

bgoProblem - 1

Option Explicit

Private Const MODULE\_NAME = "bgoProblem"

```
Dim m_sProbEvent As String
Dim m_nHandle As Long
Dim m_bAsync As Boolean
Dim m_bAllowAddPov As Boolean
Dim m_bUseLexicon As Boolean
Dim m_bHideButtons As Boolean
Dim m_iProbFilter As Integer
Dim m_iProbFilterDefault As Integer
Dim m_bPersHistAndAct As Boolean
Dim m_bAscending As Boolean
Dim m_nSortCol As Long
Dim m_bInitialized As Boolean
Dim m_bNoRefresh As Boolean
Dim GridState As TGridState
```

Private Enum cnColumns

```
cnID
cnProvNarr
cnStatus
cnModified
cnPriority
cnNotes
cnClass
cnOnset
cnEntered
cnProvider
cnICD
cnICDName
cnIEN
cnICDIEN
cnLocIEN
cnColCount ' Should always be last
End Enum
```

Implements ICSS\_SessionEvents

Implements ICSS\_PatientEvents

Implements ICSS\_EncounterEvents

Private Sub UpdateControls()

Dim b As Boolean

```
b = CheckSecurity(True) And oPatient.Handle > 0
cmd(ADD).Enabled = b
cmd(EDIT).Enabled = b And fgData.SelectedRows > 0
cmd(DELETE).Enabled = cmd(EDIT).Enabled
mnuCmd(ADD).Enabled = cmd(ADD).Enabled
mnuCmd(EDIT).Enabled = cmd(EDIT).Enabled
mnuCmd(DELETE).Enabled = cmd(DELETE).Enabled
cmdPOV.Enabled = oEncounter.Prepare(ofNotLocked Or ofValidateOnly)
cmdPOV.Visible = m_bAllowAddPov
mnuPOV.Enabled = cmdPOV.Enabled
mnuPOV.Visible = cmdPOV.Visible
mnuSetFilter.Enabled = cboProblems.ListIndex > -1
End Sub
```

Private Sub LoadProblems(Optional NoAsync As Boolean)

On Error GoTo errHandle

```
fgData.Rows = 1
m_iProbFilter = cboProblems.ListIndex
AbortAsync
If oPatient.Handle = 0 Then Exit Sub
Status "Loading Problem List Data"
```

If m\_bAsync And Not NoAsync Then

m\_nHandle = oSession.CallRPCAsync("BGOPROB GET", oPatient.Handle, Me, True)

Else

LoadGrid CallRPCList("BGOPROB GET", oPatient.Handle)

End If

```

Status
Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "LoadProblems"
End Sub

Private Sub ApplyFilter()
    On Error GoTo errHandle

    Dim i As Integer, alt As Boolean

    With fgData
        .Redraw = flexRDNone

        For i = 1 To .Rows - 1
            Select Case cboProblems.ListIndex
                Case ptAllProblems:
                    .RowHidden(i) = False

                Case ptActiveOnly:
                    If .TextMatrix(i, cnStatus) = "Active" Then
                        .RowHidden(i) = False
                    Else
                        If m_bPersHistAndAct And .TextMatrix(i, cnStatus) = "Personal History" Then
                            .RowHidden(i) = False
                        Else
                            .RowHidden(i) = True
                        End If
                    End If

                Case ptInactiveOnly:
                    .RowHidden(i) = .TextMatrix(i, cnStatus) <> "Inactive"

                Case ptPersonalHistory:
                    .RowHidden(i) = .TextMatrix(i, cnStatus) <> "Personal History"

                Case ptFamilyHistory:
                    .RowHidden(i) = .TextMatrix(i, cnStatus) <> "Family History"
            End Select

            If Not .RowHidden(i) Then
                .Cell(flexcpBackColor, i, 0, i, cnColCount - 1) = IIf(alt, .BackColorAlternate, .BackColor)
                alt = Not alt
            End If
        Next i

        .Redraw = flexRDDirect
    End With
Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "ApplyFilter"
End Sub

Private Sub DoCommand(Command As Integer)
    ' Iterate a command over selected rows
    On Error GoTo errHandle
    Dim i As Integer, Row As Integer, s As String

    m_bNoRefresh = True

    With fgData
        For i = 0 To .SelectedRows - 1
            Row = .SelectedRow(i)
            s = ""

            If Row > 0 Then
                Select Case Command
                    Case EDIT ' Edit an entry
                        s = EditProblem(Row)
                End Select
            End If
        Next i
    End With
End Sub

```

```

        Case DELETE ' Delete an entry
            s = DeleteProblem(Row)

        Case -1      ' Add as POV
            s = AddPOV(.TextMatrix(Row, cnICDIEN), .TextMatrix(Row, cnProvNarr), .TextMatrix(Row, cnOn
set))
    End Select

    ErrorCheck s
End If
Next i
End With

m_bNoRefresh = False
RefreshGrid

Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "DoCommand"
    m_bNoRefresh = False
    RefreshGrid
End Sub

Private Sub cboProblems_Click()
    ApplyFilter
End Sub

Private Sub cmdInfo_Click()
    On Error Resume Next

    If fgData.Row < 1 Then
        With frmWebSearch
            .txtSearch.Text = ""
            .Show vbModal
        End With

        Unload frmWebSearch
        Set frmWebSearch = Nothing
        Exit Sub
    End If

    With frmBrowser
        .SearchText = fgData.TextMatrix(fgData.Row, cnProvNarr)
        .ItemIEN = fgData.TextMatrix(fgData.Row, cnICDIEN)
        .LinkType = ltICD
        .ItemName = fgData.TextMatrix(fgData.Row, cnICD) & " - " & fgData.TextMatrix(fgData.Row, cnICDName
)
        .Show vbModal
    End With

    Unload frmBrowser
    Set frmBrowser = Nothing
End Sub

Private Sub cmdPOV_Click()
    DoCommand -1
End Sub

Private Sub cmd_Click(Index As Integer)
    On Error GoTo errHandle

    If Not CheckSecurity Then Exit Sub

    Select Case Index
        Case ADD
            m_iProbFilter = cboProblems.ListIndex
            AddProblem

        Case Else
            DoCommand Index
    End Select

```

```

Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "cmd_Click"
End Sub

Private Sub fgData_AfterSort(ByVal Col As Long, Order As Integer)
    GridState.iSortCol = Col
    GridState.iSortOrder = Order
    ApplyFilter
    UpdateControls
End Sub

Private Sub fgData_BeforeSort(ByVal Col As Long, Order As Integer)
    If Col = cnPriority Or Col = cnID Then
        m_nSortCol = Col
        m_bAscending = (fgData.ColSort(Col) And 1) = 0
        fgData.ColSort(Col) = IIf(m_bAscending, 1, 2)
        Order = flexSortCustom
    End If
End Sub

Private Sub fgData_Compare(ByVal Row1 As Long, ByVal Row2 As Long, Cmp As Integer)
    On Error GoTo errHandle

    If Row1 < 1 Then Exit Sub

    With fgData
        Cmp = 0
        If m_nSortCol = cnPriority Then Cmp = IntComp(StrToIntDef(.TextMatrix(Row1, cnPriority), 999), StrToIntDef(.TextMatrix(Row2, cnPriority), 999))
        If Cmp = 0 Then Cmp = CompareProblemIDs(.TextMatrix(Row1, cnID), .TextMatrix(Row2, cnID))
        If Cmp = 0 Then Cmp = DateComp(.TextMatrix(Row1, cnModified), .TextMatrix(Row2, cnModified))
        If Not m_bAscending Then Cmp = -Cmp
    End With
    Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "fgData_Compare"
End Sub

Private Sub fgData_DblClick()
    If cmd(EDIT).Enabled Then cmd_Click EDIT
End Sub

Private Sub fgData_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    On Error GoTo errHandle

    If Button = vbRightButton Then
        With fgData
            If Shift And vbShiftMask <> 0 Then
                If .MouseRow > 0 Then .IsSelected(.MouseRow) = True
            ElseIf .MouseRow > 0 Then
                .Row = .MouseRow
            Else
                .Row = -1
            End If
        End With

        UpdateControls
        PopupMenu mnuContextPopup
    End If
    Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "fgData_MouseUp"
End Sub

Private Sub fgData_SelChange()
    UpdateControls
End Sub

Private Sub ICSS_EncounterEvents_Canceled()

```

End Sub

```
Private Sub ICSS_EncounterEvents_Committed()
    UpdateControls
End Sub
```

```
Private Function ICSS_EncounterEvents_Pending(ByVal Silent As Boolean) As String
```

End Function

```
Private Sub ICSS_PatientEvents_Canceled()
```

End Sub

```
Private Sub ICSS_PatientEvents_Committed()
    On Error GoTo errHandle

    If Not m_bInitialized Then Exit Sub

    If m_sProbEvent <> "" Then
        oSession.EventUnsubscribe m_sProbEvent, Me
    End If

    If oPatient.Handle = 0 Then
        m_sProbEvent = ""
    Else
        m_sProbEvent = "PCC." & oPatient.Handle & ".PRB"
        oSession.EventSubscribe m_sProbEvent, Me
    End If

    cboProblems.ListIndex = m_iProbFilterDefault
    LoadProblems
    Exit Sub
```

```
errHandle:
    ErrorTrap MODULE_NAME, "ICSS_PatientEvents_Committed"
End Sub
```

```
Private Function ICSS_PatientEvents_Pending(ByVal Silent As Boolean) As String
```

End Function

```
Private Sub ICSS_SessionEvents_EventCallback(ByVal EventType As String, ByVal EventStub As String)
    On Error GoTo errHandle

    If EventType = m_sProbEvent Or StartsText(EventType, "REFRESH") Then RefreshGrid
    Exit Sub
```

```
errHandle:
    ErrorTrap MODULE_NAME, "ICSS_SessionEvents_EventCallback"
End Sub
```

```
Private Sub ICSS_SessionEvents_RPCCallback(ByVal Handle As Long, ByVal Data As String)
    On Error GoTo errHandle
    If Handle <> m_nHandle Then Exit Sub
    m_nHandle = 0
    LoadGrid Data
    Exit Sub
```

```
errHandle:
    ErrorTrap MODULE_NAME, "ICSS_SessionEvents_RPCCallback"
End Sub
```

```
Private Sub ICSS_SessionEvents_RPCCallbackError(ByVal Handle As Long, ByVal ErrorCode As Long, ByVal ErrorText As String)
```

End Sub

```
Private Sub mnuPOV_Click()
    cmdPOV_Click
End Sub
```

```

Private Sub mnuCmd_Click(Index As Integer)
    cmd_Click Index
End Sub

Private Sub mnuSetFilter_Click()
    If SetSysParam("BGO PL DEFAULT FILTER", CStr(cboProblems.ListIndex)) Then m_iProbFilterDefault = cbo
Problems.ListIndex
End Sub

Private Sub UserControl_Initialize()
    On Error GoTo errHandle
    InitObjects
    InitSecurity "BGO DISABLE PROB LIST EDITING", "BGOZ PROBLEM LIST EDIT"
    AppIcon.ToolTipText = AppVersion
    SetupGrid
    m_bPersHistAndAct = oUser.HasKey("@BGO PL INCLUDE PERS HIST W ACT")
    m_bAllowAddPov = True
    m_bUseLexicon = True
    m_iProbFilterDefault = Val(GetSysParam("BGO PL DEFAULT FILTER", "0"))
    cboProblems.ListIndex = m_iProbFilterDefault
    m_bAscending = True
    Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "UserControl_Initialize"
End Sub

Private Sub UserControl_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    On Error Resume Next
    UserControl.SetFocus
End Sub

Private Sub UserControl_Resize()
    On Error Resume Next

    cmd(DELETE).Move ScaleWidth - 60 - cmd(DELETE).Width
    cmd(EDIT).Move cmd(DELETE).Left - cmd(EDIT).Width - 30
    cmd(ADD).Move cmd(EDIT).Left - cmd(ADD).Width - 30
    fgData.Height = ScaleHeight - AppIcon.Top - AppIcon.Height - 20
End Sub

Private Sub UserControl_Show()
    On Error GoTo errHandle

    m_bInitialized = True
    oSession.EventSubscribe "REFRESH", Me
    ICSS_PatientEvents_Committed
    Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "UserControl_Show"
End Sub

Private Sub UserControl_Terminate()
    On Error Resume Next

    Dim i As Integer
    DoEvents
    ReleaseObjects
    ' Clean-up forms
    For i = Forms.Count - 1 To 0 Step -1
        If Not Forms(i) Is Me Then Unload Forms(i)
    Next
End Sub

Private Sub AddProblem()
    ' Add problem to problem list.
    On Error GoTo errHandle
    Dim sProbIen As String
    Dim sId As String

    If Not CheckSecurity Then Exit Sub
    sId = CallRPCStr("BGOPROB NEXTID", oPatient.Handle)

```

```

With frmAddProblem
    .txtNarrative = ""
    .txtIcd = ""
    .m_sIcdIEN = ""
    .txtIcd.Tag = ""
    .m_sFacilityIEN = CStr(oSite.Handle)
    .lblFacility.Caption = Piece(sId, "-") & "-"
    .txtProbNum = Piece(sId, "-", 2)
    .cmdNote(ADD).Visible = False
    .cmdNote(DELETE).Visible = False
    .txtNotes.Visible = True
    .fgNotes.Visible = False
    .m_bUseLexicon = m_bUseLexicon
    .Show vbModal

If .m_bOK Then
    If cboProblems.ListIndex = ptActiveOnly And Not .optStatus(ptActiveOnly).Value Then 'Active
        cboProblems.ListIndex = ptAllProblems
    ElseIf cboProblems.ListIndex = ptInactiveOnly And Not .optStatus(ptInactiveOnly).Value Then 'Inactive
        cboProblems.ListIndex = ptAllProblems
    ElseIf cboProblems.ListIndex = ptPersonalHistory And Not .optStatus(ptPersonalHistory).Value Then 'Personal History
        cboProblems.ListIndex = ptAllProblems
    ElseIf cboProblems.ListIndex = ptFamilyHistory And Not .optStatus(ptFamilyHistory).Value Then 'Family History
        cboProblems.ListIndex = ptAllProblems
    End If

    sProbIen = .m_sProbIEN

    With fgData
        .Row = .FindRow(sProbIen, , cnIEN)
        If .Row > 0 Then
            .Select .Row, 0, .Row, .Cols - 1
            If Not .RowIsVisible(.Row) Then
                .TopRow = .Row
            End If
        End If
    End With
End If
End With

Unload frmAddProblem
Set frmAddProblem = Nothing
Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "AddProblem"
End Sub

Private Function DeleteProblem(Row As Integer) As String
    Dim s As String

    If Not Confirm("Are you sure that you wish to delete the problem titled:" & vbCrLf2 & fgData.TextMatrix(Row, cnID) & " - " & fgData.TextMatrix(Row, cnProvNarr), "Delete Problem?") Then Exit Function
    s = fgData.TextMatrix(Row, cnIEN)
    DeleteProblem = CallRPCStr("BGOPROB DEL", s)
End Function

Private Function EditProblem(Row As Integer) As String
    On Error GoTo errHandle

    Dim sProbIen As String
    Dim sPrefix As String
    Dim sId As String
    Dim i As Integer

    sProbIen = fgData.TextMatrix(Row, cnIEN)
    EditProblem = ""

    With frmAddProblem

```

```

If fgData.TextMatrix(Row, cnIEN) <> "" Then
    .m_sProbiEN = fgData.TextMatrix(Row, cnIEN)
End If

.txtNotes.Visible = False
.fgNotes.Visible = True
.txtNarrative = fgData.TextMatrix(Row, cnProvNarr)
.txtPriority.Text = fgData.TextMatrix(Row, cnPriority)

If .txtPriority.Text <> "" Then
    .udPriority.Value = CInt(fgData.TextMatrix(Row, cnPriority))
End If

If fgData.TextMatrix(Row, cnStatus) = "Active" Then
    .optStatus(ptActiveOnly).Value = True
ElseIf fgData.TextMatrix(Row, cnStatus) = "Inactive" Then
    .optStatus(ptInactiveOnly).Value = True
ElseIf fgData.TextMatrix(Row, cnStatus) = "Personal History" Then
    .optStatus(ptPersonalHistory).Value = True
ElseIf fgData.TextMatrix(Row, cnStatus) = "Family History" Then
    .optStatus(ptFamilyHistory).Value = True
End If

If fgData.TextMatrix(Row, cnOnset) <> "" Then
    .vcDat.Value = fgData.TextMatrix(Row, cnOnset)
End If

.txtNotes = fgData.TextMatrix(Row, cnNotes)
.lblEntered = fgData.TextMatrix(Row, cnEntered)
.lblModified = fgData.TextMatrix(Row, cnModified)
.m_sFacilityIEN = fgData.TextMatrix(Row, cnLocIEN)
ParseProblemID fgData.TextMatrix(Row, cnID), sPrefix, sID
.lblFacility.Caption = IIf(sPrefix = "", "", sPrefix & "-")
.txtProbNum.Text = sID
.m_sOrigID = sID
.txtIcd.Text = fgData.TextMatrix(Row, cnICDName)
.txtIcd.Tag = fgData.TextMatrix(Row, cnICDIEN)
End With

m_iProbFilter = cboProblems.ListIndex
frmAddProblem.Show vbModal

If frmAddProblem.m_bOK Then
    On Error Resume Next

    If cboProblems.ListIndex = ptActiveOnly And frmAddProblem.optStatus(ptActiveOnly).Value <> True Then
        'Active
        cboProblems.ListIndex = ptAllProblems
    ElseIf cboProblems.ListIndex = ptInactiveOnly And frmAddProblem.optStatus(ptInactiveOnly).Value <> True Then
        'Inactive
        cboProblems.ListIndex = ptAllProblems
    ElseIf cboProblems.ListIndex = ptPersonalHistory And frmAddProblem.optStatus(ptPersonalHistory).Value <> True Then
        'Personal History
        cboProblems.ListIndex = ptAllProblems
    ElseIf cboProblems.ListIndex = ptFamilyHistory And frmAddProblem.optStatus(ptFamilyHistory).Value <> True Then
        'Family History
        cboProblems.ListIndex = ptAllProblems
    End If
End If

Unload frmAddProblem
Set frmAddProblem = Nothing
Exit Function

errHandle:
    ErrorTrap MODULE_NAME, "EditProblem"
End Function

Private Sub AbortAsync()
    On Error GoTo errHandle

    If m_nHandle <> 0 Then
        oSession.CallRPCAbort m_nHandle
    End If
End Sub

```



```

    m_nHandle = 0
End If

Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "AbortAsync"
End Sub

Private Sub SetupGrid()
    On Error GoTo errHandle

    GridState.iSortCol = -1

    With fgData
        .Clear
        .Rows = 1
        .Cols = cnColCount
        .FixedCols = 0
        .TextMatrix(0, cnID) = "ID"
        .TextMatrix(0, cnProvNarr) = "Provider Narrative"
        .TextMatrix(0, cnStatus) = "Status"
        .TextMatrix(0, cnModified) = "Modified"
        .ColDataType(cnModified) = flexDTDate
        .TextMatrix(0, cnPriority) = "Priority"
        .ColWidth(cnPriority) = 600
        .ColAlignment(cnPriority) = flexAlignCenterCenter
        .TextMatrix(0, cnOnset) = "Onset"
        .ColDataType(cnOnset) = flexDTDate
        .TextMatrix(0, cnNotes) = "Notes"
        .TextMatrix(0, cnClass) = "Class"
        .TextMatrix(0, cnEntered) = "Entered"
        .ColDataType(cnEntered) = flexDTDate
        .TextMatrix(0, cnProvider) = "Provider"
        .TextMatrix(0, cnICD) = "ICD"
        .TextMatrix(0, cnICDName) = "ICD Name"
        .TextMatrix(0, cnIEN) = "Ien"
        .TextMatrix(0, cnICDIEN) = "ICD Ien"
        .TextMatrix(0, cnLocIEN) = "Facility Ien"
        .ColWidth(cnID) = 750
        .ColWidth(cnProvNarr) = 2700
        .ColWidth(cnStatus) = 700
        .ColWidth(cnModified) = 1000
        .ColWidth(cnOnset) = 1000
        .ColWidth(cnNotes) = 3000
        .ColWidth(cnICD) = 700
        .ColWidth(cnICDName) = 2000
        .ColHidden(cnClass) = True
        .ColHidden(cnEntered) = True
        .ColHidden(cnProvider) = True
        .ColHidden(cnIEN) = True
        .ColHidden(cnICDIEN) = True
        .ColHidden(cnLocIEN) = True
        .ColAlignment(cnICD) = flexAlignLeftCenter
        .ColAlignment(cnOnset) = flexAlignCenterCenter
        .ColAlignment(cnModified) = flexAlignCenterCenter
    End With

    With cboProblems
        .Clear
        .AddItem "All Problems"
        .AddItem "Active Only"
        .AddItem "Inactive Only"
        .AddItem "Personal History"
        .AddItem "Family History"
        .ListIndex = ptAllProblems
    End With
Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "SetupGrid"
End Sub

```

```

Private Sub RefreshGrid()
    If m_bNoRefresh Then Exit Sub
    SaveGridState
    LoadProblems True
    RestoreGridState
End Sub

Private Sub SaveGridState()
    ' Saves selected row and sort info
    Dim i As Integer
    GridState.sRowID = U

    With fgData
        For i = 0 To .SelectedRows - 1
            If .SelectedRow(i) > 0 Then
                GridState.sRowID = GridState.sRowID & .RowData(.SelectedRow(i)) & U
            End If
        Next i
    End With
End Sub

Private Sub CancelSelection()
    fgData.Select 0, 0, 0, 0
    fgData.IsSelected(0) = False
End Sub

Private Sub RestoreGridState()
    ' Restores selected row and sort info
    Dim Row As Integer

    With fgData
        If .Rows > 1 And GridState.iSortOrder > 0 Then
            .Select 1, GridState.iSortCol, .Rows - 1
            .Sort = GridState.iSortOrder
        End If
    End With

    CancelSelection

    If GridState.sRowID <> U Then
        For Row = 1 To .Rows - 1
            .IsSelected(Row) = ContainsText(GridState.sRowID, U + .RowData(Row) + U)
        Next Row
    End If
End With
End Sub

Private Sub LoadGrid(sData As String)
    On Error GoTo errHandle
    Dim i As Integer, v() As String, s As String
    Dim bHasPriority As Boolean

    bHasPriority = False
    fgData.Rows = 1
    UpdateControls
    If sData = "" Or ErrorCheck(sData) Then Exit Sub
    v = Split(sData, vbCrLf)

    ' Number Code [1] ^ Patient IEN [2] ^ ICD Code [3] ^ Modify Date [4] ^ Class [5] ^
    ' Provider Narrative [6] ^ Date Entered [7] ^ Status [8] ^ Date Onset [9] ^
    ' Problem IEN [10] ^ Notes [11] ^ ICD9 IEN [12] ^ ICD9 Short Name [13] ^ Provider [14] ^
    ' Facility IEN [15] ^ Priority [16]

    ' e.g.,
    ' TC9^72101^472.0^05/23/2001^U^ATROPHIC RHINITIS^05/23/2001^A^^27771^^9102^CHRONIC RHINITIS^^3987^

    With fgData
        For i = 1 To UBound(v)
            s = v(i - 1)
            .Rows = .Rows + 1
            .RowData(i) = Piece(s, U, 10)
            .TextMatrix(i, cnID) = Piece(s, U) 'id
            .TextMatrix(i, cnProvNarr) = Piece(s, U, 6) 'narrative
        Next i
    End With

```

```

.TextMatrix(i, cnStatus) = IIf(Piece(s, U, 8) = "A", "Active", "Inactive") 'status
.TextMatrix(i, cnModified) = Piece(s, U, 4) 'modified
.TextMatrix(i, cnPriority) = Piece(s, U, 16) 'priority
If Piece(s, U, 16) <> "" Then bHasPriority = True
.TextMatrix(i, cnOnset) = Piece(s, U, 9) 'onset
.TextMatrix(i, cnNotes) = StrConv(Piece(s, U, 11), vbProperCase) 'notes

If Piece(s, U, 5) = "P" Then      'class
    .TextMatrix(i, cnStatus) = "Personal History"
ElseIf Piece(s, U, 5) = "F" Then
    .TextMatrix(i, cnStatus) = "Family History"
End If

.TextMatrix(i, cnEntered) = Piece(s, U, 7) 'entered
.TextMatrix(i, cnProvider) = StrConv(Replace(Piece(s, U, 14), ",", " ", vbProperCase) 'Provide
r

.TextMatrix(i, cnICD) = Piece(s, U, 3) 'icd
.TextMatrix(i, cnICDName) = Piece(s, U, 13) 'icd name
.TextMatrix(i, cnIEN) = Piece(s, U, 10) 'prob IEN
.TextMatrix(i, cnICDIEN) = Piece(s, U, 12) 'icd IEN
.TextMatrix(i, cnLocIEN) = Piece(s, U, 15) 'facility IEN
Next i

If GridState.iSortCol < 0 And .Rows > 1 Then 'Use default collation
    m_nSortCol = IIf(bHasPriority, cnPriority, cnID)
    m_bAscending = True
    .Select 1, m_nSortCol, .Rows - 1
    .Sort = flexSortCustom
End If

.AutoSize 0, 10
CancelSelection
.Redraw = True
End With

ApplyFilter
UpdateControls
Exit Sub

errHandle:
    ErrorTrap MODULE_NAME, "LoadGrid"
End Sub

Public Property Get RunAsync() As Boolean
    RunAsync = m_bAsync
End Property

Public Property Let RunAsync(ByVal Value As Boolean)
    m_bAsync = Value
End Property

Public Property Get AllowAddPOV() As Boolean
    AllowAddPOV = m_bAllowAddPov
End Property

Public Property Let AllowAddPOV(ByVal Value As Boolean)
    m_bAllowAddPov = Value
End Property

Public Property Get HideButtons() As Boolean
    HideButtons = m_bHideButtons
End Property

Public Property Let HideButtons(ByVal Value As Boolean)
    m_bHideButtons = Value
    cmdPOV.Visible = Not m_bHideButtons
    cmd(ADD).Visible = Not m_bHideButtons
    cmd(EDIT).Visible = Not m_bHideButtons
    cmd(DELETE).Visible = Not m_bHideButtons
End Property

Public Property Get UseLexicon() As Boolean
    UseLexicon = m_bUseLexicon

```

End Property

Public Property Let UseLexicon(ByVal Value As Boolean)

    m\_bUseLexicon = Value

End Property

```
VERSION 5.00
Object = "{1C0489F8-9EFD-423D-887A-315387F18C8F}#1.0#0"; "vsflex8l.ocx"
Begin VB.UserControl bgoProblem
    ClientHeight = 3975
    ClientLeft = 0
    ClientTop = 0
    ClientWidth = 7860
    ScaleHeight = 3975
    ScaleWidth = 7860
    Begin VB.CommandButton cmdInfo
        Height = 345
        Left = 2040
        Picture = (Icon)
        Style = 1 'Graphical
        TabIndex = 7
        ToolTipText = "Search Web Reference Sites"
        Top = 80
        Width = 405
    End
    Begin VSFlex8Lctl.VSFlexGrid fgData
        Align = 2 'Align Bottom
        Height = 3435
        Left = 0
        TabIndex = 3
        Top = 540
        Width = 7860
        _cx = 13864
        _cy = 6059
        Appearance = 1
        BorderStyle = 1
        Enabled = -1 'True
        BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
            Name = "MS Sans Serif"
            Size = 8.25
            Charset = 0
            Weight = 400
            Underline = 0 'False
            Italic = 0 'False
            Strikethrough = 0 'False
        EndProperty
        MousePointer = 0
        BackColor = -2147483643
        ForeColor = -2147483640
        BackColorFixed = -2147483633
        ForeColorFixed = -2147483630
        BackColorSel = -2147483635
        ForeColorSel = -2147483634
        BackColorBkg = -2147483636
        BackColorAlternate = 15790320
        GridColor = -2147483633
        GridColorFixed = -2147483632
        TreeColor = -2147483632
        FloodColor = 192
        SheetBorder = -2147483642
        FocusRect = 1
        HighLight = 1
        AllowSelection = -1 'True
        AllowBigSelection = -1 'True
        AllowUserResizing = 1
        SelectionMode = 3
        GridLines = 1
        GridLinesFixed = 2
        GridLineWidth = 1
        Rows = 50
        Cols = 10
        FixedRows = 1
        FixedCols = 1
        RowHeightMin = 0
        RowHeightMax = 0
        ColWidthMin = 0
        ColWidthMax = 0
        ExtendLastCol = -1 'True
        FormatString = ""
    End
End
```

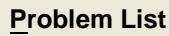
```

ScrollTrack      = 0    'False
ScrollBars      = 3
ScrollTips      = 0    'False
MergeCells      = 0
MergeCompare    = 0
AutoResize      = -1   'True
AutoSizeMode    = 1
AutoSearch      = 0
AutoSearchDelay = 2
MultiTotals     = -1   'True
SubtotalPosition= 1
OutlineBar      = 0
OutlineCol      = 0
Ellipsis        = 0
ExplorerBar     = 5
PicturesOver    = 0    'False
FillStyle       = 0
RightToLeft     = 0    'False
PictureType     = 0
TabBehavior     = 0
OwnerDraw       = 0
Editable        = 0
ShowComboButton = 1
WordWrap        = -1   'True
TextStyle       = 0
TextStyleFixed  = 0
OleDragMode     = 0
OleDropMode     = 0
ComboSearch     = 3
AutoSizeMouse   = -1   'True
FrozenRows      = 0
FrozenCols      = 0
AllowUserFreezing= 0
BackColorFrozen = 0
ForeColorFrozen = 0
WallPaperAlignment= 9
AccessibleName  = ""
AccessibleDescription= ""
AccessibleValue = ""
AccessibleRole  = 24
End
Begin VB.CommandButton cmdPOV
    Caption      = "&Set as Today's POV"
    Height       = 360
    Left         = 3960
    TabIndex     = 2
    Top          = 80
    Width        = 1575
End
Begin VB.CommandButton cmd
    Caption      = "&Delete"
    Enabled      = 0    'False
    Height       = 360
    Index        = 2
    Left         = 7110
    TabIndex     = 6
    Top          = 80
    Width        = 675
End
Begin VB.CommandButton cmd
    Caption      = "&Edit "
    Enabled      = 0    'False
    Height       = 360
    Index        = 1
    Left         = 6410
    TabIndex     = 5
    Top          = 80
    Width        = 675
End
Begin VB.CommandButton cmd
    Caption      = "&Add"
    Enabled      = 0    'False
    Height       = 360

```

```

    Index          = 0
    Left           = 5700
    TabIndex       = 4
    Top            = 80
    Width          = 675
End
Begin VB.ComboBox cboProblems
    Height         = 315
    Left           = 2520
    Style          = 2 'Dropdown List
    TabIndex       = 1
    Top            = 90
    Width          = 1365
End
Begin VB.Image AppIcon
    Height         = 480
    Left           = 30
    Picture        = (Icon)
    Stretch        = -1 'True
    Top            = 30
    Width          = 480
End
Begin VB.Label lbl
    Alignment      = 2 'Center
    Caption        = " &Problem List"
    BeginProperty Font
        Name        = "MS Sans Serif"
        Size        = 9.75
        Charset     = 0
        Weight      = 700
        Underline   = 0 'False
        Italic      = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height        = 270
    Index         = 36
    Left          = 600
    TabIndex      = 0
    Top           = 30
    Width         = 1365
End
Begin VB.Menu mnuContextPopup
    Caption        = "mnuContextPopup"
    Begin VB.Menu mnuPOV
        Caption     = "&Set as Today's POV"
    End
    Begin VB.Menu mnuLine1
        Caption     = "-"
    End
    Begin VB.Menu mnuCmd
        Caption     = "&Add New Problem"
        Index       = 0
    End
    Begin VB.Menu mnuCmd
        Caption     = "&Edit Problem"
        Index       = 1
    End
    Begin VB.Menu mnuCmd
        Caption     = "&Delete Problem"
        Index       = 2
    End
    Begin VB.Menu mnuLine2
        Caption     = "-"
    End
    Begin VB.Menu mnuSetFilter
        Caption     = "Set Current &Filter as Default"
    End
End
End
End
```



Set as Today's POV

Add

Edit

Delete

[illegible]