

```
Public Class clsChannelInfo
```

```
Private pvName As String
Private pvUnit As String
Private pvColor As System.Drawing.Color
Private pvDataFormat As String
Private pvAlarmHi2 As Single
Private pvAlarmHi1 As Single
Private pvAlarmLo1 As Single
Private pvAlarmLo2 As Single
Private pvDataMax As Single
Private pvDataMin As Single
Private pvDataNow As Single
Private pvRegister As Integer
Private pvCoefficientA As Single
Private pvCoefficientB As Single
```

```
・列挙体を宣言
```

```
Enum eAlarmLevel
    ealNormal = 0 '正常
    ealAttention = 1 '注意
    ealWarning = 2 '警報
    ealWarning2 = 3 '警報(CO2)
End Enum
```

```
Public Sub Initialize()
    pvName = ""
    pvUnit = ""
    pvColor = System.Drawing.Color.Black
    pvDataFormat = "###0.00"
    pvDataMax = 100
    pvDataMin = 0
    pvAlarmHi2 = 90
    pvAlarmHi1 = 80
    pvAlarmLo1 = 20
    pvAlarmLo2 = 10
    pvDataNow = NullData
    pvRegister = 0
    pvCoefficientA = 1
    pvCoefficientB = 0
End Sub
```

```
Public Property Name() As String
    Get
        Return pvName
    End Get
    Set(ByVal value As String)
        pvName = value
    End Set
End Property
Public Property Unit() As String
    Get
        Return pvUnit
    End Get
End Property
```

甲第3号証

```
End Get
Set(ByVal value As String)
    pvUnit = value
End Set
End Property
Public Property DataFormat() As String
Get
    Return pvDataFormat
End Get
Set(ByVal value As String)
    pvDataFormat = value
End Set
End Property
Public Function Format(ByVal vData As Single) As String
    Dim str As String, mLen As Integer, mFormat As String
    mLen = Len(pvDataFormat)
    If mLen < 5 Then mLen = 5
    If IsNullData(vData) Then
        str = Space(mLen)
    Else
        mFormat = DataFormat
        str = Right(Space(mLen) + vData.ToString(mFormat), mLen)
    End If
    Return str
End Function
Public Property Color() As System.Drawing.Color
Get
    Return pvColor
End Get
Set(ByVal value As System.Drawing.Color)
    pvColor = value
End Set
End Property
Public Property AlarmHi2() As Single
Get
    Return pvAlarmHi2
End Get
Set(ByVal value As Single)
    pvAlarmHi2 = value
End Set
End Property
Public Property AlarmHi1() As Single
Get
    Return pvAlarmHi1
End Get
Set(ByVal value As Single)
    pvAlarmHi1 = value
End Set
End Property
Public Property AlarmLo1() As Single
Get
    Return pvAlarmLo1
End Get
Set(ByVal value As Single)
    pvAlarmLo1 = value
```

```
End Set
End Property
Public Property AlarmLo2() As Single
    Get
        Return pvAlarmLo2
    End Get
    Set(ByVal value As Single)
        pvAlarmLo2 = value
    End Set
End Property
Public Property DataMax() As Single
    Get
        Return pvDataMax
    End Get
    Set(ByVal value As Single)
        pvDataMax = value
    End Set
End Property
Public Property DataMin() As Single
    Get
        Return pvDataMin
    End Get
    Set(ByVal value As Single)
        pvDataMin = value
    End Set
End Property
Public Property DataNow() As Single
    Get
        If Register >= 1 And Register <= 18 Then
            Return pvDataNow
        Else
            Return NullData
        End If
    End Get
    Set(ByVal value As Single)
        pvDataNow = value
    End Set
End Property
Public ReadOnly Property DataNowRate() As Single
    Get
        Return convDataToRate(pvDataNow)
    End Get
End Property
Public Function DataRate(ByVal vData As Single) As Single
    Return convDataToRate(vData)
End Function
Public Function convDataToRate(ByVal vData As Single) As Single
    Dim w As Single
    w = (DataMax - DataMin)
    If w <> 0 And IsNullData(vData) = False Then
        Return (vData - DataMin) / w * 100
    Else
        Return NullData
    End If
End Function
```

```
End If
End Function

Public Function convRateToData(ByVal vData As Single) As Single
    Dim w As Single
    w = (DataMax - DataMin)
    If w <> 0 And IsNullData(vData) = False Then
        Return (vData - DataMin) / w
    Else
        Return NullData
    End If
End Function

Public ReadOnly Property DataMaxStr() As String
    Get
        Return Format(pvDataMax)
    End Get
End Property
Public ReadOnly Property DataMinStr() As String
    Get
        Return Format(pvDataMin)
    End Get
End Property
Public ReadOnly Property DataNowStr() As String
    Get
        Return Format(pvDataNow)
    End Get
End Property
Public Property Register() As Integer
    Get
        Return pvRegister
    End Get
    Set(ByVal value As Integer)
        pvRegister = value
    End Set
End Property

Public Property CoefficientA() As Single
    Get
        Return pvCoefficientA
    End Get
    Set(ByVal value As Single)
        pvCoefficientA = value
    End Set
End Property
Public Property CoefficientB() As Single
    Get
        Return pvCoefficientB
    End Get
    Set(ByVal value As Single)
        pvCoefficientB = value
    End Set
End Property

Public Function GetAlarmLevel() As eAlarmLevel
```

```
If IsNullData(DataNow) = False Then
    If (DataNow >= AlarmHi2 And IsNullData(AlarmHi2) = False) Or (DataNow <= AlarmLo2
    And IsNullData(AlarmLo2) = False) Then
        Return eAlarmLevel.ealWarning
    End If
    If (DataNow >= AlarmHi1 And IsNullData(AlarmHi1) = False) Or (DataNow <= AlarmLo1
    And IsNullData(AlarmLo1) = False) Then
        Return eAlarmLevel.ealAttention
    End If
    Return eAlarmLevel.ealNormal
Else
    Return eAlarmLevel.ealNormal
End If
End Function
```

```
Public Function GetAlarmLevelColor(Optional ByVal vColorSetNo As Integer = 0) As
Drawing.Color
    Dim mColorType(,) As Drawing.Color = {{Drawing.Color.White, Drawing.Color.Yellow,
    Drawing.Color.Red},
                                           {Drawing.Color.Black, Drawing.Color.Yellow,
    Drawing.Color.Red},
                                           {Drawing.Color.White, Drawing.Color.Yellow,
    Drawing.Color.Red},
                                           {Drawing.Color.Black, Drawing.Color.Gold,
    Drawing.Color.Red}
    }
    'Goldenrod
    Select Case GetAlarmLevel()
        Case clsChannelInfo.eAlarmLevel.ealAttention
            Return mColorType(vColorSetNo, 1)
        Case clsChannelInfo.eAlarmLevel.ealWarning
            Return mColorType(vColorSetNo, 2)
        Case clsChannelInfo.eAlarmLevel.ealWarning2
            Return mColorType(vColorSetNo, 2)
        Case Else
            Return mColorType(vColorSetNo, 0)
    End Select
End Function
```

```
End Class
```

```
Imports clslibNextforest

Public Class clsMeasureData
    Public MeasDate As Date
    Public MeasData(10) As Single
    Public Alarm As Integer
    Public Flag As Integer
End Class

Public Class clsAlarmInfo
    Public OccurTime As Date
    Public RecorderNo As Integer
    Public Level As Integer
    Public Message As String
End Class

Public Class clsManlockInfo

    Public Enum eTimePitch
        etpAll = 0           ' 全部
        etpMinute01 = 1     ' 1分間隔
        etpMinute02 = 2     ' 2分間隔
        etpMinute05 = 3     ' 5分間隔
        etpMinute10 = 4     ' 10分間隔
        etpMinute20 = 5     ' 20分間隔
        etpMinute30 = 6     ' 30分間隔
        etpHour01 = 7       ' 1時間間隔
        etpHour02 = 8       ' 2時間間隔
        etpHour03 = 9       ' 3時間間隔
        etpHour06 = 10      ' 6時間間隔
        etpHour12 = 11      ' 12時間間隔
        etpDay01 = 12       ' 1日間隔
    End Enum

    Dim LibFF As New clsFolderFile

    Private pvDB As clsAccessMdb = Nothing

    Private pvTitle As String
    Private pvRecorderN As Integer = 0
    Private pvSavePitchDB(23) As Integer
    Private pvSavePitchFile As Integer
    Private pvFileName As String

    Private pvServerSMTP As String
    Private pvServerPortNo As Integer
    Private pvAccount As String
    Private pvPassword As String
    Private pvFromAddress As String

    Private pvRecorderInfo() As clsRecorderInfo
    Private pvAddressInfo() As clsAddressInfo

    Private pvLogOutFlag As Boolean
```

```
Public Enum eAccessMode
    eamLoad = 0
    eamSave = 1
End Enum

Public Sub New()
    Initialize()
End Sub

Public Sub Initialize()
    pvTitle = "無題"
    pvRecorderN = 0
    For i = 0 To 23 : pvSavePitchDB(i) = 10 : Next
    pvSavePitchFile = 1
    pvServerSMTP = "smtp.alpha-plt.jp"
    pvServerPortNo = 587
    pvAccount = "manlock%krcnet.co.jp"
    pvPassword = "kanelock2013"
    pvFromAddress = "manlock@krcnet.co.jp"
    pvFileName = ""
    ReDim pvRecorderInfo(10)
    For i = 1 To 10
        pvRecorderInfo(i) = New clsRecorderInfo
        pvRecorderInfo(i).Initialize()
        pvRecorderInfo(i).RecorderNo = i
        pvRecorderInfo(i).RecorderName = "レコーダー№" & i.ToString
    Next
    ReDim pvAddressInfo(50)
    For i = 1 To 50
        pvAddressInfo(i) = New clsAddressInfo
        pvAddressInfo(i).Initialize()
    Next
End Sub

Private Function dbOpen(ByVal vRecNo As Integer) As Boolean
    Dim mFileName As String
    mFileName = FileName
    If vRecNo >= 1 And vRecNo <= 10 Then
        mFileName = FileName & (vRecNo - 1).ToString("0")
    Else
        mFileName = FileName
    End If
    Return dbFileOpen(mFileName)
End Function

Private Function dbFileOpen(ByVal vFileName As String) As Boolean
    Dim mOpenFlag As Boolean, mFileName As String
    Dim libFF As New clsFolderFile
    If libFF.ExistsFile(vFileName) = False Then Return False
    Try
        mFileName = vFileName
        pvDB = New clsAccessMdb
        mOpenFlag = pvDB.Connect(mFileName)
        Return mOpenFlag
    Catch
        pvDB = Nothing
    End Try
End Function
```

```
Return False
End Try
End Function
Private Sub dbClose()
    pvDB.Disconnect()
    pvDB = Nothing
End Sub

Public Property Title() As String
    Get
        Return pvTitle
    End Get
    Set(ByVal Value As String)
        pvTitle = Value
    End Set
End Property
Public Property RecorderN() As Integer
    Get
        If pvRecorderN < 0 Or pvRecorderN > 10 Then pvRecorderN = 0
        Return pvRecorderN
    End Get
    Set(ByVal Value As Integer)
        If Value < 0 Or Value > 10 Then Value = 0
        pvRecorderN = Value
    End Set
End Property
Public Property RecorderInfo(ByVal vNo As Integer) As clsRecorderInfo
    Get
        If vNo >= 1 And vNo <= 10 Then
            Return pvRecorderInfo(vNo)
        Else
            Return Nothing
        End If
    End Get
    Set(ByVal Value As clsRecorderInfo)
        If vNo >= 1 And vNo <= 10 Then
            pvRecorderInfo(vNo) = Value
        Else
            pvRecorderInfo(vNo) = Nothing
            'MsgBox("RecorderInfo No." & vNo.ToString, MsgBoxStyle.OkOnly, "エラー")
            'Beep()
        End If
    End Set
End Property

Public Property AddressInfo(ByVal vNo As Integer) As clsAddressInfo
    Get
        If vNo >= 1 And vNo <= 50 Then
            Return pvAddressInfo(vNo)
        Else
            Return Nothing
        End If
    End Get
    Set(ByVal Value As clsAddressInfo)
        If vNo >= 1 And vNo <= 50 Then
```

```
        pvAddressInfo(vNo) = Value
    Else
        pvAddressInfo(vNo) = Nothing
        'MsgBox("RecorderInfo No." & vNo.ToString, MsgBoxStyle.OkOnly, "エラー")
        'Beep()
    End If
End Set
End Property
Public ReadOnly Property DataFolder() As String
    Get
        Return LibFF.GetFolderName(pvFileName)
    End Get
End Property
Public Property SavePitchDB(ByVal vHour As Integer) As Integer
    Get
        Dim mSavePitch As Integer
        mSavePitch = pvSavePitchDB(vHour)
        If mSavePitch < 1 Then mSavePitch = 10
        If mSavePitch > 60 Then mSavePitch = 60
        Return mSavePitch
    End Get
    Set(ByVal Value As Integer)
        pvSavePitchDB(vHour) = Value
    End Set
End Property
Public Property SavePitchFile() As Integer
    Get
        Return pvSavePitchFile
    End Get
    Set(ByVal Value As Integer)
        pvSavePitchFile = Value
    End Set
End Property

Public Property ServerSMTP() As String
    Get
        Return pvServerSMTP
    End Get
    Set(ByVal value As String)
        pvServerSMTP = value
    End Set
End Property
Public Property ServerPortNo() As Integer
    Get
        Return pvServerPortNo
    End Get
    Set(ByVal value As Integer)
        pvServerPortNo = value
    End Set
End Property
Public Property Account() As String
    Get
        Return pvAccount
    End Get
    Set(ByVal value As String)
```

```
        pvAccount = value
    End Set
End Property
Public Property Password() As String
    Get
        Return pvPassword
    End Get
    Set(ByVal value As String)
        pvPassword = value
    End Set
End Property
Public Property FromAddress() As String
    Get
        Return pvFromAddress
    End Get
    Set(ByVal value As String)
        pvFromAddress = value
    End Set
End Property

Public Property FileName() As String
    Get
        Return (pvFileName)
    End Get
    Set(ByVal value As String)
        pvFileName = value
    End Set
End Property

Public Function Setup(ByVal vFileName As String) As Boolean
    FileName = vFileName
    If Check() = True Then
        Return True
    Else
        FileName = ""
        Return False
    End If
End Function

Public Function Check() As Boolean
    Dim LibFF As New clsFolderFile
    If LibFF.ExistsFile(FileName) = True Then
        dbManlockInfoIO(eAccessMode.eamLoad)
        Return True
    Else
        Return False
    End If
End Function

Public Function CreateTable(ByRef rTable As DataTable, ByRef rData As Object, ByVal
vRow As Integer, ByVal vItemName As String) As Boolean
    Dim mTable As DataTable, mTableName As String, mSql As String = ""

    mTableName = "ManlockInfo"
    mTable = pvDB.ExecuteSql("select * from `" & mTableName & "` WHERE `ID`=1")
```

```
CreateTable = False
End Function

'ALTER TABLE `nf_users` ADD `CoefficientA` FLOAT NOT NULL"

Public Sub dbFieldCheck()

End Sub

Public Sub dbFieldAdd()
    Dim mSql As String
    If dbOpen(0) = True Then
        If pvDB.IsField("ChannelInfo", "CoefficientA") = False Then
            mSql = "ALTER TABLE `ChannelInfo` ADD `CoefficientA` FLOAT NOT NULL"
            pvDB.ExecuteSql(mSql)
        End If

        If pvDB.IsField("ChannelInfo", "CoefficientB") = False Then
            mSql = "ALTER TABLE `ChannelInfo` ADD `CoefficientB` FLOAT NOT NULL"
            pvDB.ExecuteSql(mSql)
        End If
        dbClose()
    End If
End Sub

Public Function dbManlockInfoIO(ByVal vMode As eAccessMode) As Boolean
    Dim mTable As DataTable, mTableName As String, mSql As String

    If dbOpen(0) = True Then
        mTableName = "ManlockInfo"
        mTable = pvDB.ExecuteSql("select * from `" & mTableName & "` WHERE `ID`=1")
        If vMode = eAccessMode.eamLoad Then
            If mTable.Rows.Count = 1 Then
                pvDB.GetTableData(mTable, Title, 0, "Title")
                pvDB.GetTableData(mTable, RecorderN, 0, "RecorderN")
                For i = 0 To 23
                    pvDB.GetTableData(mTable, SavePitchDB(i), 0, "SavePitchDB" & i.ToString("00"))
                Next
                pvDB.GetTableData(mTable, SavePitchFile, 0, "SavePitchFile")
                pvDB.GetTableData(mTable, ServerSMTP, 0, "ServerSMTP")
                pvDB.GetTableData(mTable, ServerPortNo, 0, "ServerPortNo")
                pvDB.GetTableData(mTable, Account, 0, "Account")
                pvDB.GetTableData(mTable, Password, 0, "Password")
                pvDB.GetTableData(mTable, FromAddress, 0, "FromAddress")
            End If
        Else
            pvDB.BeginTransaction()
            If mTable.Rows.Count = 0 Then
                pvDB.ExecuteSql("insert into " & mTableName & "(ID) values(1)")
            End If
            mSql = "UPDATE `" & mTableName & "` SET "

            mSql &= pvDB.AddField(Title, "Title", True)
            mSql &= pvDB.AddField(RecorderN, "RecorderN")
        End If
    End If
End Function
```

```
For i = 0 To 23
    mSql &= pvDB.AddField(SavePitchDB(i), "SavePitchDB" & i.ToString("00"))
Next
mSql &= pvDB.AddField(SavePitchFile, "SavePitchFile")
mSql &= pvDB.AddField(ServerSMTP, "ServerSMTP")
mSql &= pvDB.AddField(ServerPortNo, "ServerPortNo")
mSql &= pvDB.AddField(Account, "Account")
mSql &= pvDB.AddField>Password, "Password")
mSql &= pvDB.AddField(FromAddress, "FromAddress")

mSql &= " WHERE `ID`=1"
pvDB.ExecuteSql(mSql)
pvDB.CommitTransaction()

End If
dbClose()

For mRecNo = 1 To 10
    dbRecorderInfoIO(vMode, mRecNo)
Next

dbAddressBookIO(vMode)
Return True
End If
Return False
End Function

Public Function dbRecorderInfoIO(ByVal vMode As eAccessMode, ByVal vRecorderNo As Integer) As Boolean
    Dim mTable As DataTable, mTableName As String, mSql As String

    If vRecorderNo < 0 Or vRecorderNo > 10 Then Return False

    If dbOpen(0) = True Then
        mTableName = "RecorderInfo"
        mTable = pvDB.ExecuteSql("select * from `" & mTableName & "` WHERE `ID`=" & vRecorderNo.ToString)
        If vMode = eAccessMode.eamLoad Then
            If mTable.Rows.Count = 1 Then
                With Me.pvRecorderInfo(vRecorderNo)
                    pvDB.GetTableData(mTable, .RecorderName, 0, "Name")
                    pvDB.GetTableData(mTable, .ChannelN, 0, "ChannelN")
                    pvDB.GetTableData(mTable, .NetHostName, 0, "NetHostName")
                    pvDB.GetTableData(mTable, .NetPortNo, 0, "NetPortNo")
                    pvDB.GetTableData(mTable, .Active, 0, "Active")
                    pvDB.GetTableData(mTable, .Visible, 0, "Visible")
                End With
            End If
        Else
            pvDB.BeginTransaction()
            If mTable.Rows.Count = 0 Then
                pvDB.ExecuteSql("insert into " & mTableName & "(ID) values(" & vRecorderNo.ToString & ")")
            End If
            mSql = "UPDATE `" & mTableName & "` SET "
```

```

With Me.pvRecorderInfo(vRecorderNo)
    mSql &= pvDB.AddField(.RecorderName, "Name", True)
    mSql &= pvDB.AddField(.ChannelN, "ChannelN")
    mSql &= pvDB.AddField(.NetHostName, "NetHostName")
    mSql &= pvDB.AddField(.NetPortNo, "NetPortNo")
    mSql &= pvDB.AddField(.Active, "Active")
    mSql &= pvDB.AddField(.Visible, "Visible")
End With

mSql &= " WHERE `ID`=" & vRecorderNo.ToString
pvDB.ExecuteSql(mSql)
pvDB.CommitTransaction()
End If

'Channel情報 Read/Write
For mChNo = 1 To 10
    mTableName = "ChannelInfo"
    Dim mChID As Integer = vRecorderNo * 100 + mChNo, mTmpColor As Integer
    mTable = pvDB.ExecuteSql("select * from `" & mTableName & "` WHERE `ID`=" &
mChID.ToString)
    If vMode = eAccessMode.eamLoad Then
        If mTable.Rows.Count = 1 Then
            With Me.pvRecorderInfo(vRecorderNo).ChannelInfo(mChNo)
                pvDB.GetTableData(mTable, .Name, 0, "Name")
                pvDB.GetTableData(mTable, mTmpColor, 0, "Color") : .Color =
Color.FromArgb(mTmpColor)
                pvDB.GetTableData(mTable, .Unit, 0, "Unit")
                pvDB.GetTableData(mTable, .DataFormat, 0, "DataFormat")
                pvDB.GetTableData(mTable, .AlarmHi2, 0, "AlarmHi2")
                pvDB.GetTableData(mTable, .AlarmHi1, 0, "AlarmHi1")
                pvDB.GetTableData(mTable, .AlarmLo1, 0, "AlarmLo1")
                pvDB.GetTableData(mTable, .AlarmLo2, 0, "AlarmLo2")
                pvDB.GetTableData(mTable, .DataMax, 0, "DataMax")
                pvDB.GetTableData(mTable, .DataMin, 0, "DataMin")
                pvDB.GetTableData(mTable, .Register, 0, "Register")
                pvDB.GetTableData(mTable, .CoefficientA, 0, "CoefficientA")
                pvDB.GetTableData(mTable, .CoefficientB, 0, "CoefficientB")
                .CoefficientA = IIf(.CoefficientA = 0, 1, .CoefficientA)
                '係数Aがゼロのとき1に強制
            End With
        End If
    Else
        pvDB.BeginTransaction()
        If mTable.Rows.Count = 0 Then
            pvDB.ExecuteSql("insert into " & mTableName & "(ID) values(" & mChID.ToString
& ")")
        End If
        mSql = "UPDATE `" & mTableName & "` SET "

        With Me.pvRecorderInfo(vRecorderNo).ChannelInfo(mChNo)
            mSql &= pvDB.AddField(.Name, "Name", True)
            mTmpColor = .Color.ToArgb
            mSql &= pvDB.AddField(mTmpColor, "Color")

```

```

mSql &= pvDB.AddField(.Unit, "Unit")
mSql &= pvDB.AddField(.DataFormat, "DataFormat")
mSql &= pvDB.AddField(.AlarmHi2, "AlarmHi2")
mSql &= pvDB.AddField(.AlarmHi1, "AlarmHi1")
mSql &= pvDB.AddField(.AlarmLo1, "AlarmLo1")
mSql &= pvDB.AddField(.AlarmLo2, "AlarmLo2")
mSql &= pvDB.AddField(.DataMax, "DataMax")
mSql &= pvDB.AddField(.DataMin, "DataMin")
mSql &= pvDB.AddField(.Register, "Register")
mSql &= pvDB.AddField(.CoefficientA, "CoefficientA")
mSql &= pvDB.AddField(.CoefficientB, "CoefficientB")
End With

```

```

mSql &= " WHERE `ID`=" & mChID.ToString
pvDB.ExecuteSql(mSql)
pvDB.CommitTransaction()

```

```
End If
```

```
Next
```

```
dbClose()
```

```
Return True
```

```
End If
```

```
Return False
```

```
End Function
```

```
Public Function dbAddressBookIO(ByVal vMode As eAccessMode) As Boolean
```

```
Dim mTable As DataTable, mSql As String
```

```
Dim mTableName As String, mFieldName As String
```

```
If dbOpen(0) = True Then
```

```
For mNo = 1 To 50
```

```
mTableName = "AddressBook"
```

```
mTable = pvDB.ExecuteSql("select * from `" & mTableName & "` WHERE `ID`=" &
mNo.ToString)
```

```
If vMode = eAccessMode.eamLoad Then
```

```
If mTable.Rows.Count = 1 Then
```

```
With Me.pvAddressInfo(mNo)
```

```
pvDB.GetTableData(mTable, .Name, 0, "Name")
```

```
pvDB.GetTableData(mTable, .MailAddress, 0, "MailAddress")
```

```
For i = 1 To 10
```

```
For j = 1 To 2
```

```
mFieldName = "RecorderNo" & i.ToString("00") & "_Level" & j.ToString("0")
```

```
pvDB.GetTableData(mTable, .AlarmLevel(i, j), 0, mFieldName)
```

```
Next
```

```
Next
```

```
End With
```

```
End If
```

```
Else
```

```
pvDB.BeginTransaction()
```

```
If mTable.Rows.Count = 0 Then
```

```
pvDB.ExecuteSql("insert into " & mTableName & "(ID) values(" & mNo.ToString &
")")
```

```
End If
```

```
mSql = "UPDATE `" & mTableName & "` SET "
```

```

With Me.pvAddressInfo(mNo)
    mSql &= pvDB.AddField(.Name, "Name", True)
    mSql &= pvDB.AddField(.MailAddress, "MailAddress")
    For i = 1 To 10
        For j = 1 To 2
            mFieldName = "RecorderNo" & i.ToString("00") & "_Level" & j.ToString("0")
            mSql &= pvDB.AddField(.AlarmLevel(i, j), mFieldName)
        Next
    Next
End With
mSql &= " WHERE `ID`=" & mNo.ToString
pvDB.ExecuteSql(mSql)
pvDB.CommitTransaction()
End If
Next

dbClose()
Return True
End If
Return False
End Function

Public Function MeasureDataWriteCSV(ByVal vRecorderNo As Integer, Optional ByVal
vMeasTime As Date = NullDate) As Boolean
    Dim mRecorderInfo As clsRecorderInfo
    Dim mMeasTimeStr As String, mMeasTime As Date
    Dim mDataFolder As String, mFileNameOnly As String, mCsvFolder As String,
mFileNameCSV As String
    Dim fn As Integer, buf As String

    If vRecorderNo < 0 Or vRecorderNo > 10 Then Return False

    mRecorderInfo = Me.pvRecorderInfo(vRecorderNo)

    mMeasTime = IIf(vMeasTime = NullDate, Date.Now, vMeasTime)
    mMeasTimeStr = mMeasTime.ToString("yyyy/MM/dd HH:mm:ss")

    mDataFolder = DataFolder()
    mFileNameOnly = LibFF.GetFileNameOnly(FileName)
    mCsvFolder = mDataFolder & "\" & mFileNameOnly & "_CSV"

    If LibFF.ExistsFolder(mCsvFolder) = False Then
        LibFF.CreateDirectory(mCsvFolder)
    End If

    mFileNameCSV = mCsvFolder & "\RecNo" & vRecorderNo.ToString("00") & "_" &
mMeasTime.ToString("yyyyMMdd") & ".CSV"

    buf = mMeasTimeStr
    For mChNo = 1 To mRecorderInfo.ChannelN
        buf &= ", " & mRecorderInfo.ChannelInfo(mChNo).DataNowStr
    Next
    buf &= ", " & mRecorderInfo.GetAlarmFlag

```

```

fn = FreeFile()
FileOpen(fn, mFileNameCSV, OpenMode.Append)
PrintLine(fn, buf)
FileClose(fn)
Return True
End Function

Public Function MeasureDataWriteDB(ByVal vRecorderNo As Integer, Optional ByVal
vMeasTime As Date = NullDate, Optional ByVal vRecFlag As Boolean = False) As Boolean
    Dim mTable As DataTable, mSql As String, mTableName As String, mFieldName As String
    Dim mMeasTimeStr As String, mMeasTime As Date, mFlag As Integer, mDataNow As Single

    If vRecorderNo < 0 Or vRecorderNo > 10 Then Return False

    mMeasTime = IIf(vMeasTime = NullDate, Date.Now, vMeasTime)
    mMeasTimeStr = mMeasTime.ToString("yyyy/MM/dd HH:mm:ss")

    If dbOpen(vRecorderNo) = True Then
        pvDB.BeginTransaction()
        mTableName = "MeasureData" & vRecorderNo.ToString("00")
        mTable = pvDB.ExecuteSql("select * from `" & mTableName & "` WHERE `日時`=#" &
mMeasTimeStr & "#")

        If mTable.Rows.Count = 0 Then
            pvDB.ExecuteSql("insert into " & mTableName & "(日時) values(#" & mMeasTimeStr &
"#)")
        End If
        mSql = "UPDATE `" & mTableName & "` SET "

        Dim mChMax As Integer = Me.pvRecorderInfo(vRecorderNo).ChannelN
        For i = 1 To 10
            With Me.pvRecorderInfo(vRecorderNo).ChannelInfo(i)
                mFieldName = "Data" & i.ToString("00")
                mDataNow = IIf(i <= mChMax, .DataNow, NullDate)
                mSql &= pvDB.AddField(mDataNow, mFieldName, IIf(i = 1, True, False))
            End With
        Next
        mSql &= pvDB.AddField(Me.pvRecorderInfo(vRecorderNo).GetAlarmFlag, "Alarm")
        mFlag = IIf(vRecFlag = True, 1, 0)
        mSql &= pvDB.AddField(mFlag, "Flag")
        mSql &= " WHERE `日時`=#" & mMeasTimeStr & "#"
        pvDB.ExecuteSql(mSql)
        pvDB.CommitTransaction()
        dbClose()
        Return True
    End If
    Return False
End Function

Public Function FileImportDB(ByVal vRecorderNo As Integer, ByVal vFileName As String)
As Boolean
    Dim LibFF As New clsFolderFile
    Dim fn As Integer, buf As String, aryStrings() As String
    Dim mRtn As Boolean = False
    Dim mLineN As Integer = 0, mChN As Integer, mDmyT As Date, mBuf As String

```

```
Dim mDimSize As Integer = 1000
Dim mTime(mDimSize) As Date, mData(9, mDimSize) As Single

If LibFF.ExistsFile(vFileName) = True Then
    fn = FreeFile()
    FileOpen(fn, vFileName, OpenMode.Input)
    Do While (Not EOF(fn))
        buf = LineInput(1)
        aryStrings = Split2(buf) ' Tab or ',' 区切りを読む
        If aryStrings.Count > 1 Then
            If IsDate(aryStrings(0)) = True Then
                mDmyT = CDate(aryStrings(0))
                If mDmyT > CDate("2000/1/1") Then
                    If mLineN >= mDimSize Then
                        mDimSize += 3600
                        ReDim Preserve mData(9, mDimSize)
                        ReDim Preserve mTime(mDimSize)
                    End If

                    mTime(mLineN) = mDmyT
                    For i = 0 To 9 : mData(i, mLineN) = NullData : Next
                    mChN = aryStrings.Count - 2
                    For i = 0 To mChN - 1
                        mBuf = aryStrings(i + 2)
                        If IsNumeric(mBuf) Then
                            mData(i, mLineN) = Val(mBuf)
                        End If
                    Next
                    mLineN += 1
                End If
            End If
        End If
    Loop
    FileClose(fn)
    ReDim Preserve mData(9, mLineN - 1)
Else
    mRtn = False
End If

Dim mTable As DataTable, mSql As String, mTableName As String, mFieldName As String
Dim mMeasTimeStr As String, mMeasTime As Date, mWriteFlag As Boolean
Dim mTimeSta As Date, mTimeEnd As Date, strTimeSta As String, strTimeEnd As String
Dim mCrnt As Date, mSpan As Integer

If dbOpen(vRecorderNo) = True Then
    pvDB.BeginTransaction()

    mTableName = "MeasureData" & vRecorderNo.ToString("00")

    mTimeSta = mTime(0)
    mTimeEnd = mTime(mLineN - 1)

    ' Import 期間のデータを削除
```

```

strTimeSta = mTimeSta.ToString("yyyy/MM/dd HH:mm:ss")
strTimeEnd = mTimeEnd.ToString("yyyy/MM/dd HH:mm:ss")
mSql = "DELETE * FROM `" & mTableName & "` WHERE `日時`>=#" & strTimeSta & "# and
`日時`<=#" & strTimeEnd & "#"
pvDB.ExecuteSql(mSql)

mSpan = gManlockInfo.SavePitchDB(mTimeSta.Hour)
mCrnt = TimeFloor(mTimeSta, mSpan)

For mLineNo = 0 To mLineN - 1
    mMeasTime = mTime(mLineNo)
    mWriteFlag = True
    If mMeasTime < mTimeSta Or mMeasTime > mTimeEnd Then
        mWriteFlag = False
    ElseIf mMeasTime < mCrnt Then
        mWriteFlag = False
    End If

    If mWriteFlag = True Then
        mMeasTimeStr = mMeasTime.ToString("yyyy/MM/dd HH:mm:ss")
        mTable = pvDB.ExecuteSql("select * from `" & mTableName & "` WHERE `日時`=#" &
mMeasTimeStr & "#")
        If mTable.Rows.Count = 0 Then
            pvDB.ExecuteSql("insert into " & mTableName & "(日時) values(#" & mMeasTimeStr
& "#)")
        End If
        mSql = "UPDATE `" & mTableName & "` SET "
        For i = 1 To 10
            With Me.pvRecorderInfo(vRecorderNo).ChannelInfo(i)
                mFieldName = "Data" & i.ToString("00")
                mSql &= pvDB.AddField(mData(i - 1, mLineNo), mFieldName, IIf(i = 1, True,
False))
            End With
        Next
        mSql &= " WHERE `日時`=#" & mMeasTimeStr & "#"
        pvDB.ExecuteSql(mSql)

        mSpan = gManlockInfo.SavePitchDB(mMeasTime.Hour)
        mCrnt = TimeFloor(mMeasTime, mSpan).AddSeconds(mSpan)

    End If
Next
pvDB.CommitTransaction()

dbClose() 'pvDB.Disconnect()
Return True
End If

Return mRtn
End Function

```

```

Public Function MeasureDataReadDB(ByVal vRecorderNo As Integer, ByVal vStaet As Date,
ByVal vEnd As Date, Optional ByVal vPitch As eTimePitch = eTimePitch.etpAll, _
Optional ByVal vFlag As Integer = 0) As

```

```

        clsMeasureData()

Dim mTable As DataTable, mSql As String, mTableName As String
Dim mRecorderInfo As clsRecorderInfo
Dim mSql_WHERE As String
Dim mMeasData() As clsMeasureData

dbOpen(vRecorderNo)
If vRecorderNo < 1 Or vRecorderNo > 10 Or IsNothing(pvDB) = True Then
    ReDim mMeasData(0)
    Return mMeasData
End If

mRecorderInfo = Me.pvRecorderInfo(vRecorderNo)

mTableName = "MeasureData" & vRecorderNo.ToString("00")
mSql = "select * from `" & mTableName & "`"

mSql_WHERE = " WHERE `日時`>=#" & vStaeet.ToString("yyyy/MM/dd HH:mm:ss") & "# and " &
" `日時`<=#" & vEnd.ToString("yyyy/MM/dd HH:mm:ss") & "#"
Select Case vPitch
    Case eTimePitch.etpMinute01 ' 1分
        mSql_WHERE &= " and SECOND(日時)=0"
    Case eTimePitch.etpMinute02 ' 2分
        mSql_WHERE &= " and ("
        mSql_WHERE &= "MINUTE(日時)=0 or MINUTE(日時)=2 or MINUTE(日時)=4 or MINUTE(日時)=6
or MINUTE(日時)=8"
        mSql_WHERE &= " or MINUTE(日時)=10 or MINUTE(日時)=12 or MINUTE(日時)=14 or
MINUTE(日時)=16 or MINUTE(日時)=18"
        mSql_WHERE &= " or MINUTE(日時)=20 or MINUTE(日時)=22 or MINUTE(日時)=24 or
MINUTE(日時)=26 or MINUTE(日時)=28"
        mSql_WHERE &= " or MINUTE(日時)=30 or MINUTE(日時)=32 or MINUTE(日時)=34 or
MINUTE(日時)=36 or MINUTE(日時)=38"
        mSql_WHERE &= " or MINUTE(日時)=40 or MINUTE(日時)=42 or MINUTE(日時)=44 or
MINUTE(日時)=46 or MINUTE(日時)=48"
        mSql_WHERE &= " or MINUTE(日時)=50 or MINUTE(日時)=52 or MINUTE(日時)=54 or
MINUTE(日時)=56 or MINUTE(日時)=58"
        mSql_WHERE &= ") and SECOND(日時)=0"
    Case eTimePitch.etpMinute05 ' 5分
        mSql_WHERE &= " and ("
        mSql_WHERE &= "MINUTE(日時)=0 or MINUTE(日時)=5 or MINUTE(日時)=10 or MINUTE(日時)=15"
        mSql_WHERE &= " or MINUTE(日時)=20 or MINUTE(日時)=25 or MINUTE(日時)=30 or
MINUTE(日時)=35"
        mSql_WHERE &= " or MINUTE(日時)=40 or MINUTE(日時)=45 or MINUTE(日時)=50 or
MINUTE(日時)=55"
        mSql_WHERE &= ") and SECOND(日時)=0"
    Case eTimePitch.etpMinute10 ' 10分
        mSql_WHERE &= " and (MINUTE(日時)=0 or MINUTE(日時)=10 or MINUTE(日時)=20 or
MINUTE(日時)=30 or MINUTE(日時)=40 or MINUTE(日時)=50) and SECOND(日時)=0"
    Case eTimePitch.etpMinute20 ' 20分
        mSql_WHERE &= " and (MINUTE(日時)=0 or MINUTE(日時)=20 or MINUTE(日時)=40) and
SECOND(日時)=0"
    Case eTimePitch.etpMinute30 ' 30分
        mSql_WHERE &= " and (MINUTE(日時)=0 or MINUTE(日時)=30) and SECOND(日時)=0"
    Case eTimePitch.etpHour01 ' 1時間
        mSql_WHERE &= " and MINUTE(日時)=0 and SECOND(日時)=0"

```

```

Case eTimePitch.etpHour02      '2時間
    mSql_WHERE &= " and (HOUR(日時)=0 or HOUR(日時)=2 or HOUR(日時)=4 or HOUR(日時)=6 or
    HOUR(日時)=8 or HOUR(日時)=10 or HOUR(日時)=12 or HOUR(日時)=14 or HOUR(日時)=16 or
    HOUR(日時)=18 or HOUR(日時)=20 or HOUR(日時)=22) and MINUTE(日時)=0 and SECOND(日時)=0"
Case eTimePitch.etpHour03      '3時間
    mSql_WHERE &= " and (HOUR(日時)=0 or HOUR(日時)=3 or HOUR(日時)=6 or HOUR(日時)=9 or
    HOUR(日時)=12 or HOUR(日時)=15 or HOUR(日時)=18 or HOUR(日時)=21) and MINUTE(日時)=0
    and SECOND(日時)=0"
Case eTimePitch.etpHour06      '6時間
    mSql_WHERE &= " and (HOUR(日時)=0 or HOUR(日時)=6 or HOUR(日時)=12 or HOUR(日時)=18)
    and MINUTE(日時)=0 and SECOND(日時)=0"
Case eTimePitch.etpHour12      '12時間
    mSql_WHERE &= " and (HOUR(日時)=0 or HOUR(日時)=12) and MINUTE(日時)=0 and
    SECOND(日時)=0"
Case eTimePitch.etpDay01       '1日
    mSql_WHERE &= " and HOUR(日時)=0 and MINUTE(日時)=0 and SECOND(日時)=0"
End Select
If vFlag = 1 Then
    mSql_WHERE &= " and Flag>0"
End If

mSql &= mSql_WHERE

mSql &= " ORDER BY `日時` ASC"

pvDB.BeginTransaction()
mTable = pvDB.ExecuteSql(mSql)
pvDB.CommitTransaction()

Dim mCount As Integer = mTable.Rows.Count
If mCount > 0 Then
    ReDim mMeasData(mCount)
    For i = 1 To mCount
        mMeasData(i) = New clsMeasureData
        With mMeasData(i)
            pvDB.GetTableData(mTable, mMeasData(i).MeasDate, i - 1, "日時")
            For mChNo = 1 To 10
                pvDB.GetTableData(mTable, mMeasData(i).MeasData(mChNo), i - 1, "Data" &
                mChNo.ToString("00"))
            Next
            pvDB.GetTableData(mTable, mMeasData(i).Alarm, i - 1, "Alarm")
            pvDB.GetTableData(mTable, mMeasData(i).Flag, i - 1, "Flag")
        End With
    Next
Else
    ReDim mMeasData(0)
End If
dbClose()
Return mMeasData
End Function

Public Function AlarmWriteDB(ByVal vRecorderNo As Integer, ByVal vLevel As Integer,
ByVal vMessage As String, Optional ByVal vTime As Date = NullDate) As Boolean
    Dim mTable As DataTable, mSql As String, mTableName As String, strWHERE As String

```

```

Dim mMeasTimeStr As String, mMeasTime As Date

If vRecorderNo < 0 Or vRecorderNo > 10 Then Return False

mMeasTime = IIf(vTime = NullDate, Date.Now, vTime)
mMeasTimeStr = mMeasTime.ToString("yyyy/MM/dd HH:mm:ss")

If dbOpen(0) = True Then
    pvDB.BeginTransaction()
    mTableName = "AlarmHistory"
    strWHERE = " WHERE `RecorderNo`=" & vRecorderNo.ToString & " and `日時`=#" &
    mMeasTimeStr & "#"

    mTable = pvDB.ExecuteSql("select * from `" & mTableName & "`" & strWHERE)

    If mTable.Rows.Count = 0 Then
        pvDB.ExecuteSql("insert into " & mTableName & "(日時,RecorderNo) values(#" &
        mMeasTimeStr & "#," & vRecorderNo.ToString & ")")
    End If
    mSql = "UPDATE `" & mTableName & "` SET "
    mSql &= pvDB.AddField(vLevel, "Level", True)
    mSql &= pvDB.AddField(vMessage, "Message")

    mSql &= strWHERE
    pvDB.ExecuteSql(mSql)
    pvDB.CommitTransaction()
    dbClose()
    Return True
End If
Return False
End Function

Public Function AlarmReadDB(ByVal vRecorderNo As Integer, ByVal vStaet As Date, ByVal
vEnd As Date) As clsAlarmInfo()
    Dim mTable As DataTable, mSql As String, mTableName As String, strWHERE As String
    Dim mRecorderInfo As clsRecorderInfo
    Dim mAlarmInfo() As clsAlarmInfo

    dbOpen(0)
    If vRecorderNo < 0 Or vRecorderNo > 10 Or IsNothing(pvDB) = True Then
        ReDim mAlarmInfo(0)
        Return mAlarmInfo
    End If

    mRecorderInfo = Me.pvRecorderInfo(vRecorderNo)

    mTableName = "AlarmHistory"
    strWHERE = " WHERE `RecorderNo`=" & vRecorderNo.ToString
    strWHERE &= " and `日時`>=#" & vStaet.ToString("yyyy/MM/dd HH:mm:ss") & "#"
    strWHERE &= " and `日時`<=#" & vEnd.ToString("yyyy/MM/dd HH:mm:ss") & "#"
    mSql = "select * from `" & mTableName & "`" & strWHERE
    mSql &= " ORDER BY `日時` DESC" '昇順:ASC 降順:DESC

    mTable = pvDB.ExecuteSql(mSql)
    Dim mCount As Integer = mTable.Rows.Count

```

```
If mCount > 0 Then
    ReDim mAlarmInfo(mCount)
    For i = 1 To mCount
        mAlarmInfo(i) = New clsAlarmInfo
        With mAlarmInfo(i)
            pvDB.GetTableData(mTable, mAlarmInfo(i).OccurTime, i - 1, "日時")
            pvDB.GetTableData(mTable, mAlarmInfo(i).RecorderNo, i - 1, "RecorderNo")
            pvDB.GetTableData(mTable, mAlarmInfo(i).Level, i - 1, "Level")
            pvDB.GetTableData(mTable, mAlarmInfo(i).Message, i - 1, "Message")
        End With

        Next
    Else
        ReDim mAlarmInfo(0)
    End If

    dbClose()
    Return mAlarmInfo
End Function

Public Property LogOutFlag() As Boolean
    Get
        Return pvLogOutFlag
    End Get
    Set(ByVal Value As Boolean)
        pvLogOutFlag = Value
    End Set
End Property

End Class
```

```
Public Class clsRecorderInfo

    Private pvRecorderNo As Integer
    Private pvRecorderName As String
    Private pvChannelN As String
    Private pvNetHostName As String
    Private pvNetPortNo As Integer
    Private pvChannelInfo() As clsChannelInfo
    ''' <summary>
    '''
    ''' </summary>
    ''' <remarks></remarks>
    Private pvActive As Boolean
    Private pvVisible As Boolean

    Private pvGraphCurrentNo As Integer
    Private pvActionMode As eActionMode
    Private pvCommMode As eCommMode

    ' 列挙体を宣言
    Enum eActionMode
        esmSleep = 0      ' タイマー停止 (休止中)
        esmWakeUp = 1    ' タイマー起動 (起動中)
        esmStandBy = 2   ' 計測中 (監視中)
        esmRecording = 4 ' 収録中 (収録中)
    End Enum
    Enum eCommMode
        ecmNothing = 0   ' 無効
        ecmOffline = 1   ' オフライン
        ecmOnline = 1    ' オンライン
    End Enum

    Public Sub New()
        Initialize()
    End Sub

    Public Sub Initialize()
        pvActionMode = eActionMode.esmSleep

        RecorderName = "記録計名称"
        ChannelN = 1
        pvNetHostName = "192.168.190.999"
        pvNetPortNo = 502
        Active = False
        Visible = True
        ReDim pvChannelInfo(10)
        For i = 1 To 10
            pvChannelInfo(i) = New clsChannelInfo
            pvChannelInfo(i).Initialize()
            pvChannelInfo(i).Name = "項目" & i.ToString
        Next
        pvChannelInfo(1).Color = Color.Red
        pvChannelInfo(2).Color = Color.Green
        pvChannelInfo(3).Color = Color.Goldenrod
        pvChannelInfo(4).Color = Color.Purple
    End Sub
End Class
```

```
pvChannelInfo(5).Color = Color.Blue
pvChannelInfo(6).Color = Color.OrangeRed
pvChannelInfo(7).Color = Color.OliveDrab
pvChannelInfo(8).Color = Color.Aquamarine
pvChannelInfo(9).Color = Color.Chocolate
pvChannelInfo(10).Color = Color.SlateBlue
```

```
SetDefault()
```

```
End Sub
```

```
Public Sub SetDefault()
```

```
Me.pvChannelN = 6
```

```
With pvChannelInfo(1)
```

```
    .Name = "ゲージ圧"
```

```
    .Color = System.Drawing.Color.Red
```

```
    .DataFormat = "#0.000"
```

```
    .Unit = "MPa"
```

```
    .DataMin = 0
```

```
    .DataMax = 1
```

```
    .AlarmHi2 = NullData
```

```
    .AlarmHi1 = NullData
```

```
    .AlarmLo1 = NullData
```

```
    .AlarmLo2 = NullData
```

```
    .DataNow = .DataMin + (.DataMax - .DataMin) * 0.5
```

```
    .Register = 1
```

```
End With
```

```
With pvChannelInfo(2)
```

```
    .Name = "温度"
```

```
    .Color = System.Drawing.Color.DarkOrange
```

```
    .DataFormat = "##0.0"
```

```
    .Unit = " "
```

```
    .DataMin = 0
```

```
    .DataMax = 50
```

```
    .AlarmHi2 = NullData
```

```
    .AlarmHi1 = NullData
```

```
    .AlarmLo1 = NullData
```

```
    .AlarmLo2 = NullData
```

```
    .DataNow = .DataMin + (.DataMax - .DataMin) * 0.5
```

```
    .Register = 4
```

```
End With
```

```
With pvChannelInfo(3)
```

```
    .Name = "O2濃度"
```

```
    .Color = System.Drawing.Color.RoyalBlue
```

```
    .DataFormat = "##0.00"
```

```
    .Unit = "vol%"
```

```
    .DataMin = 0
```

```
    .DataMax = 50
```

```
    .AlarmHi2 = NullData
```

```
    .AlarmHi1 = NullData
```

```
    .AlarmLo1 = 19
```

```
    .AlarmLo2 = 18
```

```
.DataNow = .DataMin + (.DataMax - .DataMin) * 0.5  
.Register = 2  
End With
```

```
With pvChannelInfo(4)  
.Name = "O2分圧"  
.Color = System.Drawing.Color.BlueViolet  
.DataFormat = "##0.0"  
.Unit = "kPa"  
.DataMin = 0  
.DataMax = 200  
.AlarmHi2 = 160  
.AlarmHi1 = 120  
.AlarmLo1 = 19  
.AlarmLo2 = 18  
.DataNow = .DataMin + (.DataMax - .DataMin) * 0.5  
.Register = 11  
End With
```

```
With pvChannelInfo(5)  
.Name = "CO2濃度"  
.Color = System.Drawing.Color.LimeGreen  
.DataFormat = "####0"  
.Unit = "ppm"  
.DataMin = 0  
.DataMax = 10000  
.AlarmHi2 = 5000  
.AlarmHi1 = 3000  
.AlarmLo1 = NullData  
.AlarmLo2 = NullData  
.DataNow = .DataMin + (.DataMax - .DataMin) * 0.5  
.Register = 3  
End With
```

```
With pvChannelInfo(6)  
.Name = "CO2分圧"  
.Color = System.Drawing.Color.Aquamarine  
.DataFormat = "#0.000"  
.Unit = "kPa"  
.DataMin = 0  
.DataMax = 1  
.AlarmHi2 = 0.5  
.AlarmHi1 = 0.3  
.AlarmLo1 = NullData  
.AlarmLo2 = NullData  
.DataNow = .DataMin + (.DataMax - .DataMin) * 0.5  
.Register = 12  
End With
```

```
End Sub
```

```
Public Property RecorderNo() As Integer  
Get  
Return pvRecorderNo
```

```
End Get
Set(ByVal Value As Integer)
    pvRecorderNo = Value
End Set
End Property

Public Property RecorderName() As String
    Get
        Return pvRecorderName
    End Get
    Set(ByVal Value As String)
        pvRecorderName = Value
    End Set
End Property

Public Property ChannelN() As Integer
    Get
        If pvChannelN < 1 Or pvChannelN > 10 Then pvChannelN = 1
        Return pvChannelN
    End Get
    Set(ByVal Value As Integer)
        If Value < 1 Or Value > 10 Then Value = 1
        pvChannelN = Value
    End Set
End Property

Public Property GraphCurrentNo() As Integer
    Get
        Return pvGraphCurrentNo
    End Get
    Set(ByVal Value As Integer)
        pvGraphCurrentNo = Value
    End Set
End Property

Public Property NetHostName() As String
    Get
        Return pvNetHostName
    End Get
    Set(ByVal Value As String)
        pvNetHostName = Value
    End Set
End Property

Public Property NetPortNo() As Integer
    Get
        Return pvNetPortNo
    End Get
    Set(ByVal Value As Integer)
        pvNetPortNo = Value
    End Set
End Property

Public Property Active() As Boolean
    Get
```

```
Return pvActive
End Get
Set(ByVal Value As Boolean)
    pvActive = Value
End Set
End Property

Public Property Visible() As Boolean
Get
    Return pvVisible
End Get
Set(ByVal Value As Boolean)
    pvVisible = Value
End Set
End Property

Public Property ChannelInfo(ByVal vCh As Integer) As clsChannelInfo
Get
    If vCh >= 1 And vCh <= 10 Then
        Return pvChannelInfo(vCh)
    Else
        Return Nothing
    End If
End Get
Set(ByVal Value As clsChannelInfo)
    If vCh >= 1 And vCh <= 10 Then
        pvChannelInfo(vCh) = Value
    Else
        MsgBox("ChannelInfo Ch." & vCh.ToString, MsgBoxStyle.OkOnly, "エラー")
        Beep()
    End If
End Set
End Property

Public Function GetAlarmLevel() As clsChannelInfo.eAlarmLevel
Dim mAttention As Boolean = False, mWarning As Boolean = False, mWarning2 As Boolean
= False
For mCh = 1 To ChannelN
    Select Case pvChannelInfo(mCh).GetAlarmLevel
        Case clsChannelInfo.eAlarmLevel.ealWarning
            If mCh = 5 Or mCh = 6 Then
                mWarning2 = True
            Else
                mWarning = True
            End If
        Case clsChannelInfo.eAlarmLevel.ealAttention
            mAttention = True
    End Select
Next
If mWarning = True Then
    Return clsChannelInfo.eAlarmLevel.ealWarning
ElseIf mWarning2 = True Then
    Return clsChannelInfo.eAlarmLevel.ealWarning2
ElseIf mAttention = True Then
    Return clsChannelInfo.eAlarmLevel.ealAttention
```

```
End If
Return clsChannelInfo.eAlarmLevel.ealNormal
End Function

Public Function GetDataInfo() As String
Dim mAlarmLevel As clsChannelInfo.eAlarmLevel
Dim mAttention As Boolean = False, mWarning As Boolean = False
Dim buf As String

buf = RecorderName & vbCrLf
buf &= "状態:"
mAlarmLevel = GetAlarmLevel()
Select Case mAlarmLevel
Case clsChannelInfo.eAlarmLevel.ealNormal
buf &= "正常"
Case clsChannelInfo.eAlarmLevel.ealAttention
buf &= "注意"
Case clsChannelInfo.eAlarmLevel.ealWarning2
buf &= "警報(CO2)"
Case clsChannelInfo.eAlarmLevel.ealWarning
buf &= "警報"
End Select
buf &= vbCrLf

For mCh = 1 To ChannelN
buf &= " "
mAlarmLevel = pvChannelInfo(mCh).GetAlarmLevel
If mAlarmLevel <> clsChannelInfo.eAlarmLevel.ealNormal Then
Select Case mAlarmLevel
Case clsChannelInfo.eAlarmLevel.ealWarning
buf &= "警報 "
Case clsChannelInfo.eAlarmLevel.ealAttention
buf &= "注意 "
Case clsChannelInfo.eAlarmLevel.ealNormal
buf &= " "
End Select
buf &= pvChannelInfo(mCh).Name
buf &= "(" & pvChannelInfo(mCh).DataNowStr & ")"
buf &= vbCrLf
End If
Next

Return buf
End Function

Public Function GetAlarmFlag() As Integer
Dim mAttention As Boolean = False, mWarning As Boolean = False
Dim mFlag As Integer = 0
For mCh = 1 To ChannelN
Select Case pvChannelInfo(mCh).GetAlarmLevel
Case clsChannelInfo.eAlarmLevel.ealWarning
mFlag += 2 ^ (16 + mCh - 1)
Case clsChannelInfo.eAlarmLevel.ealAttention
mFlag += 2 ^ (mCh - 1)
End Select
End Function
```

```
Next
Return mFlag
End Function

'アクションモード設定
Public Property ActionMode() As eActionMode
Get
Return pvActionMode
End Get
Set(ByVal value As eActionMode)
pvActionMode = value
End Set
End Property

'アクションモード文字列取得
Public ReadOnly Property ActionModeStr() As String
Get
If Active = False Then
Return "停止中"
Else
Select Case pvActionMode
Case eActionMode.esmSleep
Return "休止中"
Case eActionMode.esmWakeUp
Return "起動中"
Case eActionMode.esmStandBy
If CommMode = eCommMode.ecmOnline Then
Return "監視中"
Else
Return CommModeStr & "(W)"
End If
Case eActionMode.esmRecording
If CommMode = eCommMode.ecmOnline Then
Return "収録中"
Else
Return CommModeStr & "(R)"
End If
Case Else
Return "?ActionMode?"
End Select
End If
End Get
End Property

'通信モード設定
Public Property CommMode() As eCommMode
Get
Return pvCommMode
End Get
Set(ByVal value As eCommMode)
pvCommMode = value
End Set
End Property

'通信モード文字列取得
Public ReadOnly Property CommModeStr() As String
```

```
Get
  Select Case pvCommMode
    Case eCommMode.ecmNothing
      Return "無効"
    Case eCommMode.ecmOffline
      Return "オフライン"
    Case eCommMode.ecmOnline
      Return "オンライン"
    Case Else
      Return "?CommMode?"
  End Select
End Get
End Property
```

```
End Class
```

```
Public Class clsTypedef
```

```
    Public Enum eDialogReturn
```

```
        edrCancel = 0
```

```
        edrOK = 1
```

```
        edrJump = 2
```

```
    End Enum
```

```
End Class
```

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class frmRegLicense
    Inherits System.Windows.Forms.Form

    'フォームがコンポーネントの一覧をクリーンアップするために dispose をオーバーライドします。
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Windows フォーム デザイナーが必要です。
    Private components As System.ComponentModel.IContainer

    'メモ: 以下のプロシージャは Windows フォーム デザイナーが必要です。
    'Windows フォーム デザイナーを使用して変更できます。
    'コード エディターを使って変更しないでください。
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Dim resources As System.ComponentModel.ComponentResourceManager = New
        System.ComponentModel.ComponentResourceManager(GetType(frmRegLicense))
        Me.Label1 = New System.Windows.Forms.Label()
        Me.Label2 = New System.Windows.Forms.Label()
        Me.txtProductID = New System.Windows.Forms.TextBox()
        Me.txtProductKEY = New System.Windows.Forms.TextBox()
        Me.btnRegister = New System.Windows.Forms.Button()
        Me.lblStatus = New System.Windows.Forms.Label()
        Me.btnClose = New System.Windows.Forms.Button()
        Me.SuspendLayout()
        '
        'Label1
        '
        Me.Label1.AutoSize = True
        Me.Label1.Font = New System.Drawing.Font("MS ゴシック", 12.0!,
        System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
        Me.Label1.Location = New System.Drawing.Point(14, 15)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(111, 16)
        Me.Label1.TabIndex = 0
        Me.Label1.Text = "プロダクトID"
        '
        'Label2
        '
        Me.Label2.AutoSize = True
        Me.Label2.Font = New System.Drawing.Font("MS ゴシック", 12.0!,
        System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
        Me.Label2.Location = New System.Drawing.Point(14, 44)
        Me.Label2.Name = "Label2"
        Me.Label2.Size = New System.Drawing.Size(120, 16)
        Me.Label2.TabIndex = 1
    End Sub
End Class
```

```
Me.Label2.Text = "プロダクトKEY"
'
'txtProductID
'
Me.txtProductID.BackColor = System.Drawing.Color.White
Me.txtProductID.Font = New System.Drawing.Font("MS ゴシック", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
Me.txtProductID.Location = New System.Drawing.Point(140, 12)
Me.txtProductID.Name = "txtProductID"
Me.txtProductID.ReadOnly = True
Me.txtProductID.Size = New System.Drawing.Size(188, 23)
Me.txtProductID.TabIndex = 3
Me.txtProductID.TabStop = False
Me.txtProductID.Text = "AAAA-AAAA-AAAA-AAAA"
'
'txtProductKEY
'
Me.txtProductKEY.Font = New System.Drawing.Font("MS ゴシック", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
Me.txtProductKEY.Location = New System.Drawing.Point(140, 41)
Me.txtProductKEY.Name = "txtProductKEY"
Me.txtProductKEY.Size = New System.Drawing.Size(188, 23)
Me.txtProductKEY.TabIndex = 1
Me.txtProductKEY.Text = "AAAA-AAAA-AAAA-AAAA"
'
'btnRegister
'
Me.btnRegister.Location = New System.Drawing.Point(196, 70)
Me.btnRegister.Name = "btnRegister"
Me.btnRegister.Size = New System.Drawing.Size(63, 23)
Me.btnRegister.TabIndex = 2
Me.btnRegister.Text = "登録"
Me.btnRegister.UseVisualStyleBackColor = True
'
'lblStatus
'
Me.lblStatus.BackColor = System.Drawing.Color.Moccasin
Me.lblStatus.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
Me.lblStatus.Font = New System.Drawing.Font("MS UI Gothic", 11.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
Me.lblStatus.Location = New System.Drawing.Point(17, 70)
Me.lblStatus.Name = "lblStatus"
Me.lblStatus.Size = New System.Drawing.Size(173, 23)
Me.lblStatus.TabIndex = 4
Me.lblStatus.Text = "ライセンスが未登録です。"
Me.lblStatus.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
'
'btnClose
'
Me.btnClose.Location = New System.Drawing.Point(265, 70)
Me.btnClose.Name = "btnClose"
Me.btnClose.Size = New System.Drawing.Size(63, 23)
Me.btnClose.TabIndex = 0
Me.btnClose.Text = "閉じる"
Me.btnClose.UseVisualStyleBackColor = True
```

```
'  
'frmRegLicense  
'  
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 12.0!)  
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font  
Me.ClientSize = New System.Drawing.Size(340, 101)  
Me.Controls.Add(Me.btnClose)  
Me.Controls.Add(Me.lblStatus)  
Me.Controls.Add(Me.btnRegister)  
Me.Controls.Add(Me.txtProductKEY)  
Me.Controls.Add(Me.txtProductID)  
Me.Controls.Add(Me.Label2)  
Me.Controls.Add(Me.Label1)  
Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog  
Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)  
Me.MaximizeBox = False  
Me.MinimizeBox = False  
Me.Name = "frmRegLicense"  
Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent  
Me.Text = "ライセンス登録"  
Me.TopMost = True  
Me.ResumeLayout(False)  
Me.PerformLayout()  
  
End Sub  
Friend WithEvents Label1 As System.Windows.Forms.Label  
Friend WithEvents Label2 As System.Windows.Forms.Label  
Friend WithEvents txtProductID As System.Windows.Forms.TextBox  
Friend WithEvents txtProductKEY As System.Windows.Forms.TextBox  
Friend WithEvents btnRegister As System.Windows.Forms.Button  
Friend WithEvents lblStatus As System.Windows.Forms.Label  
Friend WithEvents btnClose As System.Windows.Forms.Button  
End Class
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

```
Microsoft ResX Schema
```

```
Version 2.0
```

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

```
... ado.net/XML headers & schema ...
```

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader, System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter, System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">  
  <value>[base64 mime encoded serialized .NET Framework object]</value>  
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
```

```
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
  <value>[base64 mime encoded string representing a byte array form of the .NET Framework object]</value>
```

```
  <comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture. Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

mimetype: application/x-microsoft.net.object.binary.base64

value : The object must be serialized with

: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

: and then encoded with base64 encoding.

```
mimetype: application/x-microsoft.net.object.soap.base64
value   : The object must be serialized with
         : System.Runtime.Serialization.Formatters.Soap.SoapFormatter
         : and then encoded with base64 encoding.

mimetype: application/x-microsoft.net.object.bytearray.base64
value   : The object must be serialized into a byte array
         : using a System.ComponentModel.TypeConverter
         : and then encoded with base64 encoding.

-->
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
  <xsd:element name="root" msdata:IsDataSet="true">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="metadata">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
            <xsd:attribute name="name" use="required" type="xsd:string" />
            <xsd:attribute name="type" type="xsd:string" />
            <xsd:attribute name="mimetype" type="xsd:string" />
            <xsd:attribute ref="xml:space" />
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="assembly">
          <xsd:complexType>
            <xsd:attribute name="alias" type="xsd:string" />
            <xsd:attribute name="name" type="xsd:string" />
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="data">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
              <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string" use="required"
msdata:Ordinal="1" />
            <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
            <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
            <xsd:attribute ref="xml:space" />
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="resheader">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:element>
  </xsd:schema>
```

```
<xsd:attribute name="name" type="xsd:string" use="required" />
</xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>
<resheader name="resmimetype">
  <value>text/microsoft-resx</value>
</resheader>
<resheader name="version">
  <value>2.0</value>
</resheader>
<resheader name="reader">
  <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=4.0.0.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<resheader name="writer">
  <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=4.0.0.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<metadata name="Label1.Locked" type="System.Boolean, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<metadata name="Label2.Locked" type="System.Boolean, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<metadata name="txtProductID.Locked" type="System.Boolean, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<metadata name="txtProductKEY.Locked" type="System.Boolean, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<metadata name="btnRegister.Locked" type="System.Boolean, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<metadata name="lblStatus.Locked" type="System.Boolean, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<metadata name="btnClose.Locked" type="System.Boolean, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<metadata name="$this.Locked" type="System.Boolean, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<assembly alias="System.Drawing" name="System.Drawing, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
```

<data name="\$this.Icon" type="System.Drawing.Icon, System.Drawing"

mimetype="application/x-microsoft.net.object.bytearray.base64">

<value>

AAABAAQAAAAAAAEACACMEgAARgAAADAwAAABAAgAqA4AANJ6AAAgiAAAAQAIKGIAB6iQAAEBAAAAEA
CABoBQAAIpIAAIlQTkcNChoKAAAADUlRFRlIAAAEAAAABAAgGAAAAAXHkoZgAAAAFzUkdCAK7OHOkAAAAE
Z0FNQQAAsY8L/GEFAAAACXBIWXMAAA7DAAAOowHHb6hkAAB6IULEQVR4Xu29e6w321nfd84x8UUQCMfg
QFXxqhqdZebYjAWjYSELcbOkY2dCFAq/iilFCIGNSbFChXDRCGJDiWLAfSfYGMsXItAOeVQxQaEjCku
kexitW6gOLHTPwxOjCmi8PE57919Ps9an5ln1p757b3f231/e8/a7/edmTVrbms93+961mXmd9e1Rx45
2bfjx9VECSAe9rCHqxd2AdjDHq5w2AVgD3u4wmEXgD3s4QqHXQD2sIcrHHYB2MMernDYBWAPe7jCYReA
PezhCoddAPawhyscdgHYwx6ucNgFYA97uMjHf4A970EKH10AbkkgPyvm0GKWF2vplgNprsX/D8Xy4b5d
QDmyesHgGTZDnrefezVx3VF3jttjaPvHvz3cvrALwM0Mac/8d21Ay18N/OGIu5b/NzzyCAhiWwyTL6x
bVyGdtwj1z4d0UEHnm4HRslErRt3wL+Txj39UMaWA9gG3mNuOZ83o4MLLl37qOhxRF8/h6I7nie8+Yz
z/kA+H8Pty/sAnAzgwaewY113s6xj6TBnySRJQrbjQhsJ0Hybz7GLf8ou4dNYKhMlrT9nNOSMq9oZ+v7
AXF9d9zjIycPxnqnacSjtj/+ksye18M9X27M8Yv9PKPPyd8ebmfYBeBmhsmd4fKSKbjv06cJEMnWqAJ
w0MR2wiUkeO/JF1fd7sRsUXkua7ZRIjzUXM//KenkF7ECqb93EM9T2xP9XS9br+X6Zkq+j0lJQnaou3j
P587I/bwKIRdAG5i0LZnDAYuawvSFw57Y9maB0nGJF1gcnjPKTHHS+49qk/O3noP37y5FN//PHax07+
9N995OQTH/m9GR/+3ZM/+lf/e+KjH2hL4hI9jcew5BwP/9knJ3B+BShvzbtpt58bjfxslJA7SV8EgMAi
V82bQJ6gx+e+PdyusAvATQyzDfNXjZ9aPogsawoUAGA0671c7Ru1cK2ZIXsj/MdmwgeZITf40HseSPzL
X/m5xLve9sOrcl/prQpEEQPA9RCCvA+8Am+0YPziQO8PyDxgH9s8TwQWuVrS7wLwqIVdAG56wOwx/uY2
TzU5yHYyxt6IT1r+r+50YqjZp1q91OYVjzbvWABYv//+v79KfsH+JgLL2AcMT6FCj2EWho8vhSHFzmd+
sOCBatCeEplgfQ5s7QLwaIddAG56IC+rAMxkcG/WirLfhC6+y2jzB6Eq6a3VJXTF//3uf3Dyr9+7xEP/
/o0LjPsrTtJyXuG1Ro+B++L+uNf0TlIA2j0052tPyP86Ci3GPDafughkzDKkd7Tb5i0NuWdc5NAMdjnM
lwQAY24ZDvhLwnS3fnLng1TWuo3871gQ8Vff8sKTD/7yjEOEvh5UEfAa1WNQDPQW9A6qZ4CI4dE01gNy
JkIsizq9Hf6aACyaCT3sAnDrwy4ANz0q85FDZMRs4ABafQPZSdbkr634a3pAQSEZJXgIx7+968+jI+/
aBOf+vC3J7biwdolxegdJj7BsnmAZ0A+dMR68wiaAABESuULaGEXgFsfDgG4mYfSHLIys9aa8Fpz8SE+
taW1velvCSUg2lotLdbIDnEf+ugLTuFP/6/nnQtVAPKc/fz1utyX3gEeSelPACkC3SvQM+CZWydiyyAF
AP+oeUisfxeA2x12AbizGwZesF12Q28hjDrIb43fSN867qhJQaldt4i/RvhKekn86d9/9sknf7vjffee
wsff/bQlfu2Z87KDYz/x3q9PjMLwh++dorBwz3oFei8NzSvGuScRIJTsWY4g7AJwO8MuAdclK18a7dyp
12r8jy9q/FrTKwCbhf8he63ZJ8J3gn/it5568ke/1vD//W9PwCVHf/HzF/j9d/6FU/iDd/7niY/+0hcn
PvxLX5NAIBSHKgzVEED1DGwa2EfASEd3A3YBeBTLdG3FGq+4cS23n/IT//Wo0Pak0PIPKh41fCQ3bw
4Ie+6FQND/EPkX8kvZDw/+rtTz4TW6Lw//6vz5ugEJz2CFo/Ac0D+ghSBMjC9Jb00wfAutjDzQhXWgA0
sFNgX0GGRST/YahtbJ/8Iypn54WLa61Pjdc69+Ya/7y1fa3pK+nBn7y/EX0Nkn/EWUJQa/4qBh968+cu
8MGf/JyTD7zpSafwz98yskf/LOnpxhUz4BnVQj0CFIEakch+YYYIjYrf2ZxLuMvhTW9hDbXoIG9pgmQ
PsbDg3fHs4OuwCswJCG1VbnDtdVWCEAEH465uFTxMf4hTX+IeJLel17iQ/pxRrxxRr5QRWaf/uLT13A
Wh383s8+5ZQ3MJL//W94YuJ3Xj+DbfZzvEKgN1CbBtUjsLPQZsEkAJRDbPG/IwSt76AKQPMULBKKAPK3
4UTKZg/nCVdaAAgj+cFkVT3Mm/F/tltZNVILxvN1+THSwuMdrPE7+a3x12r7s8gvyenIWY08oKbfIv9I
+grJL97/hid3NOJXMXjf6z6rCcI/esrJB974tJMPvv2ZCyFQDBw9sH+gdhI2b6AZZ5I6yN5IDekrKJF5
f8Uezhd2AUgj2xCAki3N0HglthsiQpCewNi7P0/csdZfJX6v8UUVgDXin0X+ijXiilrbn4f4tfafyP+m
p3QBALMAQP73/uhnnfzmd33Wya+/6jNznbr4BWPTQBFoTYJ5tABvwL6BbBpA6Mz7LISW5yWzAhFgf59y
vQvAhcOVFwDCgVwthshuZITYh4F1Q8PwsnaK2qqSX+IfrPUL4UFuFy8A1/88ArBGfDCSvwrAeclfiV/J
39x/id9Qa/4qAIqAQkCa/+efnBYC8on8qt7A1EGl19X7WJg9qPOVKym+CEBrEuwCcH1hF4CvofvZNXjP
WH1efyLarDmZz5rF1zr6qNk01foSn9p+IQB9P/H06FfiJ738FxAiuONIABiFiBa49vR12r7Tvofi2Xg
/3jdn08oAEAB+I0f/OyTX3/5E07e9dInTEJAs2DNGwBVBOhHaf0CURYhBooAutza+sULS6GYyb8LwPnD
LgArIY0s/vpGFwAMz3n9D2YtBfkl/lqtPxF7IH8F+2r6FAAm4RSyJ+QHa+Qf06yhpt8SBETAGR92+unq
S/o18k8C8KomAAABeNf3zULwWz/6eZtCgAgAmgQIbL5oBMHjDwGYZw7y0ZRRZBDDhvRPw4mEXgJUg/XMJ
6Uv+WPOPE3og/0hmib+GJD/rlykAa9Bb0HM4T0filiAoBHoDs/vfBACiU6NXWPMD3X89AEXggRc/PtdJ
Q//Aob4BPQHENkke+d/KxqZYI3ojf64F8Ny697aHM8MuACvB7KAZkO7+1D/XspOqk1+XfyJ8RSX91v6y
jijQB3CI8JJYcl8vuMYaHEmoTQb6C+jIQwwQglrbnyJ/1PynyN9rfwXgl194T4I0iArDhowUKATkJ/n6
wbtfkk0CmwMUA56YHkCCpsAUDgG4aNgFYC3gVmaWzIaGK2qbH/LT4YeRYrAQeLPW7/tGVNKLJH+f3VfJ
XrFGZuEQ4hrwLMDaPjHOLAR6Bw4Z4hUgArXzTwGA0Onyv+JJDdb8ogjAP/9bTQSIp0lQJxGRF3oCNGf0
BNrIQLj6DBf2skmRTm9A8u8CcN6wC8AYyIrMdoYItmcjfx3mo4YSE+krOvHBGvkb+xakF514Zx+fjBC
sosqMmDczz2ItXsBeAZ4BAiBTQ9AoRAAZD87/6+zz751Zc8sdX+uv6x/BcvevzJA9/9uAQi8EsvLE2C
8Aa2+gXI98kT6HMvwCTSub4LwEXDLgCnAnlBbdJnnPwv47xbxJ/wBrxAfS5A+CuENX4m+Rf43klcAA
AgmJVOE+5+7X9KkerwPcvjXYxWD0CKoIIACKAEAAKvNB//Kdj08RqN5A7Rcgj7jfeb7A0hOgc5Cyqh7A
3gl4sbALwKlgJ9Ps9o81/0j06u6LLeKtbkGsgfQj8WvNXolvbc55IAtE/j/f+tcsv/OT95388o8+5+Tn

f+i+BYx74DXPPvmZF33tBOI5hmMVBcXCayyEIPDx/goxHoH9A3QQ0iRIIaD932v+SvoKBODnv/WeCQgB
AuJ8Aa7pfegJ1D6B11Rrttv6AmYBmOd07OFQ2AXgVGgCcO6aX/IXAdgi/1qtfx7ijzW9pJAYv/O2b01A
7re85AWJn/y2rzqI13/TF5783b/8Bblkm2M4HiHwfJwbcJ3qHYhRCBw1QAQUAGr2Bem/Y41f+B8fm+T/
uW9uIkB6BKSQPUEXj6BJgK93Ar5dwe4X7j8AlAejdU2wadNJEKjwYAAsbHIuKHNb2//RPqKM8hP3Ej8
Q7X+Vk0PAazlqcGt1SvpxS/8+PecC+NxFYjC6CUgCtU7GIUAEBvAG9AT0AR0N0HawIqgifAeRU9Bwny
BD78u1lW2VeTbv9c87PIVTCVcWvaUfal2K90uNICKdFpKsxiT4DOJKb3/od/85sL8mOAE+kPoBifYThU
66/V+GvE5/qQj9oZMtaavJIWUkMOCUJNUX2A9OPgoB3UMG1EJstz2D0BmgW0KmnN4AIQGzdfYgv+RWA
t3/D3Ymf/e9aWj0BRYD84Lp4AgwR8hwIdrsitPbvHYAWbWamvwJAWe8CQLjkAsBznX42RGCNxxEAEw
bn81/3k7/SrWiL9G/kp6AbGs7SG9NX2tvSU79yizJXdOoInaETf5LJC2TWVeFwVQhQHoHdhvwp3yrD4v
hOVjIYiA8wYQAbYAJPt/P2NNAH76b9yd24wQMMRIZyPn5fWk4tgxmJ2CUaKWJQ05/lpg2YRfeli3jKsX
LrkANMXPgrfEAXiGfcdpJg4pFdfcX6wRvBN1Fr/APHXyG+NX2v7SjxJj9Fzjxg+/RRMTqI9LKgRR/BS
POPAPuCXnAuQD23OwwMLDwHyV6/AgH3LfqIEAHY0zC6gLMItBdftap9SH/276xAREgjk5EPAnOpQhU
T4B7Q8C4d4cCU9Sj3ClhRH4iP3beTV1zuOrhkgTAH8e34KdSrWJACKeOnX6V/BjbsPDFdkXsg8gQ+yK1
PteANFS1PkZuDe/0WN+d7w+0GU4XLWTYCD1vWoh2cggDeVI9hDWPANTmgSIA9AbWRACCC4mvB/BTz70r
gTjQmYgnQfMCEahCad5wj9yrImDZt+HB5hXko/XHK6tXOlx6DyAFYCh8VtILiO189mvN9YdYkJ+aZVHz
Q3pRCV8R+3SBx7b+FvKxZIC7L/ElFESbXNyoiAMRMXBr+BwDT2HryKc6DZ6vJYllusltfkn7vZn28FxL
tmWk4eWbIJIeQ/MM5h8ugWw2FbhXvQLWzvbvqDSgCdYSANj61fxL/BQ0Qn+2fYvm8x5y89b7Y//VNBDiG
UYbaH8D5uVYVgSlv4lmaB7D8ZaY9LM01F4BWA8CAtmhkaa5/dgIFMBqMGwOy9se4Krk3BSDiMEYIfajm
Z99Ifq5RyQ/p6dyqBs19pfsehtzErNdojdXtAQLZs90NfY41iWesESNAEYCEETK8FkD5FIs7R49o1SMe1
mzAgCAjB2DwQPAVp5IiBwoGiqGJGCCA0nYDZ7n9+EL2IQApBCMChn/PYk7fcF+shAggG41H7A8hDyqp2
CiJUowi0fBkDMbsoXHIBKM/FaoL/wpCz8y9WQwVo80K22uOPcS2Iv0F+yDzW/Ge5/Zifl99aHwOunVoQ
39pe8jcx64/Qe7bbcxBfid1IvADJ87+VNEQfQi7461HxH/dFv1WpGgFgWfAI7B8wP8kjvAFewBECXHua
AhD8rc9rQARY/bkRHyaKAAAC8BWYS8qkx+wPMR5oCiBD3QBNUeOF4tiqKGaYVnp18YNmCQ4hXKVxyAWjG
unw8jKHNHaeGw1BabfaOw+QHB8gP2c9T85Neo4UglvrW+HMNFjcd0C3PmiweJB+FZ8onkcQsR5Cio6y2
9COWIZOVJcHD+201UniiptUj4BlsFoixk9AmAcN7vFYMqSE3pIfolNMDCCHAM6AZ8I//SgIDTYFXfWYK
iE0BgKdr+wMQAe6pFTz1HwCV4DKfexeASy8AWm5b8NfcWTB+zmtY/UfirwCRSPKXWv8s8nMMtT6k0OwN
1of4EEjyTwLAvyibrOVxyXs55Sj3d+NewL+tUikvUEY7Zgv010wbCFMsOSJFKJooEC77CD78uwshYKYh
y1EEWsfGU/L1YYYHYE3dxHIJkGQXg+ADkH6B9aaApyP885NgXekd5L3STOGPo2c+BX3zP1n4L13Abga
AtAX1KMQPwu+9Pozzi/51wSgegSs15r/Im3+sb0v+SGPooRnwr1xtwlhoGm5sdphcZWoc605xIhgQ9ue
O8rWhGRCXNi+gUTEkaeeI88T4oUQ2D9QvYFDIgcxcxgWsg9/QG0aKAI5KvDSNirAC0mUafnLOTm/+YoQ
2YTiLm3yZd6B/G8mP2EXgMsw8r1aDZefkwJp3K32r67/lvvvNsvzkn+tzS/5MVCiWUXuUkRo2ckXxKT3
PY22G2jcdxpn87sjosWz5rKuu5zBXyX3DK4jzKulvxZPjSkkVBtSbce32hbS2a+CN8Az2y+gCCAAigDt
eVoDIHcl/iQGscQzeNNfm72AcYIQqE0B+1KaAHSvL+613XNEZZhWMuwCcOkCRtlquTTxeEyWEK66/hp1
Jb+EN471FvkVgK02f635TxknvdVBb06LWjWX+dcMfC6LdGM1ZpbZ+ddI2w4IEFzmSssDSvtDI8SMFtkW
NbjLc3l97IZ9egTsTw+G+NjOTsLeJMADQACcQVhFgI+B8IFR5gfk0GAQPvsDnhteASMEjAh0AUAYSIMX
wNwC5waQz5wLlIacpoHfFPWR/D/B+V56RMOXBfQqXXAAgR6vpFADWMArIP9b+a+QXkBlis/5Drv+5yU+2
BzRkKktij+22ewHuvxmx5A9sJc7/SNPiEvoAgk4dB1YC95XXz9qOnKuoFMSf7Cm2s4Owi8AhTWARqE2B
dP37SEAdQB0EvIiEV6AHYKKA0e0KcA1EfoUpspY0/N18exhF4BLEJrKN2CALC9PZ81q7Q/5a8ffGumF
ZF6r+RWAQ27/TP53NIPM2rJl/vUZ3UWPIb240XD2eVr+sxL/EIRsEnxs4QmQJ4oA4NVfVhRMzU5TAJKn
CDAZqIsAglBnCPLBEYYTefnIUQEFwDx3RCBFs91SgPujDPCKEIGrGy61ACS5qJUivglAn/EXNZK1/xr5
JbHER+Q/5PrXYyr5cUmzZR81P9fnhrgrheCyhSnve84n0UJ4eX49AYSA5kd1BBCB3/nxp+T8f6YHzNg
RQBoCqQX8J2Pn98YLE0B8/50X0C1BftcdgG4vB5ALqOwY5VCxwghIK/6jsN+Z5J/o/YnDvIrAJxDA2S4
T/JT83N9APG5v8uU7zW0rIdY9hcEogaGhIoAAsAQIQJjftWmwHtf8+R08ekArAJA34ACMI0I9PcE8AIo
I87B+RCWSS8Asjfy7VFMyZCE62rFS63AERRU8gpAmGAjvtb+x8SgPOQf6z9OZ7zQX5qN2q6rPnD4+Da
jQzXMTpZhvEA8j4vV2iPVAUGPC+f056Z/KB21hNABJgforDAU4DvCOIF5GiAAvD8NhSIANAp+//5mPT
C2BeAB8iYViQpgDnoRzGvgCqAQUG7iSXNewCcAnCugC0jqtX3J+aogpAJf5E/u72H6r9FQ30Vaf3YuRL
8j+YBpjeSN5sx6UMTQCaCMzDsDzv2ugAokn+KQJ8KtXhQVz/BH0CATsDEYX/+Vsf17MDedEIEaApQN1R
nghALYf0wHJSEHfE3zLsAnAJQhUAa/7sBY7CP6vz71TNv0F8Y00v+TmmzvKz/dlczziUqQCQ5QwBJ1Eu
Y4gHzCHOqP1jVfD0LAudc7jmlQuAsIgywAugLwARgPTU/OkN9CFCXxRixCDfEXjDE9vbgqUvQDGMlgwG
0AyB+r1/eArVbq5SuBQCUAAsP8DzTdna09U6oqAUq+asAVPJXAAaik5wcyqgDU2p+a69df/5fs9ffNNAQn
DS6EqJK/5Xb8rwBcVpuDZXTcXoJHjLX4ezDLIsVjI8s3ChHP2hSg4BhQTr8qgBQ80N+3hEgjrCk6Qxk
XoCzAzkhQoJnUTtheZ68p7ifFKRuJ1c1HL0ATEQvWAhaaf/T+UenkD3Pkr8Sf0sA/IUCyV/b/hqbtb9T
UbPnGRL0+2zkr31N7X8FjI9HTPD0TQBY5hyBMjJQmWJ6AXTy0ReQ3wh4fiCaBPULIYTAiUF4ATkkGF5A
7QtA8PXIWL9AlEpcDwIw28kVKIEvCkFYAJWF7UsRKYdf9QyCsBZ5Jf4VQDYJ/lJP9U0pbbJKb64/nkv
sTrVhjWvL7sA9OYNj9jzIjtkiEBvFuXIQIilaOCI2j/jiIDzAhAAavxRAOgM5IUiZhMyMQgRYIoxAKAZ

cz7On30BITgEiqF6AMtyuTrhUguABkftj5EpABgXxjG6/UIBGMmPe4kAQP5a+9vWlPx0NHJNrJ6czSGx
LgAJIjNcZgHg2ekA7CJA6OWBAEB+CEigba4XkPkYQkqe6gXwcVC/FERTgCYAnYIKAF4BzQDEgjkBvCNQ
BQCPz1EZyiaoh2gdk1UA0mauWLjCagDC3aQ2xv2vvf8Yxyb5BwGA+ID1WvsDzuUv7eD6c535BR9cftDb
+pdMADKPNwP7eG5EoIcuAHPG6YIT3b2A2hSgP4Uyoi+AHxzVC6gCAPkBCtQDSMeLRXoBlCNliphwzkkA
6IPiCuGWBpu5YuFSCMBYiCKVPgq7uf8PpAdg+m+RH0h+2pGV/Nb+VQBq7Q8cb9bFpQPsrj8MDgJMyH/t
71jDnM9rgXiemyFYn5L/myhkvSRmZkXkk14aJGWCEKJqXwCzA5kYhADkC0F9chBLBIA5AXgIjBrQDMAL
GAVAD2AcDci/hclcrXDpBWB0/w8JAAZD55HkHwWgkp/0GCfuKuS35z97mTHya4w3Nw8g3d1VAWg41rDM
67XQ9vG41PbkAeKYCE+AcksVAMQ24k7ndqaPCV/Ac2A+7/3iUnyFICo/RGA2hdA84BJQbwgXjyA7Azs
LwnhReBRjKMB2Rzp9y+uWjh6ASak0Rj911+CzwoD673/Dv+dVfuv1fxgdP9JT81CpxUly6LnP2u5Vutz
L4QwL/6bsVY91GF+bP7mbxAQmcWEQsT22Bda2508/uDbn3nyG6/9vBwNQAcY3V8EwKnB9APQDGA40H4A
RAAB4Dycj3JCqGmmTf6IZiu7ABxliELE5c7ahZoXIYhnCsNCAGj3naftb+1fiU9NsYAnMe2PzULrMW6
/72NqwAcf97evEBozOCvhyja21YsPcF4FEhAtTcEBgvGF1/tQngtWUAPZnP8AbnjwJAOD4bEqEAPj9
RbyAZi+7AFwoAXDKb/EC7ADE7bM2qQJQ19cEAPKDJ/7i52f737F/0mNQP//i/3Lq/MogEJupvd/NexeA
OZATovkAzTvK8qNJEN4TXgCCDVFpCiCy5DVeAM0AiJ7fDUQE+qQgBcCXg+wHqF4AZYON4F3QzFh0Bnby
7wJwLIGCa4RzOa33DkA6/2irY0SSHSKpQAJQya8AQH7AORCTX/o77cc5Ezi55z88kN62JfXfo9oKrWsm
Eiri2AQgZ0b2MrMvwm5AanFeEHIokGZANgH6NOEUg9uv0JsPwCjAYhAFQDKy87AXQAuhQDMJKsFaM/y
KAAQfVMAGvxA81cBIB7yky7blG/8ynRRMSbalFxl1781A66eMZ0KZMGQDUZnu/I/hBu0vENM8arsDHQ0
4F2veFL7GbH+PkA2AforwggB+5g6zJeF6Afg3YD6CXfEm3MiArsAtHBJBKAsOzAkOwARADuTIPCaAiYl
P6RXBIiz/U9azkPNhHGuu5NtabiKhpWhlEeihMilhQeQeZfC+VA2A9Y6A3/1NZ+f7XyGA+0IRAAGf84Q
/Ia784dG+NUhpgUrAmsCgGjvAnDJBYCaRAGgBqHmHokvMBLai2PND6oAkO49//DpKQB4Ffb+a0zKpTDs
AtBRaptzHuVWF4DmuVFDI650sKYAhMfFSAD9APluQNT600hAkD8FIcQg3wugH+D1T8wPhSgANTuqADsv
jt6jXQD65jGEZiwnQ+jj6yZpUML9HwUAY1gjv65/Jf5CACKdAkBNRPu/jis3423hyhJ+DGRDRQnLXfy1
4UE8AGcGkrCpAiG0iC6gGcDcf0cDqgAAxIHxg+k11ANA2BEAPDfKDa+iDQU6IWgOlt1VKcMjFgCmpQVv
P5eIQ0lMwohsQ64JQJj/QwD4wEQVANxIzkHvP4ak+z+nKcelG/onXfVANlSUsNwF/fvcgPhf7w0vAA8L
wuJx6QUoADkDMLwAiI8YsPT7AKMAUMY23Vq/TftMwWpAlY+WV63sJlIAMBRENEkD6refyYIA1MxVACT7
KQEI8o8CAPkVgJwDUATgnd/1pHn4L2oqXEK6Hbl+Q7+hqx7IhjOyoiWZBSCHCUMAYfPKz9GAd377U1MA
wM9+6xMWAgAQgBwa/Ma7s5mAADafgNEA5wIoAHGU6blF2eWmWf5eUwAcRRgFoNw75M9apAtAdyOd/juS
vwpAbf8zdat5s/YoAgB4RVUB4NzT8N8gALMhsRRXLJzzsBE/BSDLfM+qewEIAO8G/MRff9JCAN7wdfOL
QU4Iyn6AF/Qpwa+ZBcD5AAhAtgkOAWEmKPaRzYBeVrsA3MlhukVI32b9NSGI7Sw4dsV/zMGpfdQguOd1
CvB5BABjAZBfAbATkPQYURWARP2ZUbYpAPZXAOP2YCCfoH0VAPMKAcDLQgAAnXgKwI/91TYFmI4/hwOz
KfC8u9qPhnQBoBnAUKACQD+Cnbet+bYuAHP5Xe5w1AIweQGXlX9hRQAO6Nr+F5X8CgDEXlgeAKAcwpDp
Ij1TUukDaB1JTQDS44gb0ICqITVj3gVgK7S8ogx77U8eZTLG6YZ7znAgTQCgADg1OD2A/oUGBIAMXG+
E8DPjysAeG6jANDHsAvAUTcBrkMAOPHH2n9NANJwugCYZika81BSXHQyoGpIcUMFxu3B0PJqFoAPFAFg
2A6PCwH4jR96xkIAIP8hAWAocEsAKD86AvHeCMtyuxrhyASA+5wJleTvse0R+G8pAL4EtOkBBcr5mUMu
ne5q3wDpeDe9CoDj/yk+eQ97uNGQJIX/CAB5TK89AkD7/V+89AtyFACyZ6+/AvC8+Z0AJGmXG9COWHwv
IARAEUBIGMKdXgq6xhDulRToIxeAFjMh3wVgOm77CbCLCkAlPq7jKACs8716BWCATsqoQxGAqlaL3OyQ
+Rf5yRbdfQB68BUA2/tbAoAHsCUATCzaBaCFKysAlfySG8JXAQA2AQD7EADmAGCUnL95AFyTa7ewC8CN
hXDEscUAES2vhp8EQGgCdA6AmcBwA52AZjD1ROA3ulXBQCS0+EH8SU/bUcFQJHAKGhd51hyGNauALcm
pABEUAAyCuTDK2sCkJK8Kj/a/Hwm9LgHIPoBdAI4gcJ83UQB6B6ACYK2vANiBBP1JowBgjm4C3AXg5gdz
77weQH4uPAQADwD4XQB+LEQBObx3D+B02AWgC0B1+wFvkwHcfcgXg/W9qAkAvMgLARJImANYgLewCcGMh
cy/+u24B6C8EIQbtSvAuAFthF4BBAFqNMQsAnshZHsAuADczkH/g3ALQmwCnBKCPFBudgRiB/sowBwu
hQDYAY8atMLcEADJX4AAUDtAelx+BKChCQDCAPkPCYDzEQwa8B6uL5hztQ9gFACHAF25MDsBiUcA3v19
n706FLgPay7DlRAACnxLAKjdFYBG+mYwvE8OziMATl+V+LsA3FiI3Mv16AHUYUq/9oEQAAgP3F0AiIA
dATyebBdALbd1fMAMne/JgCt9j8tAAiDamAfWokmQL+fQv5dAK4/mHOHBGAXctAFID2CaA4wU5BXghGA
7AfIcmz9ALsALMOVFIaQAlUAMBSGjnAb3/e6z0ogAL4avAvA7Qnm3Jl9ALz+C/md9GsCwKfBZgH4nPQC
dgFYhl0AugBQ+0t6RADXkXWEYReA2xsy5+K/KgDk+aoAQP5BAGgCVA/AdwJ2ATgdjlgAgmTGxH+5fkYf
AB1+awKAYYwCgOG09uMTp28DYES7ANYG0Av2PE2A2gdgnAJAP8Cvv+ozJyHfFIB9ItAxPLj32K2jB9YQ
gBSBfa4K87QAQH7IviYAdPThJioAkL8KAOQ/LABcd0n+XQCUN/S8i3+jAPAmON8ElOxJ/j4KMHoACsDY
EbgLwByOSADmAuL/XOv/seAZ9AQQgUMCIPkBNxSKAIYi+TEclucVgHbdPdx4YEALCEl5rggAPw7Chz8h
fwoA7j8CEELgNjMB+YgQ5UHAH9FZEYB8mUv7umIFeWkEgP/PEoCx9lcA6CDS7WfsWAEAVQCoRfgi8FoT
oF13DzcjrAkAea4HgABMHsAgAIB5AHwWjMlA9gPsArAerqQAKAJ4BKsCEFAA8AzO9gB2AbiZYRQA8vrC
AvAdjzt54MUhAi9djgTsArAMV0oAILzf+FMA2gJkA5v738mfAvDyJyToE6DmGAUAA9oF4NaEm+IBIADR
DJimBiCAUH6U9y4Ac7hSAGD5BSIwCgDuIpD85xeAdg97uDNhIgKQowBnCACeHOXoXIBdaOZwZQUA8i8E
4DVPXpAf1xFUAAadud4E8Lp7uPEQ5bgiAKudgM/t8WBWIDvAiIAOSFORQB4I3AXgCsoAIzpr3oAIQA5
bHRAALb7ALxuC/sw4I0EfyIslhsCAMEhOkN/KQAdfAxk9ADoB9CTo/y2BYDyChvDiK5Q2D2ALgDUeriL

6TiiAH1dAQbnCYDEX8MezhSUjgDT9wBiWwHAQ8ADmJObXc jx4M4 jAFepvK6sAOgBNND+KQDg1194T4Ie
ZAVgbwLc jrAuAPYB2ATIPoBoAlyPADAbcBeAFq68BwCx6SE+JAC0Hc8rALXWF3u4SGBEBUI+fLAPIAXg
vm0BYDag/QCUP3MBEctQAeC3AXYB2AVgEgDai3QagUkAXtUEAFQBSOOZPgnW76ETfseNgB99eTDL8DwC
kO3/IgCAXwhGAMAOANseQCP/LgB3bLh1AoB7iJso6SsUgDUPQAHgmlx7Dzc jUJIiwHYfAAKQH/+obwOG
COgBIABMB0YanBG45gGMApDvBOWcCkeGjGdc6Xy3rLXfk3s4VmcSRityQwAcBTgkANQaGA7rdAg2w2nf
C1QAnAfAdXgDEeNJs0722nIFaYa7EpkefFUB8nbsA3AegJ8Eg/AIwOgB0FGIAPzzv9VeDHIY0FoTgLIJ
1sf1Ib9WRmCZEYswbB51uCMEQPevhuV2axfSOUTsXNO3nwHLt/8iAnHYFQAKnCMgdPhVWKvj/kN2jAXQ
eaTx2IGE8TBZiHcB6u8C8Nty/IQVPzIp2G5xH1vEtW3jXF/HMv02zpvuRuDziLU0FwHnoHzGc9Z19pO3
/DIQAUcRwHoAEL0KAHMBUGD+xt0nb/vG2QMACn12BL7pKfkDo5MAhBfXBKDZ4S0uWzbtZyahwnSszpqZ
HX24owVg jmuFgQBYCG1fFAwuODVpbAHcRowIZf/X7/0HwdDU2hQ8ZBe26XENITlkx2gAX5RhSVwaTwgE
5GepAHBezv8f/slv7riF4Ofdqf35OTYE4P7v7QIQRP9pyB7Ep8YHbAP2VwHAo0MAKgt+2Q17qD8Rn15c
iMCqAHT7wugk/y4ANzlsCcAccP+jtkepI7Cr7W+eAcVF6STxQ9GpNTAcSMpXZChwftSTWhzSs7RGz97/
EAA6AMEvfvDxWcuwDvnZNwoArigG1N5FACG4EXCvxwiff8RaWrGWfg2mpQwRXAQA95+ywUNDpAUkp8ff
bYcAHQa0IxARqCKOF8d1EBO9uWyqYVFPy9hb9wC6PWB1k2uXI9wRAkA4LActMFqBtGBB5JhxqPRM/ndk
wVJrZCFHYQMIz0QeoABUIRDUEMK4ms4aBCAugCYG/Qy5DPdyDaat4Jil+GMD+bwWP4J05wECW9ep+SE/
YLYmIgDwBgDrNZ42v8Rn3XgFnmMQFM5vUwC7QQhofuBFphBkhROVDB4AdobNlc3y6MMDIwBjGAWAgkj0
QkgQHeTH3aezCDWnQKmdJ7c/4Bdhge1/v/R7HtBx6PHAENbruvn3mucFngmiolDdbEic6wGkGbdvBBL2
IuA4rg3BzwsFAA+gegY26/TyEBEEXtRCOzXUQTSyNIWw/aodUAPdBaettPjCkcmABFhtLuC/K2jqNX6
FCQFSk3MNwD5CfBPvu/eBK//5khAgHX2ffr3n33y4Ie+aIjP18AxZ4Fz07pQUcVE4ZgE4x81EfiN137e
gmxAEvuWomDfWtyINXJcBjBkLX6EzaezkMRkvdTOYoxzGyggbisOEN1zm4Z4RnAgvn0Fgs5BhIDjyB/y
HBEA/OIzQ43YefakCEj4/J//hjYAdnrMInBHCsB6pnYBAkUwKz+anzF5x4k/8MZGxEPwCMzUXwnKONGV
/CMq+cUnfuup03JNAEBig5H4TETJyShBfj4/7ktIGDGGCTDgSoCbhdom5hqJ3laeMMA7Pca5fkFAzLGN
XuPdV7dBbd9vgf18LxBQ21cBYP1nv/UJcbyA612Ad37Xk7JZQNORvgfsyuaAHX/wJL0CRaHb6S4ANynU
DB0zdmqHJR7KwqGzD/Lr81PzQ/xKZMGk4f2yL2C9vhX4R7/WiP0n739qpmdbSL/CiUTOkeB81ugQm6FF
Zw7a59BGHP48EZ6OKA jvyAM1E8he7WEpMg2vvfalPd9TnPs7OMY0U9pyTo9x+1D8dHw/dz3nrYD3MN6H
cWvHsJ9hQH4ejCXpyFvdfb0GREJhIALipgjq14AIUJnYHGDIfttjVCarf5sFlyQciQBQ+8/kZ9iGDj9r
fsnPD35a61fir0FyQlS2SSvhr5ILBQRI9EBwJzeC985CxpUBnYgMPzERhZoeg6TXeqqdutECft++rt8o
nCzDcm2fOCuedc7DfBmPbcfh63oFcTk2HzIUzTWHLtx6hz3tWnB30eP/dV7Euqr+YzQUh72zdDXwqgO
+W/eUw4IQZaloknaOiJApUJTADvLYUKsMPLrKx2efzhYASg jftb+1NIFBAFK/kBRK6EXRXXTZfgkhcD
gbzU3rjph4gP6cVI/hSAqOH5tqCgtvDI5Cf2kblXJoUA5RQIyoRloDhJzC+puGV2ZqmYmlfvMn/4JgJ
z20w/e1Avvj1f8QcdzLm5/z2Ek4qdUhmf0q1Ce2YNnTBKScEWM8gRSVEIL8fkCUDWwkJ+BsT0QAE8uK
J+3wcoU7XgAargXaVfQh++jxz0J6TZvWS60P+Xh1Ia3Dd6MAALYhMfsd8kNECOM5VgGQ+JIf0oMF6eMc
QNjB2wOJT6cWbVQMDXeVeetsW8SiSDCWvwWIFMS6oz4tTQVW+e5UYz3sAb2QX5qf8kPkSlPypKytt8G
WPZ0vFIukwGEBKvKiTLdphgixL5ytiDvC0TF04YFL1UL4M4RgJH0Lc54lHfu+LPXnx7/rP2jYG3zo/gQ
VgFwmK52xEFma+ysocPlak/J4+v/GulvYSX9KObnx8TCTeTisD0cOFuUvrbRgUQn1pe99p4UT2BNXgO
a3nPV+H+ug6mGrxjJk9pjWfpees9un4WxnMewnhMvs9cfmr+KgJVAMhnanA6gpn6Ddn16MayRyTwFPAA
3vB17bqItf4AniUja9gZ8wNoBmB/cIUowcsS7ggBGMkPlvHMzmoCQO2PKuP++5KP5EftKWzICikpZegu
FrV0B+3Eua3YvhnoczeB80E4fn0HhSAPFcIAB181ES29amZss0ZeMs3PG4CNRjLRudXNA1qmjVkr1wh
wXmOuV0Y74Vtnz03qdX7/bOd99/3G7eFPFcg2/7hOSkWkJ+8Y/QAQaeGz3Z/F2x2F2r4ZbIAMakTnvAgA
IkJZIdqIgh0Bza9wWFAvoA8CXIpwJALAdiAKgJ5ZXw+1408BoH1PzU3BownGcWKH2ADkpAcesK92FtmT
b/+AwKioWbbghB68Ca5rzzPDSxO+/anbOGv/jpO3/w+NsjJeIKyQFwFwaI8yxROzB3+zOnd4Z1RTqT1
2Nf9N/fkeWie4QVYdgoA07XtDPSFtMsS7ggBOBQmAdftU/n+Owv/Pj3ZButDvvrRxp2h4wIAAVpLYx7
KNg2DjHAYDgGAbDmmGr9IL7kZ0ov1xtJdf8nldCJTazCs1DX3R7TPBogTlneKfcxgvuqQpBiEARmyfg+
I1rFnfK1CSbYtsxrEwAPAA+EfgGEgnlcTw9gGg3I/qjLIwFHJwB6AFUAID/uP20+anCn2CIAFGgtfIER
UMikQSxsLqR7GDU+52E2IT8tDmhulLn7XH/EmtHalrzq+JkXfwlibd8h1GPMZ2d8sk7+KggSkdPq5dUy
Vxgocz01mmcKAM0KRwTWBIAO6OyLukQ+wPEIAOtdAJz2W6f70vmHANB+VwAgNgVJgdoUALiGjvsK0uI2
01bkeM79ifd+/ckfvvfb0wgA18Qgf/Lbvmovr/+mL5ywtl3T1e1HE2v3d9a+ix5j3NoxZ8XXfygAZeAb
g5QJQgDxGf9HBChvi00ITC1zkGX8mienfUBwZg3i/qcAPK/1A9CUqAJgE2AXgEchKAD82QcwCoCz96oA
4LZTyAtEDZ8ufeyvbXdrA4yDfXgWnJsafyQ9Bv13//IXnPzAs/98LgXbFXVfxaF9Ow5DMcg5+1EuCjPC
gAgAmgFJ3ChXyxdY7jTxiKfMITkCYJ9CFQC8iLumQBhh2CMicDnCEXkA6wJAjYDDgE7wyU7ALgC22xks
2kIVALwAyK/Lj2FhdJD2b9/7hMT/9Bf/3LnhMeC7/+LjT77tqfdkPOs1fjyugv0j1tJdFF5/3Dzu3F5L
5z6fa4xf06bGH9pn3Ivubc/8nf/F3VM85UG5IAQFFfugOWBTAGJb7tiIze06dkF502/g8Gx2JsaSjgDN
BLzGogqwC8CjFJoA9GnAgwJA7Y8AMH6f47xdABQBSK0RCOLZT9sfY6BdyFIBwLX8ib/+pDS4b/lP7zn5
5s9tYB2k0f9nYZQB1uu2wGj/9hd8RiIN+FR8W6/H3A5wr/W+2PaevJ9xWxg/7vOc3/00IGksa35sHcO6
+8ZzQfoF+fs+y4FywRvQE2BJByFEhtju3nbOUqaCckfo8RQQC/OAFAGGGBEA+4gQAM7LpLNZAIL8dzBf

LhqORgDyyz8bAlBrf8buAeSmds/Ove40AuOcomtbEZcPEUij6bX/3/vqx2WtI+kBRghxMXSMSxl2i9PI
BWNenKYff+89J9/7NEhz+riGrfhdQNeY4xrrJlUda8/Xb9ouf9phT2y/5r+5KGA+IB/X4hs8Y4uZ7m/Nh
jqvx4z62EQCvBchvyu5/8nduaxeAGUGYREAgGsPyfXsAOvEIRCQH+L79mDOMQgxLo4egDY2cID6M7o
ZQl3vAC0gMul7QHU2t8ZeykA0eZngg4uPqQHGoKdQ0z/hPgKwHv+4dNTBCh8BICaBsOD+BIJY8Q4BUb/
onuXxnWkQfgtcCx48Rdz3kYiz2lCja/HAM/jvroc40A9xnOYthLuvPB4Y4TLH1GPW9q9heUxrRika9AvQ
P4MtAGp9iM3QHODYjAJB5kp89kF29jsk7FuF1v5rApCdgLsAPBph/kDjmgDQ+Uftv5i1lZ0Aavo66w/y
W+Nb0IB2Yw4P9bF8Cp8mg03/ubZvRin5T5F7JL6oaQZiPI1+JBH7KsHq/rXzHdoH6j7TjueXdGdhPA54
zhff03saa+D47/+Sx6bIgbU0IxBhmgB4ZggAZyB8JyTsAa/NeQIA8kN6a/kKXP0EQ8SxJA4hYCIQ8wB2
AbijwtKcGnVp+H2d6qmrX+foU+tLfk1PzTGCwseYEAGILvk17lXyg5H4FWPaDok0n3sdlXTGVXKIQ/vA
u09Q2uuBxB8FgKbEi78oPKiId/37vuSek+//0s/IJfELIAYRpp4DDyz7G0KUFQDnBrBEBKqIIWQj+REG
iE5t7xwR4hECRgCwkTUBaH0Al4j9ES6FAPCiB7U/5Le2r218ie+3/23nWeM7ucfJJRiRoJ35smc8diIr
RlgJdspon2pFcCptgHjOVYXiPIRUdC4ErxukSgKW+7Ctf7NRz/13vviuIPpdJy/90nsaYp3tiu//srsn
QcA7AKRlm3Pw7JSDIWEIAOWkFwDoDwCUJ2KO+w/pK+rkJJaKASnWUglwXFVAPZ5ALc5zEOAhMj4DQFg
7J/ap938XtPnW13f94Tp+3IUtOSnwCW/nX4YEIaEQWlyrisAE2GCTElCt4uRs8/180KCLshdzrlAr0EP
AbKIU/s5vpwDut50jAIwxv/A0+O+n9HWIT95D9gmr8gf4hUAY4q+AIft2DFIGSMCEBUcV2gT1PrGYSsK
wGoTYBeA2xcUgMSGAPA2IO1/an/fxK0mtx0HVHkLgJfLmMpiWfUdawZF5M1NRICkLVsrF8Pyc9CJcmI
JMdngrUss9fXtm8GqhBNbn4VoFgu9nWXH3D8S5/xmEkszcF2IQiWj2UELDcEoY400LFL30/aQyd72kes
KwCCePuIFACHAXcBuI1hQf68v20BoPef2p+aH/JT0LWAdfcoeNr9WFOHYTh+jmFgOPQuP+cz5l7mluZ/
/GSY1DwadCWu6yOmWjewth9IhEp2IFkreV/2ZeuQJEmYQhQORr1K4TsUo64tdAmrV4wHnG7cW5C0xD
vvGMU77Ecy8EILZFFfZfFkVZKAAT8aOcGKWh3ABCgKAj7NUToOylC5auT8BuogLZEoBpHsAuALc+NNK3
UD8IsiYAtP+p/W3jKwCCOEDhbpGfjqU62Yfe5rE9nkIQ0GgLECAuyfulc22a66Juu9+4ur2CyQvwej2u
wmbESKR2bLtfwHa9b+C5G0nbtRrBi7cQ9ynRl2nmYwDbWyIwHusyz9HzgW2JLlpartOfo/cNCATAYUJ1
ohDNAcqZiV14AXh+2EC1jQpIrwAgFuudgPRHXZ5w6QRA95923Nj+Xw3U9ccwMBBrfOyGoT5q/GpCEH4S
gKyZlusp0Jgx+Lqu8bs9gnjTJSBEJ0Uj8UxclmutmDVjFwBrVmtZj20EarUvS67VrtdqZcnV0gmPbZiP
mbdHAUM0XvoMnxWP4DRe9uVtSVrANvIBlTtmkX0B3iPPoShbVpQdTtJLEy+gCsA0/yNsQlupdsI++o3A
mgDYBNgF4DaELQGgILYEIAS3YKEKBYARgVEAqClw+zGYBf17jQmZlgIwG/tE/hIHKTD+ul1JsyUCYQB
j51qwyIAub/fo5C09Rzz9vLcl1u7TPs4xPHS9r2nrOda2FYBDkPBuv/zL7zkVN6Pny8o9KcjcswJBGVKW
iAD9A6MAYA+V+CA/39bthvUtD2AXgNsYLiIAZAHIST69IFVxodozNKgAcLwCgLHG9isAlpJzCACNWS6
zsR1Y1/blgSVFIlSSzaStBq/kl9DPy9GQsz7rIub6FEH7GvXgVQtzXxm25clbBcAt8Hpe/NYzzND4q6R
2TTkgWnWST8Dr8D7F95bvUcxCGd1THkjADkL9KvtZGi0EUFFQkwx8CjHC4qAHYCrhWqsj8KAO1/BED3
3xoQolrjSma3s9Mt1o17WRg0cZJbbG1XIXajcdf4un0YECbuqQByQSDOMxIFrJ/nNKihWc732IhbSS+R
x+USLQ/Ii3qc5xHmGetJ/kjnc5ih9f54lrzHKB/6bBBzyhQBcH4AfT8IgpNCIDpIm+mQ/LsA3AHhIgL
KAAf8qgFamFmXBCZgYqABiGHYAYzCQAufvZuz4bZTN6jBIRLIyXfd2IK/HmdC2edY26xZ2G6dv1PLZu
tzgJw7XrNeu2ZK1EUdwdqim90ea3l+buw18trfoX7lvcFWlwQkbQcQ/5N8TPa/SzPD1q+nI4HejDuB55f
vCLiAPE8F26AAFgOJAJQasCEF5jtrPJD3YBeJTDeTsBmQg0zQOgGWDhBnIKcC9kwPRgXgmtHgACQG1B
DzLGY3u0Gm7bDrJgxB2L7SAHy/G45fHbwNBZHkqngMxpOmFiPdEJn0ITImYbOUcuWAI6Btlm2fdle6Kf
2/uYmdeybJumpQvs/dftnl80+SIOsF5RiZqI9D/IcX2bc7HkXK/48nZMxkU64vIZY9vr/wDkds47IP4P
Bv5eHPvKr2ieDmWJqFcBoAnAS2F4gZNNsAz4Gw4scyZpLBkxUADqMCBvpt6JfLnecCQCcPh7AIuZgKVQ
p+0uAHgAowDYB5ACQO2/QcKJaCMg/4oASBSNeeu8WzA9y+aCu49znm4u6NbzDPaQ23EJ7McANA662PV8
wuvU5/GZKhEl08tKTome8X0frJXsgHWIneSO9UkQAp13sa0AcOyrvqrhlc98TG6zz0lntXwYmwCUNwKw
eDcEG+lKX2wXAXAQ8C4Aj0I47QFsfw9AAchmQCG/UAS2BGDyAI8jWyzoc/Ea9AlXQPHafAVGLXQ6N2X
xAjDNq5dsxl7E4G2XdFc5l57B1ifOhpjCbFJA6Fe+ay7Tl7z1fcsQJx42TMDcR2u03dWzP2MVQC8fiX+
Dwb5JrKHC87S2phluusBa+mzoAgARWGKi+tw3ld+ZsxB3Dt47Ve25zpLAKa3Qwf7qPB3I+hM3gXgUQzn
FQD6API7gG9ov9RTC9MfhGAdEeA14bUmAABDZBKMB2NLcnbDxfBzm5qolEaVEFMawHF1W4In4tyV/EAY
YeBsey7iFQHi0rVP4rd9KQCxjfdYpMcdQkNsvnXvhzLXwDvxfDiDNIqCzyO47gjuzZo8SRpLkOQ1PmCt
bJyiMJG8bCsUFXnOfi7yjdQKQW534mfaSPPKiHv1M++eBIBJQWtNAARAG6m24Xa+Rdq/DL0mAPQBtI/T
7AJwy8NFPADfBsQLOPAafQK+HmxBIwB8DgWB4K0/BYAOIybrYDxpbGGAEAJpvN3AKyRnRY3P9TDOsnJc
VGrLFIAQiUkYMOpyDPFAcNE/HD/V0n00gW2Ohbz1V3Ehd37k4lufkDMi7//eJ6Y76yw4ZrtV8Bac78xz
7CgK2bzhObqw5f121xtie899UgtDQkJPrZZSmrWJ1GI9MaZRsKzrmvPegbrhIfoibLd0gf5E60MKU8F
gHJGAOj89SMx2ATQTLxnnwKwlgTYBeA2hosKQH4GLAiehrjErziPAOABMIykUV4PNFyQhIYgkCbW3Vfj
atq6grghMIhHhslQAWJEf9EGAmOwuLcY+QTzoxv1COLTK+o/iEINSU85ooCYIAR4FHgDKQBduLxXnqXV
wi3wisBrOhnbMzeC1v2SFUjyJHqch7gfinTtHDMmlx8B6E0A00znPtAEyG9Avqk1FfPzi32YTYxd3/YA
rgWaXV6GcCQCQcfG+QRgKsResK6j8hiAAkCtoADQ/udDEXiPhjliNuDT8dXghWSWIMaPcRh+3fa4Sji2
9S7wJCbI8wnrQXp38CoeVZ/2qzlr3tugABa07kOSEd6jssfvYlZ2AmGsCgEeASIDp4H9wERfZZKaGce
uITQwvgedb8/KASuR5jHIR/NR19BaZlmXEHBACA9AD4WmzYx28YaEABGkxAAjq8CEIzZBeB2hIt4AHwR

aPE9wLFAu9JTy3HMmgAwn74asUSs5J2NddxuBCat2/WYgwiDhdC6tGnEEZ/XDmIpAtT6lMK6+xCTaa3W
WknqIC1zHZz1OIKp0HVdsF3nwXOeFJPwIMgvvAo9AoRn6jgMIAitVm73PROWjrmGFgdJAXGt5pa8Le9a
XIuP58xlPWaGaVx3u44C1D4APD3Km76f/E2AtJHDAgDIXz0A5wHky0CXiPyESyMAfhF4WZAUDAMC4M99
HRKARg06kRGAdcKP29aALX1NI7KWGrYBhiuaAMwdapAfd1/yQ0LIaIcmz4NRQlhmOvKxE7wCv27r148q
+asI8Io0aUifx3xzzi0cEPD9ABOgvoI9Ab0ARoGbOe49nstbo/Ig4iVoHaRtxTxMceJ61fVuoHgACAPic
WxWAl1/NM6oeIkAY2KcNbQrAZfpp4AhHIgCHmwB+FZjCq16AhcoSMFpwSAAgPoaJAdvmZF2DbTXdvJ3p
Sht33DadZDeuGv4kCmG87ON4BCA7EEMAbotDPmt9SM/zQVRqbkmmQLCE2FmJR3omv9C2t0b3NxM4j8dD
eoUjhsDOR9OC12JsWgDOBSmqCKQX0D0Aczi8NT9aPkh4QfPg8DkaTM961lXkIfdCcw4BGD0ABaDZyCwA
Eh8vEhC3C8cJHM7tAfrfBfKz4Baw6xX0FXAMWBMAiCexqVEwctaJa4Y2k1+McY38S3jsjHVDhjzcA+1+
an3IVV1+iISRY6C0+yEqpKc/gCXblNS03yEr018Z/srpzx3+PkL9jJQ8CoQgRwRCCPy1HL6bz0dU6DNO
edg6CiEGlyN/uef6BOiorALQnnnpstdnrQIw7p+Pmb+W/Qjum7d7mXUB0AMYBYAKoNqIdiHxRRWA2gew
C8BtDKMHsPlR0CIANgMoRIgyFqwC4C//jJ2A1QNQAKpxVkj4tfgx7iz8SDdmrp+uf9T+tLUhP0aYNXjv
WIKMuPCVqJCRWjw9Heib016D/PlBzP514xGIYv3f234ZGWFREIir537bc9q5ERY8BsUDkeH+7BNABBQA
oLcz5oekdd+4H2T+Z3wTCo8Bntd9btuHQv5tNQEQAGyk2gR2AogHpeHotgQgRwG6XV6GcKk8AAvRgqzb
ZwmAfQAYkwIARuMUS0Eu42bjPW34LNeMHgFgiQBAJAhlzU/7m2fBSHHZce8hJz9pDSA/7jrpIDDkRAD8
oje/bkKHGNodBT3kfOMwBSYEIr+S1D0EPAKbFVyHX811xIFOQsTCnlaDIoAk5DoM5H85s+YR+Pzj3lh
XIufUePm8877bHrhQW15ANz3aBegxpnXuwA8yuGifQD8NNGoAiMQANIfEgCMVwGoHsBoyGvGzfYoAMlo
TwuA8RXUYtmxFoSizZ+lc9T6iBdtcdzxSn7SQH5ICzFJj9FS8/FcTnPmgyf/7WPuXoB4vp6DQJCe7+fz
8ou/okT7HxGA/FyLJduKACQhLSKFCdvgJfcvKEtP7bnzR/TrMHjSF/zyzwbz0t8y/PWjKIsRwHAVpg1
ig1s2Qnx7N8F4A4I5/EAUHUIzY+DVgFAxQHbtbAaxgHMJwFT7N2iYGJrrwjr1n8Y6xrHUCMd9OZwWxEd0
INT94YpDtHT7e83/lvvuPnn919wzkr8DpebXLYfEzNyzxof4lfx1m2X1CBADrosIcC68CfoTbA5UEaA5
UEWAe+HeubZE9FnpwAQ8I/vq8+P5v04vPS6RpI7tPKZsKxxJ/L6/gnivBVg/jwCs2Qr7GFHieugX2QXg
UQynPYD2NiAF8a63/fBCAGwGUIAUJsta0GJLAJgJyAswGLAKQBgoRio0Wpegkt99ns04ERqqIK2uPySC
hNSq+eJStEUhuDU/7jheZEmHHTU+RLWtT22+VeNL+q19CAEigDdAk4DzUgviYSgCgFEC+wRyVCGE4P4Q
K+47OwXDC8ANn56xk3bx/J3YLFMgOuEPkzfUPtbH8wTITwWg9gEoABC82os2AqhE2I8N7QLwKIfrEQAK
z8JknQLVM6gCgDFUAcBYmAY81TwBjHUUGYqR5JjfrImAabw029nxFzUnrj8EohZPl55pq5D7RY+famCI
/9bntfY4nXDsB7XmHwku6ddQ0wFEgPzgfIgK5+Y9+TpMyMiAIuCQJPCLRWCBnoYcIjDlRSFposS7DXF/
/KtbOvKJeLYB61vkn9DPQb7WJsCWAGgXgriKQwJwLeyQdwGqfR5zuBQCQBNAaAgYCxqh+ORvP3vVA+AV
2snAAhjdwvBiiXGuxY3kF6fOUaBAIDLu/Nb+EI/OuOyJfmkb6sPlhvwIAURk5p5tdVx2DJ+aHWKPPBbn
EQDiEBHEBBF4T3+DjvuA8EwUcq4ATZI6MoDXQFOAYxGB6V159v78SehhKf1JM+VVGhL/+DwzCBgn3Hap
B3ARARjtrRRwUgLBLRgJ3AbjFYRSARw70AUDsUQRYFxb2lgDYBwApMAKF8XZD1diqIbpNmum4vn0IEF9g
9NT8HA+BIBK91RALkmWNy2/XP+8x07i8Pf646Bg5Pfpr5B8JD4wf0wr2IQKcFzHCxWfo0ZEBwH2BbAoE
UWgzI1qIBkKwnYFBRJ7VvAKZryV/2AZjXrIN+envcJ/nIF5hqHF5rhBCypCyZFh3FAA9xWonazjCBGjv
Au4CcIvDaQE4PQ8gBSAK9RO/9dQUAEVgC2cJgEapqQmRk7XbaAh53Y34smwS7pTiLQMXYEA104YHMT0
GjVcb8hGbeuv2rJOxxxNA9JQ++M1QNglIo/Yihcei5jQn8D5uR/FBsJDFH9xCeAd4K0gWqThfhABZjHy
boQBRARzxJXkaxjTg61zKQDk6yga2IkCAMH1AGrbv3oEuwDcAeHcAhAewHkEgH0KwNgHgADYCSg0rLW4
jB9qqzUBmOJER8kwVGSriIIASP5E1P707cftptZHANL9j5oY8oPa8sd5KyTlGsnHuLpPAeD8djTS5Hda
ch2nAAFY6wuAhHw1mecmf0TNwXo/7nN/JbvptgSAOPKVa5/VBBg7AiU/ODwMuAvAoxAODwMiAKMIVDEw
/pPvu3chAP4gqJ2AGGS1qhG1tpLM1Rgr2RWAHRD0YzRU3GQ+w4UA0K1HLyRlBzUb2hbyU+OyjiAwDg/5
az/jcq/1/FdyjxjTuF2BADCuV+gMDAEY3zZEpLhXvrfAcWeCSJAU4YaGPDcPC8gr4RxnZ5189e8XUSv
TMe66c4SAMg+kr96Aiyz/yUEgOdfFYDLwf0MRyKAYycgXK4CsIas/QcBWBsFwiqEVAhnlch8Vm6YxH
BNj2/MTR+w/RqDGPArUAYKWLzTAcnX4sdbkhJOKxcDrdRg+gEn4NNV09TiAA9CtWfpoBXCfIK6dCCGY
ENV0Efi2JeJUBSD7VSJfK0mBebQWT/6YVzV+C+XnlIQ1+cp1KdOz5gEA3P0qAmcJAozfBeAWhqaw7X7m
dbA9CqAAbIkA8Un+FQ+Al0dpAlQBwKg0QjAaXEVNByC6Rst2FQDS13iGACEwAgCpua8qABCfWhbghNf
BQCiISA3IgAAwo067ZwA3GCuRc1Ou5ief0E/BXF4AAgAZEIAIA4dgdTEeDnmjXlMHo8w3yrW9rOenlWU
lyCeJdeD/JQpAsCUaDooEVcFwIlhYheAO+iJ1gWAEF5A9wAeeONLsmCqAFQRWMMoANQMCEd+oOSKAACN
threlno18hqfhtprf/eZtvb+VwGgZoXwtZYfXEE40owCsEbsLdR0k3j3jhWKwJgDU9I1XNTAawXYVALyA
+/vEIN8NqE2fmj3LeWhmlqeVCJLxBursVv09ApyrPg3ekBIABr5Bd6AixpytAk2wXgUqjXKwB/8v5G
8jXygzMF4EvmHuvR8Eas7dtKT5y1v/u5jp2FEMUOwNoHAKxhRmJg3DUsqTHQG0CjMteQt0v+SGMpDGe
9whsAkB+2vi8j+CnxPy8GID8fm8BKE4IHF70OBpgvoyo+03DcWntL+ElPROPkd/b/orYRQVA8CwKAF8V
3gXgNoZtAYj1a5+e+gDWBEbi+hpHmgc/9Eunn/79JgCMtyMA1BjZB3BAAKpx1n1bM01aeuK4DnMARp0h
GcSn9mQp6bOm7bWuIgdXbGtjoGuJABJ8De5fiZ9x7kMAOD9kzleRmZkYRB+BMLCENNagCgDP94ogp1OD
a96M+VL32Z43nWRn2nXONITw4bExcipPbc8yCgB2ggBwb5B9fB14FACeadMDiIpotsnjD0c1AIf6ACA3
qKsvcd8CwDEKJg+Qwqiq0VWDrNuu3Cfx6ylNR4BgBQqBCOjpx1C0/be+PyKse513YZsGCnp8RwQAIz+
vAIADpHf/eQL56f2lyTzp7RExHVx4DsMcGDH+JYgw5wAAte8AeBL6CGNNT21POdgtiaEB3b04b1V+Czn

FYD6/Qj2pQgXATjLATAV+A7iy42GIxEA0JoAFATDgNUdKNewjRgGA/AoAw2jVA1AAqH0xuvBmoGP+3D/qwBQY0LoNL5e0+pis844u+63NS3pnXjDVOY1AVhbH8k/ioD7IR8kgAwQA9Qv6HCvFZArO9mCaNyB/QD582PhXXG+Nz/nsVM7nnyoJE9EvjBqUN16QBnxuw3W8iMg/tpzVAEA3CP3Xr8G5PNU8Mz0t6x7AA91e7wc4UgEYDkMSB8Aylw9gOoFVIy1P6AfaAGgjUsnGkaEa6lxjoSt265vwWPWjnW7CoAf5IA0GcheQJj+qmmpeRuIw2ARAEiHcFDRk0FrsQXkrsS35qzxnE+XHiG/iCE4+Xc30h87hewnu3sAOXCM5G/+TuE984dd+Sv65KdvKepQC1vW94aHvCbDcB7FfyiM3Cbel8TgHy3Iu6Je+R5RgEYReCQADAlPawy7fMyhKMUGIUH8072Hsa45FdFYEsA8AAUAAyxlk7AdQlc941kH4k+xtDjJwEIEmBkCoBu6mickgXgoNTGigUioBewJgJlFUsA6jqE5XzU4Jwf4gsFgPhJlMI7sc+CdZ4DwvFMPBtiQv7yzUCel5qe/g9q+OrOV1jbs2zWR7ID43h2tlnyHKMAYCkKAPmJAHDvLs1n8pblLgCPYtgSAKcCUxClExC3s07tZH2a9DMIwZoA8Bt7GBi1EAYqaVlWAZDENW7c9rizjkUAIIBNgCoAAEMFtklHEA8heXbygE5E7l0RkPjnFQAB+SB/HZUYa346KSG6HZMMUZppCdLgNZC3PBP5yzk5N79UDNlrb329tid+e57256b93iNEp2OyioACMG6PAoAgkb+KqOIlRo+A59swgAd3AbiV4SwBGECBIAHEhxTWjmwzpluP4DweQJ0HUEksacdyv63K9FrPOu2f4mjFuS3//QAUbeepQoApONZAOsjFADSOyRIh2D1BCQ9S6EIAEjDswNqf17iWfWnb4FzQnYIQqceY/6AoUlI73Rgi09bgngNHKcA0ASAwPVa1viSfnDPgkvHFQoV+9YEoDYNFACutyUA2Wm5IgBCAeClqy0B4JsaLyUcjqAE+1f7AJIEQQaIQUHVgiQOjwAh0BtAAOwEzDbqFz8mjQe3E2JCSskMWSu5z8JI/ioAxrHNDZgGrAcwCkCrqZbtUuKshRUIgXEzkkB/AkavCEj2LRHguf1VZJ6b/KD2RwCydmC+Qqnh+QqRZB/B+wo0RyAbz6NX4vUhmQLA9ViOpJXcVQjYVgRYjmk833iuLQFYI3/ra2G9DWVm3BkC8Eja4+UID5wArIfTALA9gCoADlXziUbcomnQJwJZQ2FUGAZGBBFoCtA+Z0m/AGjby+/dAT78SVyL7xNWouzVTlEhLsjAkvf/OR+/+KMHYBMgjTRq3Rzy67PtHP6rcAIONbO1M643IqAnIALEJT9gPyT6/i/9jHT7OQ7y0/OP+5v47vba7xrhhd8G4F0F7sH2v2JUBQByMvEqiV1ISxmI3NfJzzbrwjsUles3usB4Jq4JpncBBgHQpM7GUQhBABvhjyhoqk/DEIfAHyXkiLU9WMLRyQARnwrQ8AAagdOFmIXQBmZW/j56SvE4EwLA0U47R9SpvV4aepc+re9tqWv9XP8FaFP9kt6NnOn/mOdQSFqBg0gRka41wQlXtgVjqlJu4zta5v/o0kWwNvC7KEgNTciADCbHEk+1jzC56deyA9Bk8NzvUhpTW67/2vXVOYhhpTD6AKACsX/JXk3J8CQL67L9GJ7XrdzrQlju3cF+vDUxETno0lz3VKAojDCntgSHWN/Gk7gwCkB/CR30sBCLrvAnd7w3IUYESAILqFyrKCYTtsTx7joNwVQAwmjTS3j7FmDQ2t9mHMAAI7LrbtKN5vReRqELHuJhpNVYEBiPFxdlYLoAbOj3D2b+8BtkcYz/cWEJm2uech3Ny/i0BgByAHnq9EISDglwBkOgKQsJEJLhi8BJ8SYk8hWwIGvdgc8T8qyLADuzp354wEF40ks/71rY530EB6H0azqlQBBYI8uNR8ZwLAegeQFpj0GUXgNsam4CIAAsf00ApkK0RQCopUYBkKAYlrXLIbfdy74G0hMnuelTSATH040IdQmfBt/PLSkGCULE24nMWcelp9ZPcseSdveIWhuL/E5gCIAeAOfmMYAi4HmMKe05eT4FgBoPD6K2+SU4ZGA5wo5AYO3PMYAAPBf5wnNCRp/XPGx5EQQocqfQdpJzb8C0xrMtJH7dtqx4Lp+bexgFAIijAM6wVARsVu0CcEeGswXA9v+aqhNHYeMCjgKAgVUBgNQansC4BptAXZ/jz1EAbi8FoJ2fdYmIoSIAfKomQ4V8kr/WthM6Id0mPdsKAG16zm9Nrwj4nFyT51rr/KMmh9CV6MJXk0ElP8cgXOqrAsB3Fji/xBf1+ec8m70CSV0FwHyu8aSraVk33SEBoIkFye1LUQCEInBYABir2psAtzkc7gPInvDu/qvsFqaFOwra2FGFIW0JANSjRgGQ7ECVw0lmsHNNBbgw14QkGioCgLHSAVdrYYxYoucawIgu4JQNYLAsDQnwLgMF81uyDefVUAchJQkCvvPwQAj8Zrm6/kR8sh86nlNaJ5aJqWf0vhHfOwpuU/1+O6XL82Abg/O0xrh6r2AhQB8mAXGdsqHJ4IXaJAmgBI/kkAommwJgBZUwUpMKokb7SNQRpnX1/EpXG2fbWmH/sFpnN1A23HzTUE1+TarOeHK4I8GCvUNcJAxlqKwEB6CTnFRVq2k7xB4toEkIhVACAJ+5kDUQWA4xWAEX4Ex028flzX2p8aNu+/CwDXAZK/kXbOT5pG5mfG9/whL0jrdju2pTFdO9coDvPwos99HgGoQpDL2E8e4H0pADkK8JHfO3noP34yyp5wkmYXgNsQyNgGJl6szwOA0AgA7j8CkC5cFwEV3cKlDXiWAGBkEtelcRWV7NkB2Nv9x1XjrqgGjNEqAJBGyWV0yCkAEVwQICfGy2/6MemoCoAi4DYkobasAkAfAIbP0CPn4jwAgkv+SRwiTkzkj3umGYOQmfzmdWuNzbObPzXfzBdhXpmpibJ/TjPno9ADAacFwGHWYahVAeAdizUPoH2gdikAxxyORABoAgTOEABr+qkwuwi4veYB4AZjqBhOM6hWk9tz77oGWw24xjFCOBqtHgyWxjsbLtdfFBAi7gdjtSnAiIACYM27BQWAY/ghEQguCUdwTe6Pj4msCQBLUcnvdi6d/x/EIj+p+SEBQsazcA3AM1bsjv1w84s0xq3tG2E+Wm4Aj0MB4N1WBaATn/s/JQL9+REABGRdAI671q/hqAWAgOXQJADY2TcVZEUVbIXAAbATkBrImrEZVREAUqPndhBcaJxua9gaoeep2+tG+/jmtOYAUBtjCHYGco80BfACdLW3ICkVADwARIuafiQ/gJg8H3MA6C8YBUASuE3eUS2IQ2Cp/blfQO3vdGSvMw1AkJi8mPKOPO0wruaZ+2ocIN2cvy0P5+1ZALj+6AFIfu59FDafG+D5bArAtU+nLe4CcBvCLADc39kCMNX+vZBdd5shHgy2CgCGSdsYg8FYk9QYXxEADEuys/g1zCQ/Hdfil+TXeLkHamzIM/UFBLhXx+Yl+ho0YAUAQjmfwd53is8QHTyfvQEOeTYJaIBzi7xW5DWdquRl7fyD9BIQYbNnN3krZ5nk73krRgGo+407BPJwPv7xeW3uAwLj6SkAPEclf80/85DlWQKQXsAdypmLhqMSAD8JngoAQ4BT7d+NNwWg3EWEhEAoAD4TT2MBkNi7J5JPAsjZT2gwbFOOrdNSOwx3n3j8vEpKPZeQ5CpLyAIPsfAvU8C0J+lWueEnBCZJgDnkVAJEACmAI8CwHnoM/FcmZ8de/n7Nv0pzKjkHvFa/MS6tX/rZ2jk9F15bsl+COTvYrsLaest72akgHYBAKxzBz7xLABgF5GHw4KqEOwewB0SfGLA8uGHNwTgKZMAWIggiUGhd6LYYVUFABJgsBguBgWxIUeKAOQPV3ra7kLANmDdwi2P6yBeA5206YKRB128CQUga+veF4AI2BSAZJBtQXbRyQhps7OzCwCkPtQHwLW4D97RVwBoanBOz4VYZps5anr7Uoh3Sb5/4r1fP7n+3Df3L/kkphhFwOUaJuENr00H7ZyzEGQ+R3pEJ58x7mUUAPivYw/fsRECl2cJQA5H7wJwe8JCANjeFIDuAYQBU4AWJNtToQdss44CAPkVAN4FwJCCzts+adXIz1LyZlyIAwKgSOBWKwzEJTr5M57tODfXEIQAFQGKAKSCXIBn5J4hJc9Ta+JKWOJpAigAkJHnYlnJz7V4d8FfJEYAGGokvzhPhaRXCbxJqBU/ROG+uRZt

cCDZef5KVAWhxptPY75NQAzKej2fsLw495oAIGbYBHbAqMmaALi+JQA5DLh7ALc2VMKPYUsA6N3HOCX/
GihcCEO79SwBAHwfgI+FSHRFAZIV4ruxVmGo+yCa8WnIHXoLigCEgThJ2DDesSka4bhv+gSaxJf8dnLZ
BKBpgwhwLZaQPmvmECw6HP1CTxUAjuc8kDyxMqXaNj8Ye/25b9r8tvt9RkAeWxuTzzWevDH/tqAA1HXO
VeMRazwAhU8BoKyzL6VPdZ4EoAPCpwD0bfZXAXj//X9/EoDmAcz2echejyEcjQBkzLWHziUAKv1CAKJg
UwDCsNcEAKObiI3A7e0+Pz2db+8FmVlqkJKXQGF91jeysC0XA9TyGuG60idhGBFPnNguATQFIJuEABEyy
x/NSM4MkaogAtRxGjvFC7On15CA63+gHxAuEgpeOmHmZM/nb8Jb9zK1inmWwT6j5JT+1LveNwLsAv5C/
5ld5duLqeoXxFWftn8opylAvZxSAJHO8p/MqJDzx7gOjAFQPYBeAWxyuVwAgd7p4vRRCrCfiouLjnEYBG
/EbiUzV4GPG0jUEHWG9xM/ldNw3racDUEj2uHds+k5XbcX3IRC3GPWHYVQT+8L3fnuluRACyWitDVMnK
M2LgVCDk+wQYc4UvFTndmPTp/sfxzKikScVSSH7b/JKfvEOsIBv3DMbaPRHP5rPnMuC+s7ZrfF1P9PNx
fvJXb2pLALAJnlUBUASqABDH7MoqAP/yV35uF4DbFbYFIP42RgEUAMhuoVYQ55z1LQFotVcjdix/TO5W
w0xE7nHuz2YBrwIHmjfQ9tU09bhg/nYt0lcDpgblfriv6gnYHAAQmp+dobj+IpSTocwLgCgC3HvX3SdI
L/kVACZX8Y4F16g1P7/HYM0P+blHmxqi5k0KW39W84ml8ebJmC63u+BunQsYz5L8swkwCgCekeSH4L7H
UIXAdZpDvI05CkDrBGROymkBOFYROBOBIO03BADjxZAPRAP4jfwPAJEG93LNADAcAhRq/5HoNwo7BzVk
zq0R/8DTIy7a5AgBBgy4D0SgdqgxvZahNgilJ6A3ADkhqP8Rq2Iwo7nws2vf3PsKSS/xeceCT6dxDa71
O2/71kwb6f675W089940nA+ozIoJmZ33uts6XkTq5u3i4XwHwGJZVADzO/TbXzhKANFhVBKC+TwGw1/QA
igAs+wCYlt7ss5J/3Wbv/HCPBMACHGFh4wIuBOC75g9pVgFoNX+g1+YjML40wLJfgwQYsCDN1D70OZ6H
GtF9GHMSKe6higAGTdma4ikC1sh6A3oEAAJvYe27grx0DeGB56rnX/MyYw+eLTP04xlMEVgKqHkz5o+o
RjBmUzkeFICWtszns2P2vAKgCNTXq9nGXrY8gBwFyPdSwqjkX7fZOz8ciQC0qcDnEQAKck0AaOtWAWC5
FIAG5SgAHRAUwtbtSlqW7NdINURgMR7vsYqALjHrfCq8tQfyID4GDokUwQwTIWAmr16BHofCoLEHqFY
sE5aj/VcAOIzHCnx6d2vbX47+6oA5L0/o4tcz4PzgDxL8kd+1KV53PK5Eb56Ah5PetJQjqMAAAUackN6
7ASbGAUGReHMJkALlfynbfY4wlELAMqeZO4CQOFRkIkuBhb0JAA/Fu5vtJu3BAB3XXKLU4Tvtfq03dNg
ig5L+ERPdzFGcD3R4jhX005Dp5bl/rhPAAKhpH0DiACotbZQDHDnhXfg7RjOhbj8zIu+NpseXM95BOZV
7ewDktNn5xnMN/Oi5UHzeFIgAkn2QK5HGrcFx5mn6VlEXTy8T6GYxOK0B5CfOisCioM1Dda1Dz0A4rc8
gLTdtMkwdgG4yeF6BICx80zh7YU64jwCgMFUAdBANTTAuvE1zbrNTTa4/PV4MArAZMAdPxBxnpNzAc5L
7zdtbciHkSeiRkYIICligFfAdxLsLKyluTX8WmuTjvQQ3tqe83HeWuubRlwf0i3J2Z6d/PO5Ww0951fN
i7pdib4QgL6/Cqp56DlbfS4CkPn0tMdm96oHYBMA99+PmWoflJtIQAhFArAW17yggEPYBeAWxpuVADS
pdtQeAo2x7oPCMDLv/yek5dF7SwoyZqxtXVqAGCc22291W4u5zT1+KUAzPFdAOK47BTknF/R15muTUKy
xowIVmNQv/KjV0DNTf7YRBAQHdQ4a3rEw9rejd4Q45MvTt6xkw6yvYJ7nAjanjnvR4ugxJW8db1uj4RP
EQhwfCW/+VNBnlRwHoRSovIAAMN8uPYKGMqfBcCO40MCEFAKUWbyBeAmh4sIAOPRkAwAlAEwHYD6yo+
AsBwGL3hCgCTy6oANMMqhOwkHYkr+BS4YqFREiz7Z4KvHU+8YHse4/Y8iMASMJGmEwZycM/cu9AJEBD6
PLCmb5Del6M4v21tiFUJaj9FEjnvz07694/xF2kC/AMxHmu9pyzIJhfikTNT9YzvwMtrgkj6y/v98C1
IT8zHqsHwBRpbIL5D1sCAPFPsiIW2/MAdgG4peFGBKAq/CQEXQAO5FEA+Bz3qgBgpN3QqgBonE1E0sQ6
xjwLwDrhjRvjLwLQ9mHQ1eibscf5+xJwHgjCdF5qXchpZyHPomdwEXCM+TC93xDX8H7qPeElIUIESih
55L3nX0asYx7zTwK4Dwkh9W3gc9Y8yFzPPOn7WvP3tKwdfuwnfcU6RUA8qR6AFUAwEh+BYB5EocFYH0i
0C4ANylcrwBAbshOgY4CQByz4nDxzhKA7APA2Dow6EruheHFdhoyy2FfhcY9Hu++JHgsF5Ak7ockhWxs
KwDpese90/6lb4BnoSPMjjtrdWv2EXaa5XsCUdtL+Lw++cF23u98PdaJj3/G+5KMLJPw/Xk2wFqbdID
8yOfk3wZ1is4t3kH8ppxLuK5/+oB2AmYAhC2gB3wWwqJWfCAsB8qkTUBOM+7ALsA3KRwvQKAelOQVQD0
BixsOoG2mgAYzWzc1diasc5kXRqrxlyNOg17iJu2x2NiexSA8XhrfWvYJF2/n3Sz+31vAXFwyA533zgF
ryJj6vWdtNnkgPh5n0sBSMJ5P0UAM10XryaOowjkdciPPKY9z5QfsTT/3ea8swAgQC1fLI8UpBAAXuWB
oKx5SUppoH0PWFcAID/u/ygA9I/sAnCbQs3ImqF+EmxLACA1tTuFKShs11V8+gTyhaA3tLfeEABeiV14
AN1QwWx4ff2Z8zbGy099Ycy4tqTnJ7/53T+WbBNPGoDBT9t9f8axD2NPQ27nJw4YB1jPdnHfrqQC3If3
g2da5yaoXsIhkC7T9ueTXEKSNeJ1D4F9STzTtPtPxHa9vw15fsk9Y75G3APHin4M6zZJ6j247TnHPoD0
AF7bPrCK6/9P/+zJt97+DXcnsIms/cMu6ER2qvrZArB/EegWhesVAF5kUQDSxeuuXRUAfJ82YArA4AFA
fgWAGk/j06g1bH7jTxFgv+SW4Bdb3wB0v9tJ/jjebdImWA8gKBLfOO9jE5VYwz70lZxSuJ97Pwv1PAdx
ID15t3ZfKZDXKKeJ46b0tVz9GeB7GwrPckEeiAhZtmUieNHAcAuEACBbej+S34qEwSA/RzPKMByThAe
QBOA04Q2NxsOQwAe5kushwVA9z9r+y4C1vyQH8WnWUABQ37dQl6Lpcag7YwAVFJilBhbq3nm2toaknVI
LdH8IdAE5+jrSWrjOyZh4JhYTsLQkefkGh2kUyAmMpT9Yjx00iziDsBnYKmIeJ9uey/Tdr8v02T6wCRy
HZBaTOFpx9R17td88L70tHI/27Gs4uw63sAoAHxcFfHHJurvLWivCINTf+wI+2DOQBWA068Dhy2mraZ5
HnW4FAJA4SEALfSCgtTl011a21EAcSbd05qrnETs7jDA8KxpspYqxlanXvKwPhE8lmwTL/E1dLcVHNID
mhBZ+Xxisk06RYLjJYHnm84b+4kXxrPOMZuCO0Prdeq95XU5X40bt3saz8GzCLc9xvTTOQDbAe4n1ykD
zhnHpaCQX7GcnqvHsaQ87I/YFICwB8iv+0+Fkjl/3f2vAnCoCXctewCXIVw6AbCDB1QBoOBpA24JAJ1H
kwBgWGF8r+4igIFlbe86zyDARGKMEOOofbmMbc4j8cZ0pgGVJB1l74HjKuLc0z6vE8s1SP68do8j/Uj0
CkgomXnu6R5i2+uaZrqp4Ri289z1HD4X2yy9Xr2G5/C8KQCcD0S8eZbphzZmuEijMOotbQkAnmHtT2VA
2/8sARgnAjXy7wJwS8INC0AQ3tq+4m0vuPvKp557V7p8KQD96zkQe65DQHA4NJiW6A0YpAlDft6p1rF
q76qGaXG+OpnNcOW+Bot2xotYJ24PIZssy+2WRJX4TELDPCBTO+5FYDpPLGeZOvXrZCIRvEv06hG4PaeJ

PCnHANbBkCL3e6jPaN6y7jGkIW29b7d5pnod74tzkOYiAmDtZ8gR5Oc7EacEINIqAHUeQNoHKLZ6zOE4
BCCI/wi9rxsCQIFB7izgIDtYCMDX35UCgIdAgZ8SgD4LLXvQMS4NNgIfBMfIMNbYhuDEAdbdfuVXhmd0
7VeGd5BGHETNfRgtxu92pGU9jTrSSri8drn+JOGYifii308VJdJl2jg+sXKuus1SmL7uq0QDVQB8rkbm
nl/9GVk37tT2cE6uK/ndlouYtsMj8LogPZka6z5TGwnoHxthynKk56e9mAoAt2cfADaCrWTPf7T9FQDJ
f5YApAewC8CtCwvi9/X2DvZDp34YhCmeFC69thSqtF3bn9+EYBKA7gGg/KSF/MAv6NIEYGotApBGB1mC
yJILdx8vQIKJJPYwzRJSi0kUikAs0nSCCEQhxaLfg+nzmJom7kcBIE6i5TqkCEAskQLgtSONS45TnNj2
msDzKVjWxKyBjs8T++szJ4gLTpde0nJn6bp9H+uIQxUe7r1uZ3mEQDBak8/YwJy04RLAaBJf+kRAF4Z
x04Qe2cBIgW/1UAGCECW30AJAigAA/vAnDrwnkFgIJB3fEAIDXqbm2feH4TgoUHEAJA4a4JALVGjoch
sdIon9WmNHxb+xq0xg+qYed6gWklgmDIihGksRL0a43p635g/HRM3ENNi7hkfidZMOQ57euox7K9OL48
t+tj2jFdjXPbuOlcyUARnpFmo41zyKvRRkFJkEIB4ChQMhPU47ypPwUAGzEmYDZ8Re1/ykBCOivPICV
PgAEgNofvuwCcIvCugAwD6D9OvBHP3BYAN76vC4A3RMAb7nv7vwCLh2CFC5eQ20CUPsJAM4BoC05jSsX
48b4JKXE1Pg15kzbDboa/2pcgPXxfInYV691ar/3BPo18UYSpDdt3Rfx0zEB70XUfT7L5vP042ta1vMa
kZ788LjxfPU8bFdpGhjvP4HnEjB9P1M5LsuqCwBeADU/5cbXkJnnoQBQ7tT8oLr/CgDEp2ki6A/Y6gQk
7AJwC806AMweAALwCz/+PQsBoDCp3SE9AJCKQBUAMwCIAEZBFHYgfUkYVBgPhiXxqUWUiLg0ToFRrhZ
NBelp9F4NdpuuKfiAhnXzzcaufuoua29qRVHZNp+juk4z1Vgfn2X2/3YjOfaHUnK+kw1XT+WZT4/6Yjr
93CWAPA8eTxxPd89PtGJ73aeh/uKcJA/ANuUVxIfIYhlyM9vHVC+2AcTvyh3xvyn2r9P/U2EADcDfBKA
137eJAA0NUcBCOrvAnCrwpYA2Ak4egC07xjKcQRgagJ0VAGgY6e+D4AQUNAYEm1qO+7YxjPAAFhqaABD
TCJ2YLRJkoBxpF/s7xjjcrsaeEk77v+hr7zn5LWB1wThWbrun9nRNRIFLQT0n6TIt6ca0XMv4cT/r/V6m
7b5vev4ep4CYV3ntHpfP5Xkivceyb3GNIgCZ/7E/RSLKgn85oGwsH+IYyCh1R7glPyJvOWMjDvstBKC7
/1sCsOYBYJW7ANYisC4A0e6i93UYBbCHlwIdBQDiU/7W57YfwhfpwQj3L4s7Dj2/mgGYDAYEJUHS9JS
+LQhWWJwKQAacaxr/Biohj6Ry30BjlmLW5wHmKCOwFziVyACwm33mb7uz2t14kvOH3LWA+nrtTzepSat
5xvj6zGvjf07ndfp8dOzBcgf1+vxnsM155rIH/D+yS8FQHGMfIAiDVjH9afmB5QzJGeqOEPflfy4+gqA
5B8FoHYC5jBgNkd3AbhloWbsIpnZrPQBpCtVewCy3xfl5z0mgQhA/jd3DwDwSziMGGAahasYJ76/9BJj
QBoXv7WPMCRZw0hntNqpGW0zYI02jb2kZzs07q/7sraL/ZohQ4BMvxQH0EhydxLsmnG+dtfBwAm1TrG
/fU8xnku7odrN8zh10sBj127rsTfgscB8ohzsmRf5k/kfxUARJvPulNWAMKyvD/KEQJTrtmhF2Iv+afe
/7AXyF/H/vUAefGM7W0PgHkALSzs80jd8QhABjsA9gHwQUyaAGsCYF9AJf8bvq6tIwIYBIWpiwgodNqJ
uI4AY8LIOCYN8CswVAwbItyVBmrtKAFmgp9GTseIZ5leQxcCYG0mJB/353ohDsJz1W2uWZcXwXyv83Ov
pQP1OoBnG9OcBlzPcxg3eUzda0KQKUfkyPICEh+Xn7LFLihnvD4FwJ1/VACV/MDfUDgoAGGOWuRon8cY
jkgAQnlLE6AKgH0ACADtFYeAMPynAABdfGaEOR0U4t8fNYfrAE+AdBgbnQ/DTo3MrZaEfBirBAfU9GNN
fgoeX85TyV9d5TxvIM9LvChpK3luJjv1hiJhSnj11LewrleTimCivguj6790E5Q1KG9ywl1lH+nkg
u+QHeH1s+/IP6fUAslP49f5IymkBWHRKmxRilza53GGIxEASBQAZwIuBOAFM/khPm1+535XEafwBEJH
kmM5ByKAIWE0GFm6nnQQ8jYaY8+9FmLZXP0w3I4001pbFWDop+LjPJxjFA6xtc94gIg4P17jklhVZFy/
ILKdbrj+eZHHR8QfAsfUwt+8ZbzfJgBQACA8gPCUbZzPQOJLeoGtKBgAAdD1F+cVANZ2AbjJYUsA8q+M
AlAwiyZAFHaSutf4LDEICp2OHlxDvwZMAadoxDE1vQJAZYAHgJFR27SxZpZtWirDhs4ORBgmRDqGv86N
OJ7z5PkKchgyLuzPNMN+9zkhZjpfXD/vwfsY1+tyiJuOq+uxL2fm9Xsgvi7X7o99wjkVLut6XS7W+3k5
PuNim/F98t8xfGWAPhrEOif5hABQ/sQjAGzTJKB51342bR4RoLmgGFDWOWA4OnRWEyDt8g7izvWEIXGA
Ftb6ACC0fQDU+JIZY6CQce346atP/vazE/wwBr+GQ0HTFkwRsNaIc1ArYBDs06AgqXMDDiGbChtYS39e
rB0/nl8c2ncI9TjX6/LRhOp8gG3H+an5edkHqRouVwd5IT8/f0Z5W/aU00JAHyBnALJy6ALgbyvuAvAo
hkp619nKMGsPrc4DgLQKGOsnBqAg+e27j7/7aSeffn+9E9hGFBAHagFdr46D+Bgdv7+PgLAPedhx5wDy
W8NTAeAB5Pf+ussPqSE6hK9lzzr2QNk6OsASO1EAPvTmz01BUACws1EAVIBcvYO4cz3haDyA/D8EYHwX
YPQA0v37jndJwzgyB8YIER/4+Un6CoyBwseY9AIQBN1GPGMCoNYg0OH542vOGv/jja0V7cRZWr8JG0f
vcnyCaGm3Nmm7P7tLz41Qbn//jv/QoLyxiuA7PT6IxSAeQDEQf4zBaCzX9usNnqM4WgEgMB3AWoToE4E
ylo8an9IjdtIPAJAgSeZo7YXFLbAkEhPrcKx1CIMETkuTG2RhtInjiRCXIAGtAamShzaBzzfecCchVPx
/b4w/hrPtaZluW/OUWH8IYzp63aNd5/rNwvmHeWRLrvt+RBoyiWFIIScPKCvh3KHZB9405MWZW/5cw7y
nuMB8doK50UA8DZOCwDT0ue3AUcbPbZwVB5AFQB+ysomAC5gtv+6ALDtHHCMJgs5vASEQ1DTAZuRqG14
V4DjOve6kkH6973usxLncJbi0dDicR8xtgrbl04jnfS8zu05/snt+nzceM3zgeOX2632I18mBFkE+9ew
lqbGGb8V5zHXg/Gc3HMKcxCd8mN4DwHAM0LI7dmH3Ja9ZS1SSOJ4RJFtmgKUsQLAPioEXjXGzpyCUL8J
uAvATQ81Q6cmZp2Ia48sBIBOQN72otAnDyCAISAAFCREXkAw1ArjsetH0iQAcTwdiTQnaFciHAiBE0dq
zZ3bPU6hcFLJGkyzOEcBtdta/BrWzn8ecCz5MoJaby2+gJriK9594/ZauovC89iLb81fBUBBZz91D2yO
kQ7YjEAAID/QAwB4HFQE6wLAVOCZ+OJYw1EJwOgBOAyIywlZKTQFIPsGopAlkjsUB+7ohoDLiLGMavCP
/8pdOQU4pxSHNwCmc/RjE57Pc9btEefZX7GWZglj+vMeb5qenmfmgldn+KN85x12zTj/sR5rreGOJYy
sEwsF/ZVAWBJf8B4vGWIKJAO+7AJgGcE+fEwiK8CsJgHwDspKQLFPru9HmO4IwSgZuLWOkEBcB4AP2dN
MwDFdiSAGoPMiAIFmcYQRPdjEKzj9mXBYwA9DWPkGF9OI+b9gRAAPAHEALB+FrBseY6bhXreeh3jjHd9
DeNxFOU9x3juiprW5Ri3lq7CY2oahJqXvyhza3gFQFj2egKsU/tp5A/QHMAzsqmChOE62v/8itIOEegj
v3dy7VN/1r4G1NVRc9potdnjCkciACwj40MAKAgKxI5aft8PAaDGp+AQAFSeQkYEiB+ByuebYrFOmlr7


```
Imports clslibNextforest

Public Class frmRegLicense

    Private Sub btnClose_Click(sender As System.Object, e As System.EventArgs) Handles
btnClose.Click
        Me.Close()
    End Sub

    Private Sub frmRegLicense_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load
        Dim mProtect = New clsProtect()
        Dim mProductID As String = mProtect.GetProductID(gProductTYPE)
        txtProductID.Text = mProductID
        txtProductKEY.Text = gProductKEY

        CheckLicense()

    End Sub

    Private Function CheckLicense() As Boolean
        Dim mProtect = New clsProtect()
        Dim mProductID As String = txtProductID.Text
        Dim mProductKEY As String = txtProductKEY.Text
        Dim rtn As Integer = mProtect.CheckProduct(mProductID, mProductKEY)
        Dim mLicence As Boolean = False
        If rtn < gLicenseLevel Then
            lblStatus.Text = "○ライセンス登録済み○"
            gProductKEY = txtProductKEY.Text
            btnRegister.Visible = False
            mLicence = True
        Else
            lblStatus.Text = " ◆ライセンス未登録◆ "
            btnRegister.Visible = True
            mLicence = False
        End If
        Return mLicence
    End Function

    Private Sub btnRegister_Click(sender As System.Object, e As System.EventArgs) Handles
btnRegister.Click
        If CheckLicense() = False Then
            lblStatus.Text = " ▼KEYが違います▼ "
            Beep()
        End If
    End Sub
End Class
```

Module mdlCtrlSub01

```
Public Sub SetCtrlTextBox(ByRef rMe As Form, ByVal vName As String, ByVal vStr As String)
    Dim cs As Control() = rMe.Controls.Find(vName, True)
    If cs.Length > 0 Then
        CType(cs(0), TextBox).Text = vStr
    End If
End Sub
```

```
Public Sub SetCtrlLabel(ByRef rMe As Form, ByVal vName As String, ByVal vStr As String)
    Dim cs As Control() = rMe.Controls.Find(vName, True)
    If cs.Length > 0 Then
        CType(cs(0), Label).Text = vStr
    End If
End Sub
```

```
Public Sub SetCtrlSize(ByRef rMe As Form, ByVal vName As String, ByVal vWidth As
Double, ByVal vHeight As Double)
    Dim cs As Control() = rMe.Controls.Find(vName, True)
    If cs.Length > 0 Then
        cs(0).Width = vWidth
        cs(0).Height = vHeight
    End If
End Sub
```

```
Public Function SearchCtrl(ByRef rMe As Control, ByVal vName As String) As Control
    Dim cs As Control() = rMe.Controls.Find(vName, True)
    If cs.Length > 0 Then
        SearchCtrl = cs(0)
    Else
        SearchCtrl = Nothing
    End If
End Function
```

...

```
''' <summary>
'''     指定したコントロール内に含まれるコントロールを指定した名前で検索します。</summary>
''' <param name="hParent">
'''     検索対象となる親コントロール。</param>
''' <param name="stName">
'''     検索するコントロールの名前。</param>
''' <returns>
'''     取得したコントロールのインスタンス。取得できなかった場合は Nothing。</returns>
'''
```

```
Public Function FindControl(ByVal hParent As Control, ByVal stName As String) As Control
    ' hParent 内のすべてのコントロールを列挙する
    For Each cControl As Control In hParent.Controls
        ' 列挙したコントロールにコントロールが含まれている場合は再帰呼び出しする
        If cControl.HasChildren Then
            Dim cFindControl As Control = FindControl(cControl, stName)

            ' 再帰呼び出し先でコントロールが見つかった場合はそのまま返す
            If Not cFindControl Is Nothing Then
```

```
        Return cFindControl
    End If
End If

' コントロール名が合致した場合はそのコントロールのインスタンスを返す
If cControl.Name = stName Then
    Return cControl
End If
Next cControl
Return Nothing
End Function

''' <summary>
''' フォームに配置されているコントロールを名前で探す
''' (フォームクラスのフィールドをフィールド名で探す)
''' </summary>
''' <param name="frm">コントロールを探すフォーム</param>
''' <param name="name">コントロール(フィールド)の名前</param>
''' <returns>見つかった時は、コントロールのオブジェクト。
'''  見つからなかった時は、null(VB.NETではNothing)。</returns>
Public Function FindControlByFieldName( _
    ByVal frm As Form, ByVal name As String) As Object
' まずプロパティ名を探し、見つからなければフィールド名を探す
Dim t As System.Type = frm.GetType()

Dim pi As System.Reflection.PropertyInfo = _
    t.GetProperty(name, _
        System.Reflection.BindingFlags.Public Or _
        System.Reflection.BindingFlags.NonPublic Or _
        System.Reflection.BindingFlags.Instance Or _
        System.Reflection.BindingFlags.DeclaredOnly)

If Not pi Is Nothing Then
    Return pi.GetValue(frm, Nothing)
End If

Dim fi As System.Reflection.FieldInfo = _
    t.GetField(name, _
        System.Reflection.BindingFlags.Public Or _
        System.Reflection.BindingFlags.NonPublic Or _
        System.Reflection.BindingFlags.Instance Or _
        System.Reflection.BindingFlags.DeclaredOnly)

If fi Is Nothing Then
    Return Nothing
End If

Return fi.GetValue(frm)
End Function

End Module
```

Module mdlCtrlSub03

```
Public Enum eCtrlMode
    ecmSet = 0    '情報をコントロールにセット
    ecmGet = 1   'コントロールから情報を取得
End Enum

Public Sub CtrlIO_TextBoxToStr(ByVal vMode As eCtrlMode, ByRef rForm As Form, ByVal
vCtrlName As String, ByRef rStr As String)
    Dim mCtrl As TextBox = FindControl(rForm, vCtrlName)
    CtrlIO_TextBoxToStr(vMode, mCtrl, rStr)
End Sub

Public Sub CtrlIO_TextBoxToStr(ByVal vMode As eCtrlMode, ByRef rCtrl As TextBox, ByRef
rStr As String)
    If vMode = eCtrlMode.ecmGet Then
        rStr = rCtrl.Text
    Else
        rCtrl.Text = rStr
    End If
End Sub

Public Sub CtrlIO_NumericToDbl(ByVal vMode As eCtrlMode, ByRef rCtrl As NumericUpDown,
ByRef rData As Double, Optional vDefault As Double = NullData)
    If vMode = eCtrlMode.ecmGet Then
        rData = rCtrl.Value
    Else
        If rData < rCtrl.Minimum Or rData > rCtrl.Maximum Then
            If vDefault = NullData Then
                rCtrl.Value = rCtrl.Minimum
            Else
                rCtrl.Value = vDefault
            End If
        Else
            rCtrl.Value = rData
        End If
    End If
End Sub

Public Sub CtrlIO_NumericToInt(ByVal vMode As eCtrlMode, ByRef rCtrl As NumericUpDown,
ByRef rData As Double, Optional vDefault As Integer = Integer.MinValue)
    If vMode = eCtrlMode.ecmGet Then
        rData = rCtrl.Value
    Else
        If rData < rCtrl.Minimum Or rData > rCtrl.Maximum Then
            If vDefault = Integer.MinValue Then
                rCtrl.Value = rCtrl.Minimum
            Else
                rCtrl.Value = vDefault
            End If
        Else
            rCtrl.Value = rData
        End If
    End If
End Sub
```

End Sub

```
Public Sub CtrlIO_TextBoxToSng(ByVal vMode As eCtrlMode, ByRef rForm As Form, ByVal
vCtrlName As String, ByRef rData As Double, Optional vFormat As String = "")
    Dim mCtrl As TextBox
    mCtrl = FindControl(rForm, vCtrlName)
    CtrlIO_TextBoxToDbl(vMode, mCtrl, rData, vFormat)
End Sub
```

```
Public Sub CtrlIO_TextBoxToSng(ByVal vMode As eCtrlMode, ByRef rCtrl As TextBox, ByRef
rData As Single, Optional vFormat As String = "")
    If vMode = eCtrlMode.ecmGet Then
        rData = Val2(rCtrl.Text)
    Else
        If IsNullData(rData) = True Then
            rCtrl.Text = ""
        Else
            If vFormat = "" Then
                rCtrl.Text = rData
            Else
                rCtrl.Text = rData.ToString(vFormat)
            End If
        End If
    End If
End Sub
```

```
Public Sub CtrlIO_TextBoxToDbl(ByVal vMode As eCtrlMode, ByRef rForm As Form, ByVal
vCtrlName As String, ByRef rData As Double, Optional vFormat As String = "")
    Dim mCtrl As TextBox = FindControl(rForm, vCtrlName)
    CtrlIO_TextBoxToDbl(vMode, mCtrl, rData, vFormat)
End Sub
```

```
Public Sub CtrlIO_TextBoxToDbl(ByVal vMode As eCtrlMode, ByRef rCtrl As TextBox, ByRef
rData As Double, Optional vFormat As String = "")
    If vMode = eCtrlMode.ecmGet Then
        rData = Val2(rCtrl.Text)
    Else
        If IsNullData(rData) = True Then
            rCtrl.Text = ""
        Else
            If vFormat = "" Then
                rCtrl.Text = rData
            Else
                rCtrl.Text = rData.ToString(vFormat)
            End If
        End If
    End If
End Sub
```

```
Public Sub CtrlIO_TextBoxToDate(ByVal vMode As eCtrlMode, ByRef rForm As Form, ByVal
vCtrlName As String, ByRef rDate As Date, Optional vFormat As String = "")
    Dim mCtrl As TextBox = FindControl(rForm, vCtrlName)
```

```
CtrlIO_TextBoxToDate(vMode, mCtrl, rDate, vFormat)
End Sub

Public Sub CtrlIO_TextBoxToDate(ByVal vMode As eCtrlMode, ByRef rCtrl As TextBox, ByRef
rDate As Date, Optional vFormat As String = "")
    If vMode = eCtrlMode.ecmGet Then
        rDate = CDate2(rCtrl.Text)
    Else
        If IsNullDate(rDate) = True Then
            rCtrl.Text = ""
        Else
            If vFormat = "" Then
                rCtrl.Text = rDate
            Else
                rCtrl.Text = rDate.ToString(vFormat)
            End If
        End If
    End If
End Sub

Public Sub CtrlIO_TextBoxToInt(ByVal vMode As eCtrlMode, ByRef rCtrl As TextBox, ByRef
rData As Integer, Optional ByVal vZeroNull As Boolean = False)
    If vMode = eCtrlMode.ecmGet Then
        rData = Val2(rCtrl.Text)
    Else
        If vZeroNull = True And rData = 0 Then
            rCtrl.Text = ""
        Else
            rCtrl.Text = rData
        End If
    End If
End Sub

Public Sub CtrlIO_ComboboxToInt(ByVal vMode As eCtrlMode, ByRef rForm As Form, ByVal
vCtrlName As String, ByRef rDat As Integer, Optional ByVal vOffset As Integer = 0)
    Dim mCtrl As ComboBox = FindControl(rForm, vCtrlName)
    CtrlIO_ComboboxToInt(vMode, mCtrl, rDat, vOffset)
End Sub

Public Sub CtrlIO_ComboboxToInt(ByVal vMode As eCtrlMode, ByRef rCtrl As ComboBox,
ByRef rDat As Integer, Optional ByVal vOffset As Integer = 0)
    If vMode = eCtrlMode.ecmGet Then
        rDat = rCtrl.SelectedIndex + vOffset
    Else
        Dim mIndex As Integer = rDat - vOffset
        If mIndex < 0 Then mIndex = -1
        If mIndex >= rCtrl.Items.Count Then mIndex = -1
        rCtrl.SelectedIndex = mIndex
    End If
End Sub

Public Sub CtrlIO_ComboboxToStr(ByVal vMode As eCtrlMode, ByRef rForm As Form, ByVal
vCtrlName As String, ByRef rDat As String)
    Dim mCtrl As ComboBox = FindControl(rForm, vCtrlName)
    CtrlIO_ComboboxToStr(vMode, mCtrl, rDat)
```

```
End Sub
Public Sub CtrlIO_ComboBoxToStr(ByVal vMode As eCtrlMode, ByRef rCtrl As ComboBox,
ByRef rDat As String)
    If vMode = eCtrlMode.ecmGet Then
        rDat = rCtrl.Text
    Else
        rCtrl.Text = rDat
    End If
End Sub

Public Sub CtrlIO_CheckBoxToBool(ByVal vMode As eCtrlMode, ByRef rCtrl As CheckBox,
ByRef rData As Boolean)
    If vMode = eCtrlMode.ecmGet Then
        rData = rCtrl.Checked
    Else
        rCtrl.Checked = rData
    End If
End Sub

Public Sub CtrlIO_DateTimePickerToDate(ByVal vMode As eCtrlMode, ByRef rCtrl As
DateTimePicker, ByRef rDate As Date)
    If vMode = eCtrlMode.ecmGet Then
        rDate = rCtrl.Value
        rDate = CDate(rDate.ToString("yyyy/MM/dd"))
    Else
        rDate = CDate(rDate.ToString("yyyy/MM/dd"))
        rCtrl.Value = rDate
    End If
End Sub

Public Sub CtrlIO_Rdo2ToInt(ByVal vMode As eCtrlMode, ByRef rCtrl1 As RadioButton,
ByRef rCtrl2 As RadioButton, ByRef rDate As Integer, Optional ByVal vOffset As Integer
= 0)
    Dim mData As Integer = -1
    If vMode = eCtrlMode.ecmGet Then
        If rCtrl1.Checked = True Then
            mData = 0
        ElseIf rCtrl2.Checked = True Then
            mData = 1
        End If
        mData += vOffset
    Else
        mData = rDate - vOffset
        rCtrl1.Checked = If(mData = 0, True, False)
        rCtrl2.Checked = If(mData = 1, True, False)
    End If
End Sub

Public Sub CtrlIO_Rdo3ToInt(ByVal vMode As eCtrlMode, ByRef rCtrl1 As RadioButton,
ByRef rCtrl2 As RadioButton, ByRef rCtrl3 As RadioButton, ByRef rDate As Integer,
Optional ByVal vOffset As Integer = 0)
    Dim mData As Integer = -1
    If vMode = eCtrlMode.ecmGet Then
        If rCtrl1.Checked = True Then
            mData = 0
```

```
ElseIf rCtrl2.Checked = True Then
    mData = 1
ElseIf rCtrl3.Checked = True Then
    mData = 2
End If
mData += vOffset
rDate = mData
Else
    mData = rDate - vOffset
    rCtrl1.Checked = If(mData = 0, True, False)
    rCtrl2.Checked = If(mData = 1, True, False)
    rCtrl3.Checked = If(mData = 2, True, False)
End If
End Sub

End Module
```

Module mdlGridSub01

```
Public Enum eCtrlMode
```

```
    ecmSet = 0    '情報をコントロールにセット  
    ecmGet = 1    'コントロールから情報を取得
```

```
End Enum
```

```
Public Enum eDataType
```

```
    edtValue = 0    'Value  
    edtColor = 1    'Color  
    edtCombo = 2    'ComboBox  
    edtChkBx = 3    'CheckBox
```

```
End Enum
```

```
Public Sub Init_gridChInfo(ByRef rGrid As System.Windows.Forms.DataGridView)
```

```
    With rGrid
```

```
        .Rows.Add(10)
```

```
        For i = 1 To .Rows.Count
```

```
            .Rows(i - 1).HeaderCell.Value = "Ch." & Right(" " & i.ToString(), 2)
```

```
        Next
```

```
    End With
```

```
End Sub
```

```
Public Sub Init_gridRecorderRegister(ByRef rGrid As System.Windows.Forms.DataGridView)
```

```
    With rGrid
```

```
        .Rows.Add(10)
```

```
        For i = 1 To .Rows.Count
```

```
            .Rows(i - 1).HeaderCell.Value = i.ToString
```

```
        Next
```

```
    End With
```

```
End Sub
```

```
Public Sub Init_gridAddressList(ByRef rGrid As System.Windows.Forms.DataGridView)
```

```
    With rGrid
```

```
        .Rows.Add(50)
```

```
        For i = 1 To .Rows.Count
```

```
            .Rows(i - 1).HeaderCell.Value = i.ToString
```

```
        Next
```

```
    End With
```

```
End Sub
```

```
Public Sub Init_gridSendReport(ByRef rGrid As System.Windows.Forms.DataGridView)
```

```
    With rGrid
```

```
        .Rows.Add(50)
```

```
        For i = 1 To .Rows.Count
```

```
            .Rows(i - 1).HeaderCell.Value = i.ToString
```

```
        Next
```

```
    End With
```

```
rGrid(0, 0).Value = "管理者1"
```

```
rGrid(0, 1).Value = "管理者②"  
rGrid(0, 2).Value = "管理者3"  
rGrid(0, 3).Value = "管理者IV"  
rGrid(0, 4).Value = "管理者五"
```

End Sub

```
Public Sub Init_gridSavePitchDB(ByRef rGrid As System.Windows.Forms.DataGridView)
```

```
    With rGrid  
        .Rows.Add(1)  
    End With
```

End Sub

```
Public Sub GridItemIO(ByVal vMode As eCtrlMode, ByRef rData As Object, ByRef rGrid As  
System.Windows.Forms.DataGridView, ByVal vCol As Integer, ByVal vRow As Integer,  
    Optional ByVal vDatatype As eDataType = eDataType.edtValue,  
    Optional ByVal vFormat As String = "")
```

```
    Dim mCell As DataGridViewCell = rGrid.Item(vCol, vRow)  
    Dim mColumn As DataGridViewColumn = rGrid.Columns(vCol)  
    GridItemIO_(vMode, rData, mCell, mColumn, vDatatype, vFormat)
```

End Sub

```
Public Sub GridItemIO2(ByVal vMode As eCtrlMode, ByRef rData As Object, ByRef rGrid As  
System.Windows.Forms.DataGridView, ByVal vColName As String, ByVal vRow As Integer,  
    Optional ByVal vDatatype As eDataType = eDataType.edtValue,  
    Optional ByVal vFormat As String = "")
```

```
    Dim mCell As DataGridViewCell = rGrid.Item(vColName, vRow)  
    Dim mColumn As DataGridViewColumn = rGrid.Columns(vColName)  
    GridItemIO_(vMode, rData, mCell, mColumn, vDatatype, vFormat)
```

End Sub

```
Public Sub GridItemIO_(ByVal vMode As eCtrlMode, ByRef rData As Object,  
    ByRef rCell As DataGridViewCell, ByRef rColumn As  
    DataGridViewColumn,  
    Optional ByVal vDatatype As eDataType = eDataType.edtValue,  
    Optional ByVal vFormat As String = "")
```

```
    Select Case vDatatype  
        Case eDataType.edtValue  
            Select Case TypeName(rData)  
                Case "Double", "Single"  
                    If vMode = eCtrlMode.ecmSet Then  
                        If IsNullData(rData) = True Then  
                            rCell.Value = ""  
                        Else  
                            If vFormat <> "" Then  
                                rCell.Value = rData.ToString(vFormat)  
                            Else  
                                rCell.Value = rData  
                            End If  
                        End If  
                    Else  
                        Dim mTypeName As String  
                        mTypeName = TypeName(rCell.Value)  
                        If mTypeName = "Nothing" Then  
                            rData = NullData
```

```
ElseIf mTypeName = "String" Then
    If rCell.Value = "" Then
        rData = NullData
    Else
        Try
            rData = Double.Parse(rCell.Value)
        Catch
            rData = NullData
        End Try
    End If
Else
    rData = rCell.Value
End If
End If
Case "Date"
If vMode = eCtrlMode.ecmSet Then
    If IsNullDate(rData) = True Then
        rCell.Value = ""
    Else
        If vFormat <> "" Then
            Dim mDate As Date
            mDate = rData
            rCell.Value = mDate.ToString(vFormat)
        Else
            rCell.Value = rData
        End If
    End If
Else
    If IsDate(rCell.Value) Then
        rData = rCell.Value
    Else
        rData = NullDate
    End If
End If
Case "Integer"
If vMode = eCtrlMode.ecmSet Then
    If rData = 0 And vFormat = "#" Then
        rCell.Value = ""
    Else
        rCell.Value = rData
    End If
Else
    rData = rCell.Value
End If
Case Else
    If vMode = eCtrlMode.ecmSet Then
        rCell.Value = rData
    Else
        rData = rCell.Value
    End If
End Select
Case eDataType.edtColor
```

```
If vMode = eCtrlMode.ecmSet Then
    rCell.Style.BackColor = rData
Else
    rData = rCell.Style.BackColor
End If
Case eDataType.edtCombo
    Dim mColumn As System.Windows.Forms.DataGridViewComboBoxColumn
    Dim mCell As System.Windows.Forms.DataGridViewComboBoxCell
    mColumn = rColumn
    mCell = rCell
    If vMode = eCtrlMode.ecmSet Then
        Dim mIndex As Integer = rData
        If mIndex < 0 Then mIndex = 0
        If mIndex > mColumn.Items.Count - 1 Then mIndex = mColumn.Items.Count - 1
        mCell.Value = mColumn.Items(mIndex).ToString
    Else
        Dim mStrValue As String
        mStrValue = rCell.Value
        Dim mIndex As Integer = 0
        For i = 0 To mColumn.Items.Count - 1
            If mStrValue = mColumn.Items(i).ToString Then
                mIndex = i
                Exit For
            End If
        Next
        rData = mIndex
    End If
Case eDataType.edtChkBx
    '         Dim mCell As System.Windows.Forms.data
    If vMode = eCtrlMode.ecmSet Then
        rCell.Value = rData
    Else
        rData = rCell.Value
    End If

End Select

End Sub
End Module
```

Module mdlMath01

′ディスプレイ単位をフォント単位に変換 [ピクセル(ドット)→ポイント(pt)]

```
Public Function convPixelToPoint(ByVal vPixel As Integer) As Integer
    Return vPixel * (72.0 / 96.0)
End Function
```

′フォント単位をディスプレイ単位に変換 [ポイント(pt)→ピクセル(ドット)]

```
Public Function convPointToPixel(ByVal vPoint As Integer) As Integer
    Return vPoint * (96.0 / 72.0)
End Function
```

```
Public Function cvByteToFloat(ByRef rData() As Byte, ByVal vStartPos As Integer) As Single
```

```
    Dim byteArray As Byte() = New Byte(4) {}
    byteArray(0) = rData(vStartPos + 1)
    byteArray(1) = rData(vStartPos + 0)
    byteArray(2) = rData(vStartPos + 3)
    byteArray(3) = rData(vStartPos + 2)
    cvByteToFloat = BitConverter.ToSingle(byteArray, 0)
```

```
End Function
```

′時間を指定秒で切り捨て

```
Public Function TimeFloor(ByVal vTime As Date, ByVal vSec As Integer) As Date
```

```
    Dim mDaySec As Integer, mDaySec0 As Integer
    Dim mHour As Integer, mMinute As Integer, mSecond As Integer
```

```
    mDaySec = vTime.Hour * 3600 + vTime.Minute * 60 + vTime.Second
    mDaySec0 = Math.Floor(mDaySec / vSec) * vSec
    mHour = Math.Floor(mDaySec0 / 3600)
    mMinute = Math.Floor((mDaySec0 Mod 3600) / 60)
    mSecond = (mDaySec0 Mod 60)
```

```
    Dim mTimeStr0 As String _
        = vTime.ToString("yyyy/MM/dd") & " " & _
            mHour.ToString("00") & ":" & mMinute.ToString("00") & ":" &
            mSecond.ToString("00")
```

```
    Return Date.Parse(mTimeStr0)
```

```
End Function
```

′日付(年月日)の部分だけを取得する

```
Public Function GetDate(ByVal vDate As DateTime) As DateTime
```

```
    Return vDate.Date
```

```
End Function
```

′時刻の部分だけを取得する

```
Public Function GetTime(ByVal vDate As DateTime) As TimeSpan
```

```
    Return vDate.TimeOfDay
```

```
End Function
```

′文字列が時間か?

```
Public Function IsTime(ByVal vTime As String) As Boolean
```

```
    Dim rtn As TimeSpan
    Return TimeSpan.TryParse(vTime, rtn)
```

```
End Function
```

′文字列を時間に変換

```
Public Function CTime(ByVal vTime As String) As TimeSpan
```

```
    Dim rtn As TimeSpan
```

```
If TimeSpan.TryParse(vTime, rtn) = True Then
    Return rtn
Else
    Return TimeSpan.Zero
End If
End Function

'日跨ぎ処理を調整
Public Function DayStraddle(ByVal vTimeS As Date, ByVal vTimeE As Date) As Date
    Dim mTimeS As Date = vTimeS
    Dim mTimeE As Date = vTimeE
    If mTimeS >= mTimeE Then
        mTimeE = mTimeS.Date.AddDays(1) + mTimeE.TimeOfDay
    End If
    Return mTimeE
End Function

'ゴミ排除(Single制度)
Public Function DelDust(ByVal vData As Double) As Double
    Return Double.Parse(vData.ToString("0.####E-000"))
End Function

'NullDataデータの判定(ゴミ排除)
Public Function IsNullData(ByVal vData As Double) As Boolean
    Return IIf(DelDust(vData) = NullData, True, False)
End Function

Public Function IsNullDate(ByVal vDate As Date) As Boolean
    Dim mDate As Date, mDate0 As Date
    mDate = vDate
    If mDate = NullDate Or mDate = mDate0 Then
        Return True
    Else
        Return False
    End If
End Function

Public Function Str2(ByVal vData As Double, Optional vFormat As String = "") As String
    Dim buf As String
    If IsNullData(vData) = False Then
        If vFormat = "" Then
            buf = vData.ToString
        Else
            buf = vData.ToString(vFormat)
        End If
    Else
        buf = ""
    End If
    Return buf
End Function

Public Function Val2(ByVal vStr As String) As Double
    Dim tmp As Double
    If vStr = "" Then
        tmp = NullData
    End If
End Function
```

```
Else
    tmp = Val(vStr)
End If
Return tmp
End Function

Public Function Math_Add(ByVal vDat1 As Double, ByVal vDat2 As Double) As Double
    If IsNullData(vDat2) = True Then
        Return vDat1
    ElseIf IsNullData(vDat1) = True Then
        Return vDat2
    Else
        Return vDat1 + vDat2
    End If
End Function

Public Function CDate2(ByVal vStr As String) As Date
    Dim tmp As Date
    If vStr = "" Then
        tmp = NullDate
    Else
        If IsDate(vStr) = False Then
            tmp = NullDate
        Else
            tmp = CDate(vStr)
        End If
    End If
    Return tmp
End Function

Public Function Adjust(ByVal vDat As Single) As Single
    Dim rtn As Single, buf As String
    buf = vDat.ToString("0.#####E-000")
    rtn = Val(buf)
    Return rtn
End Function

End Module
```

Module mdlString

```
Public Function convDate(ByVal vStr As String) As Date
    Try
        Return Date.Parse(vStr)
    Catch ex As Exception
        Return NullDate
    End Try
End Function
```

```
Public Function convTimeSpan(ByVal vStr As String) As TimeSpan
    Try
        Return TimeSpan.Parse(vStr)
    Catch ex As Exception
        Return TimeSpan.Zero
    End Try
End Function
```

```
Public Function StrRight(ByVal stTarget As String, ByVal iLength As Integer) As String
    If iLength <= stTarget.Length Then
        Return stTarget.Substring(stTarget.Length - iLength)
    End If
    Return stTarget
End Function
```

・TAB または カンマで区切られた文字列から1次元配列を作成

```
Public Function Split2(ByVal vStr As String) As String()
    Dim mStr() As String
    If InStr(vStr, vbTab) > 0 Then
        mStr = Split(vStr, vbTab)
    Else
        mStr = Split(vStr, ",")
    End If
    Return mStr
End Function
```

・数値配列を文字列に

```
Public Function NumToStr10(ByVal vNo() As Integer) As String
    Dim buf As String = ""
    For i = 1 To 10
        buf &= vNo(i).ToString("000")
    Next
    Return buf
End Function
```

・文字列を数値配列に

```
Public Function StrToNum10(ByVal vStr As String) As Integer()
    Dim rNo(10) As Integer
    vStr &= New String("0"c, 30)
    For i = 1 To 10
        rNo(i) = Val(vStr.Substring((i - 1) * 3, 3))
    Next
    Return rNo
End Function
```

```
'正規表現で文字抜き出し "<ID>個々の文字</ID>"
```

```
Public Function RegularExpressions(ByVal vStr As String) As String
```

```
Dim fff As String, rtn As String = ""
```

```
fff = "<(h[1-6])\b[^>]*>(.*?)</\1>"
```

```
fff = "<(ID)\b[^>]*>(.*?)</ID>"
```

```
fff = "<ID[^>]*>(.*?)</ID>"
```

```
fff = "<ID>(.*?)</ID>"
```

```
Dim r As New System.Text.RegularExpressions.Regex( _
```

```
fff, _
```

```
System.Text.RegularExpressions.RegexOptions.IgnoreCase Or _
```

```
System.Text.RegularExpressions.RegexOptions.Singleline)
```

```
' Dim aaa As String = "<ID>Abv</ID><ID>XXX</ID>"
```

```
'TextBox1.Text内で正規表現と一致する対象をすべて検索
```

```
Dim mc As System.Text.RegularExpressions.MatchCollection = r.Matches(vStr)
```

```
For Each m As System.Text.RegularExpressions.Match In mc
```

```
    rtn = m.Groups(1).Value
```

```
Next
```

```
Return rtn
```

```
End Function
```

```
End Module
```

Module mdlTytedef

End Module

```
Public Class clsArrange
```

```
Private pvCellN As Integer
```

```
Private pvRows As Integer, pvCols As Integer 'どちらか指定
```

```
Private pvHeight As Double, pvWidth As Double
```

```
Public Sub New()
```

```
SetStyle(6, 3, 0)
```

```
End Sub
```

```
Public Sub SetStyle(ByVal vCells As Integer, ByVal vRows As Integer, ByVal vCols As Integer)
```

```
pvCellN = IIf(vCells > 0, vCells, 1)
```

```
If vRows > 0 Then
```

```
pvRows = vRows
```

```
pvCols = 0
```

```
ElseIf vCols > 0 Then
```

```
pvRows = 0
```

```
pvCols = vCols
```

```
Else
```

```
pvRows = 1
```

```
pvCols = 0
```

```
End If
```

```
End Sub
```

```
Public ReadOnly Property CellN() As Integer
```

```
Get
```

```
Return pvCellN
```

```
End Get
```

```
End Property
```

```
Public ReadOnly Property Rows() As Integer
```

```
Get
```

```
If pvRows > 0 Then
```

```
Rows = pvRows
```

```
Else
```

```
Rows = System.Math.Ceiling(CellN / pvCols)
```

```
End If
```

```
End Get
```

```
End Property
```

```
Public ReadOnly Property Cols() As Integer
```

```
Get
```

```
If pvCols > 0 Then
```

```
Cols = pvCols
```

```
Else
```

```
Cols = System.Math.Ceiling(CellN / pvRows)
```

```
End If
```

```
End Get
```

```
End Property
```

```
'縦方向順か
```

```
Public ReadOnly Property IsLengthwise() As Boolean
```

```
Get
```

```
Return IIf(pvRows > 0, True, False)
```

```
End Get
```

```
End Property
```

```
Public Property Width() As Double
```

```
Get
```

```
Return pvWidth
```

```
End Get
```

```
Set(ByVal value As Double)
```

```
pvWidth = value
```

```
End Set
```

```
End Property
```

```
Public Property Height() As Double
```

```
Get
```

```
Return pvHeight
```

```
End Get
```

```
Set(ByVal value As Double)
```

```
pvHeight = value
```

```
End Set
```

```
End Property
```

```
Public ReadOnly Property CellLeft(ByVal vNo As Integer) As Double
```

```
Get
```

```
If IsLengthwise = True Then
```

```
Return Int((vNo - 1) / Rows) * CellWidth
```

```
Else
```

```
Return ((vNo - 1) Mod Cols) * CellWidth
```

```
End If
```

```
End Get
```

```
End Property
```

```
Public ReadOnly Property CellTop(ByVal vNo As Integer) As Double
```

```
Get
```

```
If IsLengthwise = True Then
```

```
Return ((vNo - 1) Mod Rows) * CellHeight
```

```
Else
```

```
Return Int((vNo - 1) / Cols) * CellHeight
```

```
End If
```

```
End Get
```

```
End Property
```

```
Public ReadOnly Property CellWidth() As Double
```

```
Get
```

```
Return Width / Cols
```

```
End Get
```

```
End Property
```

```
Public ReadOnly Property CellHeight() As Double
```

```
Get
```

```
Return Height / Rows
```

```
End Get
```

```
End Property
```

```
End Class
```

```
Option Strict Off
```

```
Imports Microsoft.Office.Interop
```

```
Imports Microsoft.Office.Core
```

```
Imports System.Runtime.InteropServices
```

```
Public Class clsExcel
```

```
Public _xlsFile As String = "c:\temp\Book1.xls" ' XLSファイル名  
Public _sheetNo As Integer = 1 ' シートNo.  
Public _col As String = "A" ' データ書き込みカラム  
Public _line As String = "1" ' データ書き込み開始行  
Public _printer As String = "" ' 出力プリンター
```

```
Public _A1 As String = "" ' A1  
Public _A2 As String = "" ' A2  
Public _A3 As String = "" ' A3  
Public _A4 As String = "" ' A4  
Public _A5 As String = "" ' A5
```

```
Private objExcel As Excel.Application = Nothing 'Excel.Application = Nothing  
'ExcelApplication  
Private objWorkBooks As Object = Nothing 'Excel.Workbooks = Nothing  
'ExcelWorkBooks  
Private objWorkBook As Object = Nothing 'Excel.Workbook = Nothing  
'ExcelWorkbook
```

```
Public Property ExcelFileName() As String
```

```
Get
```

```
Return _xlsFile
```

```
End Get
```

```
Set(ByVal value As String)
```

```
_xlsFile = value
```

```
End Set
```

```
End Property
```

```
Public Property Cell(ByVal vRow As Integer, ByVal vCol As Integer) As String
```

```
Get
```

```
Return objWorkBook.Sheets(_sheetNo).Range(_col & _line + 0) = _A1
```

```
End Get
```

```
Set(ByVal value As String)
```

```
_xlsFile = value
```

```
End Set
```

```
End Property
```

```
Public Function Open() As Boolean
```

```
Try
```

```
'EXCEL開始処理
```

```
Try
```

```
objExcel = CreateObject("Excel.Application") ' New Excel.Application
```

```
Catch ex As Exception
```

```
Throw New Exception( _
```

```
"Microsoft Office Excelがインストールされていないため" & vbCrLf & _
```

```
"印刷できません。")
```

```
End Try
```

```

If Dir(_xslFile) = "" Then
    Throw New Exception( _
        "エクセルファイルが見つかりませんでした。" & vbCrLf & _xslFile)
End If

'Const xlMaximized = -4137
Const xlMinimized = -4140
'Const xlNormal = -4143
objExcel.WindowState = xlMinimized ' Excel.XlWindowState.xlMinimized
objExcel.Visible = False
objExcel.DisplayAlerts = False

' CType(objExcel.Workbooks, Excel.Workbooks)
objWorkBooks = objExcel.Workbooks
' CType(objWorkBooks.Open(_xslFile), Excel.Workbook)
objWorkBook = objWorkBooks.Open(_xslFile)

objWorkBook.Saved = True ' 保存済みとする
Return True

Catch ex As Exception
    Throw ex
End Try

End Function

Public Function Close() As Boolean

    ' EXCEL終了処理
    If Not IsNothing(objWorkBook) Then
        objWorkBook.Close(False) ' ファイルを閉じる
        Marshal.ReleaseComObject(objWorkBook) ' オブジェクト参照を解放
        objWorkBook = Nothing ' オブジェクト解放
    End If

    If Not IsNothing(objWorkBooks) Then
        Marshal.ReleaseComObject(objWorkBooks) ' オブジェクト参照を解放
        objWorkBooks = Nothing ' オブジェクト解放
    End If

    If Not IsNothing(objExcel) Then
        objExcel.Quit() ' EXCELを閉じる
        Marshal.ReleaseComObject(objExcel) ' オブジェクト参照を解放
        objExcel = Nothing ' オブジェクト解放
    End If
    System.GC.Collect() ' オブジェクトを確実に削除

    Return True

End Function

Public Sub CreateRow(ByVal vRows As Integer)
    Dim xlRange As Object ' Excel.Range
    Dim xlPasteRange As Object ' Excel.Range

```

```
Dim buf As String

With objWorkBook.Sheets(_sheetNo)

    xlRange = .Rows("5:5")
    xlRange.Copy()

    For i = 1 To vRows
        buf = (i + 5).ToString & ":" & (i + 5).ToString
        xlPasteRange = .Rows(buf)
        .Paste(xlPasteRange)
    Next

End With

End Sub

Public Sub SetCell(ByVal vRow As Integer, ByVal vCol As Integer, ByVal vStr As String)

    'EXCEL出力処理
    With objWorkBook.Sheets(_sheetNo)

        .Cells(vRow, vCol) = vStr

    End With

End Sub

Public Sub PrintOut()

    '印刷実行
    With objWorkBook.Sheets(_sheetNo)
        If _printer <> "" Then
            .PrintOut(Copies:=1, ActivePrinter:=_printer, Collate:=True)
        Else
            .PrintOut(Copies:=1, Collate:=True)
        End If
    End With

End Sub

Public Sub PrintPreview()

    '印刷プレビュー実行

    'Const xlMaximized = -4137
    'Const xlMinimized = -4140
    Const xlNormal = -4143
    objExcel.WindowState = xlNormal ' Excel.XlWindowState.xlMinimized
    objExcel.Visible = True
    objExcel.DisplayAlerts = True

    With objWorkBook.Sheets(_sheetNo)
        .PrintPreview()
    End With

End Sub
```

End Class

```
Public Class clsGraphLine

    Private pvChart As Windows.Forms.DataVisualization.Charting.Chart

    Public Sub SetChart(ByVal rChart As Windows.Forms.DataVisualization.Charting.Chart)
        pvChart = rChart
    End Sub

    Public Sub Initialize(ByVal vRecInfo As clsRecorderInfo)

        ItemCount = vRecInfo.ChannelN
        SetAxisTypeX()
        For mCh = 1 To ItemCount
            If mCh < ItemCount Then
                pvChart.Series.Item(mCh - 1).LegendText = vRecInfo.ChannelInfo(mCh).Name
                pvChart.Series.Item(mCh - 1).Color = vRecInfo.ChannelInfo(mCh).Color
            Else
                pvChart.Series.Item(mCh - 1).LegendText = " "
                pvChart.Series.Item(mCh - 1).Color = Color.White
            End If
        Next

    End Sub

    Public Sub SetAxisTypeX()
        pvChart.ChartAreas(0).AxisX.LabelStyle.Format = "dd HH:mm"
        pvChart.Series(0).XValueType = DataVisualization.Charting.ChartValueType.DateTime
    End Sub

    Public Property ItemCount() As Integer
        Get
            Return pvChart.Series.Count
        End Get
        Set(ByVal value As Integer)
            Dim mItem As System.Windows.Forms.DataVisualization.Charting.Series
            pvChart.Series.Clear()
            For i = 1 To value
                mItem = pvChart.Series.Add("Item" + i.ToString("00"))
                mItem.YAxisType = DataVisualization.Charting.AxisType.Secondary
                mItem.ChartType = DataVisualization.Charting.SeriesChartType.Line
                mItem.XValueType = DataVisualization.Charting.ChartValueType.DateTime
                mItem.YValueType = DataVisualization.Charting.ChartValueType.Double
            Next
            ' 軸表示用ダミー
            mItem = pvChart.Series.Add("Dummy")
            mItem.YAxisType = DataVisualization.Charting.AxisType.Primary
            mItem.ChartType = DataVisualization.Charting.SeriesChartType.Line
            mItem.XValueType = DataVisualization.Charting.ChartValueType.DateTime
            mItem.YValueType = DataVisualization.Charting.ChartValueType.Double
            ' DataAdd(pvChart.Series.Count - 1, Date.Now, 0)
        End Set
    End Property
End Class
```

```
Private Function IsItemNo(ByVal vNo As Integer) As Boolean
    Return IIf(vNo >= 0 And vNo < ItemCount, True, False)
End Function

Public Property ItemName(ByVal vNo As Integer) As String
    Get
        If IsItemNo(vNo) Then
            Return pvChart.Series.Item(vNo).Name
        Else
            Return "Nothing"
        End If
    End Get
    Set(ByVal value As String)
        If IsItemNo(vNo) Then
            pvChart.Series.Item(vNo).Name = value
        Else
            Beep()
        End If
    End Set
End Property

Public Sub SetAxisX(ByVal vMin As Double, ByVal vMax As Double)
    pvChart.ChartAreas(0).AxisX.Minimum = vMin
    pvChart.ChartAreas(0).AxisX.Maximum = vMax
End Sub

Public Sub SetAxisTime(ByVal vMin As Date, ByVal vMax As Date)
    pvChart.ChartAreas(0).AxisX.Minimum = vMin.ToOADate
    pvChart.ChartAreas(0).AxisX.Maximum = vMax.ToOADate
End Sub

Public Sub SetAxisY(ByVal vMin As Double, ByVal vMax As Double, Optional ByVal vFormat
As String = "")
    pvChart.ChartAreas(0).AxisY.Minimum = DelDust(vMin)
    pvChart.ChartAreas(0).AxisY.Maximum = DelDust(vMax)
    pvChart.ChartAreas(0).AxisY.LabelStyle.Format = vFormat
End Sub

Public Sub SetAxisTitle(ByVal vStr As String, ByVal vColor As System.Drawing.Color)
    pvChart.ChartAreas(0).AxisY.Title = vStr
    pvChart.ChartAreas(0).AxisY.TitleForeColor = vColor
End Sub

Public Sub SetAxisY2(ByVal vMin As Double, ByVal vMax As Double)
    pvChart.ChartAreas(0).AxisY2.Minimum = DelDust(vMin)
    pvChart.ChartAreas(0).AxisY2.Maximum = DelDust(vMax)
End Sub

Public Sub DrawLineWidth(ByVal vItemNo As Integer, ByVal vWidth As Integer)
    If IsItemNo(vItemNo) Then
        pvChart.Series(vItemNo).BorderWidth = vWidth
    End If
End Sub
```

```
Public Sub DataAdd(ByVal vItemNo As Integer, ByVal vTime As DateTime, ByVal vData As Double)
    Dim idx As Integer
    If IsItemNo(vItemNo) Then
        idx = pvChart.Series(vItemNo).Points.AddXY(vTime, vData)
        If IsNullData(vData) = True Then 'NullDataはEmpty値として扱う
            pvChart.Series(vItemNo).Points(idx).IsEmpty = True
        End If
    End If
End Sub

Public Function GetPointN(ByVal vItemNo As Integer) As Integer
    Dim mPointN As Integer = 0
    If IsItemNo(vItemNo) Then
        mPointN = pvChart.Series(vItemNo).Points.Count
    End If
    Return mPointN
End Function

Public Sub DataDel0(ByVal vItemNo As Integer)
    If IsItemNo(vItemNo) Then
        pvChart.Series(vItemNo).Points.RemoveAt(0)
    End If
End Sub

Public Sub OldDataDel(ByVal vItemNo As Integer)
    Dim mX0 As Double, mX1 As Double
    If IsItemNo(vItemNo) Then
        mX0 = pvChart.Series(vItemNo).Points(0).XValue
        mX1 = pvChart.ChartAreas(0).AxisX.Minimum
        Dim mT0 As Date = Date.FromOADate(mX0)
        Dim mT1 As Date = Date.FromOADate(mX1)
        If mX0 < mX1 Then
            pvChart.Series(vItemNo).Points.RemoveAt(0)
        End If
    End If
End Sub

Public Sub DataClear(Optional ByVal vItemNo As Integer = -1)
    Dim mStaNo As Integer, mEndNo As Integer
    If vItemNo = -1 Then
        mStaNo = 0
        mEndNo = ItemCount - 1
    Else
        mStaNo = vItemNo
        mEndNo = vItemNo
    End If
    For mItenNo = mStaNo To mEndNo
        If IsItemNo(mItenNo) Then
            pvChart.Series(mItenNo).Points.Clear()
        End If
    Next
End Sub
```

```
Public Sub TestData()  
    Dim i As Integer, index As Integer  
  
    For i = 1 To 50  
        index = pvChart.Series(0).Points.AddXY(i, i)  
  
        pvChart.Series(0).Points(index).Label = i.ToString("a00")  
    Next  
  
End Sub  
  
End Class
```

```
Public Class clsSendMail
```

```
Private pvCharset As String      '文字コード(charset)
Private pvServerName As String   'サーバー名hostname
Private pvPortNo As Integer      'ポート番号
Private PvUser As String         'ユーザID
Private pvPassword As String     'パスワード
Private pvFromAddress As String  '送信元<from@xxx.com>"
Private pvToAddress As String    '送信先<to@xxx.com>"
Private pvSubject As String      '件名
Private pvBody As String         '本文
```

```
Public Sub New()
```

```
    Initialize()
```

```
End Sub
```

```
Public Sub Initialize()
```

```
    pvCharset = "ISO-2022-JP"
    pvServerName = "smtp.lolipop.jp"
    pvPortNo = 587
    PvUser = "m11117535-test"      ' "user"
    pvPassword = "test0123"       ' "pass"
    pvFromAddress = "test@nextforest.jp"      ' "送信元<fr@xxx.com>"
    pvToAddress = "nextforest@my.email.ne.jp" ' "送信先<to@xxx.com>"
    pvSubject = "タイトル test タイトルです。"
    pvBody = "本文 test 本文です。"
```

```
End Sub
```

```
Public Property ServerName As String
```

```
    Get
```

```
        Return pvServerName
```

```
    End Get
```

```
    Set(ByVal Value As String)
```

```
        pvServerName = Value
```

```
    End Set
```

```
End Property
```

```
Public Property PortNo As Integer
```

```
    Get
```

```
        Return pvPortNo
```

```
    End Get
```

```
    Set(ByVal Value As Integer)
```

```
        pvPortNo = Value
```

```
    End Set
```

```
End Property
```

```
Public Property UserID As String
```

```
    Get
```

```
        Return PvUser
```

```
    End Get
```

```
Set(ByVal Value As String)
    PvUser = Value
End Set
End Property

Public Property Password As String
Get
    Return pvPassword
End Get
Set(ByVal Value As String)
    pvPassword = Value
End Set
End Property

Public Property FromAddress As String
Get
    Return pvFromAddress
End Get
Set(ByVal Value As String)
    pvFromAddress = Value
End Set
End Property

Public Property ToAddress As String
Get
    Return pvToAddress
End Get
Set(ByVal Value As String)
    pvToAddress = Value
End Set
End Property

Public Property Subject As String
Get
    Return pvSubject
End Get
Set(ByVal Value As String)
    pvSubject = Value
End Set
End Property

Public Property Body As String
Get
    Return pvBody
End Get
Set(ByVal Value As String)
    pvBody = Value
End Set
End Property

''' <summary>
''' メール送信
''' </summary>
Public Function SendMail() As Boolean
```

```
' TcpClientソケット通信定義
Dim stream As System.Net.Sockets.NetworkStream
Dim client As System.Net.Sockets.TcpClient

Try
    ' SMTPサーバーに接続
    client = New System.Net.Sockets.TcpClient
    client.Connect(pvServerName, pvPortNo)
Catch ex As Exception
    Return False
End Try

' SSL無しの実装-----
' 通信開始(SSL無し)
stream = client.GetStream()
' -----

' SMTPサーバー接続時のレスポンス受信
Dim rstr As String
rstr = StreamWriteAndRead(stream, "", pvCharset)
If rstr.StartsWith("2") = False Then
    Throw New Exception("エラー:" & rstr)
    Return False
End If

' EHLOの送信
Dim pcname As String = System.Net.Dns.GetHostName()
rstr = StreamWriteAndRead(stream, "EHLO " & pcname & vbCrLf, pvCharset)
If rstr.StartsWith("2") = False Then
    Throw New Exception("エラー:" & rstr)
    Return False
End If

' SMTP AUTH 認証の実装-----
' AUTH PLAINの送信(LOGINやCRAM-MD5はNonMailNet.dllをお試しください)
rstr = StreamWriteAndRead(stream, "AUTH PLAIN" & vbCrLf, pvCharset)
If rstr.StartsWith("502") = False Then
    If rstr.StartsWith("3") = False Then
        Throw New Exception("エラー:" & rstr)
    End If
    ' ユーザIDとパスワードの生成
    Dim upas As String = PvUser & ControlChars.NullChar & _
        PvUser & ControlChars.NullChar & pvPassword
    ' ユーザIDとパスワードの送信
    rstr = StreamWriteAndRead(stream, Enc_Base64_1(upas, pvCharset) & vbCrLf, pvCharset)
    If rstr.StartsWith("2") = False Then
        'Throw New Exception("エラー:" & rstr)
        Return False
    End If
End If
' -----

' MAIL FROMの送信
pvFromAddress = pvFromAddress.Trim()
```

```
Dim fa As String = New System.Net.Mail.MailAddress(pvFromAddress).Address
rstr = StreamWriteAndRead(stream, "MAIL FROM:<" & fa & ">" & vbCrLf, pvCharset)
If rstr.StartsWith("2") = False Then
    'Throw New Exception("エラー:" & rstr)
    Return False
End If

' RCPT TOの送信
pvToAddress = pvToAddress.Trim()
Dim ta As String = New System.Net.Mail.MailAddress(pvToAddress).Address
rstr = StreamWriteAndRead(stream, "RCPT TO:<" & ta & ">" & vbCrLf, pvCharset)
If rstr.StartsWith("2") = False Then
    'Throw New Exception("エラー:" & rstr)
    Return False
End If

' DATAの送信
rstr = StreamWriteAndRead(stream, "DATA" & vbCrLf, pvCharset)
If rstr.StartsWith("2") = False And rstr.StartsWith("3") = False Then
    'Throw New Exception("エラー:" & rstr)
    Return False
End If

' 本文DATAの作成(From)
Dim data As String = ""
Dim frd As String = New System.Net.Mail.MailAddress(pvFromAddress).DisplayName
data &= "From: " & Enc_Base64_2(frd, pvCharset) & "<" & fa & ">" & vbCrLf

' 本文DATAの作成(To)
Dim tod As String = New System.Net.Mail.MailAddress(pvToAddress).DisplayName
data &= "To: " & Enc_Base64_2(tod, pvCharset) & "<" & ta & ">" & vbCrLf

' 本文DATAの作成(Subject)
data &= "Subject: " & Enc_Base64_2(pvSubject, pvCharset) & vbCrLf

' 本文DATAの作成(MIME-Version)
data &= "MIME-Version: 1.0" & vbCrLf

' 本文DATAの作成(Content-Type) -- 添付ファイル無しの時 --
data &= "Content-Type: text/plain; charset="" & pvCharset & "" & vbCrLf
' (参考)添付ファイル有りの時は(Content-Type: multipart/mixed; boundary="XXX")

' 本文DATAの作成(Content-Transfer-Encoding)
data &= "Content-Transfer-Encoding: 7bit" & vbCrLf & vbCrLf

' 本文DATAの作成(本文)
data &= pvBody & vbCrLf

' 本文DATAの作成(.を..に変換)
data = data.Replace(vbCrLf & "." & vbCrLf, vbCrLf & ".." & vbCrLf)

' 本文DATAの作成(.を付加)
data &= vbCrLf & "." & vbCrLf

' 本文DATAの送信
```

```
rstr = StreamWriteAndRead(stream, data, pvCharset)
If rstr.StartsWith("2") = False Then
    'Throw New Exception("エラー:" & rstr)
    Return False
End If

' QUITの送信
rstr = StreamWriteAndRead(stream, "QUIT" & vbCrLf, pvCharset)
If rstr.StartsWith("2") = False Then
    'Throw New Exception("エラー:" & rstr)
    Return False
End If

Return True
End Function

''' <summary>
''' メールサーバとの送受信
''' </summary>
''' <param name="stm">メールサーバ通信ストリーム</param>
''' <param name="req">リクエスト(コマンド)</param>
''' <param name="cst">文字コード</param>
''' <returns>レスポンス</returns>
''' <remarks></remarks>
Private Function StreamWriteAndRead( _
    ByVal stm As Object, _
    ByVal req As String, _
    ByVal cst As String) As String

    ' メールサーバへリクエスト送信
    If req <> "" Then
        Dim sdat() As Byte
        sdat = System.Text.Encoding.GetEncoding(cst).GetBytes(req)
        stm.Write(sdat, 0, sdat.Length)
        stm.Flush()
    End If

    ' メールサーバからのレスポンス受信
    Dim rstr As String = ""
    Dim rdat() As Byte
    ReDim rdat(1024 * 1024 - 1)
    Dim l As Integer = stm.Read(rdat, 0, rdat.Length)
    If l > 0 Then
        ReDim Preserve rdat(l - 1)
        rstr = System.Text.Encoding.GetEncoding(cst).GetString(rdat)
    Else
        Throw New Exception("Read Error.")
    End If

    ' レスポンス応答
    Return rstr
End Function

''' <summary>
```

```
''' 指定文字コードのBASE64インコード
''' </summary>
''' <param name="str">オリジナル文字列</param>
''' <returns>インコード結果文字列</returns>
''' <param name="cst">文字コード</param>
''' <remarks></remarks>
Private Function Enc_Base64_1(ByVal str As String, ByVal cst As String) As String
    If str = "" Then Return ""
    Dim enc As System.Text.Encoding = System.Text.Encoding.GetEncoding(cst)
    Return Convert.ToBase64String(enc.GetBytes(str))
End Function

''' <summary>
''' 指定文字コードのBASE64インコード
''' </summary>
''' <param name="str">オリジナル文字列</param>
''' <param name="cst">文字コード</param>
''' <returns>インコード結果文字列</returns>
''' <remarks>"=?ISO-2022-JP?B?"~"?="フォーマット</remarks>
Private Function Enc_Base64_2(ByVal str As String, ByVal cst As String) As String
    If str = "" Then Return ""
    Dim enc As System.Text.Encoding = System.Text.Encoding.GetEncoding(cst)
    Return "=?" & cst & "?B?" & Convert.ToBase64String(enc.GetBytes(str)) & "=?"
End Function

End Class
```

```
Imports System.Net.Sockets
```

```
Public Class clsTcpIP
```

```
    ' Inherits TcpClient
```

```
    Private objClient As System.Net.Sockets.TcpClient
```

```
    Private objStm As System.Net.Sockets.NetworkStream
```

```
    Private pvHostName As String
```

```
    Private pvPortNo As Integer
```

```
    Private pvTimeOut As New clsTimeOut
```

```
Public Sub New()
```

```
    pvTimeOut.Reset(36000)
```

```
End Sub 'New
```

```
Public Property HostName
```

```
    Set(ByVal value)
```

```
        pvHostName = value
```

```
    End Set
```

```
    Get
```

```
        HostName = pvHostName
```

```
    End Get
```

```
End Property
```

```
Public Property PortNo
```

```
    Set(ByVal value)
```

```
        pvPortNo = value
```

```
    End Set
```

```
    Get
```

```
        PortNo = pvPortNo
```

```
    End Get
```

```
End Property
```

```
Public Function TcpConnect2(ByVal vHostName As String, ByVal vPortNo As Integer) As Boolean
```

```
    HostName = vHostName
```

```
    PortNo = vPortNo
```

```
    Return TcpConnect3()
```

```
End Function
```

```
    ' ソケット接続
```

```
Public Function TcpConnect3() As Boolean
```

```
    Try
```

```
        objClient = New System.Net.Sockets.TcpClient(HostName, PortNo)
```

```
        ' ソケットストリーム取得
```

```
        objStm = objClient.GetStream()
```

```
        Return True
```

```
    Catch
```

```
        Return False
```

```
    End Try
```

```
End Function
```

・ ソケットクローズ

```
Public Function TcpClose() As Boolean
    Try
        objClient.Close()
        objStm.Close()
        Return True
    Catch
        Return False
    End Try
End Function

Public Function Write(ByVal buffer() As Byte) As Boolean
    Try
        objStm.Write(buffer, 0, buffer.Length)
        pvTimeOut.Reset(5)
        Return True
    Catch
        Return False
    End Try
End Function

Public Function GetConnected() As Boolean
    Try
        Return objClient.Connected
    Catch
        Return False
    End Try
End Function

Public Function Receive(ByRef rData() As Byte) As Integer
    Dim mReceive As Integer
    Try
        mReceive = objClient.Available()
        If mReceive > 0 Then
            Dim dat As Byte() = New Byte(mReceive - 1) {}
            objStm.Read(dat, 0, dat.GetLength(0))
            rData = dat
            pvTimeOut.Reset(36000)
            Return dat.Length
        Else
            If pvTimeOut.IsTimeOut = True Then
                pvTimeOut.Reset(36000)
                Return -2
            Else
                Return 0
            End If
        End If
    Catch
        Return -1
    End Try

End Function

End Class
```

```
Public Class clsTestData

    Private pvMin As Single
    Private pvMax As Single
    Private pvNow As Single
    Private pvIncDec As Boolean 'true:プラス方向 false:マイナス方向
    Private cRandom As System.Random

    Public Sub New()

        pvMin = 0
        pvMax = 100
        pvNow = 50

        pvIncDec = True

    End Sub

    Public Sub Initialize(Optional ByVal vMin As Single = 0, Optional ByVal vMax As Single
= 100, Optional ByVal vChNo As Integer = 0)
        cRandom = New System.Random(Date.Now.Second * 20 + vChNo)

        Min = vMin
        Max = vMax
        pvNow = Min + (Max - Min) * cRandom.NextDouble()
        pvIncDec = IIf(cRandom.NextDouble() < 0.5, False, True)

    End Sub

    Public Property Min() As Single
        Get
            Return pvMin
        End Get
        Set(ByVal value As Single)
            pvMin = value
        End Set
    End Property

    Public Property Max() As Single
        Get
            Return pvMax
        End Get
        Set(ByVal value As Single)
            pvMax = value
        End Set
    End Property

    Public Property Now() As Single
        Get
            Return pvNow
        End Get
        Set(ByVal value As Single)
            pvNow = value
        End Set
    End Property

End Class
```

```
Public Function GetData() As Single
    Dim mRnd As Single, mData As Single
    mRnd = ((cRandom.NextDouble + cRandom.NextDouble) / 2 - 0.3) * IIf(pvIncDec = False,
    1, -1)
    '    pvNow += ((Max - Min) / 600) * mRnd
    pvNow += ((Max - Min) / 120) * mRnd
    If pvNow < Min Then pvIncDec = False
    If pvNow > Max Then pvIncDec = True
    mData = pvNow
    Return mData
End Function
```

```
End Class
```

```
Public Class clsTimeOut

    Private pvActionDateTime As Date
    Private pvOutTime As TimeSpan 'タイムアウトの時間

    Public Sub New()
        pvOutTime = New TimeSpan(0, 0, 10)
        Reset()
    End Sub

    Public WriteOnly Property DefaultTime
        Set(ByVal value)
            Dim mSec As Integer
            mSec = IIf(value < 0, 0, value)
            pvOutTime = New TimeSpan(0, 0, mSec)
        End Set
    End Property

    Public Sub Reset(Optional ByVal vSec As Single = -1)
        If vSec = -1 Then
            pvOutTime = TimeSpan.MaxValue
        Else
            pvOutTime = New TimeSpan(0, 0, 0, 0, vSec * 1000)
        End If
        pvActionDateTime = Date.Now
    End Sub

    Public Function IsTimeOut() As Boolean
        Return IIf((Date.Now - pvActionDateTime) > pvOutTime, True, False)
    End Function

End Class
```

```
Imports WatchManlock.clsTypedef
```

```
Public Class frmAddressBook
```

```
Public Enum eCtrlMode
```

```
    ecmSet = 0 '情報をコントロールにセット  
    ecmGet = 1 'コントロールから情報を取得
```

```
End Enum
```

```
Public Rtn As eDialogReturn
```

```
Private Sub frmAddressBook_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
    LoadLocation(Me, "AddressBook")  
    Init_gridAddressList(gridAddressList)  
    Rtn = eDialogReturn.edrCancel
```

```
End Sub
```

```
Private Sub frmAddressBook_FormClosing(ByVal sender As System.Object, ByVal e As  
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
```

```
    SaveLocation(Me, "AddressBook")
```

```
End Sub
```

```
Private Sub frmAddressBook_Resize(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Resize
```

```
    If Me.WindowState = FormWindowState.Normal Or Me.WindowState =  
FormWindowState.Maximized Then
```

```
        Me.gridAddressList.Left = 5  
        Me.gridAddressList.Top = 5  
        Me.gridAddressList.Width = Me.ClientSize.Width - 10  
        Me.gridAddressList.Height = Me.ClientSize.Height - pnlOkCancel.Height - 15
```

```
        pnlOkCancel.Top = Me.gridAddressList.Top + Me.gridAddressList.Height + 5  
        pnlOkCancel.Left = Me.gridAddressList.Left + Me.gridAddressList.Width -  
pnlOkCancel.Width
```

```
    End If
```

```
    ' Me.ClientSize = New System.Drawing.Size(mWidth, mHeight) Then
```

```
End Sub
```

```
Private Sub DataIO_AddInfo(ByVal vMode As eCtrlMode)
```

```
    Dim mRow As Integer, mCol As Integer
```

```
    For i = 1 To 50
```

```
        mRow = i - 1
```

```
        With gManlockInfo.AddressInfo(i)
```

```
            GridItemIO(vMode, .Name, gridAddressList, 0, mRow)
```

```
            GridItemIO(vMode, .MailAddress, gridAddressList, 1, mRow)
```

```
            For mRecNo = 1 To 10
```

```
                For j = 1 To 2
```

```
                    mCol = 2 + (mRecNo - 1) * 2 + (j - 1)
```

```
                    GridItemIO(vMode, .AlarmLevel(mRecNo, j), gridAddressList, mCol, mRow,  
eDataType.edtChkBx)
```

```
        Next
    Next
End With
Next
End Sub

Private Sub frmAddressBook_Shown(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Shown
    DataIO_AddInfo(eCtrlMode.ecmSet)
End Sub

Private Sub cmdOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdOK.Click
    DataIO_AddInfo(eCtrlMode.ecmGet)
    gManlockInfo.dbAddressBookIO(clsManlockInfo.eAccessMode.eamSave)
    Rtn = eDialogReturn.edrOK
    Me.Hide()
End Sub

Private Sub cmdCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdCancel.Click
    Rtn = eDialogReturn.edrCancel
    Me.Hide()
End Sub

End Class
```

```
Imports clslibNextforest
Imports WatchManlock.clsTypedef

Public Class frmMain

    ' 列挙体を宣言
    Enum eActionMode
        esmStop = 0
        esmRun = 1
    End Enum

    Private pvDataFileOpen As Boolean
    Private pvActionMode As eActionMode
    Private pvPaperlessRecorder(10) As frmPaperlessRecorder
    Private pvRearrangement As Boolean

    Private LibFF As New clsFolderFile

    ' フォームNew
    Public Sub New()
        ' この呼び出しはデザイナーが必要です。
        InitializeComponent()
        pvActionMode = eActionMode.esmStop

        ' InitializeComponent() 呼び出しの後で初期化を追加します。
        For i = 1 To 2
            NwePaperlessRecorder(i)
        Next
    End Sub

    Private Sub NwePaperlessRecorder(ByVal vNo As Integer)
        pvPaperlessRecorder(vNo) = New frmPaperlessRecorder
        pvPaperlessRecorder(vNo).RecorderNo = vNo
    End Sub

    ' フォームFinalize
    Protected Overrides Sub Finalize()
        For i = 1 To 10
            If pvPaperlessRecorder(i).IsHandleCreated Then
                pvPaperlessRecorder(i).Close()
                pvPaperlessRecorder(i).Dispose()
            End If
        Next
        MyBase.Finalize()
    End Sub

    ' バージョン情報
    Private Sub バージョン情報aToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles バージョン情報aToolStripMenuItem.Click
        Dim fVerInfo As New frmVerInfo()
        fVerInfo.ShowDialog(Me)
        fVerInfo.Dispose()
        LicenseCheck()
    End Sub

    ' メインフォーム読み込み
    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles MyBase.Load
    Me.HelpProvider1.HelpNamespace = GetAppFolder() & "WatchManlock.chm"
    LoadLocation(Me, "FormMain")
    Initialize()

    LicenseCheck()
    ' AxVSFlexGrid1.set_TextMatrix(3, 3, "aaa")
End Sub
Private Sub LicenseCheck()
    If CheckLicense() = True Or CheckTrial() = True Then
        msMainMenu.BackColor = SystemColors.Control
        ToolBarMain.BackColor = SystemColors.Control
    Else
        msMainMenu.BackColor = Color.Red
        ToolBarMain.BackColor = Color.Red
    End If
End Sub

'プログラム終了
Private Sub mnuFileExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFileExit.Click
    Me.Close()
End Sub
'ファイル開く
Private Sub mnuFileOpne_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFileOpne.Click
    Dim mFileName As String
    OpenDbFileDialog.Title = "開くファイルを選択してください"
    OpenDbFileDialog.Filter = "マンロック環境監視 (*.mew;*.mdb) | *.MEW;*.MDB | すべてのファイル (*.*) | *.*"
    OpenDbFileDialog.FileName = gManlockInfo.FileName
    If OpenDbFileDialog.ShowDialog() = DialogResult.OK Then
        mFileName = OpenDbFileDialog.FileName
        gDefaultFolder = LibFF.GetFolderName(mFileName)
        gManlockInfo.FileName = mFileName
        CheckDataFile()
        pvRearrangement = True
    End If
End Sub
'ファイル新規作成
Private Sub mnuFileNew_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFileNew.Click
    Dim libFF As New clsFolderFile
    Dim mFolder As String, mSaveFileName As String
    Dim mAppFolder As String, mBaseDataBaseFile As String

    mAppFolder = GetAppFolder()
    mBaseDataBaseFile = mAppFolder & "\WatchManlock.db0"

    If libFF.ExistsFile(mBaseDataBaseFile) = False Then
        MsgBox("新規データを作成出来ません!", MsgBoxStyle.OkOnly, "Error (Not Base)")
        Return
    End If

    mFolder = gDefaultFolder
    SaveDbFileDialog.FileName = mFolder & "\*.mew"
```

```
SaveDbFileDialog.Filter = "マンロック環境監視 (*.mew)|*.MEW|すべてのファイル (*.*)|*.*"
' SaveDbFileDialog.Filter = "マンロック環境監視 (*.mew;*.mdb)|*.MEW;*.MDB|すべてのファイル (*.*)|*.*"
If SaveDbFileDialog.ShowDialog() = DialogResult.OK Then
    mSaveFileName = SaveDbFileDialog.FileName
    If libFF.FileCopy(mBaseDataBaseFile, mSaveFileName) = False Then
        MsgBox("新規データを作成出来ません!", MsgBoxStyle.OkOnly, "Create Error")
        Return
    End If
    Dim mSaveFileName2 As String
    For i = 0 To 9
        mSaveFileName2 = mSaveFileName + i.ToString("0")
        If libFF.FileCopy(mBaseDataBaseFile, mSaveFileName2) = False Then
            MsgBox("新規データを作成出来ません!", MsgBoxStyle.OkOnly, "Create Error")
            Return
        End If
    Next
    gDefaultFolder = libFF.GetFolderName(mSaveFileName)
    gManlockInfo = New clsManlockInfo
    gManlockInfo.Initialize()
    gManlockInfo.FileName = mSaveFileName
    gManlockInfo.dbManlockInfoIO(clsManlockInfo.eAccessMode.eamSave)
    If CheckDataFile() = True Then
        If ShowPropertyMain() = True Then
            For i = gManlockInfo.RecorderN To 1 Step -1
                pvPaperlessRecorder(i).Show()
                pvPaperlessRecorder(i).Activate()
                pvPaperlessRecorder(i).tcRoot.SelectTab("ViewSettingChannel")
            Next
        End If
    End If
    pvRearrangement = True
End If

End Sub

'フォームを閉じる
Private Sub frmMain_FormClosing(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
    For i = 1 To 10
        If IsNothing(pvPaperlessRecorder(i)) = False Then
            If pvPaperlessRecorder(i).IsHandleCreated Then
                pvPaperlessRecorder(i).pbClose = True
                pvPaperlessRecorder(i).Close()
                'pvPaperlessRecorder(i).Dispose()
            End If
        End If
    Next
    SaveLocation(Me, "FormMain")
    RegSave()
End Sub

'プロパティ表示
Private Sub mnuPropertyMain_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuPropertyMain.Click, tsbProperty.Click
    ShowPropertyMain()
End Sub
```

```

Private Function ShowPropertyMain() As Boolean
    Dim rtn As Boolean = False
    Dim fProperty As New frmPropertyMain
    fProperty.ShowDialog(Me)
    If fProperty.Rtn = eDialogReturn.edrOK Then
        gManlockInfo.dbManlockInfoIO(clsManlockInfo.eAccessMode.eamSave)
        UpdateForm()
        rtn = True
    End If
    fProperty = Nothing
    Return rtn
End Function

```

・アドレス帳表示

```

Private Sub mnuAddressBook_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuAddressBook.Click, tsbAddressBook.Click
    Dim fAddBook As New frmAddressBook
    fAddBook.ShowDialog(Me)
    If fAddBook.Rtn = eDialogReturn.edrOK Then
        '        gManlockInfo.Save()
    End If
    fAddBook = Nothing
End Sub

```

・指定レコーダー表示

```

Private Sub tsbPaperlessRecorder00_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) _
    Handles tsbPaperlessRecorder01.Click, tsbPaperlessRecorder02.Click,
    tsbPaperlessRecorder03.Click, tsbPaperlessRecorder04.Click,
    tsbPaperlessRecorder05.Click, _
        tsbPaperlessRecorder06.Click, tsbPaperlessRecorder07.Click,
        tsbPaperlessRecorder08.Click, tsbPaperlessRecorder09.Click,
        tsbPaperlessRecorder10.Click, _
        btnPaperlessRecorder01.Click, btnPaperlessRecorder02.Click,
        btnPaperlessRecorder03.Click, btnPaperlessRecorder04.Click,
        btnPaperlessRecorder05.Click, _
        btnPaperlessRecorder06.Click, btnPaperlessRecorder07.Click,
        btnPaperlessRecorder08.Click, btnPaperlessRecorder09.Click,
        btnPaperlessRecorder10.Click, _
        mnuViewRecorder01.Click, mnuViewRecorder02.Click, mnuViewRecorder03.Click,
        mnuViewRecorder04.Click, mnuViewRecorder05.Click, _
        mnuViewRecorder06.Click, mnuViewRecorder07.Click, mnuViewRecorder08.Click,
        mnuViewRecorder09.Click, mnuViewRecorder10.Click
    If gLicense = False And CheckTrial() = False Then
        MsgBox("ライセンス登録をお願いします。" & vbLf & "使用を続けると著作権侵害になります。",
        MsgBoxStyle.OkOnly, "警告<<Warning>>")
    End If

    For mRecNo = 1 To 10
        If sender.Name = "tsbPaperlessRecorder" & mRecNo.ToString("00") Or _
            sender.Name = "btnPaperlessRecorder" & mRecNo.ToString("00") Or _
            sender.Name = "mnuViewRecorder" & mRecNo.ToString("00") Then
            gManlockInfo.RecorderInfo(mRecNo).Visible = True
            If pvPaperlessRecorder(mRecNo).WindowState = FormWindowState.Minimized Then
                pvPaperlessRecorder(mRecNo).WindowState = FormWindowState.Normal
                pvPaperlessRecorder(mRecNo).Show()
            End If
        End If
    Next

```

```

    pvPaperlessRecorder(mRecNo).Activate()
    gManlockInfo.dbRecorderInfoIO(clsManlockInfo.eAccessMode.eamSave, mRecNo)
    Exit For
End If
Next
End Sub
'レコーダー表示
Private Sub ViewRecorderCount(ByVal vRecorderN As Integer)
    Dim mCtrl As Control, mHeight As Integer, mWidth As Integer, mName As String
    mHeight = 0

    mnuViewRecorder01.Visible = IIf(1 <= vRecorderN, True, False)
    'MenuStripに対して、検索が効かない。後で調査
    mnuViewRecorder02.Visible = IIf(2 <= vRecorderN, True, False)
    mnuViewRecorder03.Visible = IIf(3 <= vRecorderN, True, False)
    mnuViewRecorder04.Visible = IIf(4 <= vRecorderN, True, False)
    mnuViewRecorder05.Visible = IIf(5 <= vRecorderN, True, False)
    mnuViewRecorder06.Visible = IIf(6 <= vRecorderN, True, False)
    mnuViewRecorder07.Visible = IIf(7 <= vRecorderN, True, False)
    mnuViewRecorder08.Visible = IIf(8 <= vRecorderN, True, False)
    mnuViewRecorder09.Visible = IIf(9 <= vRecorderN, True, False)
    mnuViewRecorder10.Visible = IIf(10 <= vRecorderN, True, False)

    For i = 1 To 10
        If i > vRecorderN Then
            gManlockInfo.RecorderInfo(i).ActionMode = clsRecorderInfo.eActionMode.esmSleep
        End If

        mName = "tsbPaperlessRecorder" + i.ToString("00")
        ToolBarMain.Items(mName).Visible = IIf(i <= vRecorderN, True, False)
        mName = "pnlRecorderInfo_No" + i.ToString("00")
        mCtrl = SearchCtrl(Me, mName)
        If IsNothing(mCtrl) = False Then
            If i <= vRecorderN Then
                mCtrl.Visible = True
                mHeight = mCtrl.Location.Y + mCtrl.Size.Height + 6
            Else
                mCtrl.Visible = False
            End If
        End If
        If i <= vRecorderN Then
            If IsNothing(pvPaperlessRecorder(i)) = True Then
                NwePaperlessRecorder(i)
            End If
            pvPaperlessRecorder(i).Show()
            pvPaperlessRecorder(i).Activate()
        Else
            If IsNothing(pvPaperlessRecorder(i)) = False Then
                pvPaperlessRecorder(i).Hide()
            End If
        End If
    Next
    If vRecorderN = 0 Then
        pnlNotDataMessage.Visible = True
    End If

```

```
pnlNotDataMessage.Top = 40
pnlNotDataMessage.Left = 5
Me.Height = 230
Me.Width = 420
Else
    pnlNotDataMessage.Visible = False
    'Me.Width = pnlRecorderInfo_No01.Width + SystemInformation.f
    'Me.Height = mHeight + SystemInformation.CaptionHeight
    mWidth = pnlRecorderInfo_No01.Width + 10
    Me.ClientSize = New System.Drawing.Size(mWidth, mHeight)
End If
Me.Activate()
End Sub

Public Function GetRecorderForm(ByVal vNo As Integer) As frmPaperlessRecorder
    If vNo >= 1 And vNo <= 10 Then
        Return pvPaperlessRecorder(vNo)
    End If
    Return Nothing
End Function

'フォーム更新
Private Sub UpdateForm()
    Me.Text = IIf(gLicense = False, "◆ライセンス無◆ ", "") & "マンロック環境監視システム[" &
        gManlockInfo.Title & "]"
    ViewRecorderCount(gManlockInfo.RecorderN)
    UpdateRecorder()
End Sub

Private Sub mnuViewUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuViewUpdate.Click
    UpdateForm()
End Sub

'フォームを最初表示処理
Private Sub frmMain_Shown(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Shown
    CheckDataFile()
    pvRearrangement = True
    tmrTask.Enabled = True
End Sub

Private Function GetViewRecN() As Integer
    Dim mRecorderN As Integer = gManlockInfo.RecorderN
    Dim mViewN As Integer = 0
    For i = 1 To mRecorderN
        If gManlockInfo.RecorderInfo(i).Visible = True Then mViewN += 1
    Next
    Return mViewN
End Function

Private Sub WindowTileArrange()
    Dim Rect As Rectangle = Screen.PrimaryScreen.WorkingArea
    Dim mRecorderN As Integer = gManlockInfo.RecorderN
    Dim mViewN As Integer = GetViewRecN()
    Dim mDevN As Integer, mWidth As Single, mHeight As Single
```

```
Dim mCols As Integer, mRows As Integer, mNo As Integer

If mViewN < 1 Or mViewN > 10 Then Exit Sub
Select Case mViewN
    Case 1, 2
        mDevN = mViewN : mCols = mViewN : mRows = 1
    Case 3, 4
        mDevN = 4 : mCols = 2 : mRows = 2
    Case 5, 6
        mDevN = 6 : mCols = 3 : mRows = 2
    Case 7, 8, 9
        mDevN = 9 : mCols = 3 : mRows = 3
    Case 10
        mDevN = 12 : mCols = 3 : mRows = 4
End Select
mWidth = Rect.Width / mCols
mHeight = Rect.Height / mRows

mNo = 0
For i = 1 To mRecorderN
    If gManlockInfo.RecorderInfo(i).Visible = True Then
        With pvPaperlessRecorder(i)
            mNo += 1
            .WindowState = FormWindowState.Normal
            .Top = Int((mNo - 1) / mCols) * mHeight
            .Left = ((mNo - 1) Mod mCols) * mWidth
            .Height = mHeight
            .Width = mWidth
            .Activate()
        End With
    End If
Next
If mViewN < mDevN Then
    mNo += 1
    Me.Top = Int((mNo - 1) / mCols) * mHeight
    Me.Left = ((mNo - 1) Mod mCols) * mWidth
End If

For i = 1 To mRecorderN
    If gManlockInfo.RecorderInfo(i).Visible = True Then
        pvPaperlessRecorder(i).Activate()
    Exit For
End If
Next

End Sub
'ウィンドウ整列(左右に並べて表示)
Private Sub mnuWindowTileVerticalTwo_Click(sender As System.Object, e As System.EventArgs) Handles mnuWindowTileVerticalTwo.Click
    WindowTileArrange()
End Sub
'ウィンドウ整列(重ねて表示)
Private Sub mnuWindowCascade_Click(sender As System.Object, e As System.EventArgs) Handles mnuWindowCascade.Click
    Dim mCaptionHeight As Integer = SystemInformation.CaptionHeight
```

```
Dim mRecorderN As Integer = gManlockInfo.RecorderN
Dim mLeft As Integer = 0, mTop As Integer = 0
Me.Left = mLeft : Me.Top = mTop
For i = 1 To mRecorderN
    With pvPaperlessRecorder(i)
        mLeft += mCaptionHeight
        mTop += mCaptionHeight
        .WindowState = FormWindowState.Normal
        .Top = mTop
        .Left = mLeft
        .Width = 800
        .Height = 600
        .Activate()
    End With
Next
End Sub

'データファイルチェック
Private Function CheckDataFile() As Boolean
    Dim mFileName As String
    mFileName = gManlockInfo.FileName
    If LibFF.ExistsFile(mFileName) = True Then
        gManlockInfo.dbFieldAdd() 'フィールドを追加
        pvDataFileOpen = gManlockInfo.dbManlockInfoIO(clsManlockInfo.eAccessMode.eamLoad)
        UpdateForm() 'フォーム更新
    Else
        pvDataFileOpen = False
    End If
    ViewRecorderCount(If(pvDataFileOpen = True, gManlockInfo.RecorderN, 0))
    ToolBarMain.Visible = pvDataFileOpen
    mnuView.Visible = pvDataFileOpen
    mnuPropertyMain.Visible = pvDataFileOpen
    Return pvDataFileOpen
End Function

'Recorder情報更新
Private Sub UpdateRecorder()
    Dim mName As String
    For mRecNo = 1 To gManlockInfo.RecorderN
        mName = "pnlRecorderInfo_No" + mRecNo.ToString("00")
        pnlRecorderInfo(Me, mName, gManlockInfo.RecorderInfo(mRecNo), True)
        pvPaperlessRecorder(mRecNo).ViewUpdate()
    Next
    WindowTileArrange()
End Sub

Private Sub mnuModeRun_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuModeRun.Click
    pvActionMode = eActionMode.esmRun
End Sub

Private Sub mnuModeStop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuModeStop.Click
    pvActionMode = eActionMode.esmStop
End Sub
```

```
Private Sub UpdateMenu()  
    Me.mnuModeRun.Checked = IIf(pvActionMode = eActionMode.esmRun, True, False)  
    Me.mnuModeStop.Checked = IIf(pvActionMode = eActionMode.esmStop, True, False)  
End Sub  
  
Private Sub tmrTask_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles tmrTask.Tick  
    Dim mName As String, mNameEye As String, mCtrl As Control  
    tslTimeNow.Text = Date.Now.ToString("yyyy:MM:dd") & vbCrLf &  
    Date.Now.ToString("H:mm:ss") + " "  
  
    If pvRearrangement = True Then  
        WindowTileArrange()  
        pvRearrangement = False  
    End If  
  
    For mRecNo = 1 To gManlockInfo.RecorderN  
        mName = "pnlRecorderInfo_No" + mRecNo.ToString("00")  
        pnlRecorderInfo(Me, mName, gManlockInfo.RecorderInfo(mRecNo))  
        pvPaperlessRecorder(mRecNo).Visible = gManlockInfo.RecorderInfo(mRecNo).Visible  
        mNameEye = mName + "_ViewSW" 'pnlRecorderInfo_No01_RecSW  
        mCtrl = SearchCtrl(Me, mNameEye)  
        If Not IsNothing(mCtrl) Then  
            With mCtrl  
                If gManlockInfo.RecorderInfo(mRecNo).Visible = True Then  
                    mCtrl.BackColor = Color.Gold  
                Else  
                    mCtrl.BackColor = Color.DimGray  
                End If  
            End With  
        End If  
    Next  
End Sub  
  
Private Sub pnlRecorderInfo_NoXX_ViewSW_Click(sender As System.Object, e As  
System.EventArgs) Handles _  
    pnlRecorderInfo_No01_ViewSW.Click, pnlRecorderInfo_No02_ViewSW.Click,  
    pnlRecorderInfo_No03_ViewSW.Click, pnlRecorderInfo_No04_ViewSW.Click,  
    pnlRecorderInfo_No05_ViewSW.Click,  
    pnlRecorderInfo_No06_ViewSW.Click, pnlRecorderInfo_No07_ViewSW.Click,  
    pnlRecorderInfo_No08_ViewSW.Click, pnlRecorderInfo_No09_ViewSW.Click,  
    pnlRecorderInfo_No10_ViewSW.Click  
  
    Dim mCtrlName As String = sender.name  
    Dim mRecNo As Integer = Integer.Parse(mCtrlName.Substring(18, 2))  
    Dim mRecInfo As clsRecorderInfo = gManlockInfo.RecorderInfo(mRecNo)  
    Dim mVisible As Boolean = mRecInfo.Visible  
    Dim mActive As Boolean = mRecInfo.Active  
  
    If mVisible = True Then  
        If mActive = True Then  
            If MsgBox("記録計(" & mRecInfo.RecorderName & ")の収録を中止して非表示にします。",  
MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then  
                mRecInfo.Active = False  
                mRecInfo.ActionMode = clsRecorderInfo.eActionMode.esmSleep  
            End If  
        End If  
    End If  
End Sub
```

```
        pvPaperlessRecorder(mRecNo).ActionMode = clsRecorderInfo.eActionMode.esmSleep
    End If
End If
mRecInfo.Visible = False
Else
    mRecInfo.Visible = True
End If
gManlockInfo.dbRecorderInfoIO(clsManlockInfo.eAccessMode.eamSave, mRecNo)
End Sub

Private Sub cmdRecSW_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles pnlRecorderInfo_No01_RecSW.Click, pnlRecorderInfo_No02_RecSW.Click,
    pnlRecorderInfo_No03_RecSW.Click, pnlRecorderInfo_No04_RecSW.Click,
    pnlRecorderInfo_No05_RecSW.Click,
        pnlRecorderInfo_No06_RecSW.Click, pnlRecorderInfo_No07_RecSW.Click,
        pnlRecorderInfo_No08_RecSW.Click, pnlRecorderInfo_No09_RecSW.Click,
        pnlRecorderInfo_No10_RecSW.Click
    Dim mRecNo As Integer, mCtrlName As String
    mCtrlName = sender.Name
    mRecNo = Integer.Parse(mCtrlName.Substring(18, 2))
    Select Case gManlockInfo.RecorderInfo(mRecNo).ActionMode
        Case clsRecorderInfo.eActionMode.esmStandBy
            gManlockInfo.RecorderInfo(mRecNo).ActionMode =
                clsRecorderInfo.eActionMode.esmRecording
            gManlockInfo.MeasureDataWriteDB(mRecNo)
        Case clsRecorderInfo.eActionMode.esmRecording
            gManlockInfo.RecorderInfo(mRecNo).ActionMode =
                clsRecorderInfo.eActionMode.esmRecording
            'clsRecorderInfo.eActionMode.esmStandBy // 収録は停止する事はない様にする
            gManlockInfo.MeasureDataWriteDB(mRecNo)
        Case Else
            Beep()
    End Select
End Sub

Private Sub mnuHelp_Click(sender As System.Object, e As System.EventArgs) Handles
mnuHelp.Click
    Help.ShowHelp(Me, Me.HelpProvider1.HelpNamespace, HelpNavigator.Topic)
End Sub

Private Sub pnlRecorderInfo_NoXX_Lamp_Click(sender As System.Object, e As
System.EventArgs) _
    Handles pnlRecorderInfo_No01_Lamp.Click, pnlRecorderInfo_No02_Lamp.Click,
    pnlRecorderInfo_No03_Lamp.Click, pnlRecorderInfo_No04_Lamp.Click,
    pnlRecorderInfo_No05_Lamp.Click,
        pnlRecorderInfo_No06_Lamp.Click, pnlRecorderInfo_No07_Lamp.Click,
        pnlRecorderInfo_No08_Lamp.Click, pnlRecorderInfo_No09_Lamp.Click,
        pnlRecorderInfo_No10_Lamp.Click
    Dim mCtrlName As String = sender.name
    Dim mRecNo As Integer = Integer.Parse(mCtrlName.Substring(18, 2))
    Dim mRecInfo As clsRecorderInfo = gManlockInfo.RecorderInfo(mRecNo)
    Dim mActive As Boolean = mRecInfo.Active
    If mActive = True Then
        If MsgBox("記録計(" & mRecInfo.RecorderName & ")を停止しますか?", MsgBoxStyle.YesNo) =
            MsgBoxResult.Yes Then
```

```
mRecInfo.Active = False
mRecInfo.ActionMode = clsRecorderInfo.eActionMode.esmSleep
pvPaperlessRecorder(mRecNo).ActionMode = clsRecorderInfo.eActionMode.esmSleep
End If
Else
mRecInfo.Active = True
pvPaperlessRecorder(mRecNo).ActionMode = clsRecorderInfo.eActionMode.esmWakeUp
End If
gManlockInfo.dbRecorderInfoIO(clsManlockInfo.eAccessMode.eamSave, mRecNo)
pvPaperlessRecorder(mRecNo).ViewUpdate()
End Sub

Private Sub chkLogOut_CheckedChanged(sender As System.Object, e As System.EventArgs)
Handles chkLogOut.CheckedChanged
gManlockInfo.LogOutFlag = chkLogOut.Checked
End Sub

End Class
```

```
Imports clslibNextforest
Imports WatchManlock.clsTypedef
Imports WatchManlock.clsManlockInfo

Public Class frmPaperlessRecorder
    Public Enum eCtrlMode
        ecmSet = 0 ' 情報をコントロールにセット
        ecmGet = 1 ' コントロールから情報を取得
    End Enum

    Private pvRecorderNo As Integer
    Private pvTcpMode As eTcpMode
    Private pvUpdateFlag As Integer
    Private LibFF As New clsFolderFile

    Private pvViewMode As eViewMode = eViewMode.evmWave ' eViewMode.evmNumber
    ' eViewMode.evmWave

    Private pvDispTime As DateTime = Date.MinValue
    Private pvViewGraphN As Integer, pvCurrentGraphNo As Integer
    Private pvTestData() As clsTestData

    Private pvGraphUpdatetime As Date ' グラフを更新した時刻
    Private pvGraphDrawLine(10) As Boolean

    Private pvRecTimeDB As Date, pvRecTimeFile As Date

    Private pvGpathName() As String = {"pnlDataGraphA_No01_Chart",
    "pnlDataGraphA_No02_Chart", "pnlDataGraphA_No03_Chart", _
    "pnlDataGraphB_No01_Chart", "pnlDataGraphB_No02_Chart",
    "pnlDataGraphB_No03_Chart",
    "chartMeasData"}

    Private pvGraphCurrentChNo As Integer

    Private pvAlarmLevel As clsChannelInfo.eAlarmLevel ' 現在のアラームレベル
    Private pvOccurTimeAttention As Date ' 最後に注意を発生した時間
    Private pvOccurTimeWarning As Date ' 最後に警報を発生した時間
    Private pvRefreshGraphCount As Integer = 0 ' 定期的にグラフをRefreshする

    Public pbClose As Boolean = False

    Dim tcp As New clsTcpIP
    Dim pvTcpOnLine As Boolean
    Dim pvTimeOut As New clsTimeOut
    Dim pvCommDataType As Integer
    Dim pvFormWidth As Integer, pvFormHeight As Integer, pvFormTop As Integer, pvFormLeft
    As Integer

    ' ハードコピーオブジェクト
    Dim objHardCopy As New NonHCopyNet.HardCopyClass

    Dim objExcel As clsExcel
```

′ 列挙体を宣言

```
Enum eViewMode
    evmWave = 0
    evmNumber = 1
    evmBar = 2
    evmMulti = 3
    evmList = 4
    evmGraph = 5
    evmAlarm = 6
    evmSettingChannel = 7
    evmSettingCommon = 8
```

End Enum

```
Enum eTcpMode
    etmNothing = 0
    etmGetMeas = 1
    etmGetComp = 2
    etmWait = 3
    etmWaitConnect = 4
```

End Enum

```
Enum eActionIO
    eioGet = 0
    eioSet = 1
```

End Enum

Public Sub New()

InitializeComponent() ′ この呼び出しはデザイナーが必要です。 InitializeComponent()
呼び出しの後で初期化を追加します。

```
pvTcpMode = eTcpMode.etmNothing
pvViewGraphN = 1
pvCurrentGraphNo = 1
For i = 1 To 10
    pvGraphDrawLine(i) = True
Next
```

```
ReDim pvTestData(20)
For i = 1 To 20
    pvTestData(i) = New clsTestData
    pvTestData(i).Initialize(, , i)
Next
pvTestData(1).Initialize(0, 1, 1)
pvTestData(2).Initialize(15, 25, 2)
pvTestData(3).Initialize(0, 7000, 3)
pvTestData(4).Initialize(-30, 70, 4)
pvTestData(11).Initialize(0, 180, 11)
pvTestData(12).Initialize(0, 0.7, 12)
```

End Sub

′ アクションモード設定

```
Public Property ActionMode() As clsRecorderInfo.eActionMode
    Get
        Return MyRecorderInfo.ActionMode
    End Get
    Set(ByVal value As clsRecorderInfo.eActionMode)
        MyRecorderInfo.ActionMode = value
```

```
Select Case MyRecorderInfo.ActionMode
  Case clsRecorderInfo.eActionMode.esmSleep
    tmrTask.Enabled = False
    tmrTest.Enabled = False      'テストデータ生成
    tmrTcpTask.Enabled = False
    tmrTcpReceive.Enabled = False
    TcpIp_Close()
    SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltStop)
  Case clsRecorderInfo.eActionMode.esmWakeUp
    tmrTask.Enabled = True
    tmrTest.Enabled = True      'テストデータ生成
    tmrTcpTask.Enabled = True
    tmrTcpReceive.Enabled = True
  Case clsRecorderInfo.eActionMode.esmStandBy,
  clsRecorderInfo.eActionMode.esmRecording
    tmrTask.Enabled = True
    tmrTest.Enabled = True      'テストデータ生成
    tmrTcpTask.Enabled = True
    tmrTcpReceive.Enabled = True
End Select
UpdateSaveButton(Me, Me.RecorderNo)
lblStatus.Text = MyRecorderInfo.ActionModeStr
End Set
End Property

'ステータス文字列取得
Public Function GetStatusStr() As String
  If MyRecorderInfo.Active = False Then
    Return "停止中"
  Else
    If chkComm.Checked = False Then 'オンライン
      Select Case pvTcpMode
        Case eTcpMode.etmWaitConnect
          Return "通信異常(接続)"
        Case eTcpMode.etmWait
          Return "通信異常(応答)"
      End Select
      Return ""
    Else 'オフラインステータス
      pnlStatus.BackColor = Color.Orange
      Return ""
    End If
  End If
End Function
Public Property RecorderNo() As Double
  Get
    Return pvRecorderNo
  End Get
  Set(ByVal value As Double)
    pvRecorderNo = value
  End Set
End Property
Public Property GraphCurrentNo() As Integer
  Get
    Return MyRecorderInfo.GraphCurrentNo
```

```
End Get
Set(ByVal value As Integer)
    MyRecorderInfo.GraphCurrentNo = value
End Set
End Property
'Myレコーダー情報
Private Function MyRecorderInfo() As clsRecorderInfo
    If pvRecorderNo >= 1 And pvRecorderNo <= 10 Then
        Return gManlockInfo.RecorderInfo(pvRecorderNo)
    Else
        Return New clsRecorderInfo '取り合えずデフォルトを返す
    End If
End Function

Private Sub btnViewMode_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewMode.Click
    If MyRecorderInfo.Active = True Then
        Select Case pvViewMode
            Case eViewMode.evmWave
                pvViewMode = eViewMode.evmNumber
            Case eViewMode.evmNumber
                pvViewMode = eViewMode.evmBar
            Case eViewMode.evmBar
                pvViewMode = eViewMode.evmMulti
            Case eViewMode.evmMulti
                pvViewMode = eViewMode.evmWave
            Case eViewMode.evmList
                pvViewMode = eViewMode.evmWave
            Case eViewMode.evmGraph
                pvViewMode = eViewMode.evmWave
            Case eViewMode.evmAlarm
                pvViewMode = eViewMode.evmWave
            Case eViewMode.evmSettingChannel
                pvViewMode = eViewMode.evmWave
            Case eViewMode.evmSettingCommon
                pvViewMode = eViewMode.evmWave
        End Select
    Else
        pvViewMode = eViewMode.evmList
    End If
    updateViewMode()
End Sub

Private Sub tcRoot_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tcRoot.SelectedIndexChanged
    Select Case tcRoot.SelectedTab.Name
        Case "ViewWave"
            pvViewMode = eViewMode.evmWave
        Case "ViewNumberA"
            pvViewMode = eViewMode.evmNumber
        Case "ViewBar"
            pvViewMode = eViewMode.evmBar
        Case "ViewMulti"
            pvViewMode = eViewMode.evmMulti
        Case "ViewList"
            pvViewMode = eViewMode.evmList
    End Select
End Sub
```

```
cobTypeList.SelectedIndex = -1
cobTypeList.SelectedIndex = 0
Case "ViewGraph"
pvViewMode = eViewMode.evmGraph
cobTypeGraph.SelectedIndex = -1
cobTypeGraph.SelectedIndex = 0
Case "ViewAlarm"
pvViewMode = eViewMode.evmAlarm
cobTypeAlarm.SelectedIndex = -1
cobTypeAlarm.SelectedIndex = 0
Case "ViewSettingChannel"
Me.Width = 1120 : Me.Height = 680
pvViewMode = eViewMode.evmSettingChannel
Case "ViewSettingCommon"
Me.Width = 730 : Me.Height = 500
pvViewMode = eViewMode.evmSettingCommon
End Select
updateViewMode()

End Sub

Private Sub tcRoot_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles tcRoot.Click
If gLicense = False And CheckTrial() = False Then
MsgBox("ライセンス登録をお願いします。" & vbLf & "使用を続けると著作権侵害になります。",
MsgBoxStyle.OkOnly, "警告<<Warning>>")
End If
UpdateRootTag()
updateViewMode()
End Sub

Private Sub UpdateRootTag()
Select Case tcRoot.SelectedIndex
Case 0
PanelNumber_Resize(Me, spltViewWave.Panel2, MyRecorderInfo.ChannelN, 2, 0,
"pnlDataNumB", pvRecorderNo)
Case 1
PanelNumber_Resize(Me, ViewNumberA, MyRecorderInfo.ChannelN, 0, 2, "pnlDataNumA",
pvRecorderNo)
Case 2
PanelBar_Resize(Me, ViewBar, MyRecorderInfo.ChannelN, "pnlDataBarA", pvRecorderNo)
Case 3
PanelBar_Resize(Me, spltViewMulti2.Panel2, MyRecorderInfo.ChannelN,
"pnlDataBarB", pvRecorderNo)
PanelNumber_Resize(Me, spltViewMulti1.Panel2, MyRecorderInfo.ChannelN, 2, 0,
"pnlDataNumC", pvRecorderNo)
End Select
Select Case tcRoot.SelectedIndex
Case 7
Me.Width = 1120 : Me.Height = 680
Case 8
Me.Width = 730 : Me.Height = 500
Case Else
Me.Size = New Size(pvFormWidth, pvFormHeight)
'Me.Top = pvFormTop
'Me.Left = pvFormLeft
```

```
Me.Location = New Point(pvFormLeft, pvFormTop)
End Select
RefreshGraph()
End Sub

Private Sub updateViewMode()
Select Case pvViewMode
Case eViewMode.evmWave
    btnViewMode.Image = WatchManlock.My.Resources.Resources.menuWave
    tcRoot.SelectedTab = ViewWave
    Me.spltViewWave.Visible = MyRecorderInfo.Active
Case eViewMode.evmNumber
    btnViewMode.Image = WatchManlock.My.Resources.Resources.menuNumber
    tcRoot.SelectedTab = ViewNumberA
    Me.ViewNumberA.Visible = MyRecorderInfo.Active
Case eViewMode.evmBar
    btnViewMode.Image = WatchManlock.My.Resources.Resources.menuBar
    tcRoot.SelectedTab = ViewBar
    Me.ViewBar.Visible = MyRecorderInfo.Active
Case eViewMode.evmMulti
    btnViewMode.Image = WatchManlock.My.Resources.Resources.menuMulti
    tcRoot.SelectedTab = ViewMulti
    Me.spltViewMulti1.Visible = MyRecorderInfo.Active
Case eViewMode.evmList
    btnViewMode.Image = WatchManlock.My.Resources.Resources.menuList
    tcRoot.SelectedTab = ViewList
Case eViewMode.evmGraph
    btnViewMode.Image = WatchManlock.My.Resources.Resources.menuGraph
    tcRoot.SelectedTab = ViewGraph
Case eViewMode.evmSettingChannel
    btnViewMode.Image = WatchManlock.My.Resources.Resources.menuSetting
    tcRoot.SelectedTab = ViewSettingChannel
Case eViewMode.evmSettingCommon
    btnViewMode.Image = WatchManlock.My.Resources.Resources.menuSetting
    tcRoot.SelectedTab = ViewSettingCommon
End Select
End Sub

Private Sub tcRoot_Resize()
' Dim mArrange As New clsArrange
With tcRoot
' .Top = pnlMenu.Height + 25
.Height = Me.ClientSize.Height - .Top
'.Left = 3
.Width = Me.ClientSize.Width - .Left
End With
End Sub

Private Sub frmPaperlessRecorder_Resize(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Resize
If Me.WindowState = FormWindowState.Minimized Then Exit Sub

If Me.WindowState = FormWindowState.Normal Then
If Me.Width < 400 Then Me.Width = 400
If Me.Height < 256 Then Me.Height = 256
```

```
End If

With tcRoot
    .Height = Me.ClientSize.Height - .Top
    .Width = Me.ClientSize.Width - .Left
End With
If tcRoot.SelectedIndex < 7 Then
    pvFormWidth = Me.Width
    pvFormHeight = Me.Height
End If
End Sub

Private Sub frmPaperlessRecorder_Move(sender As System.Object, e As System.EventArgs)
Handles MyBase.Move
    If tcRoot.SelectedIndex < 7 Then
        pvFormTop = Me.Top
        pvFormLeft = Me.Left
    End If
End Sub

Private Sub ViewNumberA_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ViewNumberA.Resize
    PanelNumber_Resize(Me, ViewNumberA, MyRecorderInfo.ChannelN, 0, 2, "pnlDataNumA",
pvRecorderNo)
End Sub

Private Sub spltViewWave_Panell1_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles spltViewWave.Panell1.Resize

    pnlDataGraphA_No01_Chart.Width = spltViewWave.Panell1.Width
    pnlDataGraphA_No01_Chart.Height = spltViewWave.Panell1.Height -
pnlDataGraphA_No01_Chart.Top

    ' PanelGraph_Resize(Me, spltViewWave.Panell1, pvViewGraphN, "pnlDataGraphA",
pvRecorderNo)
    ' pnlTimeWidth.Left = 5
    ' pnlTimeWidth.Top = spltViewWave.Panell1.Height - cobTimeWidth.Height - 5
End Sub

Private Sub spltViewMulti2_Panell1_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles spltViewMulti2.Panell1.Resize
    pnlDataGraphB_No01_Chart.Width = spltViewMulti2.Panell1.Width
    pnlDataGraphB_No01_Chart.Height = spltViewMulti2.Panell1.Height -
pnlDataGraphB_No01_Chart.Top

    ' PanelGraph_Resize(Me, spltViewMulti2.Panell1, pvViewGraphN, "pnlDataGraphB",
pvRecorderNo)
End Sub

Private Sub spltViewWave_Panel2_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles spltViewWave.Panel2.Resize
    PanelNumber_Resize(Me, spltViewWave.Panel2, MyRecorderInfo.ChannelN, 2, 0,
"pnlDataNumB", pvRecorderNo)
End Sub

Private Sub spltViewMulti1_Panel2_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles spltViewMulti1.Panel2.Resize
    PanelNumber_Resize(Me, spltViewMulti1.Panel2, MyRecorderInfo.ChannelN, 2, 0,
```

```
"pnlDataNumC", pvRecorderNo)
End Sub

Private Sub ViewBar_Resize(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ViewBar.Resize
    PanelBar_Resize(Me, ViewBar, MyRecorderInfo.ChannelN, "pnlDataBarA", pvRecorderNo)
End Sub

Private Sub spltViewMulti2_Panel2_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles spltViewMulti2.Panel2.Resize
    PanelBar_Resize(Me, spltViewMulti2.Panel2, MyRecorderInfo.ChannelN, "pnlDataBarB",
pvRecorderNo)
End Sub

Private Sub frmPaperlessRecorder_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ' AlarmInfoIO(eActionIO.eioSet)
    Me.HelpProvider1.HelpNamespace = GetAppFolder() & "WatchManlock.chm"

    Init_gridChInfo(gridChInfo)
    Init_gridRecorderRegister(gridRecorderRegister)
    Init_gridSendReport(gridAddressList)

    SetCombo_TimePitch(cboListPitch)
    SetCombo_TimePitch(cboListPitchDay)
    SetCombo_TimePitch(cboListPitchWeek)
    SetCombo_TimePitch(cboListPitchMonth)
    SetCombo_OutputSettingType(cobTypeList)

    SetCombo_TimePitch(cboGrphPitch)
    SetCombo_TimePitch(cboGrphPitchDay)
    SetCombo_TimePitch(cboGrphPitchWeek)
    SetCombo_TimePitch(cboGrphPitchMonth)
    SetCombo_OutputSettingType(cobTypeGraph)

    SetCombo_OutputSettingType(cobTypeAlarm)

    ClearRecorderRegister() : chkAutoData.Checked = False ' True

    LoadLocation(Me, "PaperlessRecorder" + Me.RecorderNo.ToString("00"))

    pvViewMode = GetRootTab(RecorderNo)

    'tmrCtrl.Enabled = True
    'tmrTcpIP.Enabled = False ' True

    LicenseCheck()
End Sub

Private Sub LicenseCheck()
    If CheckLicense() = True Or CheckTrial() = True Then
        msMainMenu.BackColor = SystemColors.Control
        pnlMenu.BackColor = SystemColors.Control
    Else
        msMainMenu.BackColor = Color.Red
        pnlMenu.BackColor = Color.Red
    End If
End Sub
```

```
End If
End Sub

Private Sub frmPaperlessRecorder_Shown(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Shown
    ' pvViewMode = IIf(MyRecorderInfo.Active = True, eViewMode.evmWave,
eViewMode.evmList)
    updateViewMode()
    ViewUpdate()
    RefreshGraph()
    ActionMode = clsRecorderInfo.eActionMode.esmWakeUp
    cobTimeWidth.SelectedIndex = 4
    cboListPitch.SelectedIndex = 0
    rdoItemAxis01.Checked = True

    If gDebug = True Then 'デバッグ時
        chkComm.Checked = True '通信をOFF
        chkAutoData.Checked = True '自動データ作成
    End If
End Sub

End Sub

Private Sub frmPaperlessRecorder_Activated(sender As System.Object, e As
System.EventArgs) Handles MyBase.Activated
    If MyRecorderInfo.Active = False Then
        If (pvViewMode >= eViewMode.evmWave And pvViewMode <= eViewMode.evmMulti) Then
            pvViewMode = eViewMode.evmList
        End If
    End If
    updateViewMode()
End Sub

Private Sub mnuViewUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuViewUpdate.Click
    ViewUpdate()
    RefreshGraph()
End Sub
'フォーム更新
Public Sub ViewUpdate()
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
    SetChannelName()
    InitGraph()
    tcRoot_Resize()
    DataIO(eCtrlMode.ecmSet)
    Me.Text = IIf(gLicense = False, "◆ライセンス無◆ ", "") & gManlockInfo.Title & "[" &
mRecInfo.RecorderName & "]"
    Me.lblRecorderName.Text = mRecInfo.RecorderName
    UpdateRootTag()
    updateViewMode()

End Sub
'チャンネル名設定
Private Sub SetChannelName()
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
    Dim mCtrl As Control
```

```
    pnlGparhDrawLineSetting.Height = 4 + mRecInfo.ChannelN * 19
    For i = 1 To 10
        If i <= mRecInfo.ChannelN Then
            gridDataList.Columns(i).Visible = True
            gridDataList.Columns(i).HeaderText = mRecInfo.ChannelInfo(i).Name
            mCtrl = SearchCtrl(Me, "chkItemDraw" & i.ToString("00"))
            If Not IsNothing(mCtrl) Then
                mCtrl.Text = mRecInfo.ChannelInfo(i).Name
                mCtrl.BackColor = mRecInfo.ChannelInfo(i).Color
            End If
        Else
            gridDataList.Columns(i).Visible = False
        End If
        tsViewItems.Items("tsbViewItem" + i.ToString("00")).Visible = IIf(i <=
            mRecInfo.ChannelN, True, False)
    Next

End Sub

Private Sub frmPaperlessRecorder_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
    SaveLocation(Me, "PaperlessRecorder" + Me.RecorderNo.ToString("00"))

    SaveRootTab(RecorderNo, pvViewMode)

    If e.CloseReason = CloseReason.UserClosing And pbClose = False Then
        Me.Hide()
        MyRecorderInfo.Visible = False
        gManlockInfo.dbRecorderInfoIO(clsManlockInfo.eAccessMode.eamSave, pvRecorderNo)
        e.Cancel = True
    End If

End Sub

Private Sub mnuHelp_Click(sender As System.Object, e As System.EventArgs) Handles
mnuHelp.Click
    Help.ShowHelp(Me, Me.HelpProvider1.HelpNamespace, HelpNavigator.Topic)
End Sub

・レジスタデータ取得
Private Function GetRegisterData(ByVal vChNo As Integer) As Single
    Dim mData As Single, mRow As Integer, mCol As Integer, buf As String
    If vChNo >= 1 And vChNo <= 20 Then
        mRow = ((vChNo - 1) Mod 10)
        mCol = ((vChNo - 1) \ 10)
        Try
            buf = gridRecorderRegister.Item(mCol, mRow).Value
            If Trim(buf) = "" Then
                mData = NullData
            Else
                mData = Single.Parse(gridRecorderRegister.Item(mCol, mRow).Value.ToString)
            End If
        Catch ex As Exception
            mData = NullData
        End Try
    End If
End Function
```

```
End Try
Else
    mData = NullData
End If
Return mData
End Function

Private Sub picRevolvingLight_Click(sender As System.Object, e As System.EventArgs)
Handles picRevolvingLight.Click
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
    Dim mActive As Boolean = mRecInfo.Active
    If mActive = True Then
        If MsgBox("記録計(" & mRecInfo.RecorderName & ")を停止しますか?", MsgBoxStyle.YesNo) =
MsgBoxResult.Yes Then
            MyRecorderInfo.Active = False
            ActionMode = clsRecorderInfo.eActionMode.esmSleep
            SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltStop)
        End If
    Else
        MyRecorderInfo.Active = True
        ActionMode = clsRecorderInfo.eActionMode.esmWakeUp
    End If
    gManlockInfo.dbRecorderInfoIO(clsManlockInfo.eAccessMode.eamSave, pvRecorderNo)
    UpdateRootTag()
    updateViewMode()
End Sub

Private Sub pnlMenu_Resize(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles pnlMenu.Resize
    Resize_pnlMenu(Me)
End Sub

Private Sub バージョン情報aToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles バージョン情報aToolStripMenuItem.Click
    Dim fVerInfo As New frmVerInfo()
    fVerInfo.ShowDialog(Me)
    fVerInfo.Dispose()
    LicenseCheck()
End Sub

'ソケットオープン
Private Function TcpIp_Connect() As Boolean
    'サーバーのホスト名とポート番号
    Dim host As String = txtNetHostName.Text.ToString
    Dim port As Integer = Convert.ToInt32(txtNetPortNo.Text)
    Dim rtn As Boolean
    pvTcpOnLine = False

    CommStatusAdd("接続開始")
    pnlStatus.BackColor = Color.Yellow
    lblStatus.Text = "通信接続中"
    Me.Refresh()
    rtn = tcp.TcpConnect2(host, port)
    If rtn = True Then
```

```

    lampTcpIP_Connect.FillColor = System.Drawing.Color.Blue
    CommStatusAdd("接続")
    pnlStatus.BackColor = gPanelBabkColor
    lblStatus.Text = "接続完了"
    pvTcpOnLine = True
    MyRecorderInfo.CommMode = clsRecorderInfo.eCommMode.ecmOnline
Else
    lampTcpIP_Connect.FillColor = System.Drawing.Color.Red
    CommStatusAdd("接続 失敗")
    pvTimeOut.Reset(gWaitTimeTCP)
    pvTcpMode = eTcpMode.etmWaitConnect
    CommStatusAdd("Wait " & gWaitTimeTCP.ToString & "秒")
    TcpIp_Close()
    MyRecorderInfo.CommMode = clsRecorderInfo.eCommMode.ecmOffline
End If
Return rtn
End Function
'ソケットクローズ
Private Sub TcpIp_Close()
    Dim rtn As Boolean
    rtn = tcp.TcpClose()
    If rtn = True Then
        lampTcpIP_Connect.FillColor = System.Drawing.Color.Gray
        CommStatusAdd("切断")
    Else
        lampTcpIP_Connect.FillColor = System.Drawing.Color.Magenta
        CommStatusAdd("切断 失敗")
    End If
    pvTcpOnLine = False
End Sub
Private Sub cmdTcpIp_Connect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdTcpIp_Connect.Click
    TcpIp_Connect()
End Sub
Private Sub cmdTcpIp_Close_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdTcpIp_Close.Click
    TcpIp_Close()
End Sub
Private Sub TcpIp_GetData(ByVal mTcpMode As eTcpMode)
    Dim sendBytes As Byte()
    '文字列をByte型配列に変換
    If mTcpMode = eTcpMode.etmGetMeas Then
        sendBytes = {&H0, &H0, &H0, &H0, &H0, &H6, &H1, &H4, &H0, &H0, &H0, &H14}
        '常に電圧10Ch転送
    ElseIf mTcpMode = eTcpMode.etmGetComp Then
        sendBytes = {&H0, &H0, &H0, &H0, &H0, &H6, &H1, &H4, &HB, &HB8, &H0, &H10}
        '常に演算8Ch転送
    Else
        Exit Sub
    End If
    'データを送信する
    Try
        If tcp.Write(sendBytes) = True Then
            lampTcpIP_Status.FillColor = System.Drawing.Color.Yellow
            pvTcpMode = mTcpMode

```

```
        pvTimeOut.Reset(3)
    Else
        lampTcpIP_Status.FillColor = System.Drawing.Color.Red
    End If
Catch
    lampTcpIP_Status.FillColor = System.Drawing.Color.Red
End Try

End Sub

Private Sub cmdTcpIp_GetDataMeas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdTcpIp_GetDataMeas.Click
    TcpIp_GetData(eTcpMode.etmGetMeas)
End Sub

Private Sub cmdTcpIp_GetDataComp_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdTcpIp_GetDataComp.Click
    TcpIp_GetData(eTcpMode.etmGetComp)
End Sub

Private Sub tmrTcpTask_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tmrTcpTask.Tick
    Dim rtn As Boolean

    If chkComm.Checked = True Or MyRecorderInfo.Active = False Then '回線オフライン
        pnlStatus.BackColor = Color.Orange
        pvTcpMode = eTcpMode.etmNothing
        txtTcp_Status.Text = "+++++++"
        LogWrite("tmrTcpTask_Tick 01 " + "+++++++")

        Exit Sub
    End If

    Select Case pvTcpMode
        Case eTcpMode.etmNothing
            txtTcp_Status.Text = "-----"
            LogWrite("tmrTcpTask_Tick 02a " + "-----")
        Case eTcpMode.etmGetMeas
            txtTcp_Status.Text = "GetMeas"
            LogWrite("tmrTcpTask_Tick 02b " + "GetMeas")
        Case eTcpMode.etmGetComp
            txtTcp_Status.Text = "GetComp"
            LogWrite("tmrTcpTask_Tick 02c " + "GetComp")
        Case eTcpMode.etmWait
            txtTcp_Status.Text = "Wait"
            LogWrite("tmrTcpTask_Tick 02d " + "Wait")
        Case Else
            txtTcp_Status.Text = "???????"
            LogWrite("tmrTcpTask_Tick 02e " + "???????" )
    End Select

    If pvTcpMode <> eTcpMode.etmNothing Then Exit Sub

    rtn = tcp.GetConnected
    pvTcpOnline = rtn
    lampTcpIP_Connect2.FillColor = IIf(rtn = True, System.Drawing.Color.RoyalBlue,
```

```
System.Drawing.Color.Silver)
If rtn = False Then TcpIp_Connect() : Exit Sub

If pvCommDataType = 0 Then
    LogWrite("tmrTcpTask_Tick 03a ")
    TcpIp_GetData(eTcpMode.etmGetMeas)
    pvCommDataType = 1
    MyRecorderInfo.CommMode = clsRecorderInfo.eCommMode.ecmOnline
    LogWrite("tmrTcpTask_Tick 04a ")
Else
    LogWrite("tmrTcpTask_Tick 03b ")
    TcpIp_GetData(eTcpMode.etmGetComp)
    pvCommDataType = 0
    MyRecorderInfo.CommMode = clsRecorderInfo.eCommMode.ecmOnline
    LogWrite("tmrTcpTask_Tick 04b ")
End If
End Sub

Private Sub tmrTcpIP_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles tmrTcpReceive.Tick
    Dim mReceiveN As Integer, mReceive() As Byte : ReDim mReceive(0)
    Dim mFloatData As Single
    mReceiveN = tcp.Receive(mReceive)

    LogWrite("tmrTcpIP_Tick 01 mReceiveN:" + mReceiveN.ToString)
    If mReceiveN <= 0 Then
        If pvTcpMode = eTcpMode.etmWaitConnect And pvTimeOut.IsTimeOut = False Then
            Dim t As Date = Date.Now
            If ((t.Millisecond / 200) Mod 2) < 1 Then
                pnlStatus.BackColor = gPanelBabkColor
            Else
                pnlStatus.BackColor = Color.Red
            End If
            lblStatus.Text = "通信異常"
            MyRecorderInfo.CommMode = clsRecorderInfo.eCommMode.ecmOffline
        End If

        Dim mClearRecorderRegister As Boolean = False
        If (pvTcpMode = eTcpMode.etmGetMeas Or pvTcpMode = eTcpMode.etmGetComp) And
pvTimeOut.IsTimeOut Then
            CommStatusAdd(If(pvTcpMode = eTcpMode.etmGetMeas, "測定値取得", "演算値取得") & "
タイムアウト")
            pvTimeOut.Reset(10)
            pvTcpMode = eTcpMode.etmWait
            CommStatusAdd("Wait 10秒")
            TcpIp_Close()
            Exit Sub
        ElseIf (pvTcpMode = eTcpMode.etmWait Or pvTcpMode = eTcpMode.etmWaitConnect) And
pvTimeOut.IsTimeOut Then
            CommStatusAdd("Wait End")
            pvTcpMode = eTcpMode.etmNothing
            TcpIp_Close()
            Exit Sub
        End If
    End If
End Sub
```

```

If CDate2(txtLastUpdate.Text).AddSeconds(gDataLatchTime) < Date.Now Then
    ClearRecorderRegister() 'レコーダーデータクリア
End If

Exit Sub
Else

    ' ソケット受信
    Dim dat As Byte() = New Byte(mReceiveN - 1) {}
    Dim mReceiveSize As Integer, mDataN As Integer, mDataN2 As Integer

    mReceiveSize = dat.GetLength(0)
    dat = mReceive
    LogWrite("tmrTcpIP_Tick 02 mReceiveSize:" + mReceiveSize.ToString)

    If mReceiveSize >= 9 Then
        mDataN = mReceive(8) / 4
        mDataN2 = (mReceiveSize - 9) / 4
        LogWrite("tmrTcpIP_Tick 03 mDataN:" + mDataN.ToString)
        LogWrite("tmrTcpIP_Tick 03 mDataN2:" + mDataN2.ToString)

        If mDataN = mDataN2 Then
            If pvTcpMode = eTcpMode.etmGetMeas Then ' And mDataN = 4 Then
                For i = 0 To mDataN - 1
                    mFloatData = cvByteToFloat(mReceive, i * 4 + 9)
                    gridRecorderRegister.Item(0, i).Value = mFloatData.ToString
                Next
                ComCount1.Text = Convert.ToInt32(Val(ComCount1.Text)) + 1
            ElseIf pvTcpMode = eTcpMode.etmGetComp Then 'And mDataN = 2 Then
                For i = 0 To mDataN - 1
                    mFloatData = cvByteToFloat(mReceive, i * 4 + 9)
                    gridRecorderRegister.Item(1, i).Value = mFloatData.ToString
                Next
                ComCount2.Text = Convert.ToInt32(Val(ComCount2.Text)) + 1
            End If
            pvTcpMode = eTcpMode.etmNothing
            lampTcpIP_Status.FillColor = System.Drawing.Color.Lime
            txtLastUpdate.Text = Date.Now.ToString("yyyy/MM/dd HH:mm:ss")
        End If
    End If
    LogWrite("tmrTcpIP_Tick 04")
End Sub

' 通信ステータス
Private Sub CommStatusAdd(ByVal vStr As String)
    Dim t As Date = Date.Now
    If lslCommStatus.Items.Count > 20 Then
        lslCommStatus.Items.RemoveAt(0)
    End If
    lslCommStatus.Items.Add(t.ToString("yyyy/MM/dd H:mm:ss") & ":" & vStr)
    lslCommStatus.TopIndex = lslCommStatus.Items.Count - 1
End Sub

Private Sub mnuViewGpraphN1_Click(ByVal sender As System.Object, ByVal e As

```

```
System.EventArgs)
    pvViewGraphN = 1
End Sub

Private Sub mnuViewGpraphN2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    pvViewGraphN = 2
End Sub

Private Sub mnuViewGpraphN3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    pvViewGraphN = 3
End Sub

Private Sub tsbViewGraph1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tsbViewGraph1.Click
    pvViewGraphN = 1
    pvCurrentGraphNo = 1
End Sub

Private Sub tsbViewGraph2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tsbViewGraph2.Click
    pvViewGraphN = 2
    pvCurrentGraphNo = 1
End Sub

Private Sub tsbViewGraph3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tsbViewGraph3.Click
    pvViewGraphN = 3
    pvCurrentGraphNo = 1
End Sub

Private Sub btnDataSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDataSave.Click

    If ActionMode = clsRecorderInfo.eActionMode.esmStandBy Then
        ActionMode = clsRecorderInfo.eActionMode.esmRecording
        gManlockInfo.MeasureDataWriteDB(Me.pvRecorderNo) '収録状態変化時のデータを記録
    ElseIf ActionMode = clsRecorderInfo.eActionMode.esmRecording Then
        ActionMode = clsRecorderInfo.eActionMode.esmRecording '
        clsRecorderInfo.eActionMode.esmStandBy //収録は停止する事はない様にする
        gManlockInfo.MeasureDataWriteDB(Me.pvRecorderNo) '収録状態変化時のデータを記録
    Else
        Beep()
    End If
End Sub

Private Sub gridChInfo_CellClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles gridChInfo.CellClick

    Dim dgv As DataGridView = CType(sender, DataGridView)
    If e.ColumnIndex < 0 Or e.RowIndex < 0 Then Exit Sub

    If dgv.Columns(e.ColumnIndex).Name = "ItemColor" Then '色変更
        ColorDlg.Color = dgv(e.ColumnIndex, e.RowIndex).Style.BackColor
        If ColorDlg.ShowDialog() = DialogResult.OK Then
            dgv(e.ColumnIndex, e.RowIndex).Style.BackColor = ColorDlg.Color
        End If
    End If
End Sub
```

```
End If
dgv.CurrentCell = Nothing
End If

End Sub

Private Sub SetChannelN(ByVal vChannelN As Integer)
    For i = 1 To 10
        gridChInfo.Rows(i - 1).Visible = IIf(i <= vChannelN, True, False)
    Next
End Sub

Private Sub DataIO(ByVal vMode As eCtrlMode)
    Dim mRecInfo As clsRecorderInfo

    mRecInfo = MyRecorderInfo()

    CtrlIO_TextBoxToStr(vMode, txtRecorderName, mRecInfo.RecorderName)
    CtrlIO_TextBoxToStr(vMode, txtNetHostName, mRecInfo.NetHostName)
    CtrlIO_TextBoxToInt(vMode, txtNetPortNo, mRecInfo.NetPortNo)

    CtrlIO_NumericToInt(vMode, numWaitTimeTCP, gWaitTimeTCP)
    CtrlIO_NumericToInt(vMode, numDataLatchTime, gDataLatchTime)

    DataIO_ChInfo(vMode)

End Sub

Private Sub DataIO_ChInfo(ByVal vMode As eCtrlMode)
    Dim mRecInfo As clsRecorderInfo
    Dim mChN As Integer, mRow As Integer, mCol As Integer, mRecNo As Integer

    mRecInfo = gManlockInfo.RecorderInfo(pvRecorderNo)

    CtrlIO_ComboBoxToInt(vMode, cboChannelN, mRecInfo.ChannelN, 1)

    mChN = mRecInfo.ChannelN
    SetChannelN(mChN)

    For i = 1 To 10
        mRow = i - 1
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).Name, gridChInfo, 0, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).Unit, gridChInfo, 1, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).DataFormat, gridChInfo, 2, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).Color, gridChInfo, 3, mRow,
            eDataType.edtColor)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).DataMin, gridChInfo, 4, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).DataMax, gridChInfo, 5, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).AlarmLo2, gridChInfo, 6, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).AlarmLo1, gridChInfo, 7, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).AlarmHi1, gridChInfo, 8, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).AlarmHi2, gridChInfo, 9, mRow)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).Register, gridChInfo, 10, mRow,
            eDataType.edtCombo)
        GridItemIO(vMode, mRecInfo.ChannelInfo(i).CoefficientA, gridChInfo, 11, mRow)
```

```
GridItemIO(vMode, mRecInfo.ChannelInfo(i).CoefficientB, gridChInfo, 12, mRow)

tsViewItems.Items("tsbViewItem" & i.ToString("00")).Text =
mRecInfo.ChannelInfo(i).Name
mRecInfo.ChannelInfo(i).CoefficientA = IIf(mRecInfo.ChannelInfo(i).CoefficientA =
0, 1, mRecInfo.ChannelInfo(i).CoefficientA) '係数Aがゼロのとき1に強制
```

```
Next
```

```
For i = 1 To 50
    mRow = i - 1
    With gManlockInfo.AddressInfo(i)
        GridItemIO(vMode, .Name, gridAddressList, 0, mRow)
        mRecNo = pvRecorderNo
        For j = 1 To 2
            mCol = j
            GridItemIO(vMode, .AlarmLevel(mRecNo, j), gridAddressList, mCol, mRow,
                eDataType.edtChkBx)
        Next
    End With
Next
```

```
End Sub
```

```
Private Sub cmdApply_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdApply.Click
    DataIO(eCtrlMode.ecmGet)
    gManlockInfo.dbRecorderInfoIO(clsManlockInfo.eAccessMode.eamSave, pvRecorderNo)
    gManlockInfo.dbAddressBookIO(clsManlockInfo.eAccessMode.eamSave)
    DataIO(eCtrlMode.ecmSet)

    ViewUpdate()
End Sub
```

```
Private Sub cboChannelN_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cboChannelN.TextChanged
    SetChannelN(Integer.Parse(cboChannelN.Text))
End Sub
```

```
Private Sub cmdSampleChInfo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSampleChInfo.Click
    gManlockInfo.RecorderInfo(pvRecorderNo).SetDefault()
    DataIO_ChInfo(eCtrlMode.ecmSet)
End Sub
```

```
Private Sub tmrTest_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles tmrTest.Tick
    Dim mData As Single, mRow As Integer, mCol As Integer
    If chkAutoData.Checked = True Then
        For i = 1 To 18
            mData = pvTestData(i).GetData()
            mRow = ((i - 1) Mod 10)
```

```
mCol = ((i - 1) \ 10)
gridRecorderRegister.Item(mCol, mRow).Value = mData.ToString("0.000")
Next
End If
End Sub

'タスクタイマー
Private Sub tmrTask_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles tmrTask.Tick
    Dim mNowTime As DateTime = DateTime.Now
    Dim mNowTimeStr As String = mNowTime.ToString("yyyy/MM/dd HH:mm:ss")
    Dim mNowTime0 As DateTime = Date.Parse(mNowTimeStr)

    LogWrite("tmrTask_Tick 01 ")
    If pvTcpMode <> eTcpMode.etmWaitConnect Then
        lblStatus.Text = MyRecorderInfo.ActionModeStr
    End If

    If pnlTime.Tag <> mNowTime.ToString("H:mm:ss") Then
        UpdatePanelTime(Me, "pnlTime")
        '        Beep()
        pvRefreshGraphCount += 1
        If pvRefreshGraphCount > gGraphRefreshInterval Then '1時間毎にグラフをRefresh
            LogWrite("tmrTask_Tick 02a RefreshGraph1")
            RefreshGraph()
            LogWrite("tmrTask_Tick 02b RefreshGraph2")
        End If
    End If

    If MyRecorderInfo.Active = False Then
        If MyRecorderInfo.RecorderNo = 1 Then
            SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltStop)
        ElseIf MyRecorderInfo.RecorderNo = 3 Then
            SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltStop)
        Else
            SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltStop)
        End If
    End If

    Exit Sub
End If

Dim mChN As Integer = MyRecorderInfo.ChannelN
For mChNo = 1 To mChN
    With MyRecorderInfo.ChannelInfo(mChNo)
        Dim mA As Single = 1, mB As Single = 0
        If IsNullData(.CoefficientA) = False And IsNullData(.CoefficientB) = False Then
            mA = IIf(.CoefficientA = 0, 1, .CoefficientA)
            mB = .CoefficientB
        End If
        .DataNow = GetRegisterData(.Register) * mA + mB '係数処理
    End With
Next

If mNowTime >= pvRecTimeDB Then
```

```

Dim mMeasTime As Date, mSpan As Integer, mRecFlag As Boolean
mSpan = gManlockInfo.SavePitchDB(mNowTime.Hour)
mMeasTime = TimeFloor(mNowTime, mSpan)
mRecFlag = IIf(ActionMode = clsRecorderInfo.eActionMode.esmRecording, True, False)
LogWrite("tmrTask_Tick 03a MeasureDataWriteDB1")
gManlockInfo.MeasureDataWriteDB(Me.pvRecorderNo, mMeasTime, mRecFlag)
LogWrite("tmrTask_Tick 03b MeasureDataWriteDB2")
pvRecTimeDB = mMeasTime.AddSeconds(mSpan)
End If
If ActionMode = clsRecorderInfo.eActionMode.esmRecording Then
If mNowTime >= pvRecTimeFile Then
Dim mMeasTime As Date, mSpan As Integer
mSpan = gManlockInfo.SavePitchFile
mMeasTime = TimeFloor(mNowTime, mSpan)
LogWrite("tmrTask_Tick 04a MeasureDataWriteCSV1")
gManlockInfo.MeasureDataWriteCSV(Me.pvRecorderNo)
LogWrite("tmrTask_Tick 04b MeasureDataWriteCSV2")
pvRecTimeFile = mMeasTime.AddSeconds(mSpan)
' RealGraphUpdate()
End If
End If

```

・グラス更新

```

If pvGraphUpdatetime.ToString("yyyyMMddhhmmss") < mNowTime.ToString("yyyyMMddhhmmss")
Then
LogWrite("tmrTask_Tick 05a RealGraphUpdate1")
RealGraphUpdate()
LogWrite("tmrTask_Tick 05b RealGraphUpdate2")
Dim mSec As Integer = GetGraphTimeWidth.TotalSeconds / 2000
mSec = IIf(mSec < 1, 1, mSec)
pvGraphUpdatetime = mNowTime.AddSeconds(mSec)
PanelNumber_Resize(Me, ViewNumberA, MyRecorderInfo.ChannelN, 0, 2, "pnlDataNumA",
pvRecorderNo)
PanelNumber_Resize(Me, splViewWave.Panel2, MyRecorderInfo.ChannelN, 2, 0,
"pnlDataNumB", pvRecorderNo)
PanelNumber_Resize(Me, splViewMulti1.Panel2, MyRecorderInfo.ChannelN, 2, 0,
"pnlDataNumC", pvRecorderNo)
LogWrite("tmrTask_Tick 05c RealGraphUpdate3")
End If

```

・アラーム処理 SSSSSSSSSSSSSSSSSSSSS

```

LogWrite("tmrTask_Tick 06a Alarm")
pnlRecorderInfo2(Me, pvRecorderNo) '数値とバーを更新
Dim mAlarmLevel As clsChannelInfo.eAlarmLevel
Dim mMessage As String
mAlarmLevel = MyRecorderInfo.GetAlarmLevel
Select Case mAlarmLevel
Case clsChannelInfo.eAlarmLevel.ealWarning
SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltRed1)
Case clsChannelInfo.eAlarmLevel.ealWarning2
SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltMagental)
Case clsChannelInfo.eAlarmLevel.ealAttention
SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltYellow1)
Case Else

```

```

    SetRevolvingLight(Me, "picRevolvingLight", eRevolvingLightType.eltBlue0)
End Select
LogWrite("tmrTask_Tick 06b Alarm")
If mAlarmLevel = clsChannelInfo.eAlarmLevel.ealWarning And _
    (pvAlarmLevel = clsChannelInfo.eAlarmLevel.ealNormal Or pvAlarmLevel =
    clsChannelInfo.eAlarmLevel.ealAttention) Then
If pvOccurTimeWarning.AddMinutes(5) < Date.Now Then
    mMessage = MyRecorderInfo.GetDataInfo
    gManlockInfo.AlarmWriteDB(pvRecorderNo, 2, mMessage)
    For i = 1 To 50
        With gManlockInfo.AddressInfo(i)
            If .AlarmLevel(pvRecorderNo, 2) = True And .MailAddress <> "" Then
                SendMail(.MailAddress, mMessage)
            End If
        End With
    Next

End If
ElseIf mAlarmLevel = clsChannelInfo.eAlarmLevel.ealAttention And pvAlarmLevel =
clsChannelInfo.eAlarmLevel.ealNormal Then
If pvOccurTimeAttention.AddMinutes(5) < Date.Now Then
    mMessage = MyRecorderInfo.GetDataInfo
    gManlockInfo.AlarmWriteDB(pvRecorderNo, 1, mMessage)
    For i = 1 To 50
        With gManlockInfo.AddressInfo(i)
            If .AlarmLevel(pvRecorderNo, 1) = True And .MailAddress <> "" Then
                SendMail(.MailAddress, mMessage)
            End If
        End With
    Next
End If
End If
pvAlarmLevel = mAlarmLevel
'アラーム処理 EEEEEEEEEEEEEEEEEEEEEEE
LogWrite("tmrTask_Tick 06c Alarm")

If MyRecorderInfo.ActionMode = clsRecorderInfo.eActionMode.esmWakeUp Then
    MyRecorderInfo.ActionMode = clsRecorderInfo.eActionMode.esmRecording
    '起動時に収録モードにする
    '          MyRecorderInfo.ActionMode = clsRecorderInfo.eActionMode.esmStandBy
End If

End Sub

Private Function SendMail(ByVal vAddress As String, ByVal vMessage As String) As Boolean
    Dim mMail As New clsSendMail

    mMail.ServerName = gManlockInfo.ServerSMTP
    mMail.PortNo = gManlockInfo.ServerPortNo
    mMail.UserID = gManlockInfo.Account
    mMail.Password = gManlockInfo.Password
    mMail.FromAddress = gManlockInfo.FromAddress
    mMail.ToAddress = vAddress
    mMail.Subject = "マンロック警報メール(RecNo:" & pvRecorderNo.ToString("00") & ")"
    mMail.Body = vMessage

```

```
Return mMail.SendMail()

End Function

Private Sub gridRecorderRegister_CellEndEdit(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles gridRecorderRegister.CellEndEdit
    Dim mRow As Integer, mCol As Integer, mNo As Integer
    Dim dgv As DataGridView = CType(sender, DataGridView)

    If e.ColumnIndex < 0 Or e.RowIndex < 0 Then Exit Sub
    mRow = e.RowIndex : mCol = e.ColumnIndex
    mNo = mCol * 10 + mRow + 1
    pvTestData(mNo).Now = dgv(e.ColumnIndex, e.RowIndex).Value

End Sub

Private Sub cmdDataCentre_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdDataCentre.Click
    DataCentre()
End Sub

Private Sub DataCentre()
    Dim mRow As Integer, mCol As Integer
    For i = 1 To 20
        pvTestData(i).Now = (pvTestData(i).Min + pvTestData(i).Max) / 2
        mRow = (i - 1) Mod 10 : mCol = (i - 1) \ 10
        gridRecorderRegister.Item(mCol, mRow).Value = pvTestData(i).Now.ToString("0.0000")
    Next
End Sub

Private Sub ClearRecorderRegister()
    Dim mRow As Integer, mCol As Integer
    For i = 1 To 20
        pvTestData(i).Now = (pvTestData(i).Min + pvTestData(i).Max) / 2
        mRow = (i - 1) Mod 10 : mCol = (i - 1) \ 10
        gridRecorderRegister.Item(mCol, mRow).Value = ""
    Next
End Sub

Private Sub ViewList_Resize(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ViewList.Resize
    Dim mHeight As Integer, mWidth As Integer
    mHeight = ViewList.Height
    mWidth = ViewList.Width
    pnlDataListSetting.Height = 52
    With gridDataList
        .Left = 5
        .Top = pnlDataListSetting.Height
        .Width = mWidth - 10
        .Height = mHeight - pnlDataListSetting.Height
    End With
End Sub

Private Sub ViewGraph_Resize(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ViewGraph.Resize
    Dim mHeight As Integer, mWidth As Integer
```

```
mHeight = ViewGraph.Height
mWidth = ViewGraph.Width
pnlGraphSetting.Height = 52
With chartMeasData
    .Left = 5
    .Top = pnlGraphSetting.Height
    .Width = mWidth - 10
    .Height = mHeight - pnlGraphSetting.Height
End With

With pnlGparhDrawLineSetting
    .Top = mHeight / 2
    .Left = mWidth - .Width - 10
End With

End Sub

Private Sub ViewAlarm_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ViewAlarm.Resize
    Dim mHeight As Integer, mWidth As Integer
    mHeight = ViewAlarm.Height
    mWidth = ViewAlarm.Width
    pnlAlarmSetting.Height = 52
    With Me.gridAlarmList
        .Left = 5
        .Top = pnlAlarmSetting.Height
        .Width = mWidth - 10
        .Height = mHeight - pnlAlarmSetting.Height
    End With

End Sub

Private Sub cmdReadListData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdReadListData.Click
    ReadListData()
End Sub

Private Sub ReadListData()
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
    Dim mDataS As Date, mDataE As Date, mIndex As Integer, mPitch As Integer

    mDataS = Date.Parse(dtpListStartD.Value.ToString("yyyy/MM/dd")) +
convTimeSpan(txtListStartT.Text)
    mDataE = Date.Parse(dtpListEndD.Value.ToString("yyyy/MM/dd")) +
convTimeSpan(txtListEndT.Text)
    mPitch = cboListPitch.SelectedIndex

    Dim mMeasData() As clsMeasureData
    mMeasData = gManlockInfo.MeasureDataReadDB(pvRecorderNo, mDataS, mDataE, mPitch, 1)
    '収録データのみ読み込む
    Dim mCount As Integer = mMeasData.Length - 1
    With gridDataList
        .Rows.Clear()
        For i = 1 To mCount
            mIndex = .Rows.Add(1)
            .Item(0, mIndex).Value = mMeasData(i).MeasDate.ToString("yyyy/MM/dd HH:mm:ss")
        Next
    End With
End Sub
```

```
For mChNo = 1 To mRecInfo.ChannelN
    .Item(mChNo, mIndex).Value =
        mRecInfo.ChannelInfo(mChNo).Format(mMeasData(i).MeasData(mChNo))
Next
.Item(11, mIndex).Value = mMeasData(i).Alarm.ToString("X8")
Next
End With

End Sub

Private Sub cmdReadAlarm_Click(sender As System.Object, e As System.EventArgs) Handles
cmdReadAlarm.Click
    ReadAlarm()
End Sub

Private Sub ReadAlarm()
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
    Dim mDataS As Date, mDataE As Date, mIndex As Integer, buf As String, mLevel As Integer

    mDataS = Date.Parse(dtpAlarmStartD.Value.ToString("yyyy/MM/dd")) +
convTimeSpan(txtAlarmStartT.Text)
    mDataE = Date.Parse(dtpAlarmEndD.Value.ToString("yyyy/MM/dd")) +
convTimeSpan(txtAlarmEndT.Text)

    Dim mAlarmInfo() As clsAlarmInfo
    mAlarmInfo = gManlockInfo.AlarmReadDB(pvRecorderNo, mDataS, mDataE)
    Dim mCount As Integer = mAlarmInfo.Length - 1
    With gridAlarmList
        .Rows.Clear()
        For i = 1 To mCount
            mIndex = .Rows.Add(1)
            .Item(0, mIndex).Value = mAlarmInfo(i).OccurTime.ToString("yyyy/MM/dd HH:mm:ss")
            mLevel = mAlarmInfo(i).Level
            buf = IIf(mLevel = 1, "注意", IIf(mLevel = 2, "警報", ""))
            .Item(1, mIndex).Value = buf
            buf = mAlarmInfo(i).Message
            buf = Replace(buf, vbCrLf, "|")
            buf = Replace(buf, " ", "") : buf = Replace(buf, " ", "")
            .Item(2, mIndex).Value = buf
        Next
    End With
End Sub

Private Sub cmdFileExport_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdFileExport.Click
    Dim libFF As New clsFolderFile
    Dim mFolder As String, mSaveFileName As String
    Dim fn As Integer, buf As String
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()

    mFolder = gDefaultFolder
    SaveFileDialog.FileName = mFolder & "\*.csv"
    SaveFileDialog.Filter = "CSVファイル (*.csv)|*.csv|すべてのファイル (*.*)|*.*"
    If SaveFileDialog.ShowDialog() = DialogResult.OK Then
        mSaveFileName = SaveFileDialog.FileName
    End If
End Sub
```

```
fn = FreeFile()
FileOpen(fn, mSaveFileName, OpenMode.Output)
buf = "Time"
For j = 1 To mRecInfo.ChannelN
    buf &= "," & mRecInfo.ChannelInfo(j).Name & "(" & mRecInfo.ChannelInfo(j).Unit &
        ")"
Next
PrintLine(fn, buf)

With gridDataList
    For i = 0 To .Rows.Count - 1
        buf = ""
        For j = 0 To mRecInfo.ChannelN '.Columns.Count - 1
            buf &= IIf(j = 0, "", ",") & .Item(j, i).Value
        Next
        PrintLine(fn, buf)
    Next
End With
FileClose(fn)
End If

End Sub

Private Sub InitGraph()
    Dim mGL As clsGraphLine
    Dim mName As String, mCtrl As Control

    For Each grphName As String In pvGpathName
        mName = grphName
        mCtrl = SearchCtrl(Me, mName)
        If Not IsNothing(mCtrl) Then
            mGL = New clsGraphLine
            mGL.SetChart(mCtrl)
            mGL.Initialize(MyRecorderInfo)
            mGL.SetAxisY2(0, 100)
        End If
    Next
End Sub

' グラフ再描画
Private Sub RefreshGraph()
    Dim mGL As clsGraphLine
    Dim mChart As System.Windows.Forms.DataVisualization.Charting.Chart
    Dim mName As String, mCtrl As Control
    Dim mTimeStart As Date, mTimeEnd As Date, t As Date
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()

    mTimeEnd = Date.Now
    mTimeStart = mTimeEnd - GetGraphTimeWidth()

    Dim mMeasData() As clsMeasureData
    mMeasData = gManlockInfo.MeasureDataReadDB(pvRecorderNo, mTimeStart, mTimeEnd)
    Dim mCount As Integer = mMeasData.Length - 1
```

```
If GraphCurrentNo < 1 Then GraphCurrentNo = 1
If GraphCurrentNo > mRecInfo.ChannelN Then GraphCurrentNo = mRecInfo.ChannelN

For mGrNo = 0 To 5 Step 3
    mName = pvGpathName(mGrNo)
    mCtrl = SearchCtrl(Me, mName)
    If Not IsNothing(mCtrl) Then
        mGL = New clsGraphLine
        mChart = mCtrl
        mGL.SetChart(mCtrl)

        mGL.SetAxisY(mRecInfo.ChannelInfo(GraphCurrentNo).DataMin,
            mRecInfo.ChannelInfo(GraphCurrentNo).DataMax,
            mRecInfo.ChannelInfo(GraphCurrentNo).DataFormat)
        'buf = mRecInfo.ChannelInfo(GraphCurrentNo).Name & "(" &
            mRecInfo.ChannelInfo(GraphCurrentNo).Unit & ")"
        'mGL.SetAxisTitle(buf, mRecInfo.ChannelInfo(GraphCurrentNo).Color)
        txtGraphUnitA.Text = "(" & mRecInfo.ChannelInfo(GraphCurrentNo).Unit & ")"
        txtGraphUnitB.Text = "(" & mRecInfo.ChannelInfo(GraphCurrentNo).Unit & ")"
        For mChNo = 1 To mRecInfo.ChannelN
            mGL.DrawLineWidth(mChNo - 1, IIf(mChNo = GraphCurrentNo, 2, 1))
        Next

        If mGrNo < 6 Then
            mGL.SetAxisX(mTimeStart.ToOADate(), mTimeEnd.ToOADate())
            mGL.DataClear()
            For mChNo = 1 To mRecInfo.ChannelN
                For i = 1 To mCount
                    t = mMeasData(i).MeasDate
                    mGL.DataAdd(mChNo - 1, t,
                        mRecInfo.ChannelInfo(mChNo).DataRate(mMeasData(i).MeasDate(mChNo)))
                Next
            Next
        End If

        '軸を表示するためDummyのデータを設定
        If mGL.GetPointN(mRecInfo.ChannelN) < 1 Then
            mGL.DataAdd(mRecInfo.ChannelN, t, 0)
        End If
    End If
Next
GraphDrawLineUpdate()
pvRefreshGraphCount = 0
End Sub

Private Sub cmdReadGrphData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdReadGrphData.Click
    ReadGrphData()
End Sub

Private Sub ReadGrphData()
    Dim mGL As New clsGraphLine
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
    Dim mDataS As Date, mDataE As Date, mPitch As Integer, t As Date
```

```

Cursor.Current = Cursors.WaitCursor '秒時計

mDataS = Date.Parse(dtpGrphStartD.Value.ToString("yyyy/MM/dd")) +
convTimeSpan(txtGrphStartT.Text)
mDataE = Date.Parse(dtpGrphEndD.Value.ToString("yyyy/MM/dd")) +
convTimeSpan(txtGrphEndT.Text)
mPitch = cboGrphPitch.SelectedIndex

Dim mMeasData() As clsMeasureData
mMeasData = gManlockInfo.MeasureDataReadDB(pvRecorderNo, mDataS, mDataE, mPitch, 1)
'収録データのみ入力
Dim mCount As Integer = mMeasData.Length - 1

mGL.SetChart(chartMeasData)
mGL.SetAxisTypeX()

mGL.SetAxisX(mDataS.AddSeconds(-1).ToOADate(), mDataE.AddSeconds(1).ToOADate())

mGL.DataClear()
For i = 1 To mCount
    t = mMeasData(i).MeasDate
    For mChNo = 1 To mRecInfo.ChannelN
        mGL.DataAdd(mChNo - 1, t,
            mRecInfo.ChannelInfo(mChNo).DataRate(mMeasData(i).MeasDate(mChNo)))
    Next
Next
'軸を表示するためDummyのデータを設定
If mGL.GetPointN(mRecInfo.ChannelN) < 1 Then mGL.DataAdd(mRecInfo.ChannelN, t, 0)

chartMeasData.Titles("描画期間").Text = "描画期間:" & mDataS.ToString("yyyy/MM/dd
HH:mm:ss") & " ~ " & mDataE.ToString("yyyy/MM/dd HH:mm:ss")
chartMeasData.Titles("タイトル").Text = gManlockInfo.Title & "(" &
MyRecorderInfo.RecorderName & ")"

SetGraphViewLine()

Cursor.Current = Cursors.Default 'マウスカーソル元に戻す
End Sub

Private Sub SetGraphViewLine(Optional ByVal vLineWidth As Integer = 1)
    Dim mGL As New clsGraphLine
    Dim mCtrlCheckBox As CheckBox, mCtrlRdoBtn As RadioButton
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
    Dim mGraphCurrentNo As Integer, mDrawLine(10) As Boolean, buf As String

    If chartMeasData.Series.Count = 0 Then Exit Sub

    mGL.SetChart(chartMeasData)

    mGraphCurrentNo = 1
    For mChNo = 1 To mRecInfo.ChannelN
        mCtrlRdoBtn = SearchCtrl(Me, "rdoItemAxis" & mChNo.ToString("00"))
        If Not IsNothing(mCtrlRdoBtn) Then
            If mCtrlRdoBtn.Checked = True Then mGraphCurrentNo = mChNo
        End If
    End For

```

```
mCtrlCheckBox = SearchCtrl(Me, "chkItemDraw" & mChNo.ToString("00"))
If Not IsNothing(mCtrlCheckBox) Then
    mDrawLine(mChNo) = mCtrlCheckBox.Checked
End If
If mChNo < chartMeasData.Series.Count Then
    chartMeasData.Series.Item(mChNo - 1).Enabled = mDrawLine(mChNo)
End If
Next

mGL.SetAxisY(mRecInfo.ChannelInfo(mGraphCurrentNo).DataMin,
mRecInfo.ChannelInfo(mGraphCurrentNo).DataMax,
mRecInfo.ChannelInfo(mGraphCurrentNo).DataFormat)
buf = mRecInfo.ChannelInfo(mGraphCurrentNo).Name & "(" &
mRecInfo.ChannelInfo(mGraphCurrentNo).Unit & ")"
mGL.SetAxisTitle(buf, mRecInfo.ChannelInfo(mGraphCurrentNo).Color)

For mChNo = 1 To mRecInfo.ChannelN
    mGL.DrawLineWidth(mChNo - 1, IIf(mChNo = mGraphCurrentNo, vLineWidth * 2,
vLineWidth))
Next

chartMeasData.ChartAreas(0).AxisX.MajorGrid.LineWidth = vLineWidth
chartMeasData.ChartAreas(0).AxisY.MajorGrid.LineWidth = vLineWidth

chartMeasData.ChartAreas(0).AxisX.LineWidth = vLineWidth * 2
chartMeasData.ChartAreas(0).AxisY.LineWidth = vLineWidth * 2

End Sub

Private Sub GraphPrint()
    Dim bakWidth As Integer = chartMeasData.Width
    Dim bakHeight As Integer = chartMeasData.Height
    Dim mPrintLineWidth As Integer
    chartMeasData.Width = 2950 / 5 : chartMeasData.Height = 2100 / 5 'A4横サイズに

    mPrintLineWidth = gPrintLineWidth
    If mPrintLineWidth < 1 Then mPrintLineWidth = 1
    If mPrintLineWidth > 20 Then mPrintLineWidth = 20

    SetGraphViewLine(mPrintLineWidth)

    With chartMeasData.Printing
        '用紙サイズを指定する
        Dim pPaperSz As System.Drawing.Printing.PaperKind =
        Printing.PaperKind.A4 'A4 サイズの定数
        Dim pkSize As System.Drawing.Printing.PaperSize
        '★ サポートされている用紙サイズの一覧を取得
        For Each pkSize In .PrintDocument.PrinterSettings.PaperSizes
            '指定の用紙サイズがサポートされているか
            If pkSize.Kind = pPaperSz Then
                '指定の用紙サイズが見つかったら用紙サイズを設定する
                .PrintDocument.DefaultPageSettings.PaperSize = pkSize
            End If
        Next
        '用紙方向横向き／縦向きで印刷
    End With
End Sub
```

```

        .PrintDocument.DefaultPageSettings.Landscape = True      ' 横向き

        .PrintDocument.DefaultPageSettings.Margins.Left = 20   '
        .PrintDocument.DefaultPageSettings.Margins.Left
        .PrintDocument.DefaultPageSettings.Margins.Right = 20  '
        .PrintDocument.DefaultPageSettings.Margins.Right
        .PrintDocument.DefaultPageSettings.Margins.Top = 20    '
        .PrintDocument.DefaultPageSettings.Margins.Top
        .PrintDocument.DefaultPageSettings.Margins.Bottom = 20 '
        .PrintDocument.DefaultPageSettings.Margins.Bottom

        ' ページ設定ダイアログボックスを表示
        '          .PageSetup()
        ' 印刷プレビューを表示
        .PrintPreview()
        ' 'True で印刷のダイアログを表示して印刷(False で直ちに印刷)
        '.Print(True)
    End With
    chartMeasData.Width = bakWidth
    chartMeasData.Height = bakHeight
    SetGraphViewLine()
End Sub

Private Sub chkItemDrawXX_CheckedChanged(sender As System.Object, e As
System.EventArgs) _
    Handles rdoItemAxis01.CheckedChanged, rdoItemAxis02.CheckedChanged,
    rdoItemAxis03.CheckedChanged, rdoItemAxis04.CheckedChanged,
    rdoItemAxis05.CheckedChanged,
        rdoItemAxis06.CheckedChanged, rdoItemAxis07.CheckedChanged,
        rdoItemAxis08.CheckedChanged, rdoItemAxis09.CheckedChanged,
        rdoItemAxis10.CheckedChanged,
        chkItemDraw01.CheckedChanged, chkItemDraw02.CheckedChanged,
        chkItemDraw03.CheckedChanged, chkItemDraw04.CheckedChanged,
        chkItemDraw05.CheckedChanged,
        chkItemDraw06.CheckedChanged, chkItemDraw07.CheckedChanged,
        chkItemDraw08.CheckedChanged, chkItemDraw09.CheckedChanged,
        chkItemDraw10.CheckedChanged
    Dim mCtrlName As String, mItemNo As Integer, mCheckBox As CheckBox
    mCtrlName = sender.Name
    If mCtrlName.Length < 13 Then Exit Sub
    mItemNo = Integer.Parse(mCtrlName.Substring(11, 2))
    If mCtrlName.Substring(0, 3) = "rdo" Then
        For i = 1 To 10
            mCheckBox = SearchCtrl(Me, "chkItemDraw" & i.ToString("00"))
            If Not IsNothing(mCheckBox) Then
                mCheckBox.Checked = IIf(i = mItemNo, True, False)
            End If
        Next
    End If
    SetGraphViewLine()
End Sub

Private Sub RealGraphUpdate()

```

```
Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
Dim mGL As clsGraphLine
Dim mName As String, mCtrl As Control
Dim mTimeStart As Date, mTimeEnd As Date

mTimeEnd = Date.Now
mTimeStart = mTimeEnd - GetGraphTimeWidth()

For mGrNo = 0 To 5 Step 3
    mName = pvGpathName(mGrNo)
    mCtrl = SearchCtrl(Me, mName)
    If Not IsNothing(mCtrl) Then
        mGL = New clsGraphLine
        mGL.SetChart(mCtrl)
        mGL.SetAxisX(mTimeStart.ToOADate(), mTimeEnd.ToOADate())
        For mChNo = 1 To mRecInfo.ChannelN
            mGL.DataAdd(mChNo - 1, mTimeEnd, mRecInfo.ChannelInfo(mChNo).DataNowRate)
            mGLOldDataDel(mChNo - 1)
        Next
    End If
Next

End Sub

Private Function GetGraphTimeWidth() As TimeSpan
    Select Case cobTimeWidth.SelectedIndex
        Case 0
            Return New TimeSpan(2, 0, 0, 0)
        Case 1
            Return New TimeSpan(1, 0, 0, 0)
        Case 2
            Return New TimeSpan(0, 12, 0, 0)
        Case 3
            Return New TimeSpan(0, 6, 0, 0)
        Case 4
            Return New TimeSpan(0, 3, 0, 0)
        Case 5
            Return New TimeSpan(0, 2, 0, 0)
        Case 6
            Return New TimeSpan(0, 1, 0, 0)
        Case 7
            Return New TimeSpan(0, 0, 30, 0)
        Case 8
            Return New TimeSpan(0, 0, 20, 0)
        Case 9
            Return New TimeSpan(0, 0, 10, 0)
        Case Else
            Return New TimeSpan(0, 6, 0, 0)
    End Select
End Function

Private Sub cobTimeWidth_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cobTimeWidth.SelectedIndexChanged
    RefreshGraph()
End Sub
```

軸変更

```
Private Sub pnlDataNum_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) _
    Handles pnlDataNumB_Ch01.Click, pnlDataNumB_Ch02.Click,
    pnlDataNumB_Ch03.Click, pnlDataNumB_Ch04.Click, pnlDataNumB_Ch05.Click,
    pnlDataNumB_Ch06.Click, pnlDataNumB_Ch07.Click,
    pnlDataNumB_Ch08.Click, pnlDataNumB_Ch09.Click,
    pnlDataNumB_Ch10.Click,
    pnlDataNumB_Ch01_Data.Click, pnlDataNumB_Ch02_Data.Click,
    pnlDataNumB_Ch03_Data.Click, pnlDataNumB_Ch04_Data.Click,
    pnlDataNumB_Ch05_Data.Click,
    pnlDataNumB_Ch06_Data.Click, pnlDataNumB_Ch07_Data.Click,
    pnlDataNumB_Ch08_Data.Click, pnlDataNumB_Ch09_Data.Click,
    pnlDataNumB_Ch10_Data.Click,
    pnlDataNumB_Ch01_Name.Click, pnlDataNumB_Ch02_Name.Click,
    pnlDataNumB_Ch03_Name.Click, pnlDataNumB_Ch04_Name.Click,
    pnlDataNumB_Ch05_Name.Click,
    pnlDataNumB_Ch06_Name.Click, pnlDataNumB_Ch07_Name.Click,
    pnlDataNumB_Ch08_Name.Click, pnlDataNumB_Ch09_Name.Click,
    pnlDataNumB_Ch10_Name.Click,
    pnlDataNumB_Ch01.Click, pnlDataNumC_Ch02.Click,
    pnlDataNumC_Ch03.Click, pnlDataNumC_Ch04.Click,
    pnlDataNumC_Ch05.Click,
    pnlDataNumC_Ch06.Click, pnlDataNumC_Ch07.Click,
    pnlDataNumC_Ch08.Click, pnlDataNumC_Ch09.Click,
    pnlDataNumC_Ch10.Click,
    pnlDataNumC_Ch01_Data.Click, pnlDataNumC_Ch02_Data.Click,
    pnlDataNumC_Ch03_Data.Click, pnlDataNumC_Ch04_Data.Click,
    pnlDataNumC_Ch05_Data.Click,
    pnlDataNumC_Ch06_Data.Click, pnlDataNumC_Ch07_Data.Click,
    pnlDataNumC_Ch08_Data.Click, pnlDataNumC_Ch09_Data.Click,
    pnlDataNumC_Ch10_Data.Click,
    pnlDataNumC_Ch01_Name.Click, pnlDataNumC_Ch02_Name.Click,
    pnlDataNumC_Ch03_Name.Click, pnlDataNumC_Ch04_Name.Click,
    pnlDataNumC_Ch05_Name.Click,
    pnlDataNumC_Ch06_Name.Click, pnlDataNumC_Ch07_Name.Click,
    pnlDataNumC_Ch08_Name.Click, pnlDataNumC_Ch09_Name.Click,
    pnlDataNumC_Ch10_Name.Click

    Dim mName As String
    mName = sender.Name
    GraphCurrentNo = Integer.Parse(mName.Substring(14, 2)) 'pnlDataNumB_Ch01_Data
    RefreshGraph()
    PanelNumber_Resize(Me, splViewWave.Panel2, MyRecorderInfo.ChannelN, 2, 0,
    "pnlDataNumB", pvRecorderNo)
    PanelNumber_Resize(Me, splViewMultil.Panel2, MyRecorderInfo.ChannelN, 2, 0,
    "pnlDataNumC", pvRecorderNo)
End Sub
```

```
Private Sub tsbViewItemXX_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) _
    Handles tsbViewItem01.Click, tsbViewItem02.Click, tsbViewItem03.Click,
    tsbViewItem04.Click, tsbViewItem05.Click,
    tsbViewItem06.Click, tsbViewItem07.Click, tsbViewItem08.Click,
```

```

        tsbViewItem09.Click, tsbViewItem10.Click

    Dim mName As String = sender.Name
    Dim ItemNo As Integer = Integer.Parse(mName.Substring(11, 2)) 'tsbViewItem01
    pvGraphDrawLine(ItemNo) = Not pvGraphDrawLine(ItemNo)
    GraphDrawLineUpdate()
End Sub

Private Sub GraphCheckDrawXX_Click(sender As System.Object, e As System.EventArgs) _
    Handles pnlDataNumB_Ch01_Draw.Click, pnlDataNumB_Ch02_Draw.Click,
    pnlDataNumB_Ch03_Draw.Click, pnlDataNumB_Ch04_Draw.Click,
    pnlDataNumB_Ch05_Draw.Click,
        pnlDataNumB_Ch06_Draw.Click, pnlDataNumB_Ch07_Draw.Click,
        pnlDataNumB_Ch08_Draw.Click, pnlDataNumB_Ch09_Draw.Click,
        pnlDataNumB_Ch10_Draw.Click,
        pnlDataNumC_Ch01_Draw.Click, pnlDataNumC_Ch03_Draw.Click,
        pnlDataNumC_Ch05_Draw.Click, pnlDataNumC_Ch07_Draw.Click,
        pnlDataNumC_Ch09_Draw.Click,
        pnlDataNumC_Ch02_Draw.Click, pnlDataNumC_Ch04_Draw.Click,
        pnlDataNumC_Ch06_Draw.Click, pnlDataNumC_Ch08_Draw.Click,
        pnlDataNumC_Ch10_Draw.Click

    Dim mName As String = sender.Name
    Dim ItemNo As Integer = Integer.Parse(mName.Substring(14, 2)) 'tsbViewItem01
    pvGraphDrawLine(ItemNo) = Not pvGraphDrawLine(ItemNo)
    GraphDrawLineUpdate()
End Sub
'グラフ描画ラインの更新
Private Sub GraphDrawLineUpdate()
    Dim mChart1 As Windows.Forms.DataVisualization.Charting.Chart
    Dim mChart2 As Windows.Forms.DataVisualization.Charting.Chart
    Dim mDrawCheck As Control, mColor As System.Drawing.Color

    mChart1 = pnlDataGraphA_No01_Chart
    mChart2 = pnlDataGraphB_No01_Chart

    For i = 1 To MyRecorderInfo.ChannelN
        mChart1.Series.Item(i - 1).Enabled = pvGraphDrawLine(i)
        mChart2.Series.Item(i - 1).Enabled = pvGraphDrawLine(i)

        mColor = IIf(pvGraphDrawLine(i) = True, MyRecorderInfo.ChannelInfo(i).Color,
            Color.White)

        mDrawCheck = SearchCtrl(Me, "pnlDataNumB_Ch" & i.ToString("00") & "_Draw")
        'pnlDataNumB_Ch01_Draw
        If Not IsNothing(mDrawCheck) Then mDrawCheck.BackColor = mColor
        mDrawCheck = SearchCtrl(Me, "pnlDataNumC_Ch" & i.ToString("00") & "_Draw")
        'pnlDataNumC_Ch01_Draw
        If Not IsNothing(mDrawCheck) Then mDrawCheck.BackColor = mColor
    Next

    tsbViewItem01.Image = IIf(pvGraphDrawLine(1) = True,
        WatchManlock.My.Resources.Resources.Blue01,
        WatchManlock.My.Resources.Resources.Yellow01)
    tsbViewItem02.Image = IIf(pvGraphDrawLine(2) = True,
        WatchManlock.My.Resources.Resources.Blue02,
        WatchManlock.My.Resources.Resources.Yellow02)

```

```
tsbViewItem03.Image = IIf(pvGraphDrawLine(3) = True,
WatchManlock.My.Resources.Resources.Blue03,
WatchManlock.My.Resources.Resources.Yellow03)
tsbViewItem04.Image = IIf(pvGraphDrawLine(4) = True,
WatchManlock.My.Resources.Resources.Blue04,
WatchManlock.My.Resources.Resources.Yellow04)
tsbViewItem05.Image = IIf(pvGraphDrawLine(5) = True,
WatchManlock.My.Resources.Resources.Blue05,
WatchManlock.My.Resources.Resources.Yellow05)
tsbViewItem06.Image = IIf(pvGraphDrawLine(6) = True,
WatchManlock.My.Resources.Resources.Blue06,
WatchManlock.My.Resources.Resources.Yellow06)
tsbViewItem07.Image = IIf(pvGraphDrawLine(7) = True,
WatchManlock.My.Resources.Resources.Blue07,
WatchManlock.My.Resources.Resources.Yellow07)
tsbViewItem08.Image = IIf(pvGraphDrawLine(8) = True,
WatchManlock.My.Resources.Resources.Blue08,
WatchManlock.My.Resources.Resources.Yellow08)
tsbViewItem09.Image = IIf(pvGraphDrawLine(9) = True,
WatchManlock.My.Resources.Resources.Blue09,
WatchManlock.My.Resources.Resources.Yellow09)
tsbViewItem10.Image = IIf(pvGraphDrawLine(10) = True,
WatchManlock.My.Resources.Resources.Blue10,
WatchManlock.My.Resources.Resources.Yellow10)
```

End Sub

```
Private Sub btnPrintScreen_Click(sender As System.Object, e As System.EventArgs)
Handles btnPrintScreen.Click, mnuPrintScreen.Click
```

```
Select Case tcRoot.SelectedTab.Name
Case "ViewList"
ListPrint()
Case "ViewGraph"
GraphPrint()
Case Else
' 用紙方向を横向きに設定
Dim objPrinter As New Printing.PrintDocument
objPrinter.DefaultPageSettings.Landscape = True

' 用紙サイズをA4に設定
For Each psz As Printing.PaperSize In objPrinter.PrinterSettings.PaperSizes
If psz.Kind = Printing.PaperKind.A4 Then
objPrinter.DefaultPageSettings.PaperSize = psz
Exit For
End If
Next
' 通常使うプリンタへのスクリーン全体のハードコピー
' objHardCopy.HardCopy(False, objPrinter)
objHardCopy.HardCopy(True, objPrinter)
End Select
End Sub
```

```
Private Sub chkComm_CheckedChanged(sender As System.Object, e As System.EventArgs)
Handles chkComm.CheckedChanged
```

```
If chkComm.Checked = True Then
    chkAutoData.Visible = True
    cmdDataCentre.Visible = True
Else
    chkAutoData.Visible = False
    cmdDataCentre.Visible = False
End If
End Sub

Private Sub pd_PrintPage(ByVal sender As Object, _
    ByVal e As System.Drawing.Printing.PrintPageEventArgs)
    '画像を読み込む
    Dim img As Image = Image.FromFile("test.jpg")
    '画像を描画する
    e.Graphics.DrawImage(img, e.MarginBounds)
    '次のページがないことを通知する
    e.HasMorePages = False
    '後始末をする
    img.Dispose()
End Sub

Private Sub cobTypeList_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cobTypeList.SelectedIndexChanged
    pnlListSettingType1.Visible = False
    pnlListSettingType2.Visible = False
    pnlListSettingType3.Visible = False
    pnlListSettingType4.Visible = False
    Select Case cobTypeList.SelectedIndex
    Case 0      'フリー
        With pnlListSettingType1
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadListData.Left = .Left + .Width + 10
            dtpListStartD.Value = DayTop(Date.Now) : txtListStartT.Text = "0:00"
            dtpListEndD.Value = DayTop(Date.Now).AddDays(1) : txtListEndT.Text = "0:00"
            cboListPitch.SelectedIndex = eTimePitch.etcMinute10 '10分間隔
            ReadListData()
        End With
    Case 1      '日報
        With pnlListSettingType2
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadListData.Left = .Left + .Width + 10
            dtpListDay.Value = DayTop(Date.Now)
            dtpListStartD.Value = dtpListDay.Value : dtpListEndD.Value =
            dtpListDay.Value.AddDays(1)
            txtListStartT.Text = "0:00" : txtListEndT.Text = "0:00"
            cboListPitchDay.SelectedIndex = -1
            cboListPitchDay.SelectedIndex = eTimePitch.etcMinute10 '10分間隔
            ReadListData()
        End With
    Case 2      '週報
        With pnlListSettingType3
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadListData.Left = .Left + .Width + 10
            dtpListWeek.Value = DayTop(Date.Now).AddDays(-6)
            dtpListStartD.Value = dtpListWeek.Value : dtpListEndD.Value =
```

```
    dtpListWeek.Value.AddDays(7)
    txtListStartT.Text = "0:00" : txtListEndT.Text = "0:00"
    cboListPitchWeek.SelectedIndex = -1
    cboListPitchWeek.SelectedIndex = eTimePitch.etpHour03 '3時間間隔
    ReadListData()
End With
Case 3 '年報
With pnlListSettingType4
    .Visible = True : .Top = 26 : .Left = 7
    cmdReadListData.Left = .Left + .Width + 10
    dtpListMonth.Value = MonthTop(Date.Now)
    dtpListStartD.Value = dtpListMonth.Value : dtpListEndD.Value =
    dtpListMonth.Value.AddMonths(1)
    txtListStartT.Text = "0:00" : txtListEndT.Text = "0:00"
    cboListPitchMonth.SelectedIndex = -1
    cboListPitchMonth.SelectedIndex = eTimePitch.etpHour03 '3時間間隔
    ReadListData()
End With
End Select

    cmdFileExport.Left = cmdReadListData.Left + cmdReadListData.Width + 10
End Sub

Private Sub cmdTypeGraph_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cobTypeGraph.SelectedIndexChanged
    pnlGraphSettingType1.Visible = False
    pnlGraphSettingType2.Visible = False
    pnlGraphSettingType3.Visible = False
    pnlGraphSettingType4.Visible = False
    Select Case cobTypeGraph.SelectedIndex
    Case 0 'フリー
        With pnlGraphSettingType1
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadGrphData.Left = .Left + .Width + 10
            dtpGrphStartD.Value = DayTop(Date.Now) : txtGrphStartT.Text = "0:00"
            dtpGrphEndD.Value = DayTop(Date.Now).AddDays(1) : txtGrphEndT.Text = "0:00"
            cboGrphPitch.SelectedIndex = eTimePitch.etpAll '全部
            ReadGrphData()
        End With
    Case 1 '日報
        With pnlGraphSettingType2
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadGrphData.Left = .Left + .Width + 10
            dtpGraphDay.Value = DayTop(Date.Now)
            dtpGrphStartD.Value = dtpGraphDay.Value : dtpGrphEndD.Value =
            dtpGraphDay.Value.AddDays(1)
            txtGrphStartT.Text = "0:00" : txtGrphEndT.Text = "0:00"
            cboGrphPitchDay.SelectedIndex = -1
            cboGrphPitchDay.SelectedIndex = eTimePitch.etpAll '全部
            ReadGrphData()
        End With
    Case 2 '週報
        With pnlGraphSettingType3
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadGrphData.Left = .Left + .Width + 10
```

```
    dtpGraphWeek.Value = DayTop(Date.Now).AddDays(-6)
    dtpGrphStartD.Value = dtpGraphWeek.Value : dtpGrphEndD.Value =
    dtpGraphWeek.Value.AddDays(7)
    txtGrphStartT.Text = "0:00" : txtGrphEndT.Text = "0:00"
    cboGrphPitchWeek.SelectedIndex = -1
    cboGrphPitchWeek.SelectedIndex = eTimePitch.etpHour01 '1時間間隔
    ReadGrphData()
End With
Case 3 '年報
With pnlGraphSettingType4
    .Visible = True : .Top = 26 : .Left = 7
    cmdReadGrphData.Left = .Left + .Width + 10
    dtpGraphMonth.Value = MonthTop(Date.Now)
    dtpGrphStartD.Value = dtpGraphMonth.Value : dtpGrphEndD.Value =
    dtpGraphMonth.Value.AddMonths(1)
    txtGrphStartT.Text = "0:00" : txtGrphEndT.Text = "0:00"
    cboGrphPitchMonth.SelectedIndex = -1
    cboGrphPitchMonth.SelectedIndex = eTimePitch.etpHour01 '1時間間隔
    ReadGrphData()
End With
End Select
End Sub

Private Sub cobTypeAlarm_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cobTypeAlarm.SelectedIndexChanged
    pnlAlarmSettingType1.Visible = False
    pnlAlarmSettingType2.Visible = False
    pnlAlarmSettingType3.Visible = False
    pnlAlarmSettingType4.Visible = False
    Select Case cobTypeAlarm.SelectedIndex
    Case 0 'フリー
        With pnlAlarmSettingType1
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadAlarm.Left = .Left + .Width + 10
            dtpAlarmStartD.Value = DayTop(Date.Now) : txtAlarmStartT.Text = "0:00"
            dtpAlarmEndD.Value = DayTop(Date.Now).AddDays(1) : txtAlarmEndT.Text = "0:00"
            ReadAlarm()
        End With
    Case 1 '日報
        With pnlAlarmSettingType2
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadAlarm.Left = .Left + .Width + 10
            dtpAlarmDay.Value = DayTop(Date.Now)
            dtpAlarmStartD.Value = dtpAlarmDay.Value : dtpAlarmEndD.Value =
            dtpAlarmDay.Value.AddDays(1)
            txtAlarmStartT.Text = "0:00" : txtAlarmEndT.Text = "0:00"
            ReadAlarm()
        End With
    Case 2 '週報
        With pnlAlarmSettingType3
            .Visible = True : .Top = 26 : .Left = 7
            cmdReadAlarm.Left = .Left + .Width + 10
            dtpAlarmWeek.Value = DayTop(Date.Now).AddDays(-6)
            dtpAlarmStartD.Value = dtpAlarmWeek.Value : dtpAlarmEndD.Value =
            dtpAlarmWeek.Value.AddDays(7)
```

```
txtAlarmStartT.Text = "0:00" : txtAlarmEndT.Text = "0:00"
ReadAlarm()
End With
Case 3      '年報
With pnlAlarmSettingType4
.Visible = True : .Top = 26 : .Left = 7
cmdReadAlarm.Left = .Left + .Width + 10
dtpAlarmMonth.Value = MonthTop(dtpAlarmStartD.Value)
dtpAlarmStartD.Value = dtpAlarmMonth.Value : dtpAlarmEndD.Value =
dtpAlarmMonth.Value.AddMonths(1)
txtAlarmStartT.Text = "0:00" : txtAlarmEndT.Text = "0:00"
ReadAlarm()
End With
End Select
End Sub

Private Sub dtpListDay_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpListDay.ValueChanged
dtpListStartD.Value = dtpListDay.Value : dtpListEndD.Value =
dtpListDay.Value.AddDays(1)
txtListStartT.Text = "00:00" : txtListEndT.Text = "00:00"
End Sub

Private Sub dtpListWeek_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpListWeek.ValueChanged
dtpListStartD.Value = dtpListWeek.Value : dtpListEndD.Value =
dtpListWeek.Value.AddDays(1)
txtListStartT.Text = "00:00" : txtListEndT.Text = "00:00"
End Sub

Private Sub dtpListMonth_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpListMonth.ValueChanged
dtpListStartD.Value = MonthTop(dtpListMonth.Value)
dtpListEndD.Value = MonthTop(dtpListMonth.Value).AddMonths(1)
txtListStartT.Text = "00:00" : txtListEndT.Text = "00:00"
End Sub

Private Sub dtpGraphDay_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpGraphDay.ValueChanged
dtpGrphStartD.Value = dtpGraphDay.Value : dtpGrphEndD.Value =
dtpGraphDay.Value.AddDays(1)
txtGrphStartT.Text = "00:00" : txtGrphEndT.Text = "00:00"
End Sub

Private Sub dtpGraphWeek_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpGraphWeek.ValueChanged
dtpGrphStartD.Value = dtpGraphWeek.Value : dtpGrphEndD.Value =
dtpGraphWeek.Value.AddDays(7)
txtGrphStartT.Text = "00:00" : txtGrphEndT.Text = "00:00"
End Sub

Private Sub dtpGraphMonth_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpGraphMonth.ValueChanged
dtpGrphStartD.Value = MonthTop(dtpGraphMonth.Value) : dtpGrphEndD.Value =
MonthTop(dtpGraphMonth.Value).AddMonths(1)
```

```
txtGrphStartT.Text = "00:00" : txtGrphEndT.Text = "00:00"
End Sub

Private Sub cboListPitchDay_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cboListPitchDay.SelectedIndexChanged
    cboListPitch.SelectedIndex = cboListPitchDay.SelectedIndex
End Sub

Private Sub cboListPitchWeek_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cboListPitchWeek.SelectedIndexChanged
    cboListPitch.SelectedIndex = cboListPitchWeek.SelectedIndex
End Sub

Private Sub cboListPitchMonth_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cboListPitchMonth.SelectedIndexChanged
    cboListPitch.SelectedIndex = cboListPitchMonth.SelectedIndex
End Sub

Private Sub cboGrphPitchDay_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cboGrphPitchDay.SelectedIndexChanged
    cboGrphPitch.SelectedIndex = cboGrphPitchDay.SelectedIndex
End Sub

Private Sub cboGrphPitchWeek_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cboGrphPitchWeek.SelectedIndexChanged
    cboGrphPitch.SelectedIndex = cboGrphPitchWeek.SelectedIndex
End Sub

Private Sub cboGrphPitchMonth_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cboGrphPitchMonth.SelectedIndexChanged
    cboGrphPitch.SelectedIndex = cboGrphPitchMonth.SelectedIndex
End Sub

Private Sub dtpAlarmDay_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpAlarmDay.ValueChanged
    dtpAlarmStartD.Value = dtpAlarmDay.Value : dtpAlarmEndD.Value =
    dtpAlarmDay.Value.AddDays(1)
    txtAlarmStartT.Text = "00:00" : txtAlarmEndT.Text = "00:00"
End Sub

Private Sub dtpAlarmWeek_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpAlarmWeek.ValueChanged
    dtpAlarmStartD.Value = dtpAlarmWeek.Value : dtpAlarmEndD.Value =
    dtpAlarmWeek.Value.AddDays(7)
    txtAlarmStartT.Text = "00:00" : txtAlarmEndT.Text = "00:00"
End Sub

Private Sub dtpAlarmMonth_ValueChanged(sender As System.Object, e As System.EventArgs)
Handles dtpAlarmMonth.ValueChanged
    dtpAlarmStartD.Value = MonthTop(dtpAlarmMonth.Value) : dtpAlarmEndD.Value =
    MonthTop(dtpAlarmMonth.Value).AddMonths(1)
    txtAlarmStartT.Text = "00:00" : txtAlarmEndT.Text = "00:00"
End Sub

Private Sub ListPrint()
    Dim mAppFolder As String, mExcelFileName As String
    Dim mRecInfo As clsRecorderInfo = MyRecorderInfo()
    Dim mFolder As String, mTempExcel As String

    If IsNothing(objExcel) = True Then
        objExcel = New clsExcel
```

```
End If

mAppFolder = GetAppFolder()
mExcelFileName = mAppFolder & "PrnOutList.xls"

If LibFF.ExistsFile(mExcelFileName) = False Then
    MsgBox("EXCELファイルを作成出来ません!", MsgBoxStyle.OkOnly, "Error (Not EXCEL)")
    Return
End If

mFolder = gDefaultFolder
mTempExcel = mFolder & "\TempExcel00.xls"
If LibFF.FileCopy(mExcelFileName, mTempExcel) = False Then
    MsgBox("新規EXCELファイルを作成出来ません!", MsgBoxStyle.OkOnly, "Create Error")
    Return
End If

objExcel.ExcelFileName = mTempExcel
objExcel.Open()

objExcel.CreateRow(gridDataList.RowCount - 1)

objExcel.SetCell(1, 2, gManlockInfo.Title & "(" & MyRecorderInfo.RecorderName & ")")

For j = 1 To mRecInfo.ChannelN
    objExcel.SetCell(3, 2 + j, mRecInfo.ChannelInfo(j).Name)
    objExcel.SetCell(4, 2 + j, mRecInfo.ChannelInfo(j).Unit)
Next

With gridDataList
    For i = 0 To .Rows.Count - 1
        For j = 0 To mRecInfo.ChannelN ' .Columns.Count - 1
            objExcel.SetCell(5 + i, 2 + j, .Item(j, i).Value)
        Next
    Next
End With

objExcel.PrintOut() '印刷
' objExcel.PrintPreview() 'プレビュー (プログラムが停止する)

objExcel.Close()
objExcel = Nothing

End Sub

Private Sub LogWrite(ByVal vMessage As String)
    If gManlockInfo.LogOutFlag = True Then
        LogFileWrite(RecorderNo, vMessage)
    End If
End Sub

Private Sub mnuImportFile_Click(sender As System.Object, e As System.EventArgs) Handles
mnuImportFile.Click
    Dim mFileName As String, mFolder As String
    OpenFileDialog.Title = "データファイルを選択してください"
    OpenFileDialog.Filter = "CSVデータファイル (*.csv) | *.CSV | すべてのファイル (*.*) | *.*"
```

```
mFileName = gDefaultFolderUSB & "\*.csv"
mFolder = LibFF.GetFolderName(mFileName)
OpenDbFileDialog.InitialDirectory = mFolder
OpenDbFileDialog.FileName = "" 'mFileName

If OpenDbFileDialog.ShowDialog() = DialogResult.OK Then
    mFileName = OpenDbFileDialog.FileName
    gDefaultFolderUSB = LibFF.GetFolderName(mFileName)
    gManlockInfo.FileImportDB(pvRecorderNo, mFileName)

    MyRecorderInfo()
End If

End Sub

Private Sub mnuClose_Click(sender As System.Object, e As System.EventArgs) Handles
mnuClose.Click
    Me.Close()
End Sub
End Class
```

```
Imports WatchManlock.clsTypedef
```

```
Public Class frmPropertyMain
```

```
Public Enum eCtrlMode
```

```
    ecmSet = 0 '情報をコントロールにセット  
    ecmGet = 1 'コントロールから情報を取得
```

```
End Enum
```

```
Private cset As String = "ISO-2022-JP" '文字コード  
Private serv As String = "smtp.lolipop.jp" 'サーバー名hostname  
Private port As Integer = 587 'ポート番号  
Private user As String = "m11117535-test" 'ユーザID  
Private pass As String = "test0123" 'パスワード  
Private fadr As String = "test@nextforest.jp" '送信元<fr@xxx.com>  
Private tadr As String = "nextforest@my.email.ne.jp" '送信先<to@xxx.com>  
Private subj As String = "タイトル test タイトルです。"  
Private body As String = "本文 test 本文です。"
```

```
Public Rtn As eDialogReturn
```

```
Private Sub cmdOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles cmdOK.Click
```

```
    DataIO(eCtrlMode.ecmGet)  
    Rtn = eDialogReturn.edrOK  
    Me.Hide()
```

```
End Sub
```

```
Private Sub cmdCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles cmdCancel.Click
```

```
    Rtn = eDialogReturn.edrCancel  
    Me.Hide()
```

```
End Sub
```

```
Private Sub frmPropertyMain_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
    With Me.cobRecorderN  
        .Items.Clear()  
        For i = 1 To 10 : .Items.Add(Str(i)) : Next  
    End With
```

```
    Init_gridSavePitchDB(Me.gridSavePitchDB)
```

```
    DataIO(eCtrlMode.ecmSet)
```

```
End Sub
```

```
Private Sub DataIO(ByVal vMode As eCtrlMode)
```

```
    CtrlIO_TextBoxToStr(vMode, txtTitle, gManlockInfo.Title)
```

```
    If gManlockInfo.RecorderN < 1 And vMode = eCtrlMode.ecmSet Then '初期時処理
```

```
        Dim Dummy As Integer = 1  
        CtrlIO_ComboBoxToInt(vMode, Me.cobRecorderN, Dummy, 1)
```

```
    Else
```

```
        If gManlockInfo.RecorderN < 1 Then  
            For i = 1 To 10
```

```
        SaveRootTab(i, frmPaperlessRecorder.eViewMode.evmSettingChannel)
    Next
End If
CtrlIO_ComboboxToInt(vMode, Me.cobRecorderN, gManlockInfo.RecorderN, 1)
End If

For i = 0 To 23
    GridItemIO(vMode, gManlockInfo.SavePitchDB(i), gridSavePitchDB, i, 0)
Next

CtrlIO_TextBoxToStr(vMode, txtServerSMTP, gManlockInfo.ServerSMTP)
CtrlIO_TextBoxToStr(vMode, txtServerPortNo, gManlockInfo.ServerPortNo)
CtrlIO_TextBoxToStr(vMode, txtAccount, gManlockInfo.Account)
CtrlIO_TextBoxToStr(vMode, txtPassword, gManlockInfo.Password)
CtrlIO_TextBoxToStr(vMode, txtFromAddress, gManlockInfo.FromAddress)

CtrlIO_TextBoxToStr(vMode, txtPrintLineWidth, gPrintLineWidth)

gGraphRefreshInterval = IIf(gGraphRefreshInterval < 60, 60, gGraphRefreshInterval)
gGraphRefreshInterval = IIf(gGraphRefreshInterval > 3600, 3600, gGraphRefreshInterval)
CtrlIO_TextBoxToStr(vMode, txtGraphRefreshInterval, gGraphRefreshInterval)
gGraphRefreshInterval = IIf(gGraphRefreshInterval < 60, 60, gGraphRefreshInterval)
gGraphRefreshInterval = IIf(gGraphRefreshInterval > 3600, 3600, gGraphRefreshInterval)

Dim mCtrl As Control
For i = 1 To gManlockInfo.RecorderN
    mCtrl = SearchCtrl(Me, "chkActiveNo" + i.ToString("00"))
    If Not IsNothing(mCtrl) Then
        CtrlIO_CheckBoxToBool(vMode, mCtrl, gManlockInfo.RecorderInfo(i).Active)
    End If
Next

End Sub

Private Sub cmdSendTest_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSendTest.Click

    Dim mMail As New clsSendMail

    mMail.ServerName = txtServerSMTP.Text
    mMail.PortNo = txtServerPortNo.Text
    mMail.UserID = txtAccount.Text
    mMail.Password = txtPassword.Text
    mMail.FromAddress = txtFromAddress.Text
    mMail.ToAddress = txtToAddress.Text
    mMail.Subject = "警報送信用テストメール"
    mMail.Body = "マンロック警報" & vbCrLf _
        & "テストメール" & vbCrLf _
        & "TestMail" & vbCrLf _
        & "注意警報テストメール" & vbCrLf

    Dim rtn As Boolean
    rtn = mMail.SendMail()

    If rtn = True Then
```

```
MsgBox("◎テストメール送信成功◎", MsgBoxStyle.OkOnly, "テストメール確認")
Else
MsgBox("×テストメール送信失敗×", MsgBoxStyle.OkOnly, "テストメール確認")
End If

End Sub

Private Sub cobRecorderN_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cobRecorderN.SelectedIndexChanged
Dim mRecN As Integer, mName As String, mCtrl As Control
mRecN = cobRecorderN.SelectedIndex + 1
For i = 1 To 10
mName = "chkActiveNo" + i.ToString("00")
mCtrl = SearchCtrl(Me, mName)
If Not IsNothing(mCtrl) Then
If i <= mRecN Then
mCtrl.Visible = True
Else
mCtrl.Visible = False
End If
End If
Next
End Sub

Private Sub txtPrintLineWidth_Validating(sender As System.Object, e As
System.ComponentModel.CancelEventArgs) Handles txtPrintLineWidth.Validating
If Val(txtPrintLineWidth.Text) < 1 Or Val(txtPrintLineWidth.Text) > 20 Then
Beep()
e.Cancel = True
End If
End Sub

End Class
```

```
Public Class frmPropertyPaperlessRecorder
```

```
    Private Sub frmPropertyPaperlessRecorder_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    End Sub
```

```
End Class
```

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class frmTemp
    Inherits System.Windows.Forms.Form

    'フォームがコンポーネントの一覧をクリーンアップするために dispose をオーバーライドします。
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Windows フォーム デザイナーが必要です。
    Private components As System.ComponentModel.IContainer

    'メモ: 以下のプロシージャは Windows フォーム デザイナーが必要です。
    'Windows フォーム デザイナーを使用して変更できます。
    'コード エディターを使って変更しないでください。
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.pnlRecorderInfo_No10 = New System.Windows.Forms.Panel()
        Me.TextBox2 = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_RecSW = New System.Windows.Forms.Button()
        Me.Button1 = New System.Windows.Forms.Button()
        Me.pnlRecorderInfo_No10_Ch10_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch10_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch09_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch09_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch08_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch08_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch07_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch07_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch06_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch06_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch05_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch05_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch04_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch04_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch03_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch03_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch02_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch02_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Ch01_ItemName = New System.Windows.Forms.Label()
        Me.pnlRecorderInfo_No10_Ch01_ItemData = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_RecorderName = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No10_Lamp = New System.Windows.Forms.PictureBox()
        Me.pnlRecorderInfo_No09 = New System.Windows.Forms.Panel()
        Me.TextBox1 = New System.Windows.Forms.TextBox()
        Me.pnlRecorderInfo_No09_RecSW = New System.Windows.Forms.Button()
        Me.btnPaperlessRecorder01 = New System.Windows.Forms.Button()
        Me.pnlRecorderInfo_No09_Ch10_ItemName = New System.Windows.Forms.Label()
    End Sub
End Class
```

```
Me.pnlRecorderInfo_No09_Ch10_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch09_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch09_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch08_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch08_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch07_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch07_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch06_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch06_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch05_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch05_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch04_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch04_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch03_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch03_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch02_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch02_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Ch01_ItemName = New System.Windows.Forms.Label()
Me.pnlRecorderInfo_No09_Ch01_ItemData = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_RecorderName = New System.Windows.Forms.TextBox()
Me.pnlRecorderInfo_No09_Lamp = New System.Windows.Forms.PictureBox()
Me.pnlRecorderInfo_No10.SuspendLayout()
CType(Me.pnlRecorderInfo_No10_Lamp,
System.ComponentModel.ISupportInitialize).BeginInit()
Me.pnlRecorderInfo_No09.SuspendLayout()
CType(Me.pnlRecorderInfo_No09_Lamp,
System.ComponentModel.ISupportInitialize).BeginInit()
Me.SuspendLayout()
'
'pnlRecorderInfo_No10
'
Me.pnlRecorderInfo_No10.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
Me.pnlRecorderInfo_No10.Controls.Add(Me.TextBox2)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_RecSW)
Me.pnlRecorderInfo_No10.Controls.Add(Me.Button1)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch10_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch10_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch09_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch09_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch08_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch08_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch07_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch07_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch06_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch06_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch05_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch05_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch04_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch04_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch03_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch03_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch02_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch02_ItemData)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch01_ItemName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Ch01_ItemData)
```

```
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_RecorderName)
Me.pnlRecorderInfo_No10.Controls.Add(Me.pnlRecorderInfo_No10_Lamp)
Me.pnlRecorderInfo_No10.Location = New System.Drawing.Point(198, 326)
Me.pnlRecorderInfo_No10.Name = "pnlRecorderInfo_No10"
Me.pnlRecorderInfo_No10.Size = New System.Drawing.Size(765, 80)
Me.pnlRecorderInfo_No10.TabIndex = 40
'
'TextBox2
'
Me.TextBox2.BackColor = System.Drawing.SystemColors.Window
Me.TextBox2.Font = New System.Drawing.Font("MS UI Gothic", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
Me.TextBox2.Location = New System.Drawing.Point(415, 4)
Me.TextBox2.Name = "TextBox2"
Me.TextBox2.ReadOnly = True
Me.TextBox2.Size = New System.Drawing.Size(340, 23)
Me.TextBox2.TabIndex = 38
Me.TextBox2.Text = "ステータス"
'
'pnlRecorderInfo_No10_RecSW
'
Me.pnlRecorderInfo_No10_RecSW.BackgroundImage =
Global.WatchManlock.My.Resources.Resources.menuSaveOff
Me.pnlRecorderInfo_No10_RecSW.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch
Me.pnlRecorderInfo_No10_RecSW.Location = New System.Drawing.Point(377, 1)
Me.pnlRecorderInfo_No10_RecSW.Name = "pnlRecorderInfo_No10_RecSW"
Me.pnlRecorderInfo_No10_RecSW.Size = New System.Drawing.Size(32, 32)
Me.pnlRecorderInfo_No10_RecSW.TabIndex = 38
Me.pnlRecorderInfo_No10_RecSW.UseVisualStyleBackColor = True
'
'Button1
'
Me.Button1.Image = Global.WatchManlock.My.Resources.Resources.PaperlessRecorder02
Me.Button1.Location = New System.Drawing.Point(5, 11)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(42, 45)
Me.Button1.TabIndex = 46
Me.Button1.UseVisualStyleBackColor = True
'
'pnlRecorderInfo_No10_Ch10_ItemName
'
Me.pnlRecorderInfo_No10_Ch10_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch10_ItemName.Location = New System.Drawing.Point(695, 38)
Me.pnlRecorderInfo_No10_Ch10_ItemName.Name = "pnlRecorderInfo_No10_Ch10_ItemName"
Me.pnlRecorderInfo_No10_Ch10_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch10_ItemName.TabIndex = 68
Me.pnlRecorderInfo_No10_Ch10_ItemName.Text = "Item10"
'
'pnlRecorderInfo_No10_Ch10_ItemData
'
Me.pnlRecorderInfo_No10_Ch10_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch10_ItemData.Location = New System.Drawing.Point(697, 53)
Me.pnlRecorderInfo_No10_Ch10_ItemData.Name = "pnlRecorderInfo_No10_Ch10_ItemData"
Me.pnlRecorderInfo_No10_Ch10_ItemData.ReadOnly = True
```

```
Me.pnlRecorderInfo_No10_Ch10_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch10_ItemData.TabIndex = 67
'
'pnlRecorderInfo_No10_Ch09_ItemName
'
Me.pnlRecorderInfo_No10_Ch09_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch09_ItemName.Location = New System.Drawing.Point(631, 38)
Me.pnlRecorderInfo_No10_Ch09_ItemName.Name = "pnlRecorderInfo_No10_Ch09_ItemName"
Me.pnlRecorderInfo_No10_Ch09_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch09_ItemName.TabIndex = 66
Me.pnlRecorderInfo_No10_Ch09_ItemName.Text = "Item09"
'
'pnlRecorderInfo_No10_Ch09_ItemData
'
Me.pnlRecorderInfo_No10_Ch09_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch09_ItemData.Location = New System.Drawing.Point(633, 53)
Me.pnlRecorderInfo_No10_Ch09_ItemData.Name = "pnlRecorderInfo_No10_Ch09_ItemData"
Me.pnlRecorderInfo_No10_Ch09_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch09_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch09_ItemData.TabIndex = 65
'
'pnlRecorderInfo_No10_Ch08_ItemName
'
Me.pnlRecorderInfo_No10_Ch08_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch08_ItemName.Location = New System.Drawing.Point(567, 38)
Me.pnlRecorderInfo_No10_Ch08_ItemName.Name = "pnlRecorderInfo_No10_Ch08_ItemName"
Me.pnlRecorderInfo_No10_Ch08_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch08_ItemName.TabIndex = 64
Me.pnlRecorderInfo_No10_Ch08_ItemName.Text = "Item08"
'
'pnlRecorderInfo_No10_Ch08_ItemData
'
Me.pnlRecorderInfo_No10_Ch08_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch08_ItemData.Location = New System.Drawing.Point(569, 53)
Me.pnlRecorderInfo_No10_Ch08_ItemData.Name = "pnlRecorderInfo_No10_Ch08_ItemData"
Me.pnlRecorderInfo_No10_Ch08_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch08_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch08_ItemData.TabIndex = 63
'
'pnlRecorderInfo_No10_Ch07_ItemName
'
Me.pnlRecorderInfo_No10_Ch07_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch07_ItemName.Location = New System.Drawing.Point(503, 38)
Me.pnlRecorderInfo_No10_Ch07_ItemName.Name = "pnlRecorderInfo_No10_Ch07_ItemName"
Me.pnlRecorderInfo_No10_Ch07_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch07_ItemName.TabIndex = 62
Me.pnlRecorderInfo_No10_Ch07_ItemName.Text = "Item07"
'
'pnlRecorderInfo_No10_Ch07_ItemData
'
Me.pnlRecorderInfo_No10_Ch07_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch07_ItemData.Location = New System.Drawing.Point(505, 53)
Me.pnlRecorderInfo_No10_Ch07_ItemData.Name = "pnlRecorderInfo_No10_Ch07_ItemData"
Me.pnlRecorderInfo_No10_Ch07_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch07_ItemData.Size = New System.Drawing.Size(58, 19)
```

```
Me.pnlRecorderInfo_No10_Ch07_ItemData.TabIndex = 61
'
'pnlRecorderInfo_No10_Ch06_ItemName
'
Me.pnlRecorderInfo_No10_Ch06_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch06_ItemName.Location = New System.Drawing.Point(439, 38)
Me.pnlRecorderInfo_No10_Ch06_ItemName.Name = "pnlRecorderInfo_No10_Ch06_ItemName"
Me.pnlRecorderInfo_No10_Ch06_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch06_ItemName.TabIndex = 60
Me.pnlRecorderInfo_No10_Ch06_ItemName.Text = "Item06"
'
'pnlRecorderInfo_No10_Ch06_ItemData
'
Me.pnlRecorderInfo_No10_Ch06_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch06_ItemData.Location = New System.Drawing.Point(441, 53)
Me.pnlRecorderInfo_No10_Ch06_ItemData.Name = "pnlRecorderInfo_No10_Ch06_ItemData"
Me.pnlRecorderInfo_No10_Ch06_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch06_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch06_ItemData.TabIndex = 59
'
'pnlRecorderInfo_No10_Ch05_ItemName
'
Me.pnlRecorderInfo_No10_Ch05_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch05_ItemName.Location = New System.Drawing.Point(375, 38)
Me.pnlRecorderInfo_No10_Ch05_ItemName.Name = "pnlRecorderInfo_No10_Ch05_ItemName"
Me.pnlRecorderInfo_No10_Ch05_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch05_ItemName.TabIndex = 58
Me.pnlRecorderInfo_No10_Ch05_ItemName.Text = "Item05"
'
'pnlRecorderInfo_No10_Ch05_ItemData
'
Me.pnlRecorderInfo_No10_Ch05_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch05_ItemData.Location = New System.Drawing.Point(377, 53)
Me.pnlRecorderInfo_No10_Ch05_ItemData.Name = "pnlRecorderInfo_No10_Ch05_ItemData"
Me.pnlRecorderInfo_No10_Ch05_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch05_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch05_ItemData.TabIndex = 57
'
'pnlRecorderInfo_No10_Ch04_ItemName
'
Me.pnlRecorderInfo_No10_Ch04_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch04_ItemName.Location = New System.Drawing.Point(311, 38)
Me.pnlRecorderInfo_No10_Ch04_ItemName.Name = "pnlRecorderInfo_No10_Ch04_ItemName"
Me.pnlRecorderInfo_No10_Ch04_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch04_ItemName.TabIndex = 56
Me.pnlRecorderInfo_No10_Ch04_ItemName.Text = "Item04"
'
'pnlRecorderInfo_No10_Ch04_ItemData
'
Me.pnlRecorderInfo_No10_Ch04_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch04_ItemData.Location = New System.Drawing.Point(313, 53)
Me.pnlRecorderInfo_No10_Ch04_ItemData.Name = "pnlRecorderInfo_No10_Ch04_ItemData"
Me.pnlRecorderInfo_No10_Ch04_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch04_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch04_ItemData.TabIndex = 55
```

```
'pnlRecorderInfo_No10_Ch03_ItemName
'
Me.pnlRecorderInfo_No10_Ch03_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch03_ItemName.Location = New System.Drawing.Point(247, 38)
Me.pnlRecorderInfo_No10_Ch03_ItemName.Name = "pnlRecorderInfo_No10_Ch03_ItemName"
Me.pnlRecorderInfo_No10_Ch03_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch03_ItemName.TabIndex = 54
Me.pnlRecorderInfo_No10_Ch03_ItemName.Text = "Item03"
'
'pnlRecorderInfo_No10_Ch03_ItemData
'
Me.pnlRecorderInfo_No10_Ch03_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch03_ItemData.Location = New System.Drawing.Point(249, 53)
Me.pnlRecorderInfo_No10_Ch03_ItemData.Name = "pnlRecorderInfo_No10_Ch03_ItemData"
Me.pnlRecorderInfo_No10_Ch03_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch03_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch03_ItemData.TabIndex = 53
'
'pnlRecorderInfo_No10_Ch02_ItemName
'
Me.pnlRecorderInfo_No10_Ch02_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch02_ItemName.Location = New System.Drawing.Point(183, 38)
Me.pnlRecorderInfo_No10_Ch02_ItemName.Name = "pnlRecorderInfo_No10_Ch02_ItemName"
Me.pnlRecorderInfo_No10_Ch02_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch02_ItemName.TabIndex = 52
Me.pnlRecorderInfo_No10_Ch02_ItemName.Text = "Item02"
'
'pnlRecorderInfo_No10_Ch02_ItemData
'
Me.pnlRecorderInfo_No10_Ch02_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch02_ItemData.Location = New System.Drawing.Point(185, 53)
Me.pnlRecorderInfo_No10_Ch02_ItemData.Name = "pnlRecorderInfo_No10_Ch02_ItemData"
Me.pnlRecorderInfo_No10_Ch02_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch02_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch02_ItemData.TabIndex = 51
'
'pnlRecorderInfo_No10_Ch01_ItemName
'
Me.pnlRecorderInfo_No10_Ch01_ItemName.AutoSize = True
Me.pnlRecorderInfo_No10_Ch01_ItemName.Location = New System.Drawing.Point(119, 38)
Me.pnlRecorderInfo_No10_Ch01_ItemName.Name = "pnlRecorderInfo_No10_Ch01_ItemName"
Me.pnlRecorderInfo_No10_Ch01_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No10_Ch01_ItemName.TabIndex = 50
Me.pnlRecorderInfo_No10_Ch01_ItemName.Text = "Item01"
'
'pnlRecorderInfo_No10_Ch01_ItemData
'
Me.pnlRecorderInfo_No10_Ch01_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_Ch01_ItemData.Location = New System.Drawing.Point(121, 53)
Me.pnlRecorderInfo_No10_Ch01_ItemData.Name = "pnlRecorderInfo_No10_Ch01_ItemData"
Me.pnlRecorderInfo_No10_Ch01_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No10_Ch01_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No10_Ch01_ItemData.TabIndex = 47
'
```

```
'pnlRecorderInfo_No10_RecorderName
'
Me.pnlRecorderInfo_No10_RecorderName.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No10_RecorderName.Font = New System.Drawing.Font("MS UI Gothic",
12.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(128, Byte))
Me.pnlRecorderInfo_No10_RecorderName.Location = New System.Drawing.Point(121, 4)
Me.pnlRecorderInfo_No10_RecorderName.Name = "pnlRecorderInfo_No10_RecorderName"
Me.pnlRecorderInfo_No10_RecorderName.ReadOnly = True
Me.pnlRecorderInfo_No10_RecorderName.Size = New System.Drawing.Size(250, 23)
Me.pnlRecorderInfo_No10_RecorderName.TabIndex = 37
Me.pnlRecorderInfo_No10_RecorderName.Text = "レコーダー名称"
'
'pnlRecorderInfo_No10_Lamp
'
Me.pnlRecorderInfo_No10_Lamp.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
Me.pnlRecorderInfo_No10_Lamp.Image =
Global.WatchManlock.My.Resources.Resources.lightBlue0
Me.pnlRecorderInfo_No10_Lamp.Location = New System.Drawing.Point(53, 4)
Me.pnlRecorderInfo_No10_Lamp.Name = "pnlRecorderInfo_No10_Lamp"
Me.pnlRecorderInfo_No10_Lamp.Size = New System.Drawing.Size(64, 65)
Me.pnlRecorderInfo_No10_Lamp.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
Me.pnlRecorderInfo_No10_Lamp.TabIndex = 37
Me.pnlRecorderInfo_No10_Lamp.TabStop = False
'
'pnlRecorderInfo_No09
'
Me.pnlRecorderInfo_No09.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
Me.pnlRecorderInfo_No09.Controls.Add(Me.TextBox1)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_RecSW)
Me.pnlRecorderInfo_No09.Controls.Add(Me.btnPaperlessRecorder01)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch10_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch10_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch09_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch09_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch08_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch08_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch07_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch07_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch06_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch06_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch05_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch05_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch04_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch04_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch03_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch03_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch02_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch02_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch01_ItemName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Ch01_ItemData)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_RecorderName)
Me.pnlRecorderInfo_No09.Controls.Add(Me.pnlRecorderInfo_No09_Lamp)
Me.pnlRecorderInfo_No09.Location = New System.Drawing.Point(198, 240)
```

```
Me.pnlRecorderInfo_No09.Name = "pnlRecorderInfo_No09"
Me.pnlRecorderInfo_No09.Size = New System.Drawing.Size(765, 80)
Me.pnlRecorderInfo_No09.TabIndex = 39
'
' TextBox1
'
Me.TextBox1.BackColor = System.Drawing.SystemColors.Window
Me.TextBox1.Font = New System.Drawing.Font("MS UI Gothic", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
Me.TextBox1.Location = New System.Drawing.Point(415, 4)
Me.TextBox1.Name = "TextBox1"
Me.TextBox1.ReadOnly = True
Me.TextBox1.Size = New System.Drawing.Size(340, 23)
Me.TextBox1.TabIndex = 38
Me.TextBox1.Text = "ステータス"
'
' pnlRecorderInfo_No09_RecSW
'
Me.pnlRecorderInfo_No09_RecSW.BackgroundImage =
Global.WatchManlock.My.Resources.Resources.menuSaveOff
Me.pnlRecorderInfo_No09_RecSW.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch
Me.pnlRecorderInfo_No09_RecSW.Location = New System.Drawing.Point(377, 1)
Me.pnlRecorderInfo_No09_RecSW.Name = "pnlRecorderInfo_No09_RecSW"
Me.pnlRecorderInfo_No09_RecSW.Size = New System.Drawing.Size(32, 32)
Me.pnlRecorderInfo_No09_RecSW.TabIndex = 38
Me.pnlRecorderInfo_No09_RecSW.UseVisualStyleBackColor = True
'
' btnPaperlessRecorder01
'
Me.btnPaperlessRecorder01.Image =
Global.WatchManlock.My.Resources.Resources.PaperlessRecorder01
Me.btnPaperlessRecorder01.Location = New System.Drawing.Point(5, 11)
Me.btnPaperlessRecorder01.Name = "btnPaperlessRecorder01"
Me.btnPaperlessRecorder01.Size = New System.Drawing.Size(42, 45)
Me.btnPaperlessRecorder01.TabIndex = 46
Me.btnPaperlessRecorder01.UseVisualStyleBackColor = True
'
' pnlRecorderInfo_No09_Ch10_ItemName
'
Me.pnlRecorderInfo_No09_Ch10_ItemName.AutoSize = True
Me.pnlRecorderInfo_No09_Ch10_ItemName.Location = New System.Drawing.Point(695, 38)
Me.pnlRecorderInfo_No09_Ch10_ItemName.Name = "pnlRecorderInfo_No09_Ch10_ItemName"
Me.pnlRecorderInfo_No09_Ch10_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No09_Ch10_ItemName.TabIndex = 68
Me.pnlRecorderInfo_No09_Ch10_ItemName.Text = "Item10"
'
' pnlRecorderInfo_No09_Ch10_ItemData
'
Me.pnlRecorderInfo_No09_Ch10_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No09_Ch10_ItemData.Location = New System.Drawing.Point(697, 53)
Me.pnlRecorderInfo_No09_Ch10_ItemData.Name = "pnlRecorderInfo_No09_Ch10_ItemData"
Me.pnlRecorderInfo_No09_Ch10_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No09_Ch10_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No09_Ch10_ItemData.TabIndex = 67
```

```
'pnlRecorderInfo_No09_Ch09_ItemName
```

```
Me.pnlRecorderInfo_No09_Ch09_ItemName.AutoSize = True  
Me.pnlRecorderInfo_No09_Ch09_ItemName.Location = New System.Drawing.Point(631, 38)  
Me.pnlRecorderInfo_No09_Ch09_ItemName.Name = "pnlRecorderInfo_No09_Ch09_ItemName"  
Me.pnlRecorderInfo_No09_Ch09_ItemName.Size = New System.Drawing.Size(39, 12)  
Me.pnlRecorderInfo_No09_Ch09_ItemName.TabIndex = 66  
Me.pnlRecorderInfo_No09_Ch09_ItemName.Text = "Item09"
```

```
'pnlRecorderInfo_No09_Ch09_ItemData
```

```
Me.pnlRecorderInfo_No09_Ch09_ItemData.BackColor = System.Drawing.SystemColors.Window  
Me.pnlRecorderInfo_No09_Ch09_ItemData.Location = New System.Drawing.Point(633, 53)  
Me.pnlRecorderInfo_No09_Ch09_ItemData.Name = "pnlRecorderInfo_No09_Ch09_ItemData"  
Me.pnlRecorderInfo_No09_Ch09_ItemData.ReadOnly = True  
Me.pnlRecorderInfo_No09_Ch09_ItemData.Size = New System.Drawing.Size(58, 19)  
Me.pnlRecorderInfo_No09_Ch09_ItemData.TabIndex = 65
```

```
'pnlRecorderInfo_No09_Ch08_ItemName
```

```
Me.pnlRecorderInfo_No09_Ch08_ItemName.AutoSize = True  
Me.pnlRecorderInfo_No09_Ch08_ItemName.Location = New System.Drawing.Point(567, 38)  
Me.pnlRecorderInfo_No09_Ch08_ItemName.Name = "pnlRecorderInfo_No09_Ch08_ItemName"  
Me.pnlRecorderInfo_No09_Ch08_ItemName.Size = New System.Drawing.Size(39, 12)  
Me.pnlRecorderInfo_No09_Ch08_ItemName.TabIndex = 64  
Me.pnlRecorderInfo_No09_Ch08_ItemName.Text = "Item08"
```

```
'pnlRecorderInfo_No09_Ch08_ItemData
```

```
Me.pnlRecorderInfo_No09_Ch08_ItemData.BackColor = System.Drawing.SystemColors.Window  
Me.pnlRecorderInfo_No09_Ch08_ItemData.Location = New System.Drawing.Point(569, 53)  
Me.pnlRecorderInfo_No09_Ch08_ItemData.Name = "pnlRecorderInfo_No09_Ch08_ItemData"  
Me.pnlRecorderInfo_No09_Ch08_ItemData.ReadOnly = True  
Me.pnlRecorderInfo_No09_Ch08_ItemData.Size = New System.Drawing.Size(58, 19)  
Me.pnlRecorderInfo_No09_Ch08_ItemData.TabIndex = 63
```

```
'pnlRecorderInfo_No09_Ch07_ItemName
```

```
Me.pnlRecorderInfo_No09_Ch07_ItemName.AutoSize = True  
Me.pnlRecorderInfo_No09_Ch07_ItemName.Location = New System.Drawing.Point(503, 38)  
Me.pnlRecorderInfo_No09_Ch07_ItemName.Name = "pnlRecorderInfo_No09_Ch07_ItemName"  
Me.pnlRecorderInfo_No09_Ch07_ItemName.Size = New System.Drawing.Size(39, 12)  
Me.pnlRecorderInfo_No09_Ch07_ItemName.TabIndex = 62  
Me.pnlRecorderInfo_No09_Ch07_ItemName.Text = "Item07"
```

```
'pnlRecorderInfo_No09_Ch07_ItemData
```

```
Me.pnlRecorderInfo_No09_Ch07_ItemData.BackColor = System.Drawing.SystemColors.Window  
Me.pnlRecorderInfo_No09_Ch07_ItemData.Location = New System.Drawing.Point(505, 53)  
Me.pnlRecorderInfo_No09_Ch07_ItemData.Name = "pnlRecorderInfo_No09_Ch07_ItemData"  
Me.pnlRecorderInfo_No09_Ch07_ItemData.ReadOnly = True  
Me.pnlRecorderInfo_No09_Ch07_ItemData.Size = New System.Drawing.Size(58, 19)  
Me.pnlRecorderInfo_No09_Ch07_ItemData.TabIndex = 61
```

```
'pnlRecorderInfo_No09_Ch06_ItemName
'
Me.pnlRecorderInfo_No09_Ch06_ItemName.AutoSize = True
Me.pnlRecorderInfo_No09_Ch06_ItemName.Location = New System.Drawing.Point(439, 38)
Me.pnlRecorderInfo_No09_Ch06_ItemName.Name = "pnlRecorderInfo_No09_Ch06_ItemName"
Me.pnlRecorderInfo_No09_Ch06_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No09_Ch06_ItemName.TabIndex = 60
Me.pnlRecorderInfo_No09_Ch06_ItemName.Text = "Item06"
'
'pnlRecorderInfo_No09_Ch06_ItemData
'
Me.pnlRecorderInfo_No09_Ch06_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No09_Ch06_ItemData.Location = New System.Drawing.Point(441, 53)
Me.pnlRecorderInfo_No09_Ch06_ItemData.Name = "pnlRecorderInfo_No09_Ch06_ItemData"
Me.pnlRecorderInfo_No09_Ch06_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No09_Ch06_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No09_Ch06_ItemData.TabIndex = 59
'
'pnlRecorderInfo_No09_Ch05_ItemName
'
Me.pnlRecorderInfo_No09_Ch05_ItemName.AutoSize = True
Me.pnlRecorderInfo_No09_Ch05_ItemName.Location = New System.Drawing.Point(375, 38)
Me.pnlRecorderInfo_No09_Ch05_ItemName.Name = "pnlRecorderInfo_No09_Ch05_ItemName"
Me.pnlRecorderInfo_No09_Ch05_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No09_Ch05_ItemName.TabIndex = 58
Me.pnlRecorderInfo_No09_Ch05_ItemName.Text = "Item05"
'
'pnlRecorderInfo_No09_Ch05_ItemData
'
Me.pnlRecorderInfo_No09_Ch05_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No09_Ch05_ItemData.Location = New System.Drawing.Point(377, 53)
Me.pnlRecorderInfo_No09_Ch05_ItemData.Name = "pnlRecorderInfo_No09_Ch05_ItemData"
Me.pnlRecorderInfo_No09_Ch05_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No09_Ch05_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No09_Ch05_ItemData.TabIndex = 57
'
'pnlRecorderInfo_No09_Ch04_ItemName
'
Me.pnlRecorderInfo_No09_Ch04_ItemName.AutoSize = True
Me.pnlRecorderInfo_No09_Ch04_ItemName.Location = New System.Drawing.Point(311, 38)
Me.pnlRecorderInfo_No09_Ch04_ItemName.Name = "pnlRecorderInfo_No09_Ch04_ItemName"
Me.pnlRecorderInfo_No09_Ch04_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No09_Ch04_ItemName.TabIndex = 56
Me.pnlRecorderInfo_No09_Ch04_ItemName.Text = "Item04"
'
'pnlRecorderInfo_No09_Ch04_ItemData
'
Me.pnlRecorderInfo_No09_Ch04_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No09_Ch04_ItemData.Location = New System.Drawing.Point(313, 53)
Me.pnlRecorderInfo_No09_Ch04_ItemData.Name = "pnlRecorderInfo_No09_Ch04_ItemData"
Me.pnlRecorderInfo_No09_Ch04_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No09_Ch04_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No09_Ch04_ItemData.TabIndex = 55
'
'pnlRecorderInfo_No09_Ch03_ItemName
```

```
'
Me.pnlRecorderInfo_No09_Ch03_ItemName.AutoSize = True
Me.pnlRecorderInfo_No09_Ch03_ItemName.Location = New System.Drawing.Point(247, 38)
Me.pnlRecorderInfo_No09_Ch03_ItemName.Name = "pnlRecorderInfo_No09_Ch03_ItemName"
Me.pnlRecorderInfo_No09_Ch03_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No09_Ch03_ItemName.TabIndex = 54
Me.pnlRecorderInfo_No09_Ch03_ItemName.Text = "Item03"
'
'pnlRecorderInfo_No09_Ch03_ItemData
'
Me.pnlRecorderInfo_No09_Ch03_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No09_Ch03_ItemData.Location = New System.Drawing.Point(249, 53)
Me.pnlRecorderInfo_No09_Ch03_ItemData.Name = "pnlRecorderInfo_No09_Ch03_ItemData"
Me.pnlRecorderInfo_No09_Ch03_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No09_Ch03_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No09_Ch03_ItemData.TabIndex = 53
'
'pnlRecorderInfo_No09_Ch02_ItemName
'
Me.pnlRecorderInfo_No09_Ch02_ItemName.AutoSize = True
Me.pnlRecorderInfo_No09_Ch02_ItemName.Location = New System.Drawing.Point(183, 38)
Me.pnlRecorderInfo_No09_Ch02_ItemName.Name = "pnlRecorderInfo_No09_Ch02_ItemName"
Me.pnlRecorderInfo_No09_Ch02_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No09_Ch02_ItemName.TabIndex = 52
Me.pnlRecorderInfo_No09_Ch02_ItemName.Text = "Item02"
'
'pnlRecorderInfo_No09_Ch02_ItemData
'
Me.pnlRecorderInfo_No09_Ch02_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No09_Ch02_ItemData.Location = New System.Drawing.Point(185, 53)
Me.pnlRecorderInfo_No09_Ch02_ItemData.Name = "pnlRecorderInfo_No09_Ch02_ItemData"
Me.pnlRecorderInfo_No09_Ch02_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No09_Ch02_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No09_Ch02_ItemData.TabIndex = 51
'
'pnlRecorderInfo_No09_Ch01_ItemName
'
Me.pnlRecorderInfo_No09_Ch01_ItemName.AutoSize = True
Me.pnlRecorderInfo_No09_Ch01_ItemName.Location = New System.Drawing.Point(119, 38)
Me.pnlRecorderInfo_No09_Ch01_ItemName.Name = "pnlRecorderInfo_No09_Ch01_ItemName"
Me.pnlRecorderInfo_No09_Ch01_ItemName.Size = New System.Drawing.Size(39, 12)
Me.pnlRecorderInfo_No09_Ch01_ItemName.TabIndex = 50
Me.pnlRecorderInfo_No09_Ch01_ItemName.Text = "Item01"
'
'pnlRecorderInfo_No09_Ch01_ItemData
'
Me.pnlRecorderInfo_No09_Ch01_ItemData.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No09_Ch01_ItemData.Location = New System.Drawing.Point(121, 53)
Me.pnlRecorderInfo_No09_Ch01_ItemData.Name = "pnlRecorderInfo_No09_Ch01_ItemData"
Me.pnlRecorderInfo_No09_Ch01_ItemData.ReadOnly = True
Me.pnlRecorderInfo_No09_Ch01_ItemData.Size = New System.Drawing.Size(58, 19)
Me.pnlRecorderInfo_No09_Ch01_ItemData.TabIndex = 47
'
'pnlRecorderInfo_No09_RecorderName
'
```

```
Me.pnlRecorderInfo_No09_RecorderName.BackColor = System.Drawing.SystemColors.Window
Me.pnlRecorderInfo_No09_RecorderName.Font = New System.Drawing.Font("MS UI Gothic",
12.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(128, Byte))
Me.pnlRecorderInfo_No09_RecorderName.Location = New System.Drawing.Point(121, 4)
Me.pnlRecorderInfo_No09_RecorderName.Name = "pnlRecorderInfo_No09_RecorderName"
Me.pnlRecorderInfo_No09_RecorderName.ReadOnly = True
Me.pnlRecorderInfo_No09_RecorderName.Size = New System.Drawing.Size(250, 23)
Me.pnlRecorderInfo_No09_RecorderName.TabIndex = 37
Me.pnlRecorderInfo_No09_RecorderName.Text = "レコーダー名称"
'
'pnlRecorderInfo_No09_Lamp
'
Me.pnlRecorderInfo_No09_Lamp.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
Me.pnlRecorderInfo_No09_Lamp.Image =
Global.WatchManlock.My.Resources.Resources.lightBlue0
Me.pnlRecorderInfo_No09_Lamp.Location = New System.Drawing.Point(53, 4)
Me.pnlRecorderInfo_No09_Lamp.Name = "pnlRecorderInfo_No09_Lamp"
Me.pnlRecorderInfo_No09_Lamp.Size = New System.Drawing.Size(64, 65)
Me.pnlRecorderInfo_No09_Lamp.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
Me.pnlRecorderInfo_No09_Lamp.TabIndex = 37
Me.pnlRecorderInfo_No09_Lamp.TabStop = False
'
'frmTemp
'
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 12.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.ClientSize = New System.Drawing.Size(1161, 646)
Me.Controls.Add(Me.pnlRecorderInfo_No10)
Me.Controls.Add(Me.pnlRecorderInfo_No09)
Me.Name = "frmTemp"
Me.Text = "frmTemp"
Me.pnlRecorderInfo_No10.ResumeLayout(False)
Me.pnlRecorderInfo_No10.PerformLayout()
CType(Me.pnlRecorderInfo_No10_Lamp, System.ComponentModel.ISupportInitialize).EndInit()
Me.pnlRecorderInfo_No09.ResumeLayout(False)
Me.pnlRecorderInfo_No09.PerformLayout()
CType(Me.pnlRecorderInfo_No09_Lamp, System.ComponentModel.ISupportInitialize).EndInit()
Me.ResumeLayout(False)

End Sub

Friend WithEvents pnlRecorderInfo_No10 As System.Windows.Forms.Panel
Friend WithEvents TextBox2 As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_RecSW As System.Windows.Forms.Button
Friend WithEvents Button1 As System.Windows.Forms.Button
Friend WithEvents pnlRecorderInfo_No10_Ch10_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch10_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch09_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch09_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch08_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch08_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch07_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch07_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch06_ItemName As System.Windows.Forms.Label
```

```
Friend WithEvents pnlRecorderInfo_No10_Ch06_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch05_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch05_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch04_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch04_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch03_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch03_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch02_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch02_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Ch01_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No10_Ch01_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_RecorderName As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No10_Lamp As System.Windows.Forms.PictureBox
Friend WithEvents pnlRecorderInfo_No09 As System.Windows.Forms.Panel
Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_RecSW As System.Windows.Forms.Button
Friend WithEvents btnPaperlessRecorder01 As System.Windows.Forms.Button
Friend WithEvents pnlRecorderInfo_No09_Ch10_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch10_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch09_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch09_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch08_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch08_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch07_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch07_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch06_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch06_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch05_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch05_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch04_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch04_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch03_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch03_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch02_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch02_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Ch01_ItemName As System.Windows.Forms.Label
Friend WithEvents pnlRecorderInfo_No09_Ch01_ItemData As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_RecorderName As System.Windows.Forms.TextBox
Friend WithEvents pnlRecorderInfo_No09_Lamp As System.Windows.Forms.PictureBox
```

End Class

Public Class frmTemp

End Class

```
Imports System.Reflection
Imports System.Runtime.InteropServices

Public Class frmVerInfo
    Inherits System.Windows.Forms.Form

    #Region " Windows フォーム デザイナで生成されたコード "

    Public Sub New()
        MyBase.New()

        ' この呼び出しは Windows フォーム デザイナが必要です。
        InitializeComponent()

        ' InitializeComponent() 呼び出しの後に初期化を追加します。

    End Sub

    ' Form は dispose をオーバーライドしてコンポーネント一覧を消去します。
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    ' Windows フォーム デザイナが必要です。
    Private components As System.ComponentModel.IContainer

    ' メモ : 以下のプロシージャは、Windows フォーム デザイナが必要です。
    ' Windows フォーム デザイナを使って変更してください。
    ' コード エディタは使用しないでください。
    Friend WithEvents label4 As System.Windows.Forms.Label
    Friend WithEvents btnOK As System.Windows.Forms.Button
    Friend WithEvents label3 As System.Windows.Forms.Label
    Friend WithEvents label2 As System.Windows.Forms.Label
    Friend WithEvents label1 As System.Windows.Forms.Label
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents btnLicense As System.Windows.Forms.Button
    Friend WithEvents pictureBox1 As System.Windows.Forms.PictureBox
    <System.Diagnostics.DebuggerStepThrough(> Private Sub InitializeComponent()
        Dim resources As System.ComponentModel.ComponentResourceManager = New
        System.ComponentModel.ComponentResourceManager(GetType(frmVerInfo))
        Me.label4 = New System.Windows.Forms.Label()
        Me.btnOK = New System.Windows.Forms.Button()
        Me.label3 = New System.Windows.Forms.Label()
        Me.label2 = New System.Windows.Forms.Label()
        Me.label1 = New System.Windows.Forms.Label()
        Me.pictureBox1 = New System.Windows.Forms.PictureBox()
        Me.Label5 = New System.Windows.Forms.Label()
        Me.Label6 = New System.Windows.Forms.Label()
        Me.btnLicense = New System.Windows.Forms.Button()
```

```
CType(Me.pictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()  
Me.SuspendLayout()  
,  
'label4  
,  
Me.label4.Location = New System.Drawing.Point(403, 388)  
Me.label4.Name = "label4"  
Me.label4.Size = New System.Drawing.Size(118, 27)  
Me.label4.TabIndex = 11  
Me.label4.Text = " "  
,  
'btnOK  
,  
Me.btnOK.Location = New System.Drawing.Point(309, 73)  
Me.btnOK.Name = "btnOK"  
Me.btnOK.Size = New System.Drawing.Size(135, 30)  
Me.btnOK.TabIndex = 10  
Me.btnOK.Text = "OK"  
,  
'label3  
,  
Me.label3.Location = New System.Drawing.Point(495, 243)  
Me.label3.Name = "label3"  
Me.label3.Size = New System.Drawing.Size(126, 12)  
Me.label3.TabIndex = 9  
Me.label3.Text = "(株)計測リサーチコンサルタント "  
,  
'label2  
,  
Me.label2.Font = New System.Drawing.Font("MS UI Gothic", 14.25!,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(128, Byte))  
Me.label2.Location = New System.Drawing.Point(264, 47)  
Me.label2.Name = "label2"  
Me.label2.Size = New System.Drawing.Size(180, 23)  
Me.label2.TabIndex = 8  
Me.label2.Text = "Version XX.XX.XX"  
Me.label2.TextAlign = System.Drawing.ContentAlignment.MiddleRight  
,  
'label1  
,  
Me.label1.Font = New System.Drawing.Font("MS P明朝", 18.0!,  
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(128, Byte))  
Me.label1.Location = New System.Drawing.Point(147, 12)  
Me.label1.Name = "label1"  
Me.label1.Size = New System.Drawing.Size(322, 23)  
Me.label1.TabIndex = 7  
Me.label1.Text = "マンロック環境監視システム"  
,  
'pictureBox1  
,  
Me.pictureBox1.Image = Global.WatchManlock.My.Resources.Resources.TR_V500  
Me.pictureBox1.Location = New System.Drawing.Point(12, 12)  
Me.pictureBox1.Name = "pictureBox1"  
Me.pictureBox1.Size = New System.Drawing.Size(115, 109)  
Me.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom
```

```
Me.pictureBox1.TabIndex = 6
Me.pictureBox1.TabStop = False
'
'Label5
'
Me.Label5.AutoSize = True
Me.Label5.Font = New System.Drawing.Font("MS UI Gothic", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
Me.Label5.Location = New System.Drawing.Point(142, 109)
Me.Label5.Name = "Label5"
Me.Label5.Size = New System.Drawing.Size(186, 12)
Me.Label5.TabIndex = 13
Me.Label5.Text = "Copyright © 2013-2016 Nextforest "
'
'Label6
'
Me.Label6.AutoSize = True
Me.Label6.BackColor = System.Drawing.Color.White
Me.Label6.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.Label6.Font = New System.Drawing.Font("MS UI Gothic", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(128, Byte))
Me.Label6.Location = New System.Drawing.Point(12, 128)
Me.Label6.Name = "Label6"
Me.Label6.Padding = New System.Windows.Forms.Padding(10, 5, 10, 5)
Me.Label6.Size = New System.Drawing.Size(351, 48)
Me.Label6.TabIndex = 20
Me.Label6.Text = "この製品は、日本国著作権法および国際条約により保護されています。" &
Global.Microsoft.VisualBasic.ChrW(13) & Global.Microsoft.VisualBasic.ChrW(10) &
"この製品の全部または一部を無断で複製したり、無断で複製物を" & Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & "頒布すると、著作権の侵害となります" & _
"のご注意下さい。"
Me.Label6.TextAlign = System.Drawing.ContentAlignment.MiddleLeft
'
'btnLicense
'
Me.btnLicense.Location = New System.Drawing.Point(369, 128)
Me.btnLicense.Name = "btnLicense"
Me.btnLicense.Size = New System.Drawing.Size(88, 48)
Me.btnLicense.TabIndex = 21
Me.btnLicense.Text = "ライセンス登録済み"
Me.btnLicense.UseVisualStyleBackColor = True
'
'frmVerInfo
'
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 12)
Me.ClientSize = New System.Drawing.Size(467, 187)
Me.Controls.Add(Me.btnLicense)
Me.Controls.Add(Me.Label6)
Me.Controls.Add(Me.Label5)
Me.Controls.Add(Me.label4)
Me.Controls.Add(Me.btnOK)
Me.Controls.Add(Me.label3)
Me.Controls.Add(Me.label2)
Me.Controls.Add(Me.label1)
Me.Controls.Add(Me.pictureBox1)
```

```

Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
Me.MaximizeBox = False
Me.MinimizeBox = False
Me.Name = "frmVerInfo"
Me.ShowInTaskbar = False
Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent
Me.Text = "マンロック環境監視システム - バージョン情報"
CType(Me.pictureBox1, System.ComponentModel.ISupportInitialize).EndInit()
Me.ResumeLayout(False)
Me.PerformLayout()

```

End Sub

#End Region

```

Private Sub VerInfoDialog_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    ' バージョン名(AssemblyInformationalVersionAttribute属性)を取得
    Dim appVersion As String = Application.ProductVersion
    ' 製品名(AssemblyProductAttribute属性)を取得
    Dim appProductName As String = Application.ProductName
    ' 会社名(AssemblyCompanyAttribute属性)を取得
    Dim appCompanyName As String = Application.CompanyName

    ' ***** アセンブリから直接取得 *****
    Dim mainAssembly As [Assembly] = [Assembly].GetEntryAssembly()

    ' コピーライト情報を取得
    Dim appCopyright As String = "-"
    Dim CopyrightArray() As Object = _
        mainAssembly.GetCustomAttributes( _
            GetType(AssemblyCopyrightAttribute), False)
    If Not (CopyrightArray Is Nothing) AndAlso (CopyrightArray.Length > 0) Then
        appCopyright = _
            (CType(CopyrightArray(0), AssemblyCopyrightAttribute)).Copyright
    End If

    ' 詳細情報を取得
    Dim appDescription As String = "-"
    Dim DescriptionArray() As Object = _
        mainAssembly.GetCustomAttributes( _
            GetType(AssemblyDescriptionAttribute), False)
    If Not (DescriptionArray Is Nothing) AndAlso (DescriptionArray.Length > 0) Then
        appDescription = _
            (CType(DescriptionArray(0), AssemblyDescriptionAttribute)).Description
    End If

    ' ***** EXEファイルから直接取得(win32API使用) *****

    ' アプリケーション・アイコンを取得
    Dim appIcon As Icon
    Dim shinfo As New SHFILEINFO()
    Dim hSuccess As IntPtr = SHGetFileInfo( _

```

```

    mainAssembly.Location, 0, _
    shinfo, Marshal.SizeOf(shinfo), _
    SHGFI_ICON Or SHGFI_LARGEICON)
If hSuccess.Equals(IntPtr.Zero) = False Then
    appIcon = Icon.FromHandle(shinfo.hIcon)
Else
    appIcon = SystemIcons.Application
End If

' ラベルなどにバージョン情報をセット
'pictureBox1.Image = appIcon.ToBitmap()
'Text = appProductName & " のバージョン情報"
'label1.Text = appCompanyName & " " & appProductName
label2.Text = "Version " & appVersion
'label3.Text = appCopyright
'label4.Text = appDescription

' バージョン情報を取得(別バージョン)
'Dim mainAssemName As AssemblyName = mainAssembly.GetName()
'Dim appVersion2 As Version = mainAssemName.Version
'MessageBox.Show(appVersion2.ToString())

LicenseCheck()

End Sub

#Region "アイコン取得用のWin32 API"

' SHGetFileInfo関数
Private Declare Ansi Function SHGetFileInfo Lib "shell32.dll" (ByVal pszPath As String,
ByVal dwFileAttributes As Integer, ByRef psfi As SHFILEINFO, ByVal cbFileInfo As
Integer, ByVal uFlags As Integer) As IntPtr

' SHGetFileInfo関数で使用するフラグ
Private Const SHGFI_ICON As Integer = &H100 ' アイコン・リソースの取得
Private Const SHGFI_LARGEICON As Integer = &H0 ' 大きいアイコン
Private Const SHGFI_SMALLICON As Integer = &H1 ' 小さいアイコン

' SHGetFileInfo関数で使用する構造体
Private Structure SHFILEINFO
    Public hIcon As IntPtr
    Public iIcon As IntPtr
    Public dwAttributes As Integer
    <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=260)> _
    Public szDisplayName As String
    <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=80)> _
    Public szTypeName As String
End Structure

#End Region

Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnOK.Click

```

```
Me.Close()  
End Sub  
  
Private Sub LicenseCheck()  
    If CheckLicense() = True Then  
        btnLicense.Text = "ライセンス" & vbCrLf & "登録済"  
    Else  
        btnLicense.Text = "ライセンス" & vbCrLf & "未登録"  
    End If  
End Sub  
  
Private Sub btnLicense_Click(sender As System.Object, e As System.EventArgs) Handles  
btnLicense.Click  
    Dim fRegLicense As New frmRegLicense()  
    fRegLicense.ShowDialog(Me)  
    fRegLicense.Dispose()  
    LicenseCheck()  
End Sub  
End Class
```

Module mdlCtrlSub02

```
Public Sub Resize_pnlMenu(ByRef rMe As Form, Optional ByVal vName As String = "pnlMenu")
    Dim mCtrl As Control, mPosR As Integer, mPosL As Integer
    mCtrl = SearchCtrl(rMe, vName)
    If Not IsNothing(mCtrl) Then
        mCtrl = SearchCtrl(rMe, "pnlTime") : If IsNothing(mCtrl) Then Exit Sub
        mPosR = mCtrl.Left

        mCtrl = SearchCtrl(rMe, "btnPrintScreen") : If IsNothing(mCtrl) Then Exit Sub
        mCtrl.Left = mPosR - mCtrl.ClientSize.Width - 3
        mPosR = mCtrl.Left

        mCtrl = SearchCtrl(rMe, "btnDataSave") : If IsNothing(mCtrl) Then Exit Sub
        mCtrl.Left = mPosR - mCtrl.ClientSize.Width - 3
        mPosR = mCtrl.Left

        mCtrl = SearchCtrl(rMe, "picRevolvingLight") : If IsNothing(mCtrl) Then Exit Sub
        mCtrl.Left = mPosR - mCtrl.ClientSize.Width - 3
        mPosR = mCtrl.Left

        mCtrl = SearchCtrl(rMe, "pnlStatus") : If IsNothing(mCtrl) Then Exit Sub
        mCtrl.Left = mPosR - mCtrl.ClientSize.Width - 3
        mPosR = mCtrl.Left

        mCtrl = SearchCtrl(rMe, "btnViewMode") : If IsNothing(mCtrl) Then Exit Sub
        mPosL = mCtrl.Left + mCtrl.Width

        mCtrl = SearchCtrl(rMe, "PanelTitle") : If IsNothing(mCtrl) Then Exit Sub
        mCtrl.Left = mPosL + 1
        mCtrl.Width = mPosR - mPosL - 2

    End If
End Sub
```

```
Public Sub PanelTitle_Resize(ByRef rMe As Form, ByVal vArrange As clsArrange, ByVal
vName As String, ByVal rChannelInfo() As clsChannelInfo, Optional ByVal vCurrent As
Integer = 0)
    Dim mName As String, mCtrl As Control
    For i = 1 To 10
        mName = vName + "_Ch" + i.ToString("00")
        mCtrl = SearchCtrl(rMe, mName)
        If Not IsNothing(mCtrl) Then
            If i <= vArrange.CellN Then
                mCtrl.Visible = True
                mCtrl.Size = New Size(vArrange.CellWidth, vArrange.CellHeight)
                mCtrl.Location = New Point(vArrange.CellLeft(i), vArrange.CellTop(i))
                pnlDataNum_Resize(rMe, mName, rChannelInfo(i), IIf(i = vCurrent, True, False))
            Else
                mCtrl.Visible = False
            End If
        End If
    Next
End Sub
```

```
Public Sub UpdatePanelTime(ByRef rMe As Form, ByVal vDateTimeName As String, Optional
ByVal vDateName As String = "lblNowDate", Optional ByVal vTimeName As String =
"lblNowTime")
    Dim mCtrl As Control
    Dim mNowTime As DateTime = DateTime.Now
    mCtrl = SearchCtrl(rMe, vDateTimeName)
    If Not IsNothing(mCtrl) Then
        mCtrl.Tag = mNowTime.ToString("H:mm:ss")
        mCtrl = SearchCtrl(rMe, vDateName)
        If Not IsNothing(mCtrl) Then
            mCtrl.Text = mNowTime.ToString("yyyy/MM/dd")
        End If
        mCtrl = SearchCtrl(rMe, vTimeName)
        If Not IsNothing(mCtrl) Then
            mCtrl.Text = mNowTime.ToString("H:mm:ss")
        End If
    End If
End Sub
```

```
Public Sub PanelGraph_Resize(ByRef rMe As Form, ByRef rPanel As Panel, _
                            ByVal vName As String, ByVal vRecorderNo As Integer)
    Dim mName As String, mCtrl As Control
    Dim mHeight As Integer, mWidth As Integer
    mHeight = rPanel.Height
    mWidth = rPanel.Width
    If mHeight < 1 Or mWidth < 1 Then Exit Sub
    mName = vName + "_No01_Chart" 'pnlDataGraphA_No01
    mCtrl = SearchCtrl(rMe, mName)
    If Not IsNothing(mCtrl) Then
        mCtrl.Visible = True
        mCtrl.Left = 0
        mCtrl.Size = New Size(mWidth - mCtrl.Location.X, mHeight - mCtrl.Top)
    End If
End Sub
```

```
Public Sub PanelNumber_Resize(ByRef rMe As Form, ByRef rPanel As Panel, _
                              ByVal vCells As Integer, ByVal vRows As Integer, ByVal
                              vCols As Integer, _
                              ByVal vName As String, ByVal vRecorderNo As Integer)
    Dim mName As String, mCtrl As Control
    Dim mArrange As New clsArrange
    Dim mRecInfo As clsRecorderInfo

    If vCells < 1 Or vCells > 10 Then Exit Sub
    If vRecorderNo < 1 Or vRecorderNo > 10 Then Exit Sub
    mRecInfo = gManlockInfo.RecorderInfo(vRecorderNo)

    mArrange.SetStyle(vCells, vRows, vCols)
    mArrange.Height = rPanel.Height
    mArrange.Width = rPanel.Width
    If mArrange.Height < 1 Or mArrange.Width < 1 Then Exit Sub
    For i = 1 To 10
        mName = vName + "_Ch" + i.ToString("00")
        mCtrl = SearchCtrl(rMe, mName)
    End For
End Sub
```

```

If Not IsNothing(mCtrl) Then
    If i <= mArrange.CellN Then
        mCtrl.Visible = True
        mCtrl.Size = New Size(mArrange.CellWidth, mArrange.CellHeight)
        mCtrl.Location = New Point(mArrange.CellLeft(i), mArrange.CellTop(i))
        pnlDataNum_Resize(rMe, mName, mRecInfo.ChannelInfo(i),
            IIf(mRecInfo.GraphCurrentNo = i, True, False))
    Else
        mCtrl.Visible = False
    End If
End If
Next
End Sub

Public Sub PanelBar_Resize(ByRef rMe As Form, ByRef rPanel As Panel, _
    ByVal vCells As Integer, ByVal vName As String, ByVal
    vRecorderNo As Integer)
    ' ByVal rChannelInfo() As clsChannelInfo)
    Dim mName As String, mCtrl As Control
    Dim mHeight As Integer, mWidth As Integer, mWidth1 As Integer
    Dim mRecInfo As clsRecorderInfo

    If vCells < 1 Or vCells > 10 Then Exit Sub
    If vRecorderNo < 1 Or vRecorderNo > 10 Then Exit Sub
    mRecInfo = gManlockInfo.RecorderInfo(vRecorderNo)

    mHeight = rPanel.Height
    mWidth = rPanel.Width
    mWidth1 = mWidth / vCells
    If mHeight < 1 Or mWidth < 1 Then Exit Sub
    For mChNo = 1 To 10
        mName = vName + "_Ch" + mChNo.ToString("00")
        mCtrl = SearchCtrl(rMe, mName)
        If Not IsNothing(mCtrl) Then
            If mChNo <= vCells Then
                mCtrl.Visible = True
                mCtrl.Size = New Size(mWidth1, mHeight)
                mCtrl.Location = New Point(mWidth1 * (mChNo - 1), 0)
                pnlDataBar_Resize(rMe, mName, mRecInfo.ChannelInfo(mChNo), mChNo)
            Else
                mCtrl.Visible = False
            End If
        End If
    Next
End Sub
' グラフのカレント項目を設定
Public Sub SetCurrentItem(ByRef rMe As Form, ByVal vName As String, ByVal vRecorderNo
As Integer)
    Dim mName As String, mCtrl As Control, mPanelName As String
    Dim mArrange As New clsArrange
    Dim mRecInfo As clsRecorderInfo
    Dim mFontSize As Integer

    If vRecorderNo < 1 Or vRecorderNo > 10 Then Exit Sub
    mRecInfo = gManlockInfo.RecorderInfo(vRecorderNo)

```

```
For i = 1 To mRecInfo.ChannelN
    mPanelName = vName + "_Ch" + i.ToString("00")
    mName = mPanelName + "_Name" 'pnlDataNumA_Ch01_Name
    mCtrl = SearchCtrl(rMe, mName)
    If Not IsNothing(mCtrl) Then
        With mCtrl
            .Font = New Font("MSゴシック", mFontSize, FontStyle.Bold)

            End With
        End If

    Next
End Sub

End Module
```

```
Imports WatchManlock.clsManlockInfo
```

```
Module mdlCtrlSub04
```

```
Public Enum eRevolvingLightType
```

```
eltBlue0 = 0      '青停止  
eltBlue1 = 1      '青回転  
eltYellow0 = 2    '黄停止  
eltYellow1 = 3    '黄回転  
eltRed0 = 4       '赤停止  
eltRed1 = 5       '赤回転  
eltMagenta0 = 6   '紫停止  
eltMagenta1 = 7   '紫回転  
eltStop = 9       'ストップ
```

```
End Enum
```

```
Public Enum eSaveButtonType
```

```
eltSaveOff = 0    '停止  
eltSaveOn = 1     '収録
```

```
End Enum
```

```
Public Sub SetRevolvingLight(ByRef rMe As Form, ByVal vLightName As String, ByVal  
vLightType As eRevolvingLightType)
```

```
Dim mCtrl As Control, mPictureBox As Windows.Forms.PictureBox  
mCtrl = SearchCtrl(rMe, vLightName)
```

```
If Not IsNothing(mCtrl) Then
```

```
mPictureBox = mCtrl
```

```
Select Case vLightType
```

```
Case eRevolvingLightType.eltBlue0
```

```
If mPictureBox.Tag <> "eltBlue0" Then
```

```
mPictureBox.Image = WatchManlock.My.Resources.kaiblu0  
mPictureBox.Tag = "eltBlue0"
```

```
End If
```

```
Case eRevolvingLightType.eltBlue1
```

```
If mPictureBox.Tag <> "eltBlue1" Then
```

```
mPictureBox.Image = WatchManlock.My.Resources.kaiblul  
mPictureBox.Tag = "eltBlue1"
```

```
End If
```

```
Case eRevolvingLightType.eltYellow0
```

```
If mPictureBox.Tag <> "eltYellow0" Then
```

```
mPictureBox.Image = WatchManlock.My.Resources.kaiore0  
mPictureBox.Tag = "eltYellow0"
```

```
End If
```

```
Case eRevolvingLightType.eltYellow1
```

```
If mPictureBox.Tag <> "eltYellow1" Then
```

```
mPictureBox.Image = WatchManlock.My.Resources.kaiore1  
mPictureBox.Tag = "eltYellow1"
```

```
End If
```

```
Case eRevolvingLightType.eltRed0
```

```
If mPictureBox.Tag <> "eltRed0" Then
```

```
mPictureBox.Image = WatchManlock.My.Resources.kaired0  
mPictureBox.Tag = "eltRed0"
```

```
End If
```

```
Case eRevolvingLightType.eltRed1
```

```
    If mPictureBox.Tag <> "eltRed1" Then
        mPictureBox.Image = WatchManlock.My.Resources.kaired1
        mPictureBox.Tag = "eltRed1"
    End If
Case eRevolvingLightType.eltMagenta0
    If mPictureBox.Tag <> "eltMagenta0" Then
        mPictureBox.Image = WatchManlock.My.Resources.kaivi0
        mPictureBox.Tag = "eltMagenta0"
    End If
Case eRevolvingLightType.eltMagenta1
    If mPictureBox.Tag <> "eltMagenta1" Then
        mPictureBox.Image = WatchManlock.My.Resources.kaivil
        mPictureBox.Tag = "eltMagenta1"
    End If
Case eRevolvingLightType.eltStop
    '           If mPictureBox.Tag <> "eltStop" Then
mPictureBox.Image = WatchManlock.My.Resources.bmpSTOP
mPictureBox.Tag = "eltStop"
    '           End If
End Select
End If
End Sub

Public Sub SetSaveButton(ByRef rMe As Form, ByVal vButtonName As String, ByVal
vSaveButtonType As eSaveButtonType)
    Dim mCtrl As Control, mButton As Windows.Forms.Button
    mCtrl = SearchCtrl(rMe, vButtonName)

    If Not IsNothing(mCtrl) Then
        mButton = mCtrl
        Select Case vSaveButtonType
            Case eSaveButtonType.eltSaveOff
                mButton.BackgroundImage = WatchManlock.My.Resources.menuSaveOff
            Case eSaveButtonType.eltSaveOn
                mButton.BackgroundImage = WatchManlock.My.Resources.menuSaveOn
        End Select
    End If
End Sub

Public Sub SetCombo_TimePitch(ByRef rComboBox As ComboBox)
    With rComboBox
        .Items.Clear()
        .Items.Add("全部")
        .Items.Add(" 1分間隔")
        .Items.Add(" 2分間隔")
        .Items.Add(" 5分間隔")
        .Items.Add("10分間隔")
        .Items.Add("20分間隔")
        .Items.Add("30分間隔")
        .Items.Add(" 1時間間隔")
        .Items.Add(" 2時間間隔")
        .Items.Add(" 3時間間隔")
        .Items.Add(" 6時間間隔")
        .Items.Add("12時間間隔")
        .Items.Add("1日間隔")
    End With
End Sub
```

```
        .SelectedIndex = eTimePitch.etcAll
    End With
End Sub

Public Sub SetCombo_OutputSettingType(ByRef rComboBox As ComboBox)
    With rComboBox
        .Items.Clear()
        .Items.Add("フリー")
        .Items.Add("日報")
        .Items.Add("週報")
        .Items.Add("月報")
        .SelectedIndex = 0
    End With
End Sub

End Module
```

Module mdlDateTime

'指定日の[yyyy/MM/dd 0:00:00]を返す

```
Public Function DayTop(ByVal vDate As Date) As Date
    Return Date.Parse(vDate.ToString("yyyy/MM/dd"))
End Function
```

'指定日の[yyyy/MM/01 0:00:00]を返す

```
Public Function MonthTop(ByVal vDate As Date) As Date
    Return Date.Parse(vDate.ToString("yyyy/MM") & "/01")
End Function
```

End Module

```
Imports clslibNextforest
```

```
Imports WatchManlock.frmPaperlessRecorder
```

```
Module mdlForm00
```

```
Public Sub SaveLocation(ByRef rForm As Form, ByVal vName As String)
    Dim Reg As New clsRegistry
    Reg.SetOrganization(GetCompanyName)

    rForm.WindowState = FormWindowState.Normal
    Reg.SaveStr(rForm.Left, vName + "_Left", "FormLocation", GetAppName)
    Reg.SaveStr(rForm.Top, vName + "_Top")
    Reg.SaveStr(rForm.Width, vName + "_Width")
    Reg.SaveStr(rForm.Height, vName + "_Height")

```

```
End Sub
```

```
Public Sub LoadLocation(ByRef rForm As Form, ByVal vName As String)
    Dim Reg As New clsRegistry
    Reg.SetOrganization(GetCompanyName)

    Dim mLeft As Integer, mTop As Integer, mWidth As Integer, mHeight As Integer
    mLeft = Reg.LoadStr(-1, vName + "_Left", "FormLocation", GetAppName)
    mTop = Reg.LoadStr(-1, vName + "_Top")
    mWidth = Reg.LoadStr(600, vName + "_Width")
    mHeight = Reg.LoadStr(400, vName + "_Height")
    mWidth = IIf(mWidth < 400, 400, mWidth)
    mHeight = IIf(mHeight < 600, 600, mHeight)
    rForm.WindowState = FormWindowState.Normal
    rForm.Left = IIf(mLeft < -mWidth, -1, mLeft)
    rForm.Top = IIf(mTop < -mHeight, -1, mTop)
    rForm.Width = mWidth
    rForm.Height = mHeight

```

```
End Sub
```

```
Public Sub SaveRootTab(ByVal vRecNo As Integer, ByVal vIndex As eViewMode)
    Dim Reg As New clsRegistry
    Dim mIndexNo As Integer
    Reg.SetOrganization(GetCompanyName)
    mIndexNo = vIndex
    Reg.SaveStr(mIndexNo, "TabIndex", "RecorderNo" & vRecNo.ToString("00"), GetAppName)

```

```
End Sub
```

```
Public Function GetRootTab(ByVal vRecNo As Integer) As eViewMode
    Dim Reg As New clsRegistry
    Dim mIndex As eViewMode, buf As String
    Reg.SetOrganization(GetCompanyName)
    buf = Reg.LoadStr(eViewMode.evmWave, "TabIndex", "RecorderNo" &
    vRecNo.ToString("00"), GetAppName)
    mIndex = Val(buf) ' Integer.Parse(buf)
    Return mIndex
End Function
```

End Module

```
Imports System.Text
```

```
Imports System.IO
```

```
Module mdlFile
```

```
Public Sub LogFileWrite(ByVal vRecNo As Integer, ByVal vMessage As String)
```

```
    Dim mNowTime As DateTime = DateTime.Now
```

```
    Dim mNowTimeStr As String = mNowTime.ToString("yyyy/MM/dd HH:mm:ss")
```

```
    Dim FileName As String
```

```
    FileName = GetMyDocPath() + "\Log" + mNowTime.ToString("yyyyMMdd") + "_" +  
vRecNo.ToString("0") + ".txt"
```

```
    Dim writer As New StreamWriter(FileName, True)
```

```
    writer.WriteLine(mNowTime.ToString("HH:mm:ss.fff") + vbTab + vMessage)
```

```
    writer.Close()
```

```
End Sub
```

```
・実行中のアプリケーション(.exe)が保存されているパスを取得する
```

```
Public Function GetAppPath() As String
```

```
    Return My.Application.Info.DirectoryPath.ToString()
```

```
End Function
```

```
・マイドキュメントパスを取得する
```

```
Public Function GetMyDocPath() As String
```

```
    Return System.Environment.GetFolderPath(Environment.SpecialFolder.Personal)
```

```
End Function
```

```
・フルファイルパスから拡張子を取得
```

```
Public Function GetExtension(ByVal vFullPath As String) As String
```

```
    Dim ext1 As String
```

```
    ext1 = Path.GetExtension(vFullPath)
```

```
    Return ext1
```

```
End Function
```

```
・フルファイルパスからフォルダー名を取得
```

```
Public Function GetDirectoryName(ByVal vFullPath As String) As String
```

```
    Dim ext1 As String
```

```
    ext1 = Path.GetDirectoryName(vFullPath)
```

```
    Return ext1
```

```
End Function
```

```
End Module
```

```
Imports clslibNextforest
```

```
Module mdlGlobal
```

```
' {33F7ED6A-42D7-4ABE-A03C-B2EFEF81C32B}
Public Const PAI = 3.1415926536          '円周率
Public Const NullData = 1.0E-20        'データのない時の値
Public Const NullDate = #1/1/1900#     'データのない時の値

Public Const gProductTYPE = "MEWA"     'プロダクトタイプ ProductTYPE
Public Const gLicenseLevel = 3         'プロダクトレベル
Public gProductKEY As String = "      -      -      -      " 'プロダクトタイプ ProductKEY
Public gLicense As Boolean = False     'プロダクトライセンス

Public gPanelBabkColor As System.Drawing.Color 'パネルのバックカラー
```

```
Public gManlockInfo As clsManlockInfo
```

```
' Public gMeasData As clsMeasData
```

```
Public gDefaultFolder As String
Public gDebug As Boolean = False       'レジストリ対応
Public gPrintLineWidth As Integer = 3
Public gGraphRefreshInterval As Integer = 60
Public gDefaultFolderUSB As String
Public gWaitTimeTCP As Integer = 60
Public gDataLatchTime As Integer = 120
```

```
Public Sub Initialize()
```

```
Dim libFF As New clsFolderFile
gManlockInfo = New clsManlockInfo
' gMeasData = New clsMeasData
gDefaultFolder = libFF.GetMyDocFolder
gDefaultFolderUSB = libFF.GetMyDocFolder

gPanelBabkColor = Color.FromArgb(173, 162, 173)
gManlockInfo.FileName = ""
RegLoad()
CheckLicense()
End Sub
```

```
Public Function CheckLicense() As Boolean
```

```
gLicense = False
Dim mProtect = New clsProtect()
Dim mProductID As String = mProtect.GetProductID(gProductTYPE)
Dim mProductKEY As String = gProductKEY
Dim rtn As Integer = mProtect.CheckProduct(mProductID, mProductKEY)
If rtn < gLicenseLevel Then
gLicense = True
Else
gLicense = False
End If
Return gLicense
```

End Function

Public Function CheckTrial() As Boolean

Dim mNowDate As Date = Date.Now

If mNowDate > CDate("2017/1/1") And mNowDate < CDate("2017/2/1") Then

Return True

End If

Return False

End Function

Public Sub RegSave()

Dim Reg As New clsRegistry

Reg.SetOrganization(GetCompanyName)

Reg.SaveStr(gProductKEY, "ProductKEY", "Setting", GetAppName)

Reg.SaveStr(gDebug, "Debug", "Setting", GetAppName)

Reg.SaveStr(gManlockInfo.FileName, "CurrentFileName", "Setting", GetAppName)

Reg.SaveStr(gDefaultFolder, "DefaultFolder", "Setting", GetAppName)

Reg.SaveStr(gDefaultFolderUSB, "DefaultFolderUSB", "Setting", GetAppName)

Reg.SaveStr(gPrintLineWidth, "PrintLineWidth", "Setting", GetAppName)

Reg.SaveStr(gGraphRefreshInterval, "GraphRefreshInterval", "Setting", GetAppName)

Reg.SaveStr(gWaitTimeTCP, "WaitTimeTCP", "Setting", GetAppName)

Reg.SaveStr(gDataLatchTime, "DataLatchTime", "Setting", GetAppName)

End Sub

Public Sub RegLoad()

Dim Reg As New clsRegistry

Reg.SetOrganization(GetCompanyName)

gProductKEY = Reg.LoadStr(gProductKEY, "ProductKEY", "Setting", GetAppName)

gDebug = Reg.LoadStr(gDebug, "Debug", "Setting", GetAppName)

gManlockInfo.FileName = Reg.LoadStr(gManlockInfo.FileName, "CurrentFileName",
"Setting", GetAppName)

gDefaultFolder = Reg.LoadStr(gDefaultFolder, "DefaultFolder", "Setting", GetAppName)

gDefaultFolderUSB = Reg.LoadStr(gDefaultFolderUSB, "DefaultFolderUSB", "Setting",
GetAppName)

gPrintLineWidth = Reg.LoadStr(gPrintLineWidth, "PrintLineWidth", "Setting", GetAppName)

gGraphRefreshInterval = Reg.LoadStr(gGraphRefreshInterval, "GraphRefreshInterval",
"Setting", GetAppName)

gGraphRefreshInterval = IIf(gGraphRefreshInterval < 60, 60, gGraphRefreshInterval)

gGraphRefreshInterval = IIf(gGraphRefreshInterval > 3600, 3600, gGraphRefreshInterval)

gWaitTimeTCP = Reg.LoadStr(gWaitTimeTCP, "WaitTimeTCP", "Setting", GetAppName)

gDataLatchTime = Reg.LoadStr(gDataLatchTime, "DataLatchTime", "Setting", GetAppName)

End Sub

Public Function GetCompanyName()

Dim appProductName As String = Application.ProductName

GetCompanyName = appProductName

GetCompanyName = "KRC"

End Function

Public Function GetAppName()

Dim appProductName As String = Application.ProductName

GetAppName = appProductName

GetAppName = "WatchManlock"

End Function

Public Function GetAppFolder() As String

Return System.AppDomain.CurrentDomain.BaseDirectory

End Function

End Module

Module mdlPanel01

```
Public Sub pnlDataNum_Resize(ByRef rMe As Form, ByVal vPanelName As String, ByVal
vChannelInfo As clsChannelInfo, Optional ByVal vCurrent As Boolean = False)
    Dim mFrame As Control, mName As String
    Dim mCtrlName As Control, mCtrlUnit As Control, mCtrlData As Control
    Dim mFontSize As Integer, mHeight As Integer, mFrameHeight As Integer, mColorBoxWidth
As Integer = 0, mTop As Integer
    mFrame = SearchCtrl(rMe, vPanelName)

    If Not IsNothing(mFrame) Then
        mFrameHeight = mFrame.Height - 7
        Dim mColorBox As Control, mDrawCheck As Control, mWidth As Integer
        mName = vPanelName + "_Color" 'pnlDataNumA_Ch01_Color
        mColorBox = SearchCtrl(rMe, mName)
        If Not IsNothing(mColorBox) Then
            With mColorBox
                mWidth = mFrame.Width / 20
                mWidth = IIf(mWidth < 20, 20, mWidth)
                .Left = 2 : .Top = 2
                .Height = mFrameHeight - 1
                .Width = mWidth
                .BackColor = vChannelInfo.Color
                mColorBoxWidth = mWidth
            End With

            mDrawCheck = SearchCtrl(rMe, vPanelName + "_Draw") 'pnlDataNumB_Ch01_Draw
            If Not IsNothing(mDrawCheck) Then
                With mDrawCheck
                    .Left = 2 : .Top = 2
                    .Height = mColorBox.Width - 6
                    .Width = mColorBox.Width - 6
                End With
            End If
        End If

        mCtrlName = SearchCtrl(rMe, vPanelName + "_Name") 'pnlDataNumA_Ch01_Name
        If Not IsNothing(mCtrlName) Then
            mFontSize = convPixelToPoint(mFrameHeight * 0.15)
            If mFontSize < 8 Then mFontSize = 8
            With mCtrlName
                If vCurrent = True Then
                    .Font = New Font("MSゴシック", mFontSize, FontStyle.Underline Or
FontStyle.Underline)
                Else
                    .Font = New Font("MSゴシック", mFontSize)
                End If
                .Text = vChannelInfo.Name
                .Left = mColorBox.DisplayRectangle.Right + 5
                .Top = 0
            End With
        End If

        mCtrlUnit = SearchCtrl(rMe, vPanelName + "_Unit") 'pnlDataNumA_Ch01_Unit
        If Not IsNothing(mCtrlUnit) Then
```

```
mFontSize = convPixelToPoint(mFrameHeight * 0.12)
If mFontSize < 8 Then mFontSize = 8
With mCtrlUnit
    .Font = New Font("MSゴシック", mFontSize, FontStyle.Bold)
    .Text = vChannelInfo.Unit
    .Left = mFrame.Width - .Width - 5
    .Top = mFrameHeight - .Height
End With
End If

Dim mHeight0 As Integer = mFrameHeight * 0.8
Dim mHeight1 As Integer = (mFrameHeight - mCtrlUnit.Height - mCtrlName.Height)
Dim mHeight2 As Integer = ((mFrame.Width - mColorBox.Left - mColorBox.Width)) * 0.3
mHeight = IIf(mHeight1 < mHeight2, mHeight1, mHeight2)
mCtrlData = SearchCtrl(rMe, vPanelName + "_Data") 'pnlDataNumA_Ch01_Data
If Not IsNothing(mCtrlData) Then
    '        Dim mDataFrameWidth As Integer =
With mCtrlData
    mFontSize = convPixelToPoint(mHeight)
    If mFontSize < 18 Then mFontSize = 18
    .Font = New Font("MSゴシック", mFontSize, FontStyle.Bold)
    .Text = vChannelInfo.DataNowStr
    .Left = mFrame.Width - .Width - 10
    mTop = mFrameHeight - mCtrlUnit.Height - .Height
    .Top = IIf(mTop < 0, 0, mTop)
End With
End If
End If

End Sub

Public Sub pnlDataBar_Resize(ByRef rMe As Form, ByVal vPanelName As String, ByVal
vChannelInfo As clsChannelInfo, ByVal vChNo As Integer)
    Dim mFrame As Control, mCtrl As Control
    Dim mName As String, mFontSize As Integer
    Dim mTop As Single, mBottom As Single

    mFrame = SearchCtrl(rMe, vPanelName)
    If Not IsNothing(mFrame) Then
        mTop = 0 : mBottom = mFrame.Height
        mName = vPanelName + "_Name" 'pnlDataBarA_Ch01_Name
        mCtrl = SearchCtrl(rMe, mName)
        If Not IsNothing(mCtrl) Then
            With mCtrl
                mFontSize = .Font.Size
                .Font = New Font("MSゴシック", mFontSize, FontStyle.Bold)
                .Text = vChannelInfo.Name
                .Left = 2 ' + mFrame.Width / 20
                .Top = mTop + 2
                mTop = .Top + .Height
            End With
        End If

        mName = vPanelName + "_DataMax" 'pnlDataBarA_Ch01_DataMax
        mCtrl = SearchCtrl(rMe, mName)
```

```
If Not IsNothing(mCtrl) Then
    With mCtrl
        .Text = vChannelInfo.DataMaxStr
        .Left = (mFrame.Width - .Width) / 2
        .Top = mTop + 2
        mTop = .Top + .Height
    End With
End If

mName = vPanelName + "_Unit" 'pnlDataBarA_Ch01_Unit
mCtrl = SearchCtrl(rMe, mName)
If Not IsNothing(mCtrl) Then
    With mCtrl
        .Text = vChannelInfo.Unit
        .Left = mFrame.Width - .Width - 5
        .Top = mBottom - .Height - 5
        mBottom = .Top
    End With
End If

mName = vPanelName + "_Data" 'pnlDataBarA_Ch01_Data
mCtrl = SearchCtrl(rMe, mName)
If Not IsNothing(mCtrl) Then
    With mCtrl
        If mFrame.Width > 80 Then
            mFontSize = 18
        Else
            mFontSize = 18 * (mFrame.Width / 80)
            mFontSize = IIf(mFontSize < 9, 9, mFontSize)
        End If
        .Font = New Font("MSゴシック", mFontSize)
        .Text = vChannelInfo.DataNowStr
        .Refresh()
        .Left = mFrame.Width - .Width - 5
        .Top = mBottom - .Height - 2
        mBottom = .Top
    End With
End If

mName = vPanelName + "_DataMin" 'pnlDataBarA_Ch01_DataMin
mCtrl = SearchCtrl(rMe, mName)
If Not IsNothing(mCtrl) Then
    With mCtrl
        .Text = vChannelInfo.DataMinStr
        .Left = (mFrame.Width - .Width) / 2
        .Top = mBottom - .Height - 2
        mBottom = .Top
    End With
End If

mName = vPanelName + "_BarFrame" 'pnlDataBarA_Ch01_BarFrame
mCtrl = SearchCtrl(rMe, mName)
If Not IsNothing(mCtrl) Then
    With mCtrl
        .Left = 10
    End With
End If
```

```

        .Top = mTop + 2
        .Height = mBottom - mTop - 4
        .Width = mFrame.Width - 20
        '           .BackColor = pvChannelInfo.Color
    DrawBar(rMe, mName, vChannelInfo, vChNo)

```

```
End With
```

```
End If
```

```
End If
```

```
End Sub
```

```
Public Function RateY(ByVal vData As Double, ByVal vChannelInfo As clsChannelInfo) As Double
```

```
    Dim mRate As Double, mW As Double
```

```
    With vChannelInfo
```

```
        mW = (.DataMax - .DataMin)
```

```
        If vData < .DataMin Or mW <= 0 Then
```

```
            mRate = 0
```

```
        ElseIf vData > .DataMax Then
```

```
            mRate = 1
```

```
        Else
```

```
            mRate = (vData - .DataMin) / mW
```

```
        End If
```

```
    Return mRate
```

```
End With
```

```
End Function
```

```
Public Sub DrawBar(ByRef rMe As Form, ByVal vPicName As String, ByVal vChInfo As clsChannelInfo, ByVal vChNo As Integer)
```

```
    Dim mCtrl As Control, mPictureBox As PictureBox
```

```
    Dim mWidth As Integer, mHeight As Integer
```

```
    mCtrl = SearchCtrl(rMe, vPicName)
```

```
    If Not IsNothing(mCtrl) Then
```

```
        If TypeName(mCtrl) = "PictureBox" Then
```

```
            mPictureBox = mCtrl
```

```
            mWidth = mCtrl.Width : mHeight = mCtrl.Height
```

```
            With mCtrl
```

```
                Dim canvas As New Bitmap(mWidth, mHeight)
```

```
                Dim g As Graphics = Graphics.FromImage(canvas)
```

```
                Dim x1 As Integer, y1 As Integer
```

```
                Dim mX0 As Integer, mX1 As Integer, mW As Integer
```

```
                Dim mY0 As Integer, mY1 As Integer, mY2 As Integer, mY5 As Integer, mY7 As Integer, mY8 As Integer, mY9 As Integer
```

```
                Dim brsRed As New SolidBrush(Color.Red)
```

```
                Dim brsMagenta As New SolidBrush(Color.Magenta)
```

```
                Dim brsYellow As New SolidBrush(Color.Yellow)
```

```
                Dim brsDataColor As New SolidBrush(vChInfo.Color)
```

```
                Dim brsRed As New SolidBrush(Color.Red)
```

```
                Dim brsMagenta As New SolidBrush(Color.Magenta)
```

```
                Dim brsYellow As New SolidBrush(Color.Yellow)
```

```
                Dim brsDataColor As New SolidBrush(vChInfo.Color)
```

```
                Dim brsRed As New SolidBrush(Color.Red)
```

```
                Dim brsMagenta As New SolidBrush(Color.Magenta)
```

```
                Dim brsYellow As New SolidBrush(Color.Yellow)
```

```
                Dim brsDataColor As New SolidBrush(vChInfo.Color)
```

```
                Dim brsRed As New SolidBrush(Color.Red)
```

```
                Dim brsMagenta As New SolidBrush(Color.Magenta)
```

```
                Dim brsYellow As New SolidBrush(Color.Yellow)
```

```
                Dim brsDataColor As New SolidBrush(vChInfo.Color)
```

```

mY5 = mHeight - mHeight * RateY(vChInfo.DataNow, vChInfo)
g.FillRectangle(brsDataColor, 0, mY5, mWidth, mY0 - mY5)

'mY9 = 0
'mY8 = mHeight - mHeight * RateY(vChInfo.AlarmHi2, vChInfo)
'mY7 = mHeight - mHeight * RateY(vChInfo.AlarmHi1, vChInfo)
'mY2 = mHeight - mHeight * RateY(vChInfo.AlarmLo1, vChInfo)
'mY1 = mHeight - mHeight * RateY(vChInfo.AlarmLo2, vChInfo)
'mY0 = mHeight

'g.FillRectangle(brsRed, mX0, mY9, mW, mY8 - mY9)
'g.FillRectangle(brsYellow, mX0, mY8, mW, mY7 - mY8)
'g.FillRectangle(brsYellow, mX0, mY2, mW, mY1 - mY2)
'g.FillRectangle(brsRed, mX0, mY1, mW, mY0 - mY1)

'g.DrawRectangle(Pens.Black, mX0, mY9, mW, mY8 - mY9)
'g.DrawRectangle(Pens.Black, mX0, mY8, mW, mY7 - mY8)
'g.DrawRectangle(Pens.Black, mX0, mY2, mW, mY1 - mY2)
'g.DrawRectangle(Pens.Black, mX0, mY1, mW, mY0 - mY1)

'上限Hi2の描画
If IsNullData(vChInfo.AlarmHi2) = False Then
    mY8 = mHeight - mHeight * RateY(vChInfo.AlarmHi2, vChInfo)
    g.FillRectangle(IIf(vChNo = 5 Or vChNo = 6, brsMagenta, brsRed), mX0, mY9,
        mW, mY8 - mY9)
    g.DrawRectangle(Pens.Black, mX0, mY9, mW, mY8 - mY9)
End If

'上限Hi1の描画
If IsNullData(vChInfo.AlarmHi1) = False Then
    mY7 = mHeight - mHeight * RateY(vChInfo.AlarmHi1, vChInfo)
    g.FillRectangle(brsYellow, mX0, mY8, mW, mY7 - mY8)
    g.DrawRectangle(Pens.Black, mX0, mY8, mW, mY7 - mY8)
End If

'下限Lo2の描画
If IsNullData(vChInfo.AlarmLo2) = False Then
    mY1 = mHeight - mHeight * RateY(vChInfo.AlarmLo2, vChInfo)
    g.FillRectangle(IIf(vChNo = 5 Or vChNo = 6, brsMagenta, brsRed), mX0, mY1,
        mW, mY0 - mY1)
    g.DrawRectangle(Pens.Black, mX0, mY1, mW, mY0 - mY1)
End If

'下限Lo1の描画
If IsNullData(vChInfo.AlarmLo1) = False Then
    mY2 = mHeight - mHeight * RateY(vChInfo.AlarmLo1, vChInfo)
    g.FillRectangle(brsYellow, mX0, mY2, mW, mY1 - mY2)
    g.DrawRectangle(Pens.Black, mX0, mY2, mW, mY1 - mY2)
End If

For i = 1 To 9
    x1 = mWidth - mWidth / IIf(i = 5, 2, 3)
    y1 = mHeight - mHeight * (i / 10)
    g.DrawLine(Pens.Black, x1, y1, mWidth, y1)
    g.DrawLine(Pens.WhiteSmoke, x1, y1 + 1, mWidth, y1 + 1)
Next

```

```
        'リソースを解放する
        g.Dispose()
        mPictureBox.Image = canvas '作成したビットマップをPictureBoxに表示

    End With
End If
End If

End Sub

End Module
```

Module mdlPanel02

```
Public Sub pnlRecorderInfo(ByRef rMe As frmMain, ByVal vPanelName As String, ByVal
vRecorderInfo As clsRecorderInfo, Optional ByVal vRefresh As Boolean = False)
    Dim mFrame As Control, mCtrl As Control
    Dim mName As String, mViewFlag As Boolean
    mFrame = SearchCtrl(rMe, vPanelName)

    If Not IsNothing(mFrame) Then

        If vRefresh = True Then
            mName = vPanelName + "_RecorderName" 'pnlRecorderInfo_No01_RecorderName
            mCtrl = SearchCtrl(rMe, mName)
            If Not IsNothing(mCtrl) Then
                With mCtrl
                    .Text = vRecorderInfo.RecorderName
                End With
            End If
        End If

        mName = vPanelName + "_ActionMode" 'pnlRecorderInfo_No01_ActionMode
        mCtrl = SearchCtrl(rMe, mName)
        If Not IsNothing(mCtrl) Then
            With mCtrl
                .Text = vRecorderInfo.ActionModeStr
            End With
        End If

        mName = vPanelName + "_Status" 'pnlRecorderInfo_No01_Status
        mCtrl = SearchCtrl(rMe, mName)
        If Not IsNothing(mCtrl) Then
            With mCtrl
                .Text = rMe.GetRecorderForm(vRecorderInfo.RecorderNo).GetStatusStr
            End With
        End If

        mName = vPanelName + "_RecSW" 'pnlRecorderInfo_No01_RecSW
        mCtrl = SearchCtrl(rMe, mName)
        If Not IsNothing(mCtrl) Then
            With mCtrl
                If vRecorderInfo.ActionMode = clsRecorderInfo.eActionMode.esmRecording Then
                    .BackgroundImage = WatchManlock.My.Resources.menuSaveOn
                Else
                    .BackgroundImage = WatchManlock.My.Resources.menuSaveOff
                End If
            End With
        End If

        mName = vPanelName + "_Lamp" 'pnlRecorderInfo_No01_Lamp

        mCtrl = SearchCtrl(rMe, mName)
        If Not IsNothing(mCtrl) Then
            If vRecorderInfo.Active = True Then
```

```
With mCtrl
    Select Case vRecorderInfo.GetAlarmLevel
        Case clsChannelInfo.eAlarmLevel.ealNormal
            SetRevolvingLight(rMe, mName, eRevolvingLightType.eltBlue0)
        Case clsChannelInfo.eAlarmLevel.ealAttention
            SetRevolvingLight(rMe, mName, eRevolvingLightType.eltYellow1)
        Case clsChannelInfo.eAlarmLevel.ealWarning2
            SetRevolvingLight(rMe, mName, eRevolvingLightType.eltMagental)
        Case clsChannelInfo.eAlarmLevel.ealWarning
            SetRevolvingLight(rMe, mName, eRevolvingLightType.eltRed1)
    End Select
End With
Else
    SetRevolvingLight(rMe, mName, eRevolvingLightType.eltStop)
End If
End If

For mChNo = 1 To 10
    mViewFlag = IIf(mChNo <= vRecorderInfo.ChannelN, True, False)

    If vRefresh = True Then
        mName = vPanelName + "_Ch" & mChNo.ToString("00") & "_ItemName"
        'pnlRecorderInfo_No01_Ch01_ItemName
        mCtrl = SearchCtrl(rMe, mName)
        If Not IsNothing(mCtrl) Then
            With mCtrl
                .Visible = mViewFlag
                .Text = vRecorderInfo.ChannelInfo(mChNo).Name
            End With
        End If
    End If

    mName = vPanelName + "_Ch" & mChNo.ToString("00") & "_ItemData"
    'pnlRecorderInfo_No01_Ch01_ItemData
    mCtrl = SearchCtrl(rMe, mName)
    If Not IsNothing(mCtrl) Then
        With mCtrl
            .Visible = mViewFlag
            .Text = vRecorderInfo.ChannelInfo(mChNo).DataNowStr
            Dim mColor As Drawing.Color
            mColor = vRecorderInfo.ChannelInfo(mChNo).GetAlarmLevelColor(2)
            If (mChNo = 5 Or mChNo = 6) And mColor = Drawing.Color.Red Then
                'CO2の時の警告色はピンクにする
                mColor = Drawing.Color.Magenta
            End If
            .BackColor = mColor
        End With
    End If
Next

End If

End Sub
```

```
Public Sub UpdateSaveButton(ByRef rMe As frmPaperlessRecorder, ByVal vRecorderNo As Integer)
    Dim mRecorderInfo As clsRecorderInfo = gManlockInfo.RecorderInfo(vRecorderNo)

    If mRecorderInfo.ActionMode = clsRecorderInfo.eActionMode.esmRecording Then
        SetSaveButton(rMe, "btnDataSave", eSaveButtonType.eltSaveOn)
    Else
        SetSaveButton(rMe, "btnDataSave", eSaveButtonType.eltSaveOff)
    End If
    ' SetSaveButton(rMe, "btnDataSave", IIf(mRecorderInfo.ActionMode =
    clsRecorderInfo.eActionMode.esmRecording, eSaveButtonType.eltSaveOn,
    eSaveButtonType.eltSaveOff))
End Sub
```

・数値とバーを更新

```
Public Sub pnlRecorderInfo2(ByRef rMe As frmPaperlessRecorder, ByVal vRecorderNo As Integer)
    Dim mRecorderInfo As clsRecorderInfo = gManlockInfo.RecorderInfo(vRecorderNo)
    Dim mFrame As Control, mPanelName As String, mName As String, mRight As Integer
    Dim mCtrl_Data As Control, mCtrl_Item As Control, mCtrl_Unit As Control
    Dim mNumPnlName() As String = {"pnlDataNumA", "pnlDataNumB", "pnlDataNumC"}
    Dim mBarPnlName() As String = {"pnlDataBarA", "pnlDataBarB"}

    rMe.Text = gManlockInfo.Title & "[" & mRecorderInfo.RecorderName & "]"
    rMe.lblRecorderName.Text = mRecorderInfo.RecorderName

    UpdateSaveButton(rMe, vRecorderNo)

    For Each pnlName As String In mNumPnlName

        mCtrl_Item = Nothing
        mCtrl_Data = Nothing
        mCtrl_Unit = Nothing
        For mCh = 1 To mRecorderInfo.ChannelN
            mPanelName = pnlName & "_Ch" & mCh.ToString("00")
            mFrame = SearchCtrl(rMe, mPanelName)
            If Not IsNothing(mFrame) Then

                mCtrl_Item = SearchCtrl(rMe, mPanelName + "_Name") 'pnlDataNumA_Ch01_Name
                Dim mSizeItem As Size
                Dim mRectItem As Rectangle
                If Not IsNothing(mCtrl_Item) Then
                    mSizeItem = mCtrl_Item.Size
                    mRectItem = mCtrl_Item.DisplayRectangle
                End If

                mCtrl_Unit = SearchCtrl(rMe, mPanelName + "_Unit") 'pnlDataNumA_Ch01_Name
                Dim mSizeUnit As Size
                Dim mRectUnit As Rectangle
                If Not IsNothing(mCtrl_Unit) Then
                    mSizeUnit = mCtrl_Unit.Size
                    mRectUnit = mCtrl_Unit.DisplayRectangle
                End If
            End For
        Next
    End For
```

```
mCtrl_Data = SearchCtrl(rMe, mPanelName & "_Data") 'pnlDataNumA_Ch01_Data
If Not IsNothing(mCtrl_Data) Then
    With mCtrl_Data
        .Text = mRecorderInfo.ChannelInfo(mCh).DataNowStr
        .Left = mFrame.Width - 10 - .Width
        '' .Refresh()

        'Dim mDataFrameWidth As Integer = (mFrame.Width * 0.5) + 10
        'If mDataFrameWidth < .Width Then
        ' Dim mOldFontSize As Single = .Font.Size
        ' Dim mFontSize As Integer
        ' mFontSize = mOldFontSize * (mDataFrameWidth / .Width)
        ' ' .Font = New Font("MSゴシック", mFontSize)
        ' .Left = mFrame.Width - 10 - .Width

    'End If

    End With
End If

'状態により背景色を変更
Dim mColor As Drawing.Color
mColor = mRecorderInfo.ChannelInfo(mCh).GetAlarmLevelColor(3)
If (mCh = 5 Or mCh = 6) And mColor = Drawing.Color.Red Then 'CO2の時の警告色はピンクにする
    mColor = Drawing.Color.Magenta
End If
mFrame.BackColor = mColor
If mRecorderInfo.RecorderNo = 3 Then
    mColor = mColor
End If
If mColor = Color.Yellow Or mColor = Color.Gold Then
    mCtrl_Data.ForeColor = Color.Black
    mCtrl_Unit.ForeColor = Color.Black
    mCtrl_Item.ForeColor = Color.Black
Else
    mCtrl_Data.ForeColor = Color.White
    mCtrl_Unit.ForeColor = Color.White
    mCtrl_Item.ForeColor = Color.White
End If

End If
Next
Next

For Each pnlName As String In mBarPnlName

    For mCh = 1 To mRecorderInfo.ChannelN
        mPanelName = pnlName & "_Ch" & mCh.ToString("00")
        mFrame = SearchCtrl(rMe, mPanelName)
        If Not IsNothing(mFrame) > 0 Then
```

```
mName = mPanelName & "_Data" 'pnlDataBarB_Ch01_Data
mCtrl_Data = SearchCtrl(rMe, mName)
If Not IsNothing(mCtrl_Data) Then
    With mCtrl_Data
        mRight = .Left + .Width
        .Text = mRecorderInfo.ChannelInfo(mCh).DataNowStr
        .Refresh()
        .Left = mRight - .Width
    End With
End If

mName = mPanelName & "_BarFrame" 'pnlDataBarB_Ch01_BarFrame
DrawBar(rMe, mName, mRecorderInfo.ChannelInfo(mCh), mCh)
If Not IsNothing(mFrame) Then
    Dim mColor As Drawing.Color
    mColor = mRecorderInfo.ChannelInfo(mCh).GetAlarmLevelColor()
    If (mCh = 5 Or mCh = 6) And mColor = Drawing.Color.Red Then
        'CO2の時の警告色はピンクにする
        mColor = Drawing.Color.Magenta
    End If
    mFrame.BackColor = mColor
End If
End If
Next

Next

End Sub

End Module
```

Module mdlRegistry

End Module