```
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;
        }
    }
}
```

#2

```
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();
    }
}
```

#3

```
    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
```

#4

```
public void singCatSong()
{
  sing("We gave the cat to a little kid");
  sing("But the cat came back");
  sing("The very next day");
  sing("Oh the cat came back");
  sing("We thought he was a goner");
  sing("But the cat came back, he just wouldn't go away");
  sing("We sent the cat out on a boat");
  sing("But the cat came back");
  sing("The very next day");
  sing("Oh the cat came back");
  sing("We thought he was a goner");
  sing("But the cat came back, he just wouldn't go away");
}
```

```
public void singBottlesOfBeer()
{
  sing("100 bottles of beer on the wall");
  sing("100 bottles of beer");
  sing("Take one down, pass it around");
  sing("99 bottles of beer on the wall");
  sing("99 bottles of beer on the wall");
  sing("99 bottles of beer");
  sing("Take one down, pass it around");
  sing("98 bottles of beer on the wall");
}
```

```
public void singCheers()
{
  int number = 2;
  sing(number + "! ");
  number = number + 2;
  sing(number + "! ");
  number = number + 2;
  sing(number + "! ");
  number = number + 2;
  sing(number + "! ");
  sing("Who do we appreciate?");
  number = 17;
  sing(number + "! ");
  number = getNextPrime(number);
  sing(number + "! ");
  number = getNextPrime(number);
  sing(number + "! ");
  number = getNextPrime(number);
  sing(number + "! ");
  sing("These are the primes, that we find fine!");
}
```

#7

```java
public void singSong(int style, String... names)
{
  switch (style)
  {
    case 1 :
      for (String name : names)
      {
        if (name.startsWith("L", 0))
        {
          sing("Hip Hip Horray! For " + name);
        }
        else
        {
          sing("Hello " + name + ", it's nice to meet you.");
        }
      }
      break;
    case 2 :
      for (String name : names)
      {
        if (name.startsWith("am", 1))
        {
          sing("Say yeah! Say yo! Say " + name);
        }
        else
        {
          sing("Hello " + name + ", it's nice to meet you.");
        }
      }
      break;
    case 3 :
      for (String name : names)
      {
        sing("Hello " + name + ", it's nice to meet you.");
      }
      break;
  }
```

#8 – Extra Credit

```java
public void singSong(int style, String... names)
{
  switch (style)
  {
    case 1 :
      for (String name : names)
      {
        if (name.startsWith("L"))
        {
          sing("Hip Hip Horray! For " + name);
        }
        else
        {
          sing("Hello " + name + ", it's nice to meet you.");
        }
      }
      break;
    case 2 :
      for (String name : names)
      {
        if (name.contains("a"))
        {
          sing(name.toUpperCase() + "! Yay " + name + "!");
        }
        else
        {
          sing("Hello " + name + ", it's nice to meet you.");
        }
      }
      break;
    case 3 :
      for (String name : names)
      {
        sing("Hello " + name + ", it's nice to meet you.");
      }
      break;
  }
}
```

#9 – Extra Credit