

# Steps to be followed while Coding RDLC Report in C# with Pie Chart and Combobox

**Move the connection string this function “public partial class MasterForm : Form”**

```
// you modify to your requirements – these are dummy for video purpose
string connectionString = @"Server=MYLAPTOP\SQLEXPRESS;Database=PerformanceDB;User
Id=hi;Password=admin;";
```

---

**Add this code next    InitializeComponent();**  
PopulateStateComboBox();

---

**Here is the PopulateStateComboBox function:**

```
private void PopulateStateComboBox()
{
    // Clear existing items
    StateNameCbx.Items.Clear();

    // Add "ALL STATES" option
    StateNameCbx.Items.Add("ALL STATES");

    try
    {
        // Connect to the database
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            // Open the connection
            connection.Open();

            string query = "SELECT DISTINCT(State_Name) FROM [PerformanceDB].[dbo].[MonthlyResultsTbl]
ORDER BY State_Name";

            // Create a command object
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                // Execute the command and read data
                using (SqlDataReader reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        // Add State_Name to the ComboBox
                        StateNameCbx.Items.Add(reader["State_Name"].ToString());
                    }
                }
            }
        }
    }
}
```

```

catch (Exception ex)
{
    MessageBox.Show("Error: " + ex.Message);
}

// Set "ALL STATE" as the default selected item
StateNameCbx.SelectedItem = "ALL STATES";
}

```

---

**When you double click on the form you will land in MasterForm\_Load, enter this ReportProcessing(); within that function**

ReportProcessing();

---

**Here is the function for ReportProcessing:**

```

private void ReportProcessing()
{
    string mstatename = StateNameCbx.Text;

    reportViewer1.ProcessingMode = ProcessingMode.Local;
    reportViewer1.LocalReport.ReportPath =
@"C:\Users\Ganapathy\source\repos\WindowsReportForm1\WindowsReportForm1\PerformanceReport.rdlc"
;

    string query = "SELECT " +
        " Fyear, " +
        " FyearName, " +
        " Month_Year, " +
        " SUM(Revenue_Amt) AS Revenue_Amt, " +
        " SUM(Net_Result) AS Net_Result " +
        " FROM PerformanceDB.dbo.fn_MonthlyPerformanceData() ";

    if (mstatename != "ALL STATES")
    {
        query += " WHERE State_Name = @StateName ";
    }

    query += " GROUP BY Fyear, FyearName, Month_Year";

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            SqlCommand command = new SqlCommand(query, connection);
            if (mstatename != "ALL STATES")
            {
                command.Parameters.AddWithValue("@StateName", mstatename);
            }
        }
    }
}

```

```

    }

    SqlDataAdapter dataAdapter = new SqlDataAdapter(command);
    DataSet dataSet = new DataSet();
    dataAdapter.Fill(dataSet, "MonthlyResultsTbl");

    DataTable dataTable = dataSet.Tables["MonthlyResultsTbl"];

    ReportDataSource reportDataSource = new ReportDataSource("ds_FinancialReport", dataTable);
    reportViewer1.LocalReport.DataSources.Clear(); // Clear any existing data sources
    reportViewer1.LocalReport.DataSources.Add(reportDataSource);

    // Add parameters to the report
    ReportParameter[] reportParameters = new ReportParameter[]
    {
        new ReportParameter("pm_statename", mstatename)
    };
    reportViewer1.LocalReport.SetParameters(reportParameters);

    }
}
catch (InvalidCastException ex)
{
    MessageBox.Show($"Error in data conversion: {ex.Message}");
}
catch (Exception ex)
{
    MessageBox.Show($"Error loading report: {ex.Message}");
}

try
{
    reportViewer1.RefreshReport();
}
catch (Exception ex)
{
    MessageBox.Show("Error loading report: " + ex.Message);
}
}

```

---

**When you add a trigger called `SelectedIndexChanged` to Combobox, you will land here, you need to just add `ReportProcessing();` as shown below.**

```

private void StateNameCbx_SelectedIndexChanged(object sender, EventArgs e)
{
    ReportProcessing();
}

```