```
    'make sure that the entry in the text portion exists in the drop down list of the combobox
    Dim found As Boolean = False
    For i As Integer = 0 To ComboBoxAssignedRegion.Items.Count - 1
        If ComboBoxAssignedRegion.Text = ComboBoxAssignedRegion.Items(i).ToString Then
            found = True
            Exit For
        End If
    Next
    If Not found Then
        MessageBoxUtilities.MyMessageBox("Region must be selected from the drop down list.", MsgBoxStyle.OkOnly, "Unrecognized Region")
        Exit Sub
    End If

    If SQLNonQueryBactesRequest("Update tbl_BactesComputers set Region = '" & ComboBoxAssignedRegion.Text & "' " & _
      "where computername = '" & ListBoxComputers.Items(ListBoxComputers.SelectedIndex).ToString & "' ") = 0 Then

    End If
End Sub
```

```
Private Sub ButtonUpdateWorkstationAssignedUser_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonUpdateWorkstationAssignedUser.Click
    If ListBoxComputers.SelectedIndex = -1 Then
        MessageBoxUtilities.MyMessageBox("Please select a PC first.")
        Exit Sub
    End If

    'make sure that the entry in the text portion exists in the drop down list of the combobox
    Dim found As Boolean = False
    For i As Integer = 0 To ComboBoxAssignedUser.Items.Count - 1
        If ComboBoxAssignedUser.Text = ComboBoxAssignedUser.Items(i).ToString Then
            found = True
            Exit For
        End If
    Next
    If Not found Then
        MessageBoxUtilities.MyMessageBox("User must be selected from the drop down list.", MsgBoxStyle.OkOnly, "Unrecognized User")
        Exit Sub
    End If

    If SQLNonQueryBactesRequest("Update tbl_BactesComputers set UserID = '" & ComboBoxAssignedUser.Text & "' " & _
      "where computername = '" & ListBoxComputers.Items(ListBoxComputers.SelectedIndex).ToString & "' ") = 0 Then

    End If

End Sub
```

```vb
Try
    cnLocations.Open()
    Dim sdrLocations As SqlDataReader = cmLocations.ExecuteReader
    With sdrLocations
        If .HasRows Then
            While .Read()
                Dim SelectedClinic As String = .GetString(0).ToUpper()
                For i As Integer = 0 To CheckedListBoxDistroLocations.Items.Count - 1
                    If CheckedListBoxDistroLocations.Items(i).ToString = SelectedClinic Then
                        CheckedListBoxDistroLocations.SetItemChecked(i, True)
                    End If
                Next
            End While
        End If
    End With
Catch ex As Exception
    Logger.Warning(ex)

End Try
cnLocations.Close()

'Update checks in Worrkstations
Dim cnWorkstations As New SqlConnection(SessionDBs.BIMRequestConnectionString)
strsql = "Select WorkstationName " & _
"from tbl_distroWorkstations a inner join tbl_distroheader b on a.distroid = b.distroid " & _
"where name = '" & DistroGroup & "'" & _
"order by WorkstationName"
Dim cmWorkstations As New SqlCommand(strsql, cnWorkstations)

Try
    cnWorkstations.Open()
    Dim sdrWorkstations As SqlDataReader = cmWorkstations.ExecuteReader
    With sdrWorkstations
        If .HasRows Then
            While .Read()
                Dim SelectedLocation As String = .GetString(0).ToUpper()
                For i As Integer = 0 To CheckedListBoxDistroWorkstations.Items.Count - 1
                    If CheckedListBoxDistroWorkstations.Items(i).ToString = SelectedLocation Then
                        CheckedListBoxDistroWorkstations.SetItemChecked(i, True)
                    End If
                Next
            End While
        End If
    End With
Catch ex As Exception
    Logger.Warning(ex)

End Try
cnWorkstations.Close()
```

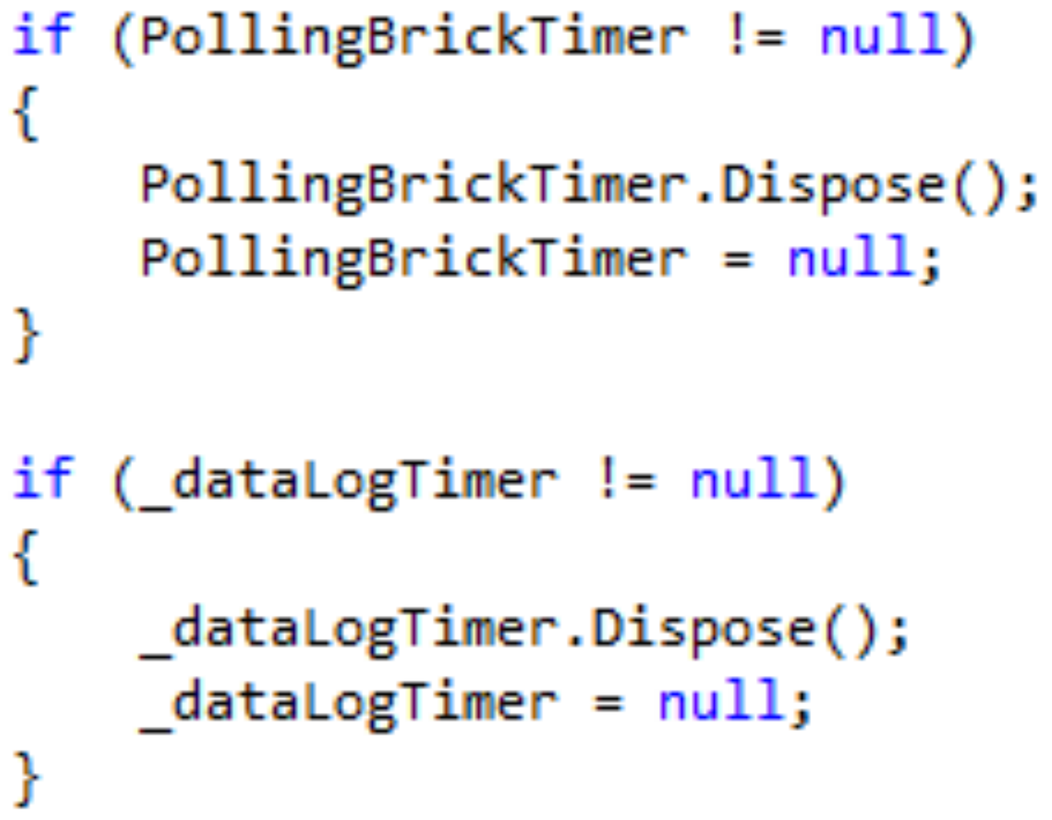
```
For i As Integer = 0 To CheckedListBoxDistroLocations.CheckedItems.Count - 1
    'Check if there is already a match... if not add

    If ReturnSQLStringBR("select clinicid from tbl_distroclinics where clinicid = '" & CheckedListBox
       "and distroid = " & DistroID & "") <> CheckedListBoxDistroLocations.CheckedItems(i).ToString
        If SQLNonQueryBactesRequest("Insert into tbl_distroclinics (Distroid, ClinicID) " & _
         "Values ( " & _
         "" & DistroID & "" & _
         ",'" & CheckedListBoxDistroLocations.CheckedItems(i).ToString & "')") = 0 Then
        End If
    End If 'if it's already there... ignore the entry
Next
'Item is not checked so delete it from the db.
If SQLNonQueryBactesRequest("delete tbl_distroclinics where " & _
 "Distroid = " & DistroID & " and " & _
 "ClinicID  in " & UnCheckedLocationsList & "") = 0 Then
End If


'Update Workstations
Dim UnCheckedWorkstationsList As String = "('"
For i As Integer = 0 To CheckedListBoxDistroWorkstations.Items.Count - 1
    If Not CheckedListBoxDistroWorkstations.GetItemChecked(i) Then
        UnCheckedWorkstationsList = UnCheckedWorkstationsList & CheckedListBoxDistroWorkstations.Item
    End If
Next
UnCheckedWorkstationsList = UnCheckedWorkstationsList & "')"

For i As Integer = 0 To CheckedListBoxDistroWorkstations.CheckedItems.Count - 1
    'Check if there is already a match... if not add

    If ReturnSQLStringBR("select workstationName from tbl_distroWorkstations where workstationName =
       "and distroid = " & DistroID & "") <> CheckedListBoxDistroWorkstations.CheckedItems(i).ToStri
        If SQLNonQueryBactesRequest("Insert into tbl_distroWorkstations (Distroid, workstationName)
         "Values ( " & _
         "" & DistroID & "" & _
         ",'" & CheckedListBoxDistroWorkstations.CheckedItems(i).ToString & "')") = 0 Then
        End If
    End If 'if it's already there... ignore the entry
Next
'Item is not checked so delete it from the db.
If SQLNonQueryBactesRequest("delete tbl_distroWorkstations where " & _
 "Distroid = " & DistroID & " and " & _
 "workstationName  in " & UnCheckedWorkstationsList & "") = 0 Then
End If
```

```
If SQLNonQueryBactesRequest("delete tbl_distroWorkstations where Distroid = " & DistroID & " ") = 0 Then
End If
If SQLNonQueryBactesRequest("delete tbl_DistroHeader where Distroid = " & DistroID & " ") = 0 Then
End If
If SQLNonQueryBactesRequest("delete tbl_DistroClinics where Distroid = " & DistroID & " ") = 0 Then
End If
```

```
    Try
        cnDistroGroup.Open()
        Dim sdrDistroGroup As SqlDataReader = cmDistroGroup.ExecuteReader
        With sdrDistroGroup
            If .HasRows Then
                While .Read()
                    ComboBoxDistroGroups.Items.Add(.GetString(0).ToUpper())
                    ComboBoxDistroGroupsMaster.Items.Add(.GetString(0).ToUpper())
                End While
            End If
        End With
    Catch ex As Exception
        Logger.Warning(ex)

    End Try
    cnDistroGroup.Close()

    PopulateDistroCheckedLists()

    LabelDistroGroupDescription.Text = "Description: "
End Sub
Private Sub PopulateDistroCheckedLists()
    'Populate Locations
    CheckedListBoxDistroLocations.Items.Clear()
    Dim cnLocations As New SqlConnection(SessionDBs.BIMReportConnectionString)
    Dim strsql As String = "Select locationName " & _
    "from tbl_Locations order by locationname "
    Dim cmLocations As New SqlCommand(strsql, cnLocations)

    Try
        cnLocations.Open()
        Dim sdrLocations As SqlDataReader = cmLocations.ExecuteReader
        With sdrLocations
            If .HasRows Then
                While .Read()
                    CheckedListBoxDistroLocations.Items.Add(.GetString(0).ToUpper(), False)
                End While
            End If
        End With
    Catch ex As Exception
        Logger.Warning(ex)

    End Try
```

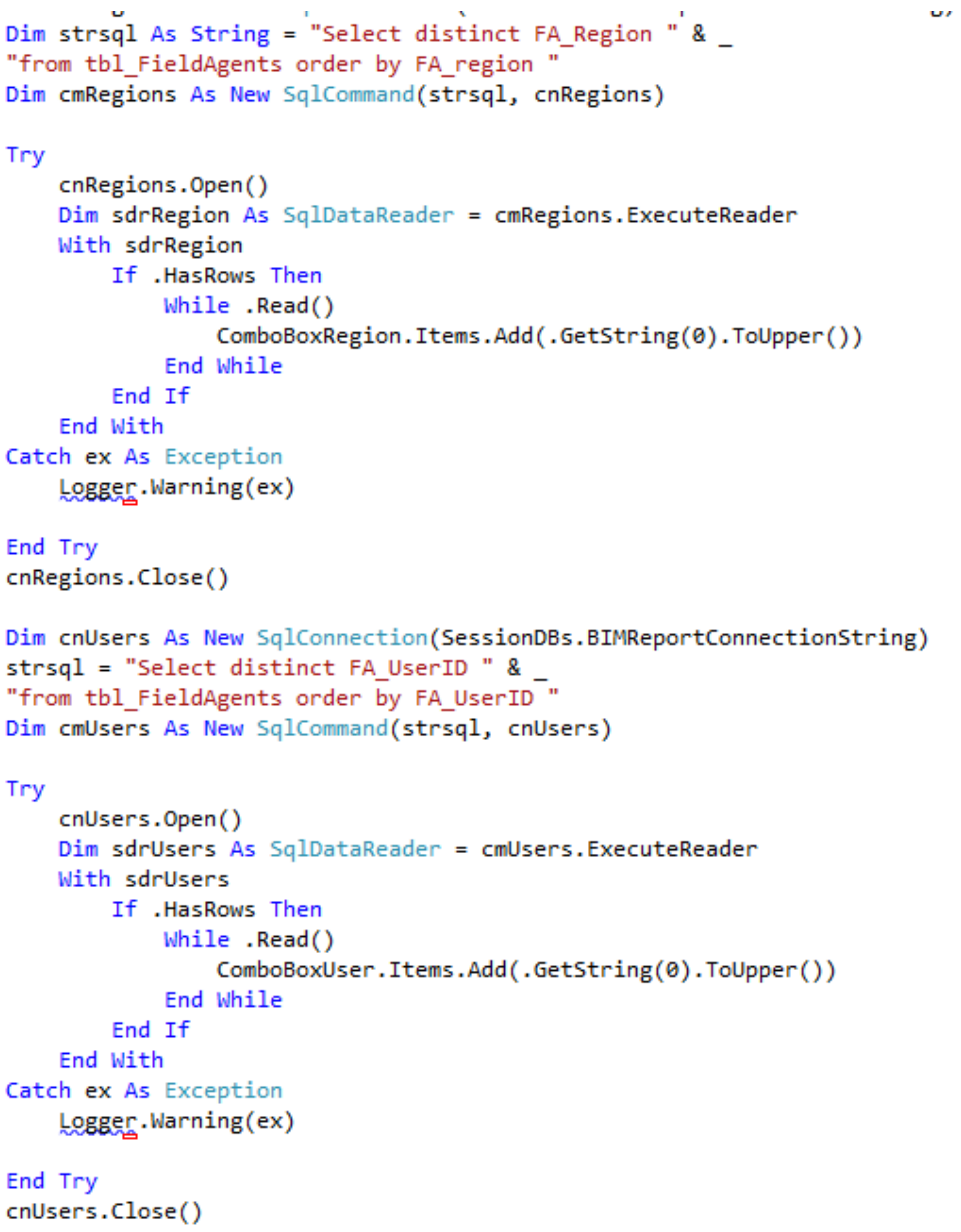
```
private void FinishStructure(CurrentGenHeap.Structure classicStructure, NextGenSourceModel.Structure nextGenStructure)
{
    // Confirm that the current and next gen structures have
    // the same number of diagrams.
    Log.Assert(classicStructure.Diagrams.Count() == nextGenStructure.NestedDiagrams.Count(), "Classic and NextGen versions of the same structure have di
    for (int i = 0; i < nextGenStructure.NestedDiagrams.Count(); ++i)
    {
        var classicDiagram = classicStructure.Diagrams[i];
        var nextGenDiagram = nextGenStructure.NestedDiagrams.ElementAt(i);
        // make this the new next gen diagram
        _virtualInstrument.PushDiagram(nextGenDiagram);
        // Visit the classic gen diagram.
        classicDiagram.AcceptVisitor(this);
        // Pop the next gen diagram that you pushed above.
        _virtualInstrument.PopDiagram();
    }
}

private void FinishFlatSequence(CurrentGenHeap.FlatSequence classicFlatSequence, NextGenVIModel.FlatSequence nextGenFlatSequence)
{
    // Confirm that the current and next gen structures have
    // the same number of diagrams.
    Log.Assert(classicFlatSequence.Sequences.Count() == nextGenFlatSequence.NestedDiagrams.Count(), "Classic and NextGen versions of the same structure
    for (int i = 0; i < nextGenFlatSequence.NestedDiagrams.Count(); ++i)
    {
        var classicDiagram = classicFlatSequence.Sequences.ElementAt(i);
        var nextGenDiagram = nextGenFlatSequence.NestedDiagrams.ElementAt(i);
        // make this the new next gen diagram
        _virtualInstrument.PushDiagram(nextGenDiagram);
        // Visit the classic gen diagram.
        classicDiagram.AcceptVisitor(this);
        // Pop the next gen diagram that you pushed above.
        _virtualInstrument.PopDiagram();
    }
}
```

```
private unsafe void DirectCommand_LSGetStatus(ref NXTObject nxtObject, PortId port, ref bool idleFlag, ref bool fatalErrorFlag, Int32* error)
{
    // josh.todo:  The requestResponseFlag is currently not used.
    byte[] inputBuffer = new byte[2];
    byte[] outputBuffer = new byte[3];

    inputBuffer[0] = (byte)NXTOpcode.LSGetStatus;
    inputBuffer[1] = (byte)port;

    fixed (byte* inputBufferPtr = &inputBuffer[0])
    {
        fixed (byte* outputBufferPtr = &outputBuffer[0])
        {
            nFANTOM100_iNXT_sendDirectCommand(nxtObject.NXTHandle, 1, inputBufferPtr, (UInt32)inputBuffer.Length, outputBufferPtr, (UInt32)outputBuffer.Length,
        }
    }

    sbyte status = (sbyte)outputBuffer[1];
    idleFlag = status == 0 ? true : false;
    fatalErrorFlag = status < 0 ? true : false;
}

private unsafe byte[] DirectCommand_LSRead(ref NXTObject nxtObject, PortId port, ref int bytesRead, ref bool lsErrorFlag, Int32* status)
{
    // josh.todo:  The requestResponseFlag is currently not used.
    byte[] inputBuffer = new byte[2];
    byte[] outputBuffer = new byte[19];

    inputBuffer[0] = (byte)NXTOpcode.LSRead;
    inputBuffer[1] = (byte)port;

    fixed (byte* inputBufferPtr = &inputBuffer[0])
    {
        fixed (byte* outputBufferPtr = &outputBuffer[0])
        {
            nFANTOM100_iNXT_sendDirectCommand(nxtObject.NXTHandle, 1, inputBufferPtr, (UInt32)inputBuffer.Length, outputBufferPtr, (UInt32)outputBuffer.Length,
        }
    }
```



```
if (PollingBrickTimer != null)
{
    PollingBrickTimer.Dispose();
    PollingBrickTimer = null;
}

if (_dataLogTimer != null)
{
    _dataLogTimer.Dispose();
    _dataLogTimer = null;
}
```

```
if (listType == LineListType.SourceHorizontal || listType == LineListType.DestinationHorizontal)
{
    for (int i = 0; i < lineSegments.Count; ++i)
    {
        var lineSegment = lineSegments[i];
        if (lineSegment.GenerationPoint.X == point.X && point.Y >= lineSegment.StartPoint.Y && point.Y <= lineSegment.EndPoint.Y)
        {
            segmentIndex = i;
            break;
        }
    }
}
else
{
    for (int i = 0; i < lineSegments.Count; ++i)
    {
        var lineSegment = lineSegments[i];
        if (lineSegment.GenerationPoint.Y == point.Y && point.X >= lineSegment.StartPoint.X && point.X <= lineSegment.EndPoint.X)
        {
            segmentIndex = i;
            break;
        }
    }
}
```



```
Dim strsql As String = "Select distinct FA_Region " & _
"from tbl_FieldAgents order by FA_region "
Dim cmRegions As New SqlCommand(strsql, cnRegions)

Try
    cnRegions.Open()
    Dim sdrRegion As SqlDataReader = cmRegions.ExecuteReader
    With sdrRegion
        If .HasRows Then
            While .Read()
                ComboBoxRegion.Items.Add(.GetString(0).ToUpper())
            End While
        End If
    End With
Catch ex As Exception
    Logger.Warning(ex)

End Try
cnRegions.Close()

Dim cnUsers As New SqlConnection(SessionDBs.BIMReportConnectionString)
strsql = "Select distinct FA_UserID " & _
"from tbl_FieldAgents order by FA_UserID "
Dim cmUsers As New SqlCommand(strsql, cnUsers)

Try
    cnUsers.Open()
    Dim sdrUsers As SqlDataReader = cmUsers.ExecuteReader
    With sdrUsers
        If .HasRows Then
            While .Read()
                ComboBoxUser.Items.Add(.GetString(0).ToUpper())
            End While
        End If
    End With
Catch ex As Exception
    Logger.Warning(ex)

End Try
cnUsers.Close()
```