

                Caption          = "Mean Time btw Contamination Events
(log10 d):"
                Height           = 495
                Left              = 120
                TabIndex          = 36
                Top               = 360
                Width             = 3135
            End
        End
    End
Begin vsOcx6LibCtl.vsElastic ela2PlantData
    Height           = 6720
    Left             = -12450
    TabIndex         = 2
    TabStop          = 0 'False
    Top              = 330
    Width           = 11205
    _ExtentX        = 19764
    _ExtentY        = 11853
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
        Name         = "MS Sans Serif"
        Size         = 8.25
        Charset      = 0
        Weight       = 400
        Underline    = 0 'False
        Italic       = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Enabled          = -1 'True
    Appearance       = 1
    MousePointer     = 0
    _ConvInfo        = 1
    Version          = 600
    BackColor        = -2147483633
    ForeColor        = -2147483630
    FloodColor       = 192
    ForeColorDisabled= -2147483631
    Caption          = ""
    Align            = 0
    Appearance       = 1
    AutoSizeChildren= 0
    BorderWidth      = 6
    ChildSpacing     = 4
    Splitter         = 0 'False
    FloodDirection   = 0
    FloodPercent     = 0
    CaptionPos       = 1
    WordWrap         = -1 'True
    MaxChildSize    = 0
    MinChildSize    = 0
    TagWidth         = 0
    TagPosition      = 0
    Style            = 0
    TagSplit         = 0 'False
    PicturePos       = 4
    CaptionStyle     = 0
    ResizeFonts     = 0 'False

```



```

GridRows      = 0
GridCols      = 0
_GridInfo     = ""
Begin VB.Frame fraPlantSize
    Caption     = "Plant Size Distribution"
    BeginProperty Font
        Name     = "MS Sans Serif"
        Size     = 8.25
        Charset  = 0
        Weight   = 700
        Underline = 0 'False
        Italic   = 0 'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor   = &H00C00000&
    Height      = 2415
    Left        = 480
    TabIndex    = 94
    Top         = 360
    Width       = 4695
    Begin VB.TextBox txtLotMassSDVerySmall
        Height    = 285
        Left      = 3600
        TabIndex  = 178
        Top       = 1800
        Width     = 735
    End
    Begin VB.TextBox txtLotMassSDSmall
        Height    = 285
        Left      = 3600
        TabIndex  = 177
        Top       = 1320
        Width     = 735
    End
    Begin VB.TextBox txtLotMassSDLarge
        Height    = 285
        Left      = 3600
        TabIndex  = 176
        Top       = 840
        Width     = 735
    End
    Begin VB.TextBox txtLotMassMeanVerySmall
        Height    = 285
        Left      = 2520
        TabIndex  = 175
        Top       = 1800
        Width     = 735
    End
    Begin VB.TextBox txtLotMassMeanSmall
        Height    = 285
        Left      = 2520
        TabIndex  = 174
        Top       = 1320
        Width     = 735
    End
    Begin VB.TextBox txtLotMassMeanLarge
        Height    = 285

```

```

        Left           = 2520
        TabIndex      = 173
        Top           = 840
        Width         = 735
    End
    Begin VB.TextBox txtFractionVerySmall
        Height        = 285
        Left          = 1440
        TabIndex      = 101
        Top           = 1800
        Width         = 735
    End
    Begin VB.TextBox txtFractionSmall
        Height        = 285
        Left          = 1440
        TabIndex      = 100
        Top           = 1320
        Width         = 735
    End
    Begin VB.TextBox txtFractionLarge
        Height        = 285
        Left          = 1440
        TabIndex      = 99
        Top           = 840
        Width         = 735
    End
    End
    Begin VB.Label lblLotMassSD
        Caption       = "Std, Dev. Lot Mass (lb)"
        Height        = 375
        Left          = 3600
        TabIndex      = 180
        Top           = 240
        Width         = 975
    End
    End
    Begin VB.Label lblLotMassMean
        Caption       = "Mean Lot Mass (lb)"
        Height        = 375
        Left          = 2520
        TabIndex      = 179
        Top           = 240
        Width         = 735
    End
    End
    Begin VB.Line Line3
        X1            = 1200
        X2            = 4560
        Y1            = 720
        Y2            = 720
    End
    End
    Begin VB.Label lblVerySmallFraction
        Caption       = "Very Small:"
        Height        = 255
        Left          = 240
        TabIndex      = 98
        Top           = 1800
        Width         = 975
    End
    End
    Begin VB.Label lblSmallFraction

```

```

Caption          = "Small:"
Height           = 255
Left             = 240
TabIndex        = 97
Top              = 1320
Width            = 615
End
Begin VB.Label lblLargeFraction
Caption          = "Large:"
Height           = 255
Left             = 240
TabIndex        = 96
Top              = 840
Width            = 615
End
Begin VB.Label lblPLantSize
Alignment        = 2 'Center
Caption          = "Fraction produced (0-1)"
Height           = 495
Left             = 1200
TabIndex        = 95
Top              = 240
Width            = 1215
End
End
Begin VB.Frame fraLotTesting
Caption          = "Product Testing "
BeginProperty Font
Name            = "MS Sans Serif"
Size            = 8.25
Charset         = 0
Weight          = 700
Underline       = 0 'False
Italic          = 0 'False
Strikethrough   = 0 'False
EndProperty
ForeColor       = &H00C00000&
Height          = 3375
Left            = 5520
TabIndex        = 77
Top             = 2880
Width           = 4695
Begin VB.Frame fraLotPositiveAction
Caption          = "Positive Result Actions"
ForeColor       = &H00C00000&
Height          = 855
Left            = 240
TabIndex        = 164
Top             = 2280
Width           = 1815
Begin VB.CheckBox chkLotPositiveActionDispose
Caption          = "Dispose product"
Height           = 255
Left             = 240
TabIndex        = 165
Top             = 360
Width            = 1455

```

```

End
End
Begin VB.Frame fraLotTestType
Caption      = "Testing Type"
ForeColor    = &H00C00000&
Height       = 975
Left         = 3000
TabIndex     = 161
Top          = 2160
Width        = 1455
Begin VB.OptionButton optLotRandom
Caption      = "Random"
Height       = 255
Left         = 120
TabIndex     = 163
Top          = 600
Width        = 1095
End
Begin VB.OptionButton optLotSystematic
Caption      = "Systematic"
ForeColor    = &H00000000&
Height       = 255
Left         = 120
TabIndex     = 162
Top          = 240
Width        = 1095
End
End
End
Begin VB.TextBox txtNLotLarge
Height       = 285
Left         = 1800
TabIndex     = 80
Top          = 840
Width        = 615
End
Begin VB.TextBox txtNLotSmall
Height       = 285
Left         = 1800
TabIndex     = 79
Top          = 1320
Width        = 615
End
Begin VB.TextBox txtNLotVerySmall
Height       = 285
Left         = 1800
TabIndex     = 78
Top          = 1800
Width        = 615
End
Begin VB.Label lblLargeNLot
Caption      = "Large Plants"
Height       = 255
Left         = 240
TabIndex     = 84
Top          = 840
Width        = 1095
End
End

```

```

Begin VB.Label lblNRTETests
    Caption      = "No. Tests / month"
    Height       = 255
    Left         = 240
    TabIndex     = 83
    Top          = 360
    Width        = 1455
End
Begin VB.Label lblSmallNLot
    Caption      = "Small Plants"
    Height       = 255
    Left         = 240
    TabIndex     = 82
    Top          = 1320
    Width        = 1095
End
Begin VB.Label lblVerySmallNLot
    Caption      = "Very Small Plants"
    Height       = 255
    Left         = 240
    TabIndex     = 81
    Top          = 1800
    Width        = 1455
End
Begin VB.Line Line2
    X1           = 240
    X2           = 2520
    Y1           = 720
    Y2           = 720
End
End
Begin VB.Frame fraFCSTesting
    Caption      = "Food Contact Surface Testing "
    BeginProperty Font
        Name      = "MS Sans Serif"
        Size      = 8.25
        Charset   = 0
        Weight    = 700
        Underline = 0   'False
        Italic    = 0   'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor    = &H00C00000&
    Height       = 3375
    Left         = 480
    TabIndex     = 21
    Top          = 2880
    Width        = 4695
    Begin VB.Frame fraFCSTestType
        Caption      = "Testing Type:"
        ForeColor    = &H00C00000&
        Height       = 975
        Left         = 3000
        TabIndex     = 158
        Top          = 2280
        Width        = 1455
        Begin VB.OptionButton optFCSRandom

```

```

Caption      = "Random"
Height      = 255
Left        = 120
TabIndex    = 160
Top         = 600
Width       = 1095
End
Begin VB.OptionButton optFCSSystematic
Caption      = "Systematic"
Height      = 255
Left        = 120
TabIndex    = 159
Top         = 240
Width       = 1095
End
End
Begin VB.Frame fraFCSPositive
Caption      = "Positive Result Actions"
ForeColor    = &H00800000&
Height      = 975
Left        = 240
TabIndex    = 155
Top         = 2280
Width       = 2295
Begin VB.CheckBox chkFCSPositiveActionTestLot
Caption      = "Test Lot "
Height      = 255
Left        = 240
TabIndex    = 157
Top         = 600
Width       = 1695
End
Begin VB.CheckBox chkFCSPositiveActionClean
Caption      = "Enhanced Cleaning"
Height      = 255
Left        = 240
TabIndex    = 156
Top         = 240
Width       = 1695
End
End
End
Begin VB.TextBox txtNFCSVerySmall
Height      = 285
Left        = 1800
TabIndex    = 75
Top         = 1800
Width       = 615
End
Begin VB.CheckBox chkHoldFCSVerySmall
Height      = 375
Left        = 3360
TabIndex    = 74
ToolTipText = "If selected, the tested lot will be
held until the results become available."
Top         = 1800
Width       = 255
End
End

```

```

Begin VB.TextBox txtNFCSSmall
    Height      = 285
    Left        = 1800
    TabIndex    = 72
    Top         = 1320
    Width       = 615
End
Begin VB.CheckBox chkHoldFCSSmall
    Height      = 375
    Left        = 3360
    TabIndex    = 71
    ToolTipText = "If selected, the tested lot will be
held until the results become available."
    Top         = 1320
    Width       = 255
End
Begin VB.CheckBox chkHoldFCSLarge
    Height      = 375
    Left        = 3360
    TabIndex    = 24
    ToolTipText = "If selected, the tested lot will be
held until the results become available."
    Top         = 840
    Width       = 255
End
Begin VB.TextBox txtNFCSLarge
    Height      = 285
    Left        = 1800
    TabIndex    = 23
    Top         = 840
    Width       = 615
End
Begin VB.Line Line1
    X1          = 240
    X2          = 4200
    Y1          = 720
    Y2          = 720
End
Begin VB.Label lblVerySmallNFCS
    Caption     = "Very Small Plants"
    Height      = 255
    Left        = 240
    TabIndex    = 76
    Top         = 1800
    Width       = 1455
End
Begin VB.Label lblSmallNFCS
    Caption     = "Small Plants"
    Height      = 255
    Left        = 240
    TabIndex    = 73
    Top         = 1320
    Width       = 1095
End
Begin VB.Label lblTestandHold
    Caption     = "Test and Hold Product?"
    Height      = 375

```

```

        Left            = 3120
        TabIndex       = 70
        Top            = 240
        Width          = 1095
    End
    Begin VB.Label lblNFCSTests
        Caption         = "No. Tests / month"
        Height          = 255
        Left            = 240
        TabIndex       = 69
        Top            = 360
        Width          = 1455
    End
    Begin VB.Label lblLargeNFCS
        Caption         = "Large Plants"
        Height          = 255
        Left            = 240
        TabIndex       = 22
        Top            = 840
        Width          = 1095
    End
End
Begin VB.Frame fraSanitation
    Caption           = "Sanitation Data"
    BeginProperty Font
        Name            = "MS Sans Serif"
        Size            = 8.25
        Charset         = 0
        Weight          = 700
        Underline       = 0   'False
        Italic          = 0   'False
        Strikethrough   = 0   'False
    EndProperty
    ForeColor         = &H00C00000&
    Height            = 2415
    Left              = 5520
    TabIndex         = 12
    Top               = 360
    Width             = 4695
    Begin VB.TextBox txtFCSTrigger
        Height          = 285
        Left            = 3000
        TabIndex       = 151
        Top            = 1800
        Width          = 975
    End
    Begin VB.TextBox txtWipeEff
        Height          = 285
        Left            = 3000
        TabIndex       = 67
        Top            = 360
        Width          = 975
    End
    Begin VB.TextBox txtSSOPEff
        Height          = 285
        Left            = 3000
        TabIndex       = 65

```



```

        Top          = 840
        Width        = 975
    End
    Begin VB.TextBox txtEnhanceEff
        Height        = 285
        Left          = 3000
        TabIndex      = 20
        Top          = 1320
        Width        = 975
    End
    Begin VB.Label lblNFCSTrigger
        Caption        = "Sequential FCS Positives to trigger
enhanced cleaning"
        Height        = 495
        Left          = 240
        TabIndex      = 150
        Top          = 1800
        Width        = 2295
    End
    Begin VB.Label lblWipeEff
        Caption        = "Wipe Down Btw Lots Efficiency (0-
1):"
        Height        = 375
        Left          = 240
        TabIndex      = 68
        Top          = 360
        Width        = 2535
    End
    Begin VB.Label lblSSOPEff
        Caption        = "End of Day Cleaning Efficiency (0-
1):"
        Height        = 375
        Left          = 240
        TabIndex      = 66
        Top          = 840
        Width        = 2535
    End
    Begin VB.Label lblEnhanceEff
        Caption        = "Enhanced Cleaning after FCS
positive (0-1):"
        Height        = 375
        Left          = 240
        TabIndex      = 19
        Top          = 1320
        Width        = 2535
    End
End
End
Begin vsOcx6LibCtl.vsElastic elalProjInfo
    Height          = 6720
    Left           = -12750
    TabIndex       = 1
    TabStop        = 0    'False
    Top            = 330
    Width          = 11205
    _ExtentX       = 19764
    _ExtentY       = 11853

```

```

BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
  Name           = "MS Sans Serif"
  Size           = 18
  Charset        = 0
  Weight         = 700
  Underline      = 0 'False
  Italic         = 0 'False
  Strikethrough  = 0 'False
EndProperty
Enabled          = -1 'True
Appearance       = 1
MousePointer     = 0
_ConvInfo        = 1
_Version         = 600
BackColor        = -2147483633
ForeColor        = -2147483630
FloodColor       = 192
ForeColorDisabled = -2147483631
Caption          = "Listeria monocytogenes Risk Assessment"
Align            = 0
Appearance       = 1
AutoSizeChildren = 0
BorderWidth      = 6
ChildSpacing     = 4
Splitter         = 0 'False
FloodDirection  = 0
FloodPercent     = 0
CaptionPos       = 3
WordWrap         = -1 'True
MaxChildSize     = 0
MinChildSize     = 0
TagWidth         = 0
TagPosition      = 0
Style            = 0
TagSplit         = 0 'False
PicturePos       = 4
CaptionStyle     = 0
ResizeFonts      = 0 'False
GridRows         = 0
GridCols         = 0
_GridInfo        = ""
Begin MSComDlg.CommonDialog dlgFile
  Left           = 600
  Top            = 5640
  _ExtentX       = 847
  _ExtentY       = 847
  _Version       = 393216
End
Begin VB.CommandButton cmdExampleData
  Caption        = "&Example Data"
  Height         = 495
  Left           = 4080
  TabIndex       = 27
  Top            = 5160
  Width          = 2055
End
Begin VB.Frame frmRunInfo

```

```

Caption          = "Model Run Information"
BeginProperty Font
  Name           = "MS Sans Serif"
  Size           = 8.25
  Charset        = 0
  Weight         = 700
  Underline      = 0 'False
  Italic         = 0 'False
  Strikethrough  = 0 'False
EndProperty
ForeColor        = &H00C00000&
Height           = 3735
Left             = 4920
TabIndex        = 3
Top             = 960
Width           = 5175
Begin VB.TextBox txtRunName
  Height         = 285
  Left          = 1560
  TabIndex      = 7
  Top           = 480
  Width         = 3255
End
Begin VB.TextBox txtAuthors
  Height         = 285
  Left          = 1560
  TabIndex      = 6
  Top           = 840
  Width         = 3255
End
Begin VB.TextBox txtDate
  Height         = 285
  Left          = 1560
  TabIndex      = 5
  Top           = 1200
  Width         = 3255
End
Begin VB.TextBox txtDescription
  Height         = 1695
  Left          = 1560
  MultiLine     = -1 'True
  TabIndex      = 4
  Top           = 1560
  Width         = 3255
End
Begin VB.Label lblAuthors
  Caption       = "Authors:"
  Height        = 255
  Left         = 360
  TabIndex     = 11
  Top          = 840
  Width        = 735
End
Begin VB.Label lblDate
  Caption       = "Date:"
  Height        = 255
  Left         = 360

```

```

        TabIndex      = 10
        Top           = 1200
        Width        = 615
    End
    Begin VB.Label lblRunName
        Caption       = "Run Name:"
        Height       = 255
        Left         = 360
        TabIndex     = 9
        Top         = 480
        Width       = 975
    End
    Begin VB.Label lblDescription
        Caption       = "Description:"
        Height       = 255
        Left         = 360
        TabIndex     = 8
        Top         = 1560
        Width       = 855
    End
    End
    End
    Begin VB.Image imgIntroPic
        Height       = 3600
        Left        = 600
        Picture     = "lmra.frx":0000
        Stretch     = -1 'True
        Top        = 1080
        Width      = 3420
    End
    End
    End
    Begin VB.Menu mnuFile
        Caption     = "&File"
        Begin VB.Menu mnuNew
            Caption = "&New"
        End
        Begin VB.Menu mnuOpen
            Caption = "&Open"
        End
        Begin VB.Menu mnuSave
            Caption = "&Save"
        End
        Begin VB.Menu mnuSaveAs
            Caption = "Save &As"
        End
        Begin VB.Menu mnuPrint
            Caption = "&Print"
            Enabled = 0 'False
        End
        Begin VB.Menu mnuExit
            Caption = "E&xit"
        End
    End
    End
    Begin VB.Menu mnuEdit
        Caption = "&Edit"
        Begin VB.Menu mnuCut
            Caption = "Cu&t"
        End
    End

```

```

        Shortcut      = ^X
    End
    Begin VB.Menu mnuCopy
        Caption      = "&Copy"
        Shortcut     = ^C
    End
    Begin VB.Menu mnuPaste
        Caption      = "&Paste"
        Shortcut     = ^V
    End
End
Begin VB.Menu mnuHelp
    Caption          = "&Help"
    NegotiatePosition= 3 'Right
    Begin VB.Menu mnuOverview
        Caption      = "&Overview"
    End
    Begin VB.Menu mnuHelpContents
        Caption      = "&Contents"
        Enabled      = 0 'False
    End
    Begin VB.Menu mnuAbout
        Caption      = "&About"
        Enabled      = 0 'False
    End
End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
' Portions of this code not directly relating to the risk assessment
are still
' incomplete.  In particular, the printing and help functions are
disabled.
' A future version may also include tracking the random number seed so
that
' specific runs can be repeated.

Option Explicit

' optional settings for debugging
Const CheckCE = False
Const CheckCEFile = "CECheck.csv"

' Global Constants

Const NSizes = 3                ' number of different plant
sizes
Const gLarge = 1                ' index for plant size
Const gSmall = 2
Const gVerySmall = 3
Const gMinLotMass = 1000#      ' Smallest mass in pounds that
can be produced
Const gMaxLotMass = 100000#

```

```

Const MAXDAY = 1000000           ' Maximum number of days. Used
to size arrays
Const nLotsPerDay = 2
Const MAXLOT = nLotsPerDay * MAXDAY
Const MINFCSLspp = 0.0000000000000001 'if conc less than this on FCS,
reset to 0
Const DaysPerMonth = 30
Const MaxDouble = 1E+300        ' Max for double precision.
Used in truncation functions.
Const MinStart = 100           ' minimum lot number to use for
selection to retail

' Global Variables
Dim ContamDay() As Boolean      ' true if contamination event
occurs that day
Dim FCSSampled() As Boolean
Dim FCSTriggered() As Boolean
Dim FCSSwabArea() As Double
Dim Fraction() As Double
Dim GIPEffic() As Double
Dim GIPFrac() As Double
'Dim HoldLot() As Integer
Dim HoldFCS() As Integer
Dim log10LMRetail() As Double
Dim lotCEAdded() As Double
Dim lotDay() As Long
Dim lotFCSArea() As Double
Dim lotFCSDisinf() As Double
Dim lotFCSLspp() As Double
Dim lotFCSResult() As Boolean   ' true if FCS contamination
detected that lot
Dim lotLM() As Double
Dim lotLMPP() As Double
Dim lotLMResult() As Boolean
Dim lotLMRetail() As Double
Dim lotLMRetailSelected() As Double
Dim lotLspp() As Double
Dim LotMassMean() As Double
Dim LotMassStdDev() As Double
Dim lotMass() As Double
Dim lotMassSampled() As Double
Dim lotRatio() As Double
Dim LotSampled() As Boolean
Dim lotSampledbyFCS() As Boolean
Dim lotTC() As Double
Dim LotToUse() As Long
Dim nLotsThisSize() As Long
Dim NLotUsedThisSize() As Long
Dim NSamplesFCS() As Long
Dim NSamplesLot() As Long
Dim PPFrac() As Double
Dim PPMaxReduct() As Double
Dim PPMinReduct() As Double
Dim PPReduct() As Double
Dim xtab() As Long

Dim CEAddMean As Double

```

```
Dim CEAddStdDev As Double
Dim CEMean As Double
Dim CEStdDev As Double
Dim DurationMean As Double
Dim DurationStdDev As Double
Dim EfficEnhance As Double
Dim EfficSOP As Double
Dim EfficWipe As Double
Dim FCSAreaLargeMax As Double
Dim FCSAreaLargeMin As Double
Dim FCSPositives As Long
Dim FCSSwabMin As Double
Dim FCSSwabMax As Double
Dim FCSSwabsNComposited As Long
Dim FCSTrigger As Long
Dim FCSPositiveActionTestLot As Long
Dim FCSPositiveActionClean As Long
Dim gCurrentFileName As String
Dim GIPEfficMin As Double
Dim GIPEfficMax As Double
Dim Growth As Double
Dim LagReportFCS As Double
Dim LagReportLot As Double
Dim LotPositiveActionDispose As Long
Dim NLots As Long
Dim NLotsUsed As Long
Dim PDetectOneFCS As Double
Dim PDetectOneLot As Double
Dim RatioMean As Double
Dim RatioStdDev As Double
Dim RTESampledMin As Double
Dim RTESampledMax As Double
Dim TCMean As Double
Dim TCStdDev As Double
Dim TransitTemperMax As Double
Dim TransitTemperMin As Double
Dim TransitTimeMin As Double
Dim TransitTimeMode As Double
Dim TransitTimeMax As Double
```

```
Private Sub CalibrateDeli()
    flxCalibration.TextMatrix(1, 1) = "7.4E-06"
    flxCalibration.TextMatrix(2, 1) = "3.7E-05"
    flxCalibration.TextMatrix(3, 1) = "2.7E-04"
    flxCalibration.TextMatrix(4, 1) = "5.5E-03"
    flxCalibration.TextMatrix(5, 1) = "1.5E+00"
    flxCalibration.TextMatrix(6, 1) = "1.1E+01"
    flxCalibration.TextMatrix(7, 1) = "7.9E+02"
    flxCalibration.TextMatrix(8, 1) = "1.4E+05"
End Sub
```

```
Private Sub CalibrateFrankfurters()
    flxCalibration.TextMatrix(1, 1) = "1.1E-05"
    flxCalibration.TextMatrix(2, 1) = "7.9E-05"
    flxCalibration.TextMatrix(3, 1) = "9.3E-04"
    flxCalibration.TextMatrix(4, 1) = "3.6E-02"
    flxCalibration.TextMatrix(5, 1) = "3.4E+00"
```

```

    flxCalibration.TextMatrix(6, 1) = "4.2E+02"
    flxCalibration.TextMatrix(7, 1) = "7.4E+04"
    flxCalibration.TextMatrix(8, 1) = "4.1E+07"
End Sub

Private Sub ClearArrays()
    ReDim ContamDay(1 To NSizes, 0 To MAXDAY)
    NLots = Val(txtNLots.Text)
    ReDim FCSSampled(1 To NSizes, 0 To NLots)
    ReDim FCSSwabArea(1 To NSizes, 0 To NLots)
    ReDim FCSTriggered(1 To NSizes, 0 To NLots)
    ReDim Fraction(1 To NSizes)
    ReDim GIPEffic(1 To NSizes, 0 To NLots)
    ReDim GIPFrac(1 To NSizes)
    ' ReDim HoldLot(1 To NSizes)
    ReDim HoldFCS(1 To NSizes)
    ReDim lotCEAdded(1 To NSizes, 0 To NLots)
    ReDim lotDay(1 To NSizes, 0 To NLots)
    ReDim lotFCSArea(1 To NSizes, 0 To NLots)
    ReDim lotFCSDisinf(1 To NSizes, 0 To NLots)
    ReDim lotFCSLspp(1 To NSizes, 0 To NLots)
    ReDim lotFCSResult(1 To NSizes, 0 To NLots)
    ReDim lotLspp(1 To NSizes, 0 To NLots)
    ReDim lotLM(1 To NSizes, 0 To NLots)
    ReDim lotLMPP(1 To NSizes, 0 To NLots)
    ReDim lotLMResult(1 To NSizes, 0 To NLots)
    ReDim lotLMRetail(1 To NSizes, 0 To NLots)
    ReDim lotLMRetailSelected(0 To NLots)
    ReDim lotMass(1 To NSizes, 0 To NLots)
    ReDim lotMassSampled(1 To NSizes, 0 To NLots)
    ReDim LotMassMean(1 To NSizes)
    ReDim LotMassStdDev(1 To NSizes)
    ReDim lotRatio(1 To NSizes, 0 To NLots)
    ReDim LotSampled(1 To NSizes, 0 To NLots)
    ReDim lotSampledbyFCS(1 To NSizes, 0 To NLots)
    ReDim lotTC(1 To NSizes, 0 To NLots)
    ReDim LotToUse(0 To NLots, 1 To 2)
    ReDim nLotsThisSize(1 To NSizes)
    ReDim NLotUsedThisSize(1 To NSizes) As Long
    ReDim NSamplesFCS(1 To NSizes)
    ReDim NSamplesLot(1 To NSizes)
    ReDim PPFrac(1 To NSizes)
    ReDim PPMaxReduct(1 To NSizes)
    ReDim PPMinReduct(1 To NSizes)
    ReDim PPReduct(1 To NSizes, 0 To NLots)
End Sub

Private Sub cmdExampleData_Click()
    ' loads default data from file
    Dim strBaseFileName As String
    If MsgBox("WARNING! You are about to erase all existing data.
Continue?", vbYesNo) = vbYes Then
        Call DataFileRead(App.Path + "\" + "Base_listeria.dat")
    End If
End Sub

Private Sub cmdRunModel_Click()

```



```

'Call LMRAModel
Screen.MousePointer = 11
Call ClearArrays
Call ScreenDataRead
Call generateCE
If optLotRandom.Value Then
    Call generateLotRandom
Else
    Call generateLotSystematic
End If
If optFCSRandom.Value Then
    Call generateFCSRandom
Else
    Call generateFCSSystematic
End If
Call LotCalc
Call PostProcess
Call SelectLotsToUse
Call OutputFileSave(txtOutputFile.Text)
'MsgBox "Simulation Finished"
Beep
Call outputCalibration
optGraphLMRetail.Value = True    'Call OutputGraph
Call OutputStatsToScreen
Screen.MousePointer = 0
End Sub

Private Sub DataFileRead(fname As String)
Dim fnum As Integer, i1 As Integer, i2 As Integer, i3 As Integer
Dim b1 As Boolean, b2 As Boolean, b3 As Boolean, b4 As Boolean
Dim strDescript As String
    fnum = FreeFile
    'Open App.Path + "\" + fname For Input As fnum
    Open fname For Input As fnum
    ' ### Project Data ###
    Input #fnum, strDescript
    txtRunName.Text = strDescript
    Input #fnum, strDescript
    txtAuthors.Text = strDescript
    Input #fnum, strDescript
    txtDate.Text = strDescript
    Input #fnum, strDescript
    txtDescription.Text = strDescript
    ' ### Plant Data ###
    Input #fnum, strDescript
    Input #fnum, Fraction(gLarge), Fraction(gSmall), Fraction(gVerySmall)
    Input #fnum, strDescript
    Input #fnum, LotMassMean(gLarge), LotMassMean(gSmall),
LotMassMean(gVerySmall)
    Input #fnum, strDescript
    Input #fnum, LotMassStdDev(gLarge), LotMassStdDev(gSmall),
LotMassStdDev(gVerySmall)
    Input #fnum, strDescript
    Input #fnum, EfficWipe, EfficSOP, EfficEnhance
    Input #fnum, strDescript
    Input #fnum, FCSTrigger
    Input #fnum, strDescript

```

```

Input #fnum, i1, i2, i3
NSamplesFCS(gLarge) = i1
NSamplesFCS(gSmall) = i2
NSamplesFCS(gVerySmall) = i3
Input #fnum, strDescript
Input #fnum, i1, i2, i3
HoldFCS(gLarge) = i1
HoldFCS(gSmall) = i2
HoldFCS(gVerySmall) = i3
Input #fnum, strDescript
Input #fnum, FCSPositiveActionTestLot, FCSPositiveActionClean,
LotPositiveActionDispose
Input #fnum, strDescript
Input #fnum, i1, i2, i3
NSamplesLot(gLarge) = i1
NSamplesLot(gSmall) = i2
NSamplesLot(gVerySmall) = i3
Input #fnum, strDescript
Input #fnum, b1, b2, b3, b4
optFCSRandom.Value = b1
optFCSSystematic.Value = b2
optLotRandom.Value = b3
optLotSystematic.Value = b4
' ### Contamination Data ###
Input #fnum, strDescript
Input #fnum, CEMean, CEStdDev
Input #fnum, strDescript
Input #fnum, DurationMean, DurationStdDev
Input #fnum, strDescript
Input #fnum, CEAddMean, CEAddStdDev
Input #fnum, strDescript
Input #fnum, TCMean, TCStdDev
Input #fnum, strDescript
Input #fnum, FCSSwabMin, FCSSwabMax, FCSSwabsNComposited
Input #fnum, strDescript
Input #fnum, RTESampledMin, RTESampledMax
' ### Post Processing Tab ###
Input #fnum, strDescript
Input #fnum, PPFrac(gLarge), PPFrac(gSmall), PPFrac(gVerySmall)
Input #fnum, strDescript
Input #fnum, PPMinReduct(gLarge), PPMinReduct(gSmall),
PPMinReduct(gVerySmall)
Input #fnum, strDescript
Input #fnum, PPMaxReduct(gLarge), PPMaxReduct(gSmall),
PPMaxReduct(gVerySmall)
Input #fnum, strDescript
Input #fnum, GIPFrac(gLarge), GIPFrac(gSmall), GIPFrac(gVerySmall)
Input #fnum, strDescript
Input #fnum, GIPEfficMin, GIPEfficMax
' ### Advanced Data Tab ###
Input #fnum, strDescript
Input #fnum, PDetectOneFCS, PDetectOneLot
Input #fnum, strDescript
Input #fnum, LagReportFCS, LagReportLot
Input #fnum, strDescript
Input #fnum, RatioMean, RatioStdDev
Input #fnum, strDescript

```

```

Input #fnum, FCSAreaLargeMin, FCSAreaLargeMax
Input #fnum, strDescript
Input #fnum, Growth
' ### Simulation Tab ###
Input #fnum, strDescript
Input #fnum, NLots
Input #fnum, strDescript
Input #fnum, gCurrentFileName
Input #fnum, strDescript
Input #fnum, b1, b2
optCalibrateDeliMeats.Value = b1
optCalibrateFrankfurters.Value = b2
Close #fnum
Call ScreenDataSet
End Sub

Private Sub DataFileSave(fname As String)
Dim fnum As Integer, i As Integer
Call ScreenDataRead
fnum = FreeFile
' Open App.Path + "\" + fname For Output As fnum
Open fname For Output As fnum
' ### Project Data ###
Write #fnum, txtRunName.Text
Write #fnum, txtAuthors.Text
Write #fnum, txtDate.Text
Write #fnum, txtDescription.Text
' ### Plant Data ###
Write #fnum, "Fraction of product produced by large, small and very
small plants"
Write #fnum, Fraction(gLarge), Fraction(gSmall), Fraction(gVerySmall)
Write #fnum, "Lot mass mean: large, small, and very small plants"
Write #fnum, LotMassMean(gLarge), LotMassMean(gSmall),
LotMassMean(gVerySmall)
Write #fnum, "Lot mass std dev: large, small, and very small plants"
Write #fnum, LotMassStdDev(gLarge), LotMassStdDev(gSmall),
LotMassStdDev(gVerySmall)
Write #fnum, "Cleaning efficiency: Wipedown, Standard, Enhanced"
Write #fnum, EfficWipe, EfficSOP, EfficEnhance
Write #fnum, "Number of sequential FCS positive to trigger enhanced
cleaning:"
Write #fnum, FCSTrigger
Write #fnum, "Number of FCS Samples / Month for Large, Small, and
Very Small Plants"
Write #fnum, NSamplesFCS(gLarge), NSamplesFCS(gSmall),
NSamplesFCS(gVerySmall)
Write #fnum, "Test and Hold on FCS Testing for Large, Small, and Very
Small Plants"
Write #fnum, HoldFCS(gLarge), HoldFCS(gSmall), HoldFCS(gVerySmall)
Write #fnum, "Actions if Positive Result : FCS Test Lot, FCS Clean,
Lot Dispose"
Write #fnum, FCSPositiveActionTestLot, FCSPositiveActionClean,
LotPositiveActionDispose
Write #fnum, "Number of Product Samples / Month for Large, Small, and
Very Small Plants"
Write #fnum, NSamplesLot(gLarge), NSamplesLot(gSmall),
NSamplesLot(gVerySmall)

```

```

Write #fnum, "FCS random and systematic sampling, Lot random and
systematic sampling"
Write #fnum, optFCSRandom.Value, optFCSSystematic.Value,
optLotRandom.Value, optLotSystematic.Value
' ### Contamination Data ###
Write #fnum, "Days between contamination events: mean and std dev"
Write #fnum, CEMean, CEStdDev
Write #fnum, "Duration of contamination events: mean and std dev"
Write #fnum, DurationMean, DurationStdDev
Write #fnum, "Added Lspp to FCS during contamintion event: mean and
std dev"
Write #fnum, CEAddMean, CEAddStdDev
Write #fnum, "Transfer coef: mean and std dev"
Write #fnum, TCMean, TCStdDev
Write #fnum, "Sampled FCS Areas: min and max, number of composites"
Write #fnum, FCSSwabMin, FCSSwabMax, FCSSwabsNComposited
Write #fnum, "Sampled mass for RTE product: min and max"
Write #fnum, RTESampledMin, RTESampledMax
' ### Post Processing Tab ###
Write #fnum, "Fraction of large, small, and very small plants that
use post processing"
Write #fnum, PPFrac(gLarge), PPFrac(gSmall), PPFrac(gVerySmall)
Write #fnum, "Min Fraction reduction for post processing by large,
small, very small plants"
Write #fnum, PPMinReduct(gLarge), PPMinReduct(gSmall),
PPMinReduct(gVerySmall)
Write #fnum, "Max Fraction reduction for post processing by large,
small, very small plants"
Write #fnum, PPMaxReduct(gLarge), PPMaxReduct(gSmall),
PPMaxReduct(gVerySmall)
Write #fnum, "Fraction of large, small, and very small plants that
use growth inhibiting packaging"
Write #fnum, GIPFrac(gLarge), GIPFrac(gSmall), GIPFrac(gVerySmall)
Write #fnum, "Min and max effectiveness of growth inhibiting
packaging"
Write #fnum, GIPEfficMin, GIPEfficMax
' ### Advanced Data Tab ###
Write #fnum, " Prob of method detecting 1 cfu for FCS (Lspp) and RTE
product (LM)"
Write #fnum, PDetectOneFCS, PDetectOneLot
Write #fnum, "Lags on reporting FCS and product results:"
Write #fnum, LagReportFCS, LagReportLot
Write #fnum, "Lm to Lspp ratio: mean and std dev"
Write #fnum, RatioMean, RatioStdDev
Write #fnum, "FCS area for Large plants: min and max"
Write #fnum, FCSAreaLargeMin, FCSAreaLargeMax
Write #fnum, "Growth"
Write #fnum, Growth
' ### Simulation Tab ###
Write #fnum, "Number of lots to simulate"
Write #fnum, NLots
Write #fnum, "Output file name:"
Write #fnum, gCurrentFileName
Write #fnum, "Product to calibrate:"
Write #fnum, optCalibrateDeliMeats.Value,
optCalibrateFrankfurters.Value
Close #fnum

```

End Sub

Private Sub Form_Load()

Dim temp As Double

Randomize

temp = RNpoisson(10#) ' initialize Poisson RN generator

Call ClearArrays

tabLMRAMain.CurrTab = 0

flxCalibration.Clear

flxCalibration.ColWidth(0) = 1000

flxCalibration.ColWidth(1) = 1200

flxCalibration.ColWidth(2) = 1200

flxCalibration.ColWidth(3) = 1200

flxCalibration.TextMatrix(0, 0) = "Percentile"

flxCalibration.TextMatrix(0, 1) = "FDA"

flxCalibration.TextMatrix(0, 2) = "Retail"

flxCalibration.TextMatrix(0, 3) = "PP Large"

flxCalibration.TextMatrix(1, 0) = "80th"

flxCalibration.TextMatrix(2, 0) = "85th"

flxCalibration.TextMatrix(3, 0) = "90th"

flxCalibration.TextMatrix(4, 0) = "95th"

flxCalibration.TextMatrix(5, 0) = "99th"

flxCalibration.TextMatrix(6, 0) = "99.5th"

flxCalibration.TextMatrix(7, 0) = "99.9th"

flxCalibration.TextMatrix(8, 0) = "99.99th"

flxCalibration.TextMatrix(9, 0) = "max"

flxOutputStats.Clear

flxOutputStats.ColWidth(0) = 1500

flxOutputStats.ColWidth(1) = 1000

flxOutputStats.ColWidth(2) = 1000

flxOutputStats.ColWidth(3) = 1000

flxOutputStats.ColWidth(4) = 1000

flxOutputStats.TextMatrix(0, 0) = "Statistic"

flxOutputStats.TextMatrix(0, 1) = "Large"

flxOutputStats.TextMatrix(0, 2) = "Small"

flxOutputStats.TextMatrix(0, 3) = "Very Small"

flxOutputStats.TextMatrix(0, 4) = "Selected"

flxOutputStats.TextMatrix(1, 0) = "No. Lots Produced"

flxOutputStats.TextMatrix(2, 0) = "No. Lots Selected"

flxOutputStats.TextMatrix(3, 0) = "No. Lots to Retail"

flxOutputStats.TextMatrix(4, 0) = "No. Lots Tested"

flxOutputStats.TextMatrix(5, 0) = " by Routine"

flxOutputStats.TextMatrix(6, 0) = " by FCS Positive"

flxOutputStats.TextMatrix(7, 0) = "No. Lots Failed"

flxOutputStats.TextMatrix(8, 0) = "Q90 (cfu/gm)"

flxOutputStats.TextMatrix(9, 0) = "Q95 (cfu/gm)"

flxOutputStats.TextMatrix(10, 0) = "Q99 (cfu/gm)"

flxOutputStats.TextMatrix(11, 0) = "No. FCS tests"

flxOutputStats.TextMatrix(12, 0) = "No. FCS tests failed"

flxContingency.Clear

flxContingency.ColWidth(0) = 1500

flxContingency.ColWidth(1) = 1500

```

flxContingency.ColWidth(2) = 1500
flxContingency.ColWidth(3) = 1500
flxContingency.TextMatrix(0, 1) = "Lot pos"
flxContingency.TextMatrix(0, 2) = "Lot neg"
flxContingency.TextMatrix(0, 3) = "Lot not tested"
flxContingency.TextMatrix(1, 0) = "FCS pos"
flxContingency.TextMatrix(2, 0) = "FCS neg"
flxContingency.TextMatrix(3, 0) = "FCS not tested"

flxContingencyLag.Clear
flxContingencyLag.ColWidth(0) = 1500
flxContingencyLag.ColWidth(1) = 1500
flxContingencyLag.ColWidth(2) = 1500
flxContingencyLag.ColWidth(3) = 1500
flxContingencyLag.TextMatrix(0, 1) = "Lot pos"
flxContingencyLag.TextMatrix(0, 2) = "Lot neg"
flxContingencyLag.TextMatrix(0, 3) = "Lot not tested"
flxContingencyLag.TextMatrix(1, 0) = "FCS pos"
flxContingencyLag.TextMatrix(2, 0) = "FCS neg"
flxContingencyLag.TextMatrix(3, 0) = "FCS not tested"

    optCalibrateDeliMeats.Value = True
End Sub

Public Sub generateCE()
'generates the initiating times and durations for a contamination event
'fills the ContamDay two dimensional vector by plant size and day
Dim i As Long, iday As Long, iDurat As Long, isize As Integer, NDays As
Long, delta As Long
Dim RunLength As Long
Dim fnum As Integer
If CheckCE Then
    fnum = FreeFile
    Open App.Path + "\" + CheckCEFile For Output As fnum
    Write #fnum, "CE Start Day", "Duration"
End If

NDays = NLots \ nLotsPerDay
For isize = 1 To NSizes
    iday = 1
    Do While iday <= NDays
        'Generate the start or the next contamination event
        delta = newCEDelta(CEMean, CESTdDev)
        iday = iday + delta
        If iday > NDays Then
            Exit Do
        End If
        ContamDay(isize, iday) = True
        iDurat = newCEDuration(DurationMean, DurationStdDev)
        If CheckCE Then
            Write #fnum, iday, iDurat
        End If
        For i = 1 To iDurat
            ContamDay(isize, iday + i) = True
        Next i
    Loop
Next isize

```

```

If CheckCE Then
    Close #fnum
End If
End Sub

Sub GenerateNRandomTests(RNDay() As Long, ByVal NSamples As Long)
Dim i As Long, j As Long
Dim aFD() As Double
'ReDim RNDay(1 To NSamples)
ReDim aFD(1 To NSamples)
Dim dTestValue As Double
Dim dOldValue As Double
Dim dOldDay As Double
Dim dDay As Double
    For i = 1 To nLotsPerDay * DaysPerMonth
        dDay = i
        dTestValue = Rnd
        For j = 1 To NSamples
            If dTestValue > aFD(j) Then
                dOldValue = aFD(j)
                dOldDay = RNDay(j)
                aFD(j) = dTestValue
                RNDay(j) = dDay
                dTestValue = dOldValue
                dDay = dOldDay
            End If
        Next j
    Next i
    'bubble sort
    For i = 1 To NSamples - 1
        For j = i + 1 To NSamples
            If RNDay(i) > RNDay(j) Then
                dOldDay = RNDay(j)
                RNDay(j) = RNDay(i)
                RNDay(i) = dOldDay
            End If
        Next j
    Next i
End Sub

Private Sub generateLotRandom()
Dim ilot As Long, isize As Long, imonth As Long, j As Long
Dim nMonths As Long
Dim RNLot(1 To nLotsPerDay * DaysPerMonth) As Long
    nMonths = Int((NLots / nLotsPerDay) / DaysPerMonth) + 1
    For isize = 1 To NSizes
        If NSamplesLot(isize) > 0 Then
            For imonth = 1 To nMonths
                Call GenerateNRandomTests(RNLot(), NSamplesLot(isize))
                For j = 1 To NSamplesLot(isize)
                    ilot = (imonth - 1) * DaysPerMonth * nLotsPerDay +
RNLot(j)
                    If ilot <= NLots Then
                        LotSampled(isize, ilot) = True
                    End If
                Next j
            Next imonth
        End If
    Next isize
End Sub

```

```

        End If
    Next isize
End Sub

Private Sub generateLotSystematic()
Dim nMonths As Long
Dim RndStartLot As Long
Dim DeltaLot As Long
Dim syslot(1 To DaysPerMonth * nLotsPerDay) As Long
Dim i As Long, ilot As Long, imonth As Long, isize As Long, j As Long
    nMonths = Int((NLots / nLotsPerDay) / DaysPerMonth) + 1
    For isize = 1 To NSizes
        If NSamplesLot(isize) > 0 Then
            DeltaLot = Int(DaysPerMonth * nLotsPerDay /
NSamplesLot(isize))
            RndStartLot = RNIntBtw(1, DeltaLot)
            syslot(1) = RndStartLot
            For j = 2 To NSamplesLot(isize)
                syslot(j) = syslot(j - 1) + DeltaLot
            Next j
            For imonth = 1 To nMonths
                For j = 1 To NSamplesLot(isize)
                    ilot = (imonth - 1) * DaysPerMonth * nLotsPerDay +
syslot(j)
                    If ilot <= NLots Then
                        LotSampled(isize, ilot) = True
                    End If
                Next j
            Next imonth
        End If
    Next isize
End Sub

Private Sub generateFCSRRandom()
Dim ilot As Long, isize As Long, imonth As Long, j As Long
Dim nMonths As Long
Dim RNLot(1 To nLotsPerDay * DaysPerMonth) As Long
nMonths = Int((NLots / nLotsPerDay) / DaysPerMonth) + 1
    For isize = 1 To NSizes
        If NSamplesFCS(isize) > 0 Then
            For imonth = 1 To nMonths
                Call GenerateNRandomTests(RNLot(), NSamplesFCS(isize))
                For j = 1 To NSamplesFCS(isize)
                    ilot = (imonth - 1) * DaysPerMonth * nLotsPerDay +
RNLot(j)
                    If ilot <= NLots Then
                        FCSSampled(isize, ilot) = True
                    End If
                Next j
            Next imonth
        End If
    Next isize
End Sub

Private Sub generateFCSSystematic()
Dim nMonths As Long
Dim RndStartFCS As Long

```



```

Dim DeltaFCS As Long
Dim sysFCS(1 To DaysPerMonth * nLotsPerDay) As Long
Dim i As Long, ilot As Long, imonth As Long, isize As Long, j As Long
    nMonths = Int((NLots / nLotsPerDay) / DaysPerMonth) + 1
    For isize = 1 To NSizes
        If NSamplesFCS(isize) > 0 Then
            DeltaFCS = Int(DaysPerMonth * nLotsPerDay /
NSamplesFCS(isize))
            RndStartFCS = RNIntBtw(1, DeltaFCS)
            sysFCS(1) = RndStartFCS
            For j = 2 To NSamplesFCS(isize)
                sysFCS(j) = sysFCS(j - 1) + DeltaFCS
            Next j
            For imonth = 1 To nMonths
                For j = 1 To NSamplesFCS(isize)
                    ilot = (imonth - 1) * DaysPerMonth * nLotsPerDay +
sysFCS(j)
                    If ilot <= NLots Then
                        FCSSampled(isize, ilot) = True
                    End If
                Next j
            Next imonth
        End If
    Next isize
End Sub

Private Sub LotCalc()
Dim CurrentDay As Long
Dim NewDay As Boolean
Dim iday As Long, ilot As Long, isize As Long, j As Long, lag As Long,
LotToTest As Long
'initial FCS area
For isize = 1 To NSizes
    lotFCSArea(isize, ilot) = NewFCSArea(isize, FCSAreaLargeMin,
FCSAreaLargeMax)
Next isize
For isize = 1 To NSizes
    FCSPositives = 0
    lotFCSLspp(isize, 0) = 0#
    CurrentDay = 0
    For ilot = 1 To NLots
        iday = Int((ilot - 1) / nLotsPerDay) + 1
        ' decide if a new day has started
        If iday <> CurrentDay Then
            CurrentDay = iday
            NewDay = True
        Else
            NewDay = False
        End If
        FCSSwabArea(isize, ilot) = NewSwabArea(FCSSwabMin, FCSSwabMax)
        ' update the FCS level, contaminate if ongoing contamination event
        If ContamDay(isize, iday) Then
            lotCEAdded(isize, ilot) = NewFCSLspp(CEAddMean, CEAddStdDev)
            lotFCSLspp(isize, ilot) = lotFCSLspp(isize, ilot - 1) +
lotCEAdded(isize, ilot)
        Else
            lotFCSLspp(isize, ilot) = lotFCSLspp(isize, ilot - 1)
        End If
    Next ilot
Next isize

```

```

End If
If lotFCSLspp(ysize, ilot) = 0 Then
    lotFCSArea(ysize, ilot) = NewFCSArea(ysize, FCSAreaLargeMin,
FCSAreaLargeMax)
Else
    lotFCSArea(ysize, ilot) = lotFCSArea(ysize, ilot - 1)
End If
If FCSSampled(ysize, ilot) Then
    lotFCSResult(ysize, ilot) = TestFCSResult(lotFCSLspp(ysize,
ilot), FCSSwabArea(ysize, ilot), PDetectOneFCS)
    If lotFCSResult(ysize, ilot) Then
        FCSPositives = FCSPositives + 1
        lag = NewLsppLag()
        If FCSPositiveActionTestLot = vbChecked Then
            If HoldFCS(ysize) = vbChecked Then
                lotSampledbyFCS(ysize, ilot) = True
            Else
                If ilot + lag * nLotsPerDay <= NLots Then
                    lotSampledbyFCS(ysize, ilot + lag * nLotsPerDay) =
True
                End If
            End If
        End If
    End If
    ' always lag the enhanced cleaning even if test and hold
product
    If (FCSPositives >= FCSTrigger) And (FCSPositiveActionClean
= vbChecked) Then
        LotToTest = ilot + lag * nLotsPerDay
        If LotToTest <= NLots Then
            FCSTriggered(ysize, LotToTest) = True
            FCSPositives = 0
        End If
    End If
End If
End If
lotDay(ysize, ilot) = iday
lotMass(ysize, ilot) = NewLotMass(LotMassMean(ysize),
LotMassStdDev(ysize))
lotMassSampled(ysize, ilot) = NewLotMassSampled(RTESampledMin,
RTESampledMax)
lotTC(ysize, ilot) = NewTC(TCMean, TCStdDev)
lotRatio(ysize, ilot) = NewRatio(RatioMean, RatioStdDev)
lotLspp(ysize, ilot) = lotTC(ysize, ilot) * (lotFCSLspp(ysize,
ilot) * lotFCSArea(ysize, ilot) / (lotMass(ysize, ilot) * 454#))
lotFCSLspp(ysize, ilot) = lotFCSLspp(ysize, ilot) * (1 -
lotTC(ysize, ilot))
lotLM(ysize, ilot) = lotRatio(ysize, ilot) * lotLspp(ysize, ilot)
' clean the food contact surface at the end of the lot production
If FCSTriggered(ysize, ilot) Then
    lotFCSLspp(ysize, ilot) = lotFCSLspp(ysize, ilot) * (1 -
EfficEnhance)
    lotFCSDisinf(ysize, ilot) = EfficEnhance
ElseIf NewDay Then
    lotFCSLspp(ysize, ilot) = lotFCSLspp(ysize, ilot) * (1 -
EfficWipe)
    lotFCSDisinf(ysize, ilot) = EfficWipe
Else

```

```

        lotFCSLspp(ysize, ilot) = lotFCSLspp(ysize, ilot) * (1 -
EfficSOP)
        lotFCSDisinf(ysize, ilot) = EfficSOP
    End If
    ' If FCS concentration drops too low, set it to 0.
    If lotFCSLspp(ysize, ilot) < MINFCSLspp Then
        lotFCSLspp(ysize, ilot) = 0
    End If
Next ilot
Next ysize
End Sub

```

```

Public Sub Moment(DatQ() As Double, n As Long, ave As Double, adev As
Double, sdev As Double, var As Double, skew As Double, curt As Double)
' Taken from Numerical Recipes
Dim j As Long, s As Double, p As Double
If n <= 1 Then
    MsgBox "N must be at least 2 in Moment"
    Exit Sub
End If
s = 0#
For j = 1 To n
    s = s + DatQ(j)
Next j
ave = s / n
adev = 0#
var = 0#
skew = 0#
curt = 0#
For j = 1 To n
    s = DatQ(j) - ave
    adev = adev + Abs(s)
    p = s * s
    var = var + p
    p = p * s
    skew = skew + p
    p = p * s
    curt = curt + p
Next j
adev = adev / n
var = var / (n - 1)
sdev = (var) ^ 0.5
If var <> 0# Then
    skew = skew / (n * sdev ^ 3)
    curt = curt / (n * var ^ 2) - 3!
Else
    MsgBox "no skew or kurtosis when zero variance"
End If
End Sub

```

```

Private Sub mnuCopy_Click()
If TypeOf Screen.ActiveControl Is TextBox Then
    Clipboard.Clear
    Clipboard.SetText Screen.ActiveControl.SelText
ElseIf TypeOf Screen.ActiveControl Is vsFlexGrid Then
    Clipboard.Clear
    Clipboard.SetText Screen.ActiveControl.Clip

```

```

End If
End Sub

Private Sub mnuCut_Click()
    If TypeOf Screen.ActiveControl Is TextBox Then
        Clipboard.Clear
        Clipboard.SetText Screen.ActiveControl.SelText
        Screen.ActiveControl.SelText = ""
    ElseIf TypeOf Screen.ActiveControl Is vsFlexGrid Then
        Screen.ActiveControl.Action = 23 'SS_ACTION_CLIPBOARD_CUT
    End If
End Sub

Private Sub mnuExit_Click()
    Unload frmMain
'    End
End Sub

Private Sub mnuNew_Click()
    If MsgBox("Warning. You are about to delete all current data.
Continue?", vbYesNo + vbExclamation + vbDefaultButton2) = vbYes Then
        ' ### Plant Data Tab ###
        txtFractionLarge.Text = ""
        txtFractionSmall.Text = ""
        txtFractionVerySmall.Text = ""
        txtLotMassMeanLarge.Text = ""
        txtLotMassMeanSmall.Text = ""
        txtLotMassMeanVerySmall.Text = ""
        txtLotMassSDLarge.Text = ""
        txtLotMassSDSmall.Text = ""
        txtLotMassSDVerySmall.Text = ""
        txtWipeEff.Text = ""
        txtSSOPEff.Text = ""
        txtEnhanceEff.Text = ""
        txtFCSTrigger.Text = ""
        txtNFCSLarge.Text = ""
        txtNFCSSmall.Text = ""
        txtNFCSTextVerySmall.Text = ""
        chkHoldFCSLarge.Value = vbUnchecked
        chkHoldFCSSmall.Value = vbUnchecked
        chkHoldFCSVerySmall.Value = vbUnchecked
        chkFCSPositiveActionTestLot.Value = vbUnchecked
        chkFCSPositiveActionClean.Value = vbUnchecked
        chkLotPositiveActionDispose.Value = vbUnchecked
        txtNLotLarge.Text = ""
        txtNLotSmall.Text = ""
        txtNLotVerySmall.Text = ""
        chkLotPositiveActionDispose.Value = vbChecked
        ' ### Contamination Tab ###
        txtCEAddMean.Text = ""
        txtCEAddStdDev.Text = ""
        txtCEMean.Text = ""
        txtCEStdDev.Text = ""
        txtDurationMean.Text = ""
        txtDurationStdDev.Text = ""
        txtTCMean.Text = ""
        txtTCStdDev.Text = ""
    End If
End Sub

```

```

txtFCSSwabMin.Text = ""
txtFCSSwabMax.Text = ""
txtFCSSwabsNComposited.Text = ""
txtRTESampledMin.Text = ""
txtRTESampledMax.Text = ""
' ### Post Processing Tab ###
txtPPFracLarge.Text = ""
txtPPFracSmall.Text = ""
txtPPFracVerySmall.Text = ""
txtPPMinReductLarge.Text = ""
txtPPMinReductSmall.Text = ""
txtPPMinReductVerySmall.Text = ""
txtPPMaxReductLarge.Text = ""
txtPPMaxReductSmall.Text = ""
txtPPMaxReductVerySmall.Text = ""
txtGIPFracLarge.Text = ""
txtGIPFracSmall.Text = ""
txtGIPFracVerySmall.Text = ""
txtGIPEfficMin.Text = ""
txtGIPEfficMax.Text = ""
' ### Advanced Data Tab ###
txtPDetectOneFCS.Text = ""
txtPDetectOneLot.Text = ""
txtLagReportFCS = ""
txtLagReportLot = ""
txtRatioMean.Text = ""
txtRatioStdDev.Text = ""
txtFCSAreaLargeMin.Text = ""
txtFCSAreaLargeMax.Text = ""
txtGrowth.Text = ""
' ### Simulation Tab ###
txtNLots.Text = ""
txtOutputFile.Text = ""
End If
End Sub

Private Sub mnuOpen_Click()
    Dim ErrHandlerOpen As Integer
    On Error GoTo ErrHandlerOpen
    dlgFile.Filter = "Data Files (*.dat)|*.dat|All Files (*.*)|*.*"
    dlgFile.DialogTitle = "Open File"
    dlgFile.Action = 1
    Call DataFileRead(dlgFile.FileName)
    gCurrentFileName = dlgFile.FileName
ErrHandlerOpen:
    Exit Sub
End Sub

Private Sub mnuOverview_click()
    MsgBox "This model performs a risk assessment for Listeria
monocytogenes based on food contact surface testing at the RTE
processing line."
End Sub

Private Sub mnuPaste_Click()
    If TypeOf Screen.ActiveControl Is TextBox Then
        Screen.ActiveControl.SelText = Clipboard.GetText()
    End If
End Sub

```

```

ElseIf TypeOf Screen.ActiveControl Is vsFlexGrid Then
    Screen.ActiveControl.Clip = Clipboard.GetText()
End If
End Sub

Private Sub mnuPrint_Click()
' this routine is incomplete and not called in the current version
Dim HoldTab As Long
HoldTab = tabLMRAMain.CurrTab
tabLMRAMain.CurrTab = 0
frmMain.PrintForm
tabLMRAMain.CurrTab = 1
frmMain.PrintForm
tabLMRAMain.CurrTab = 2
frmMain.PrintForm
tabLMRAMain.CurrTab = 3
frmMain.PrintForm
tabLMRAMain.CurrTab = 4
frmMain.PrintForm
' Print flxCalibration.Picture
' Printer.NewPage
tabLMRAMain.CurrTab = 6
frmMain.PrintForm
' Print flxOutputStats.Picture
' Printer.NewPage
Printer.EndDoc
tabLMRAMain.CurrTab = HoldTab
End Sub

Private Sub mnuSave_Click()
If gCurrentFileName <> "" Then
    Call DataFileSave(gCurrentFileName)
Else
    Call mnuSaveAs_Click
End If
End Sub

Private Sub mnuSaveAs_Click()
Dim ErrorHandlerSaveAs As Integer
On Error GoTo ErrorHandlerSaveAs
dlgFile.Filter = "Data files (*.dat)|*.dat|All Files (*.*)|*.*"
dlgFile.DialogTitle = "Save As"
dlgFile.Action = 2
Call DataFileSave(dlgFile.FileName)
gCurrentFileName = dlgFile.FileName
ErrorHandlerSaveAs:
Exit Sub
End Sub

Private Function newCEDelta(CEMean As Double, CEStdDev As Double) As Long
newCEDelta = Round(10 ^ RNNormal(CEMean, CEStdDev), 0)
End Function

Private Function newCEDuration(DurationMean As Double, DurationStdDev
As Double) As Long

```

```

    newCEDuration = Round(10 ^ RNNormal(DurationMean, DurationStdDev),
0)
End Function

Private Function NewFCSArea(ysize, FCSAreaMin As Double, FCSAreaMax As
Double) As Double
    NewFCSArea = LotMassMean(ysize) / LotMassMean(gLarge) *
RNUiform(FCSAreaMin, FCSAreaMax)
End Function

Private Function NewFCSLspp(CEAddMean As Double, CEAddStdDev As Double)
As Double
    NewFCSLspp = 10 ^ RNNormal(CEAddMean, CEAddStdDev)
End Function

Private Function NewGIPEffic(GIPEfficMin As Double, GIPEfficMax As
Double) As Double
    NewGIPEffic = RNUiform(GIPEfficMin, GIPEfficMax)
End Function

Private Function NewLotMass(LotMassMean As Double, LotMassStdDev As
Double) As Double
    NewLotMass = RNNormalTrunc(LotMassMean, LotMassStdDev, gMinLotMass,
gMaxLotMass)
End Function

Private Function NewLotMassSampled(LotMassSampledMin As Double,
LotMassSampledMax As Double) As Double
    NewLotMassSampled = RNUiform(LotMassSampledMin, LotMassSampledMax)
End Function

Private Function NewLsppLag()
    ' placeholder function in case decide later to make lag stochastic
    NewLsppLag = LagReportFCS
End Function

Private Function NewPPReduct(PPMinReduct As Double, PPMaReduct As
Double) As Double
    NewPPReduct = RNUiform(PPMinReduct, PPMaReduct)
End Function

Private Function NewRatio(RatioMean As Double, RatioStdDev As Double)
As Double
    NewRatio = RNNormalTrunc(RatioMean, RatioStdDev, 0, 1)
End Function

Private Function NewTC(TCMean As Double, TCStdDev As Double) As Double
    NewTC = 10 ^ (RNNormal(TCMean, TCStdDev))
    NewTC = mind(NewTC, 1#)
End Function

Private Function NewSwabArea(SwabAreaMin As Double, SwabAreaMax As
Double) As Double
    NewSwabArea = RNUiform(SwabAreaMin, SwabAreaMax)
End Function

Private Sub optCalibrateDeliMeats_Click()

```

```

    Call CalibrateDeli
    lblLogSSR.Caption = ""
End Sub

Private Sub optCalibrateFrankfurters_Click()
    Call CalibrateFrankfurters
    lblLogSSR.Caption = ""
End Sub

Private Sub optGraphLMRetail_Click()
    Call OutputGraph
End Sub

Private Sub optGraphLsppFCS_Click()
    Call OutputGraph
End Sub

Private Sub outputCalibration()
'ReDim log10LMRetail(1 To NLotsUsed)
Const NPercentiles = 8
Dim logssr As Double
Dim ysort() As Double
ReDim ysort(1 To NLotsUsed)
Dim ave As Double, adev As Double, sdev As Double, var As Double, skew
As Double, curt As Double
Dim i As Long
    For i = 1 To NLotsUsed
        ysort(i) = lotLMRetailSelected(i)
    Next i
    Call sort(NLotsUsed, ysort())
    flxCalibration.TextMatrix(1, 2) = Format$(ysort(Int(0.8 *
NLotsUsed)), "Scientific")
    flxCalibration.TextMatrix(2, 2) = Format$(ysort(Int(0.85 *
NLotsUsed)), "Scientific")
    flxCalibration.TextMatrix(3, 2) = Format$(ysort(Int(0.9 *
NLotsUsed)), "Scientific")
    flxCalibration.TextMatrix(4, 2) = Format$(ysort(Int(0.95 *
NLotsUsed)), "Scientific")
    flxCalibration.TextMatrix(5, 2) = Format$(ysort(Int(0.99 *
NLotsUsed)), "Scientific")
    flxCalibration.TextMatrix(6, 2) = Format$(ysort(Int(0.995 *
NLotsUsed)), "Scientific")
    flxCalibration.TextMatrix(7, 2) = Format$(ysort(Int(0.999 *
NLotsUsed)), "Scientific")
    flxCalibration.TextMatrix(8, 2) = Format$(ysort(Int(0.9999 *
NLotsUsed)), "Scientific")
    flxCalibration.TextMatrix(9, 2) = Format$(ysort(NLotsUsed),
"Scientific")
    logssr = 0#
    ' Debug.Print "Percentiles"
    For i = 1 To NPercentiles
        logssr = logssr + (log10(flxCalibration.ValueMatrix(i, 1)) -
log10(flxCalibration.ValueMatrix(i, 2))) ^ 2
    Next i
    lblLogSSR.Caption = Format$(logssr, "Scientific")
    ReDim ysort(1 To NLots)
    For i = 1 To NLots

```



```

        ysort(i) = lotLMPP(gLarge, i)
    Next i
    Call sort(NLots, ysort())
    flxCalibration.TextMatrix(1, 3) = Format$(ysort(Int(0.8 * NLots)),
"Scientific")
    flxCalibration.TextMatrix(2, 3) = Format$(ysort(Int(0.85 * NLots)),
"Scientific")
    flxCalibration.TextMatrix(3, 3) = Format$(ysort(Int(0.9 * NLots)),
"Scientific")
    flxCalibration.TextMatrix(4, 3) = Format$(ysort(Int(0.95 * NLots)),
"Scientific")
    flxCalibration.TextMatrix(5, 3) = Format$(ysort(Int(0.99 * NLots)),
"Scientific")
    flxCalibration.TextMatrix(6, 3) = Format$(ysort(Int(0.995 * NLots)),
"Scientific")
    flxCalibration.TextMatrix(7, 3) = Format$(ysort(Int(0.999 * NLots)),
"Scientific")
    flxCalibration.TextMatrix(8, 3) = Format$(ysort(Int(0.9999 *
NLots)), "Scientific")
    flxCalibration.TextMatrix(9, 3) = Format$(ysort(NLots),
"Scientific") '
End Sub

```

```

Private Sub OutputFileSave(fname As String)
Dim fnum As Integer, ilot As Long, iday As Long, isize As Long
    fnum = FreeFile
    If fname = "" Then
        'MsgBox "Warning! No Output file name provided."
    Else
        Open App.Path + "\" + fname For Output As fnum
        Write #fnum, "Plant Size", "Lot", _
            "Day", "Contamintion Day?", _
            "LotMass", "LotMassSampled", _
            "FCSArea", "FCSSanitationEffic", _
            "CEAdded", "FCSLsspAtEndLot", _
            "FCSSampled", "FCSSwabArea", _
            "FCSResult", "Sampled by Rule", _
            "Sampled by FCS", "TC", _
            "Ratio", "Lssp", _
            "LM", "LotResult", _
            "PPReduct", "LM PP", _
            "GIPEffic", "LMRetail"
        For isize = 1 To NSizes
            For ilot = 1 To NLots
                Write #fnum, isize, ilot, _
                    lotDay(isize, ilot), ContamDay(isize,
lotDay(isize, ilot)), _
                    lotMass(isize, ilot), lotMassSampled(isize,
ilot), _
                    lotFCSArea(isize, ilot), lotFCSDisinf(isize,
ilot), _
                    lotCEAdded(isize, ilot), lotFCSLssp(isize,
ilot), _
                    FCSSampled(isize, ilot), FCSSwabArea(isize,
ilot), _
                    lotFCSResult(isize, ilot), LotSampled(isize,
ilot), _

```

```

lotSampledbyFCS(ysize, ilot), lotTC(ysize,
ilot), _
lotRatio(ysize, ilot), lotLspp(ysize, ilot), _
lotLM(ysize, ilot), lotLMResult(ysize, ilot), _
PPReduct(ysize, ilot), lotLMPP(ysize, ilot), _
GIPEffic(ysize, ilot), lotLMRetail(ysize, ilot)
Next ilot
Next ysize
Close #fnum
End If
End Sub

Private Sub OutputGraph()
Dim i As Long, iused As Long, ilot As Long
Dim CumProb() As Double, ylog() As Double, ysort() As Double
ReDim CumProb(1 To NLots)
ReDim ylog(1 To NLots)
'ReDim lotLsppFCS(1 To NLots)
ReDim ysort(1 To NLots)
'Debug.Print "Graphs"
If optGraphLMRetail Then
For i = 1 To NLotsUsed
ysort(i) = lotLMRetailSelected(i)
Next i
If NLotsUsed < 32000 Then
Call sort(NLotsUsed, ysort())
ilot = 1
For iused = 1 To NLotsUsed
If ysort(iused) > 0 Then
CumProb(ilot) = (iused - 0.5) / NLotsUsed
ylog(ilot) = log10(ysort(iused))
ilot = ilot + 1
End If
Next iused
ilot = ilot - 1
Call PlotGraph(ilot, ylog(), CumProb(), "log10 Lm Conc
(cfu/gm) at Retail", "Cumulative Probability")
End If
ElseIf optGraphLsppFCS Then
For i = 1 To NLots
ysort(i) = lotFCSLspp(gLarge, i)
Next i
If NLots < 32000 Then
Call sort(NLots, ysort())
ilot = 1
For i = 1 To NLots
If ysort(i) > 0 Then
CumProb(ilot) = (i - 0.5) / NLots
ylog(ilot) = log10(ysort(i))
ilot = ilot + 1
End If
Next i
ilot = ilot - 1
Call PlotGraph(ilot, ylog(), CumProb(), "log10 Lspp Conc
(cfu/cm^2) on FCS", "Cumulative Probability")
End If
End If
End Sub

```

```

End Sub

Private Sub OutputStatsToScreen()
Const iPos = 1
Const iNeg = 2
Const iMiss = 3
Dim ilot As Long, isize As Long, i As Long, j As Long, klot As Long,
lag As Long
Dim nFCSPosLotPos As Long, nFCSPosLotNeg As Long
Dim nFCSNegLotPos As Long, nFCSNegLotNeg As Long
Dim NFCSSResult() As Long
Dim NFCSSampled() As Long
Dim NLotSampledByRule() As Long
Dim NLotSampledbyFCS() As Long
Dim NLotSampledTotal() As Long
ReDim NLotLMResult(1 To NSizes)
ReDim NFCSSResult(1 To NSizes)
ReDim NFCSSampled(1 To NSizes)
ReDim NLotSampledByRule(1 To NSizes)
ReDim NLotSampledbyFCS(1 To NSizes)
ReDim NLotSampledTotal(1 To NSizes)
ReDim NLotLMResult(1 To NSizes)
ReDim xtab(1 To 3, 1 To 3)
  For isize = 1 To NSizes
    For ilot = 1 To NLots
      If FCSSampled(isize, ilot) Then
        NFCSSampled(isize) = NFCSSampled(isize) + 1
      End If
      If lotFCSSResult(isize, ilot) Then
        NFCSSResult(isize) = NFCSSResult(isize) + 1
      End If
      If LotSampled(isize, ilot) Then
        NLotSampledByRule(isize) = NLotSampledByRule(isize) + 1
      End If
      If lotSampledbyFCS(isize, ilot) Then
        NLotSampledbyFCS(isize) = NLotSampledbyFCS(isize) + 1
      End If
      If LotSampled(isize, ilot) Or lotSampledbyFCS(isize, ilot)
Then
        NLotSampledTotal(isize) = NLotSampledTotal(isize) + 1
      End If
      If lotLMResult(isize, ilot) Then
        NLotLMResult(isize) = NLotLMResult(isize) + 1
      End If
    Next ilot
  Next isize

  flxOutputStats.TextMatrix(1, 1) = NLots
  flxOutputStats.TextMatrix(1, 2) = NLots
  flxOutputStats.TextMatrix(1, 3) = NLots

  flxOutputStats.TextMatrix(2, 1) = nLotsThisSize(gLarge)
  flxOutputStats.TextMatrix(2, 2) = nLotsThisSize(gSmall)
  flxOutputStats.TextMatrix(2, 3) = nLotsThisSize(gVerySmall)

  flxOutputStats.TextMatrix(3, 1) = NLotUsedThisSize(gLarge)
  flxOutputStats.TextMatrix(3, 2) = NLotUsedThisSize(gSmall)

```

```

flxOutputStats.TextMatrix(3, 3) = NLotUsedThisSize(gVerySmall)
flxOutputStats.TextMatrix(3, 4) = NLotsUsed

flxOutputStats.TextMatrix(4, 1) = NLotSampledTotal(gLarge)
flxOutputStats.TextMatrix(4, 2) = NLotSampledTotal(gSmall)
flxOutputStats.TextMatrix(4, 3) = NLotSampledTotal(gVerySmall)

flxOutputStats.TextMatrix(5, 1) = NLotSampledByRule(gLarge)
flxOutputStats.TextMatrix(5, 2) = NLotSampledByRule(gSmall)
flxOutputStats.TextMatrix(5, 3) = NLotSampledByRule(gVerySmall)

flxOutputStats.TextMatrix(6, 1) = NLotSampledbyFCS(gLarge)
flxOutputStats.TextMatrix(6, 2) = NLotSampledbyFCS(gSmall)
flxOutputStats.TextMatrix(6, 3) = NLotSampledbyFCS(gVerySmall)

flxOutputStats.TextMatrix(7, 1) = NLotLMResult(gLarge)
flxOutputStats.TextMatrix(7, 2) = NLotLMResult(gSmall)
flxOutputStats.TextMatrix(7, 3) = NLotLMResult(gVerySmall)

flxOutputStats.TextMatrix(8, 4) = flxCalibration.TextMatrix(3, 2)
flxOutputStats.TextMatrix(9, 4) = flxCalibration.TextMatrix(4, 2)
flxOutputStats.TextMatrix(10, 4) = flxCalibration.TextMatrix(5, 2)

flxOutputStats.TextMatrix(11, 1) = NFCSSampled(gLarge)
flxOutputStats.TextMatrix(11, 2) = NFCSSampled(gSmall)
flxOutputStats.TextMatrix(11, 3) = NFCSSampled(gVerySmall)

flxOutputStats.TextMatrix(12, 1) = NFCSResult(gLarge)
flxOutputStats.TextMatrix(12, 2) = NFCSResult(gSmall)
flxOutputStats.TextMatrix(12, 3) = NFCSResult(gVerySmall)

For ilot = 1 To NLots
    isize = gLarge
    If FCSSampled(isize, ilot) And lotFCSResult(isize, ilot) And
(LotSampled(isize, ilot) Or lotSampledbyFCS(isize, ilot)) And
lotLMResult(isize, ilot) Then
        xtab(iPos, iPos) = xtab(iPos, iPos) + 1
    ElseIf FCSSampled(isize, ilot) And lotFCSResult(isize, ilot) And
(LotSampled(isize, ilot) Or lotSampledbyFCS(isize, ilot)) And Not
lotLMResult(isize, ilot) Then
        xtab(iPos, iNeg) = xtab(iPos, iNeg) + 1
    ElseIf FCSSampled(isize, ilot) And lotFCSResult(isize, ilot) And
Not (LotSampled(isize, ilot) Or lotSampledbyFCS(isize, ilot)) Then
        xtab(iPos, iMiss) = xtab(iPos, iMiss) + 1
    ElseIf FCSSampled(isize, ilot) And Not lotFCSResult(isize, ilot)
And (LotSampled(isize, ilot) Or lotSampledbyFCS(isize, ilot)) And
lotLMResult(isize, ilot) Then
        xtab(iNeg, iPos) = xtab(iNeg, iPos) + 1
    ElseIf FCSSampled(isize, ilot) And Not lotFCSResult(isize, ilot)
And (LotSampled(isize, ilot) Or lotSampledbyFCS(isize, ilot)) And Not
lotLMResult(isize, ilot) Then
        xtab(iNeg, iNeg) = xtab(iNeg, iNeg) + 1
    ElseIf FCSSampled(isize, ilot) And Not lotFCSResult(isize, ilot)
And Not (LotSampled(isize, ilot) Or lotSampledbyFCS(isize, ilot)) Then
        xtab(iNeg, iMiss) = xtab(iNeg, iMiss) + 1
    ElseIf Not FCSSampled(isize, ilot) And (LotSampled(isize, ilot)
Or lotSampledbyFCS(isize, ilot)) And lotLMResult(isize, ilot) Then

```

```

        xtab(iMiss, iPos) = xtab(iMiss, iPos) + 1
    ElseIf Not FCSSampled(isize, ilot) And (LotSampled(isize, ilot)
Or lotSampledbyFCS(isize, ilot)) And Not lotLMResult(isize, ilot) Then
        xtab(iMiss, iNeg) = xtab(iMiss, iNeg) + 1
    ElseIf Not FCSSampled(isize, ilot) And Not (LotSampled(isize,
ilot) Or lotSampledbyFCS(isize, ilot)) Then
        xtab(iMiss, iMiss) = xtab(iMiss, iMiss) + 1
    End If
Next ilot
For i = 1 To 3
    For j = 1 To 3
        flxContingency.TextMatrix(i, j) = xtab(i, j)
    Next j
Next i
ReDim xtab(1 To 3, 1 To 3)
For ilot = 1 To NLots
    lag = LagReportFCS
    klot = ilot + lag
    If klot <= NLots Then
        If FCSSampled(isize, ilot) And lotFCSResult(isize, ilot) And
(LotSampled(isize, klot) Or lotSampledbyFCS(isize, klot)) And
lotLMResult(isize, klot) Then
            xtab(iPos, iPos) = xtab(iPos, iPos) + 1
        ElseIf FCSSampled(isize, ilot) And lotFCSResult(isize, ilot)
And (LotSampled(isize, klot) Or lotSampledbyFCS(isize, klot)) And Not
lotLMResult(isize, klot) Then
            xtab(iPos, iNeg) = xtab(iPos, iNeg) + 1
        ElseIf FCSSampled(isize, ilot) And lotFCSResult(isize, ilot)
And Not (LotSampled(isize, klot) Or lotSampledbyFCS(isize, klot)) Then
            xtab(iPos, iMiss) = xtab(iPos, iMiss) + 1
        ElseIf FCSSampled(isize, ilot) And Not lotFCSResult(isize,
ilot) And (LotSampled(isize, klot) Or lotSampledbyFCS(isize, klot)) And
lotLMResult(isize, klot) Then
            xtab(iNeg, iPos) = xtab(iNeg, iPos) + 1
        ElseIf FCSSampled(isize, ilot) And Not lotFCSResult(isize,
ilot) And (LotSampled(isize, klot) Or lotSampledbyFCS(isize, klot)) And
Not lotLMResult(isize, klot) Then
            xtab(iNeg, iNeg) = xtab(iNeg, iNeg) + 1
        ElseIf FCSSampled(isize, ilot) And Not lotFCSResult(isize,
ilot) And Not (LotSampled(isize, klot) Or lotSampledbyFCS(isize, klot))
Then
            xtab(iNeg, iMiss) = xtab(iNeg, iMiss) + 1
        ElseIf Not FCSSampled(isize, ilot) And (LotSampled(isize,
klot) Or lotSampledbyFCS(isize, klot)) And lotLMResult(isize, klot)
Then
            xtab(iMiss, iPos) = xtab(iMiss, iPos) + 1
        ElseIf Not FCSSampled(isize, ilot) And (LotSampled(isize,
klot) Or lotSampledbyFCS(isize, klot)) And Not lotLMResult(isize, klot)
Then
            xtab(iMiss, iNeg) = xtab(iMiss, iNeg) + 1
        ElseIf Not FCSSampled(isize, ilot) And Not (LotSampled(isize,
klot) Or lotSampledbyFCS(isize, klot)) Then
            xtab(iMiss, iMiss) = xtab(iMiss, iMiss) + 1
        End If
    End If
Next ilot
For i = 1 To 3

```

```

        For j = 1 To 3
            flxContingencyLag.TextMatrix(i, j) = xtab(i, j)
        Next j
    Next i
End Sub

Private Sub PlotGraph(npoints As Long, x() As Double, y() As Double,
xtitle As String, ytitle As String)
Dim i As Long
    grfOutput.GraphType = 9
    grfOutput.GraphStyle = 0
    grfOutput.BottomTitle = xtitle
    grfOutput.GraphTitle = ytitle
    ' npoints = UBound(x) - LBound(x)
    grfOutput.NumSets = 1
    grfOutput.NumPoints = npoints
    grfOutput.Color(1) = 1
    grfOutput.DrawMode = 0
    For i = 1 To npoints
        grfOutput.XPos(i) = x(i)
        grfOutput.Data(i) = y(i)
    Next i
    grfOutput.DrawMode = 3
End Sub

Public Sub PostProcess()
Dim isize As Long, ilot As Long
Dim GIPGrowth As Double
    For isize = 1 To NSizes
        For ilot = 1 To NLots
            ' reduce lm lot concentration if post processing
            If Rnd < PPFrac(isize) Then
                PPreduct(isize, ilot) = NewPPReduct(PPMinReduct(isize),
PPMaxReduct(isize))
            Else
                PPreduct(isize, ilot) = 0#
            End If
            lotLMPP(isize, ilot) = lotLM(isize, ilot) * (1# - PPreduct(isize,
ilot))
            If LotSampled(isize, ilot) Or lotSampledbyFCS(isize, ilot) Then
                lotLMResult(isize, ilot) = TestLotResult(lotLMPP(isize, ilot),
lotMassSampled(isize, ilot), PDetectOneLot)
            Else
                lotLMResult(isize, ilot) = False
            End If
            ' calculate efficiency for growth inhibiting packaging
            If Rnd < GIPFrac(isize) Then
                GIPEffic(isize, ilot) = NewGIPEffic(GIPEfficMin, GIPEfficMax)
            Else
                GIPEffic(isize, ilot) = 0#
            End If
            GIPGrowth = Growth + log10(1 - GIPEffic(isize, ilot))
            ' calculate final retail concentration
            lotLMRetail(isize, ilot) = lotLMPP(isize, ilot) * 10 ^ GIPGrowth
        Next ilot
    Next isize
End Sub

```

```

Private Sub ScreenDataRead()
' reads the data from the screens and converts to variables for the
model
' ### Plant Data Tab ###
Fraction(gLarge) = Val(txtFractionLarge.Text)
Fraction(gSmall) = Val(txtFractionSmall.Text)
Fraction(gVerySmall) = Val(txtFractionVerySmall.Text)
LotMassMean(gLarge) = Val(txtLotMassMeanLarge.Text)
LotMassMean(gSmall) = Val(txtLotMassMeanSmall.Text)
LotMassMean(gVerySmall) = Val(txtLotMassMeanVerySmall.Text)
LotMassStdDev(gLarge) = Val(txtLotMassSDLarge.Text)
LotMassStdDev(gSmall) = Val(txtLotMassSDSmall.Text)
LotMassStdDev(gVerySmall) = Val(txtLotMassSDVerySmall.Text)
EfficSOP = Val(txtSSOPEff.Text)
EfficWipe = Val(txtWipeEff.Text)
EfficEnhance = Val(txtEnhanceEff.Text)
FCSTrigger = Val(txtFCSTrigger.Text)
NSamplesFCS(gLarge) = Val(txtNFCSLarge.Text)
NSamplesFCS(gSmall) = Val(txtNFCSSmall.Text)
NSamplesFCS(gVerySmall) = Val(txtNFCSText)
If chkHoldFCSLarge.Value = vbChecked Then
    HoldFCS(gLarge) = vbChecked
Else
    HoldFCS(gLarge) = vbUnchecked
End If
If chkHoldFCSSmall.Value = vbChecked Then
    HoldFCS(gSmall) = vbChecked
Else
    HoldFCS(gSmall) = vbUnchecked
End If
If chkHoldFCSVerySmall.Value = vbChecked Then
    HoldFCS(gVerySmall) = vbChecked
Else
    HoldFCS(gVerySmall) = vbUnchecked
End If
FCSPositiveActionTestLot = chkFCSPositiveActionTestLot.Value
FCSPositiveActionClean = chkFCSPositiveActionClean.Value
LotPositiveActionDispose = chkLotPositiveActionDispose.Value
NSamplesLot(gLarge) = Val(txtNLotLarge.Text)
NSamplesLot(gSmall) = Val(txtNLotSmall.Text)
NSamplesLot(gVerySmall) = Val(txtNLotVerySmall.Text)
' ### Contamination Data Tab ###
CEMean = Val(txtCEMean.Text)
CEStdDev = Val(txtCEStdDev.Text)
DurationMean = Val(txtDurationMean.Text)
DurationStdDev = Val(txtDurationStdDev.Text)
CEAddMean = Val(txtCEAddMean.Text)
CEAddStdDev = Val(txtCEAddStdDev.Text)
TCMean = Val(txtTCMean.Text)
TCStdDev = Val(txtTCStdDev.Text)
FCSSwabMin = Val(txtFCSSwabMin.Text)
FCSSwabMax = Val(txtFCSSwabMax.Text)
FCSSwabsNComposited = Val(txtFCSSwabsNComposited.Text)
RTESampledMin = Val(txtRTESampledMin.Text)
RTESampledMax = Val(txtRTESampledMax.Text)
' ### Post processing tab ###

```

```

PPFrac(gLarge) = Val(txtPPFracLarge.Text)
PPFrac(gSmall) = Val(txtPPFracSmall.Text)
PPFrac(gVerySmall) = Val(txtPPFracVerySmall.Text)
PPMinReduct(gLarge) = Val(txtPPMinReductLarge.Text)
PPMinReduct(gSmall) = Val(txtPPMinReductSmall.Text)
PPMinReduct(gVerySmall) = Val(txtPPMinReductVerySmall.Text)
PPMaxReduct(gLarge) = Val(txtPPMaxReductLarge.Text)
PPMaxReduct(gSmall) = Val(txtPPMaxReductSmall.Text)
PPMaxReduct(gVerySmall) = Val(txtPPMaxReductVerySmall.Text)
GIPFrac(gLarge) = Val(txtGIPFracLarge.Text)
GIPFrac(gSmall) = Val(txtGIPFracSmall.Text)
GIPFrac(gVerySmall) = Val(txtGIPFracVerySmall.Text)
GIPEfficMin = Val(txtGIPEfficMin.Text)
GIPEfficMax = Val(txtGIPEfficMax.Text)
' ### Advanced Data Tab ###
PDetectOneFCS = Val(txtPDetectOneFCS.Text)
PDetectOneLot = Val(txtPDetectOneLot.Text)
LagReportFCS = Val(txtLagReportFCS.Text)
LagReportLot = Val(txtLagReportLot.Text)
RatioMean = Val(txtRatioMean.Text)
RatioStdDev = Val(txtRatioStdDev.Text)
FCSAreaLargeMin = Val(txtFCSAreaLargeMin.Text)
FCSAreaLargeMax = Val(txtFCSAreaLargeMax.Text)
Growth = Val(txtGrowth.Text)
' ### Simulation Tab ###
NLots = Val(txtNLots.Text)
gCurrentFileName = txtOutputFile.Text
End Sub

Private Sub ScreenDataSet()
' Writes the variables to the screen text boxes
' ### Plant Data Tab ###
txtFractionLarge.Text = Fraction(gLarge)
txtFractionSmall.Text = Fraction(gSmall)
txtFractionVerySmall.Text = Fraction(gVerySmall)
txtLotMassMeanLarge.Text = LotMassMean(gLarge)
txtLotMassMeanSmall.Text = LotMassMean(gSmall)
txtLotMassMeanVerySmall.Text = LotMassMean(gVerySmall)
txtLotMassSDLarge.Text = LotMassStdDev(gLarge)
txtLotMassSDSmall.Text = LotMassStdDev(gSmall)
txtLotMassSDVerySmall.Text = LotMassStdDev(gVerySmall)
txtWipeEff.Text = EfficWipe
txtSSOPEff.Text = EfficSOP
txtEnhanceEff.Text = EfficEnhance
txtFCSTrigger.Text = FCSTrigger
txtNFCSLarge.Text = NSamplesFCS(gLarge)
txtNFCSSmall.Text = NSamplesFCS(gSmall)
txtNFCSVerySmall.Text = NSamplesFCS(gVerySmall)
chkHoldFCSLarge.Value = HoldFCS(gLarge)
chkHoldFCSSmall.Value = HoldFCS(gSmall)
chkHoldFCSVerySmall.Value = HoldFCS(gVerySmall)
chkFCSPositiveActionTestLot.Value = FCSPositiveActionTestLot
chkFCSPositiveActionClean.Value = FCSPositiveActionClean
chkLotPositiveActionDispose.Value = LotPositiveActionDispose
txtNLotLarge.Text = NSamplesLot(gLarge)
txtNLotSmall.Text = NSamplesLot(gSmall)
txtNLotVerySmall.Text = NSamplesLot(gVerySmall)

```



```

' ### Contamination Data Tab ###
txtCEAddMean.Text = CEAddMean
txtCEAddStdDev.Text = CEAddStdDev
txtCEMean.Text = CEMean
txtCEStdDev.Text = CEStdDev
txtDurationMean.Text = DurationMean
txtDurationStdDev.Text = DurationStdDev
txtTCMean.Text = TCMean
txtTCStdDev.Text = TCStdDev
txtFCSSwabMin.Text = FCSSwabMin
txtFCSSwabMax.Text = FCSSwabMax
txtFCSSwabsNComposited.Text = FCSSwabsNComposited
txtRTESampledMin.Text = RTESampledMin
txtRTESampledMax.Text = RTESampledMax
' ### Post processing tab ###
txtPPFracLarge.Text = PPFrac(gLarge)
txtPPFracSmall.Text = PPFrac(gSmall)
txtPPFracVerySmall.Text = PPFrac(gVerySmall)
txtPPMinReductLarge.Text = PPMinReduct(gLarge)
txtPPMinReductSmall.Text = PPMinReduct(gSmall)
txtPPMinReductVerySmall.Text = PPMinReduct(gVerySmall)
txtPPMaxReductLarge.Text = PPMMaxReduct(gLarge)
txtPPMaxReductSmall.Text = PPMMaxReduct(gSmall)
txtPPMaxReductVerySmall.Text = PPMMaxReduct(gVerySmall)
txtGIPFracLarge.Text = GIPFrac(gLarge)
txtGIPFracSmall.Text = GIPFrac(gSmall)
txtGIPFracVerySmall.Text = GIPFrac(gVerySmall)
txtGIPEfficMin.Text = GIPEfficMin
txtGIPEfficMax.Text = GIPEfficMax
' ### Advanced Data Tab ###
txtPDetectOneFCS.Text = PDetectOneFCS
txtPDetectOneLot.Text = PDetectOneLot
txtLagReportFCS.Text = LagReportFCS
txtLagReportLot.Text = LagReportLot
txtRatioMean.Text = RatioMean
txtRatioStdDev.Text = RatioStdDev
txtFCSAreaLargeMin.Text = FCSAreaLargeMin
txtFCSAreaLargeMax.Text = FCSAreaLargeMax
txtGrowth.Text = Growth
' ### Simulation Tab ###
txtNLots.Text = NLots
txtOutputFile = gCurrentFileName

```

End Sub

Private Sub SelectLotsToUse()

Dim sum As Long, rndstart As Long

Dim i As Long, j As Long, k As Long, isize As Long, ilot As Long, iused

sum = 0

For isize = 1 To (NSizes - 1)

nLotsThisSize(isize) = Round(Fraction(isize) * NLots)

sum = sum + nLotsThisSize(isize)

Next isize

nLotsThisSize(NSizes) = max(0, NLots - sum)

k = 1

For isize = 1 To NSizes

rndstart = RNIntBtw(MinStart, NLots)

For j = 1 To nLotsThisSize(isize)

```

1)         ilot = MinStart + (rndstart + j - 1) Mod (NLots - MinStart +
LotToUse(k, 1) = isize
LotToUse(k, 2) = ilot
k = k + 1
Next j
Next isize
iused = 0
If LotPositiveActionDispose = vbChecked Then
    For ilot = 1 To NLots
        ' don't let positive lots into retail if this check box is
selected
        If Not lotLMResult(LotToUse(ilot, 1), LotToUse(ilot, 2)) Then
            iused = iused + 1
            lotLMRetailSelected(iused) = lotLMRetail(LotToUse(ilot, 1),
LotToUse(ilot, 2))
            NLotUsedThisSize(LotToUse(ilot, 1)) =
NLotUsedThisSize(LotToUse(ilot, 1)) + 1
        End If
    Next ilot
    NLotsUsed = iused
Else
    For ilot = 1 To NLots
        lotLMRetailSelected(ilot) = lotLMRetail(LotToUse(ilot, 1),
LotToUse(ilot, 2))
        NLotUsedThisSize(LotToUse(ilot, 1)) =
NLotUsedThisSize(LotToUse(ilot, 1)) + 1
    Next ilot
    NLotsUsed = NLots
End If
End Sub

```

```

Public Sub sort(n As Long, ra() As Double)
' taken from Numerical Recipes
Dim el As Long, ir As Long, i As Long, j As Long
Dim rra As Double
el = Int(n / 2) + 1
ir = n
Do
    If el > 1 Then
        el = el - 1
        rra = ra(el)
    Else
        rra = ra(ir)
        ra(ir) = ra(1)
        ir = ir - 1
        If ir = 1 Then
            ra(1) = rra
            Exit Sub
        End If
    End If
End Do
i = el
j = el + el
Do While j <= ir
    If j < ir Then If ra(j) < ra(j + 1) Then j = j + 1
    If rra < ra(j) Then
        ra(i) = ra(j)
    End If
    j = j + 1
End Do

```

```

        i = j
        j = j + j
    Else
        j = ir + 1
    End If
Loop
ra(i) = rra
Loop
End Sub

```

```

Private Function TestFCSResult(LsppConc As Double, SwabSurfArea As
Double, pDetectOneLspp As Double) As Boolean
' returns the test result on FCS
' assumes uniformly spread across entire area
' Need SurfArea = # of cm^2 constituting a sample
' Need PDetectOne = probability of culturing protocol correctly
identifying a sample containing a single bacterium
Dim LsppInSamp As Double
    LsppInSamp = (RNpoisson(SwabSurfArea * LsppConc)) 'this gives number
of bacteria in sample
    If LsppInSamp = 0 Then
        TestFCSResult = False
    Else
        If (1 - pDetectOneLspp) ^ LsppInSamp < Rnd Then
            TestFCSResult = True
        Else
            TestFCSResult = False
        End If
    End If
End Function

```

```

Private Function TestLotResult(LMConc As Double, SampleMass As Double,
pDetectOneLM As Double) As Boolean
Dim LMInSamp As Double
    LMInSamp = (RNpoisson(SampleMass * LMConc)) 'this gives number of
bacteria in sample
    If LMInSamp = 0 Then
        TestLotResult = False
    Else
        If (1 - pDetectOneLM) ^ LMInSamp < Rnd Then
            TestLotResult = True
        Else
            TestLotResult = False
        End If
    End If
End Function

```

Source Code
MathFuncs.Bas
Version 1.00

```

Attribute VB_Name = "MathFuncs"
Option Explicit

Public Function log10(x As Double) As Double
    log10 = Log(x) / Log(10#)
End Function

Public Function maxd(x As Double, y As Double) As Double
    If x > y Then
        maxd = x
    Else
        maxd = y
    End If
End Function

Public Function mind(x As Double, y As Double) As Double
    If x < y Then
        mind = x
    Else
        mind = y
    End If
End Function

Public Function maxl(x As Long, y As Long) As Long
    If x > y Then
        maxl = x
    Else
        maxl = y
    End If
End Function

Public Function minl(x As Long, y As Long) As Long
    If x < y Then
        minl = x
    Else
        minl = y
    End If
End Function

Public Function RNIntBtw(min As Long, max As Long) As Long
    RNIntBtw = Int((max - min + 1) * Rnd + min)
End Function

Public Function RNLnNormal(theta As Double, tau As Double) As Double
' See Environmental Statistics by Millard and Neerchal
' theta is the mean on the linear scale
' tau is the standard deviation on the linear scale
Dim cv As Double, mu As Double, sigma As Double
    cv = tau / theta
    mu = Log(theta / (cv ^ 2 + 1) ^ 0.5)
    sigma = (Log(cv ^ 2 + 1)) ^ 0.5
    RNLnNormal = Exp(RNNormal(mu, sigma))
End Function

Public Static Function RNPoisson(xm As Double) As Double
' Based on Numerical Recipes POIDEV function
' Returns as a floating point an integer value that is a random

```

```

' deviate drawn from a Poisson distribution of mean xm,
' using built-in rnd() as a source of uniform random deviates.
Const PI = 3.14159265358979
Dim sq As Double, alxm As Double, g As Double, oldm As Double
Dim em As Double, t As Double, y As Double
  If xm < 12# Then
    If xm <> oldm Then
      oldm = xm
      g = Exp(-xm)
    End If
    em = -1#
    t = 1#
    Do
      em = em + 1#
      t = t * Rnd()
    Loop While t > g
  Else
    If xm <> oldm Then
      oldm = xm
      sq = Sqr(2# * xm)
      alxm = Log(xm)
      g = xm * alxm - rnPoisson_gammln(xm + 1#)
    End If
    Do
      y = Tan(PI * Rnd())
      em = sq * y + xm
      Loop While em < 0#
      em = Int(em)
      t = 0.9 * (1# + y ^ 2) * Exp(em * alxm - rnPoisson_gammln(em +
1#) - g)
      Loop While Rnd() > t
    End If
    RNpoisson = em
  End Function

```