' AID 44624

' April 6, 2011

' Rockwell Automation

' FactoryTalk View 6.0 CPR9 SR3

Option Explicit

Private Const ROOT As String = "/"

' \*\*\*\* USER-CONFIGURABLE ENTRIES START HERE \*\*\*\*

'First HMI Server

Private Const HMISERVERNAME As String = "InstantFizz\_HMI" 'sample HMI server name - change to actual name

Private Const HMIAREANAME As String = "/HMI\_Area" 'sample area for HMI server - change to actual name

'First Data Server (includes Device-Based FTAE)

Private Const DATASERVERNAME As String = "RSLinx Enterprise" 'default OPC Data server name - change to actual name

Private Const DATAAREANAME As String = "/Data\_Area" 'sample

'First Tag-Based FTAE Server, if defined

Private Const FTAETAGSERVERNAME As String = "FTAETag\_Server" 'sample Tag-Based FTAE server name - change to actual name

Private Const FTAETAGAREANAME As String = "/FTAETag\_Area" 'sample area for Tag-Based FTAE server - change to actual name

Private Const NUMBEROFHMISERVERS = 1 'max = 1, modify to suit actual number of HMI servers in app

Private Const NUMBEROFDATASERVERS = 1 'max = 1, modify to suit actual number of DATA servers in app, with or without FTAE enabled

Private Const NUMBEROFFTAETAGSERVERS = 1 'max = 1, modify to suit actual number of FTAE Tag-Based servers in app

' Total number is the number of your servers + 1 for the FactoryTalk Directory.

' For example: HMI Server, Data Server (RSLinx), Tag-Based A&E = 3 + 1 for FTD

' Example total would be 4.

Private Const NUMBEROFSERVERGROUPS = 4

' \*\*\*\* USER-CONFIGURABLE ENTRIES END HERE \*\*\*\*

'max number of servers our screen is configured for.

Private Const MAXNUMBEROFSERVERGROUPS = 4

'Second HMI Server, if defined

Private Const HMISERVERNAME\_2 As String = "N/A" 'sample HMI server name - change to actual name

Private Const HMIAREANAME\_2 As String = "N/A" 'sample area for HMI server - change to actual name

'Second Data Server, if defined (includes Device-Based FTAE)

Private Const DATASERVERNAME\_2 As String = "N/A" 'default OPC Data server name - change to actual name

Private Const DATAAREANAME\_2 As String = "N/A" 'sample

'Second Tag-Based FTAE Server, if defined

Private Const FTAETAGSERVERNAME\_2 As String = "N/A" 'sample Tag-Based FTAE server name - change to actual name

Private Const FTAETAGAREANAME\_2 As String = "N/A" 'sample area for Tag-Based FTAE server - change to actual name

'Assignments of GRP constants used to display graphics

' GRP\_x assignments correspond to the graphic. If you reorganize or

' want to reassign, you'll have to go through the code reassign everywhere.

Private Const GRP\_1 = 1 'Group1 status-state indicator - FTD

' HMI Server

Private Const GRP\_2 = 2 'Group2 status-state indicator - HMI Server

' Data Server

Private Const GRP\_3 = 3 'Group3 status-state indicator - Data Server

' FTAE Tag-Based

Private Const GRP\_4 = 4 'Group4 status-state indicator - FTAE Tag-Based

' HMI Server 2

Private Const GRP\_5 = 5 'Group5 status-state indicator - HMI Server 2

' Data Server 2

Private Const GRP\_6 = 6 'Group6 status-state indicator - Data Server 2

' FTAE Tag-Based 2

Private Const GRP\_7 = 7 'Group7 status-state indicator - FTAE Tag-Based 2

Private sHMIAreaName As String 'default HMI area

Private sHMISvrName As String 'default HMI server

Private sHMIAreaName\_2 As String 'default HMI area

Private sHMISvrName\_2 As String 'default HMI server

Private sDATAreaName As String 'default Data area

Private sDATSvrName As String 'default Data server

Private sDATAreaName\_2 As String 'default Data area

Private sDATSvrName\_2 As String 'default Data server

Private sFTAETagAreaName As String 'FTAE Area

Private sFTAETagSvrName As String 'FTAE Server

Private sFTAETagAreaName\_2 As String 'FTAE Area

Private sFTAETagSvrName\_2 As String 'FTAE Server

'adding 1 to array definition since all our indexing starts with 1 instead of 0.

Private sAreaName(MAXNUMBEROFSERVERGROUPS + 1) As String

Private sServername(MAXNUMBEROFSERVERGROUPS + 1) As String

Private sFullServerName(MAXNUMBEROFSERVERGROUPS + 1) As String

Private WithEvents appStatus As Application 'app events used to update real time status

Private sHMIArray(NUMBEROFHMISERVERS + 1) As String 'array to search multiple HMI servers + 1 b/c search starts at 1

Private sDATAArray(NUMBEROFDATASERVERS + 1) As String 'array to search multiple DATA servers + 1 b/c search starts at 1

Private sFTAETagArray(NUMBEROFFTAETAGSERVERS + 1) As String 'array to search multiple DATA servers + 1 b/c search starts at 1

Private Sub Display\_AnimationStart()

Dim lGroup As Long

On Error GoTo ErrHandler

'general initialization

'appStatus ServiceDisruption/ServiceRecovery events used to update real time status

Set appStatus = Me.Application

'initialize for default servers

sHMIAreaName = HMIAREANAME

sHMISvrName = HMISERVERNAME

sFTAETagAreaName = FTAETAGAREANAME

sFTAETagSvrName = FTAETAGSERVERNAME

sDATAreaName = DATAAREANAME

sDATSvrName = DATASERVERNAME

'array only required if using more than one server of each type

'if multiple servers populate arrays with all respective servernames

If NUMBEROFHMISERVERS > 0 Then

sHMIArray(1) = sHMISvrName

HMI\_Group2.Visible = True

Else

HMI\_Group2.Visible = False

End If

If NUMBEROFDATASERVERS > 0 Then

sDATAArray(1) = sDATSvrName

Data\_Group3.Visible = True

Else

Data\_Group3.Visible = False

End If

If NUMBEROFFTAETAGSERVERS > 0 Then

sFTAETagArray(1) = sFTAETagSvrName

FTAETag\_Group4.Visible = True

Else

FTAETag\_Group4.Visible = False

End If

'set default status for GRP\_1 server: FTD

sAreaName(GRP\_1) = "Not Applicable"

sServername(GRP\_1) = "FTDirectory"

sFullServerName(GRP\_1) = sServername(GRP\_1)

'set default status for GRP\_2 server: HMI Server

sAreaName(GRP\_2) = sHMIAreaName

sServername(GRP\_2) = sHMISvrName

sFullServerName(GRP\_2) = sAreaName(GRP\_2) & ":" & sServername(GRP\_2)

'set default status for GRP\_3 server: Data Server

sAreaName(GRP\_3) = sDATAreaName

sServername(GRP\_3) = sDATSvrName

sFullServerName(GRP\_3) = sAreaName(GRP\_3) & ":" & sServername(GRP\_3)

'set default status for GRP\_4 server: FTAE Tag-Based

sAreaName(GRP\_4) = sFTAETagAreaName

sServername(GRP\_4) = sFTAETagSvrName

sFullServerName(GRP\_4) = sAreaName(GRP\_4) & ":" & sServername(GRP\_4)

For lGroup = 1 To MAXNUMBEROFSERVERGROUPS

ConfigServerStatus lGroup

ConfigServerState lGroup

Next

Exit Sub

ErrHandler:

LogDiagnosticsMessage "VBA Error (Display\_AnimationStart()): " & Hex(Err.Number) & " - " & Err.Description, ftDiagSeverityError

End Sub

Private Sub appStatus\_ServerStateChanged(ByVal Area As String, ByVal ServerName As String, ByVal ComputerName As String, ByVal State As gfxServerStateConstants)

Dim lGroup As Long

For lGroup = 1 To NUMBEROFSERVERGROUPS

ConfigServerState lGroup

Next

End Sub

Private Sub appStatus\_ServiceDisruption(ByVal Area As String, ByVal ServerName As String, ByVal ComputerName As String)

Dim lGroup As Long

For lGroup = 1 To NUMBEROFSERVERGROUPS

ConfigServerStatus lGroup

Next

End Sub

Private Sub appStatus\_ServiceRecovery(ByVal Area As String, ByVal ServerName As String, ByVal ComputerName As String)

Dim lGroup As Long

For lGroup = 1 To NUMBEROFSERVERGROUPS

ConfigServerStatus lGroup

Next

End Sub

Private Sub Display\_BeforeAnimationStop()

Set appStatus = Nothing

End Sub

Public Sub SetServerStatus(lGroup As Long, sscStatus As gfxServerStatusConstants, bPrimaryServer As Boolean)

Dim sObjectName As String

On Error GoTo ErrHandler

'determine whether primary or secondary

If bPrimaryServer Then

sObjectName = "polyPrimaryStatus"

Else

sObjectName = "polySecondaryStatus"

End If

'set status colour

If sscStatus = gfxServerStatusActive Then

Elements.Item(sObjectName & lGroup).BackColor = vbGreen

ElseIf sscStatus = gfxServerStatusStandby Then

Elements.Item(sObjectName & lGroup).BackColor = vbYellow

ElseIf sscStatus = gfxServerStatusOutOfService Then

Elements.Item(sObjectName & lGroup).BackColor = vbRed

Else

Elements.Item(sObjectName & lGroup).BackColor = vbBlack

End If

Exit Sub

ErrHandler:

LogDiagnosticsMessage "VBA Error (SetServerStatus()): " & Hex(Err.Number) & " - " & Err.Description, ftDiagSeverityError

End Sub

Public Sub ConfigServerStatus(lGroup As Long)

Dim strActiveComputerName As String

Dim sscPrimaryStatus As gfxServerStatusConstants

Dim sscSecondaryStatus As gfxServerStatusConstants

On Error GoTo ErrHandler

'Display name of application

txtApplicationName.Caption = Application.ApplicationName

'determining server status and name

GetServerStatus sFullServerName(lGroup), sscPrimaryStatus, sscSecondaryStatus, strActiveComputerName

Elements.Item("txtAreaName" & lGroup).Caption = sAreaName(lGroup)

Elements.Item("txtServerName" & lGroup).Caption = sServername(lGroup)

Elements.Item("txtComputerName" & lGroup).Caption = strActiveComputerName

SetServerStatus lGroup, sscPrimaryStatus, True

SetServerStatus lGroup, sscSecondaryStatus, False

Exit Sub

ErrHandler:

Select Case Hex(Err.Number)

Case 80040488

sscPrimaryStatus = gfxServerStatusOutOfService

sscSecondaryStatus = gfxServerStatusSecondaryNotDefined

strActiveComputerName = "Server NOT found!"

Resume Next

Case Else

LogDiagnosticsMessage "VBA Error (ConfigServerStatus()): " & Hex(Err.Number) & " - " & Err.Description, ftDiagSeverityError

End Select

End Sub

Public Sub ConfigServerState(lGroup As Long)

Dim sscPrimaryState As gfxServerStateConstants

Dim sscSecondaryState As gfxServerStateConstants

On Error GoTo ErrHandler

GetServerState sFullServerName(lGroup), sscPrimaryState, sscSecondaryState

SetServerState lGroup, sscPrimaryState, True

SetServerState lGroup, sscSecondaryState, False

Exit Sub

ErrHandler:

Select Case Hex(Err.Number)

Case 80040490

sscPrimaryState = gfxServerStateNotLoaded

sscSecondaryState = gfxServerStateSecondaryNotDefined

Resume Next

Case Else

LogDiagnosticsMessage "VBA Error (ConfigServerState()): " & Hex(Err.Number) & " - " & Err.Description, ftDiagSeverityError

End Select

End Sub

Public Sub SetServerState(lGroup As Long, sscState As gfxServerStateConstants, bPrimaryServer As Boolean)

Dim sObjectName As String

Dim bFTServer As Boolean

On Error GoTo ErrHandler

'determine whether primary or secondary

If bPrimaryServer Then

sObjectName = "txtPrimaryState"

Else

sObjectName = "txtSecondaryState"

End If

'determine whether FTserver

If StrComp(UCase(Elements.Item("txtServerName" & lGroup).Caption), "FTDIRECTORY", vbTextCompare) = 0 Then

bFTServer = True

Else

bFTServer = False

End If

With Elements.Item(sObjectName & lGroup)

Select Case sscState

Case gfxServerStateActive

.ForeColor = vbBlack

.Caption = "Active"

Case gfxServerStateActiveNoPartner

.ForeColor = vbBlack

If bFTServer Then

.Caption = "Active" 'FTServer cannot be made redundant: will never have partner

Else

.Caption = "Active - No Partner"

End If

Case gfxServerStateActiveSync

.ForeColor = vbBlack

.Caption = "Active - Synchronizing"

Case gfxServerStateCommError

.ForeColor = vbBlack

.Caption = "Communication Error"

Case gfxServerStateLoading

.ForeColor = vbBlack

.Caption = "Server Loading"

Case gfxServerStateNotInUse

.ForeColor = vbBlack

.Caption = "Out of Service - Manual"

Case gfxServerStateNotLoaded

.ForeColor = vbBlack

.Caption = "Server Not Loaded"

Case gfxServerStateOutofService

.ForeColor = vbBlack

.Caption = "Out of Service - Failure"

Case gfxServerStateReady

.ForeColor = vbBlack

.Caption = "Ready"

Case gfxServerStateReadyToBeActive

.ForeColor = vbBlack

.Caption = "Ready to be Active"

Case gfxServerStateReadyToBeStandby

.ForeColor = vbBlack

.Caption = "Ready to be Standby"

Case gfxServerStateStandby

.ForeColor = vbBlack

.Caption = "Standby"

Case gfxServerStateStandbySync

.ForeColor = vbBlack

.Caption = "Standby - Synchronizing"

Case gfxServerStateStarting

.ForeColor = vbBlack

.Caption = "Server Starting"

Case gfxServerStateSecondaryNotDefined

.ForeColor = vbWhite

If bFTServer Then

.Caption = "Not Applicable"

Else

.Caption = "Secondary Not Defined"

End If

Case Else

.ForeColor = vbWhite

.Caption = "Not Defined"

End Select

End With

Exit Sub

ErrHandler:

LogDiagnosticsMessage "VBA Error (SetServerState()): " & Hex(Err.Number) & " - " & Err.Description, ftDiagSeverityError

End Sub

Public Function SelectServer(lGroup As Long) As Boolean

Dim sTemp As String

Dim bServerSelected As Boolean

On Error GoTo ErrHandler

bServerSelected = False

'determine applicable group default area

If lGroup = 2 Then

sTemp = sHMIAreaName

ElseIf lGroup = 3 Then

sTemp = sDATAreaName

ElseIf lGroup = 4 Then

sTemp = sFTAETagSvrName

ElseIf lGroup = 5 Then

sTemp = sHMIAreaName\_2

ElseIf lGroup = 6 Then

sTemp = sDATAreaName\_2

ElseIf lGroup = 7 Then

sTemp = sFTAETagSvrName\_2

Else

sTemp = ROOT

End If

sTemp = InputBox("Enter name of Area: ", , sTemp)

'if no area defined proceed only if FTDirectory

If sTemp = "" Or sTemp = "Not Applicable" Then

sTemp = InputBox("Enter server name: ", , "FTDirectory")

If StrComp(UCase(sTemp), "FTDIRECTORY", vbTextCompare) = 0 Then

sAreaName(lGroup) = "Not Applicable"

sServername(lGroup) = sTemp

sFullServerName(lGroup) = sServername(lGroup)

bServerSelected = True

End If

Else

sAreaName(lGroup) = Trim(sTemp)

'determine applicable group default server

If lGroup = 2 Then

sTemp = sHMISvrName

ElseIf lGroup = 3 Then

sTemp = sDATSvrName

ElseIf lGroup = 4 Then

sTemp = sFTAETagSvrName

ElseIf lGroup = 5 Then

sTemp = sHMISvrName\_2

ElseIf lGroup = 6 Then

sTemp = sDATSvrName\_2

ElseIf lGroup = 7 Then

sTemp = sFTAETagSvrName\_2

Else

sTemp = "FTDirectory"

End If

sTemp = InputBox("Enter server name: ", , sTemp)

If sTemp <> "" Then

If StrComp(UCase(sTemp), "FTDIRECTORY", vbTextCompare) = 0 Then

sAreaName(lGroup) = "Not Applicable"

sServername(lGroup) = sTemp

sFullServerName(lGroup) = sServername(lGroup)

bServerSelected = True

Else

sServername(lGroup) = Trim(sTemp)

sFullServerName(lGroup) = sAreaName(lGroup) & ":" & sServername(lGroup)

bServerSelected = True

End If

End If

End If

SelectServer = bServerSelected

Exit Function

ErrHandler:

LogDiagnosticsMessage "VBA Error (SelectServer()): " & Hex(Err.Number) & " - " & Err.Description, ftDiagSeverityError

SelectServer = False

End Function

Public Sub ConfigColumnHeadings(lGroup As Long)

Dim lServ As Long

Dim bItemFound As Boolean

On Error GoTo ErrHandler

bItemFound = False

If StrComp(UCase(sServername(lGroup)), "FTDIRECTORY", vbTextCompare) = 0 Then

Elements.Item("txtServer" & lGroup).Caption = "FT Server"

Else

For lServ = 1 To NUMBEROFHMISERVERS

If StrComp(UCase(sHMIArray(lServ)), UCase(sServername(lGroup)), vbTextCompare) = 0 Then

Elements.Item("txtServer" & lGroup).Caption = "HMI Server"

bItemFound = True

End If

Next lServ

If bItemFound Then Exit Sub

For lServ = 1 To NUMBEROFDATASERVERS

If StrComp(UCase(sDATAArray(lServ)), UCase(sServername(lGroup)), vbTextCompare) = 0 Then

Elements.Item("txtServer" & lGroup).Caption = "Data Server"

bItemFound = True

End If

Next lServ

For lServ = 1 To NUMBEROFFTAETAGSERVERS

If StrComp(UCase(sFTAETagArray(lServ)), UCase(sServername(lGroup)), vbTextCompare) = 0 Then

Elements.Item("txtServer" & lGroup).Caption = "Data Server"

bItemFound = True

End If

Next lServ

If Not bItemFound Then

Elements.Item("txtServer" & lGroup).Caption = "Server"

End If

End If

Exit Sub

ErrHandler:

LogDiagnosticsMessage "VBA Error (ConfigColumnHeadings()): " & Hex(Err.Number) & " - " & Err.Description, ftDiagSeverityError

End Sub

Private Sub btnServer2\_Released()

Dim bServerSelected As Boolean

bServerSelected = SelectServer(GRP\_2)

If bServerSelected Then

ConfigColumnHeadings GRP\_2

ConfigServerStatus GRP\_2

ConfigServerState GRP\_2

End If

End Sub

Private Sub btnServer3\_Released()

Dim bServerSelected As Boolean

bServerSelected = SelectServer(GRP\_3)

If bServerSelected Then

ConfigColumnHeadings GRP\_3

ConfigServerStatus GRP\_3

ConfigServerState GRP\_3

End If

End Sub

Private Sub btnServer4\_Released()

Dim bServerSelected As Boolean

bServerSelected = SelectServer(GRP\_4)

If bServerSelected Then

ConfigColumnHeadings GRP\_4

ConfigServerStatus GRP\_4

ConfigServerState GRP\_4

End If

End Sub