Из класса Chart\_Acts:

//Формат арены и курсора

//Chart1.ChartAreas[meinAreaName].CursorX.LineColor = Color.DarkGreen;

//Chart1.ChartAreas[Num\_area].CursorX.LineDashStyle = ChartDashStyle.DashDot;

//Chart1.ChartAreas[Num\_area].CursorX.LineWidth = 2;

//Chart1.ChartAreas[Num\_area].CursorY.LineColor = Color.DarkGreen;

//Chart1.ChartAreas[Num\_area].CursorY.LineDashStyle = ChartDashStyle.DashDot;

//Chart1.ChartAreas[Num\_area].CursorY.LineWidth = 2;

//Расположение полосы прокрутки

Chart1.ChartAreas[Num\_area].AxisX.ScrollBar.IsPositionedInside = true;

Chart1.ChartAreas[Num\_area].AxisY.ScrollBar.IsPositionedInside = true;

//Показывать конечное значение на оси

//Chart1.ChartAreas[Num\_area].AxisX.LabelStyle.IsStaggered=true;

//Chart1.ChartAreas[Num\_area].AxisX.LabelStyle.IntervalType = DateTimeIntervalType.Auto;

//Chart1.ChartAreas[Num\_area].AxisX.IntervalType = DateTimeIntervalType.Minutes;

My\_Legend.DockedToChartArea = Num\_area;

Chart1.Legends.Add(My\_Legend);

////Если серия уже была, не уникальна, имя серии для нее назовем с добавлением номера, сколько раз уже строили дополнительные линии аналогичные, в конце.

//РЕШИл, ЧТО ЭТО ЛИШНЕЕ

//if (!Chart1.Series.IsUniqueName(SerName))

//{

// I++;

// SerName+=I;

//}

//Добавили серию в список графика

//Определяем, строились ли дополнительные оси.

bool noAxis=true;

foreach (var item1 in Chart1.ChartAreas)

{

if (item1.Name.Contains("1b") || item1.Name.Contains("2b"))

{

noAxis=false;

}

}

//Если нет дополнительной оси. Легенду привязываем к указанной арене.

if (noAxis)

{

Chart1.Series[SerName].Legend = chartareaName;

}

//Если есть, то и тип графика с 4 осями, на основную арену легенду наносим.

else

Chart1.Series[SerName].Legend = "Area# 1";

//ЕСЛИ на допоплнительных осях строим в первый раз.

bool IsarenaReadyLeft = false;

foreach (var item1 in myChart.ChartAreas)

{

if (item1.Name.Contains("1b"))

{

IsarenaReadyLeft = true;

}

}

if (ISsecond == "Yes" && !IsarenaReadyLeft && lrAxis=="left")

{

//Теперь переносим ее.

this.CreateSecondYAxis("L", Chart1, Chart1.ChartAreas[chartareaName], Chart1.Series[SerName], 4, 0);

}

if (ISsecond == "Yes" && IsarenaReadyLeft)

{

//Теперь переносим ее.

// Добавим линию, и назначим ее в созданную область.

Chart1.Series[SerName].ChartArea = Chart1.ChartAreas[chartareaName].Name + "1a";

}

bool IsarenaReadyRight = false;

foreach (var item2 in Chart1.ChartAreas)

{

if (item2.Name.Contains("2b"))

{

IsarenaReadyRight = true;

}

}

if (ISsecond == "Yes" && !IsarenaReadyRight && lrAxis == "right")

{

//Теперь переносим ее.

this.CreateSecondYAxis("R", Chart1, Chart1.ChartAreas[chartareaName], Chart1.Series[item.KKS], 4, 0);

}

if (ISsecond == "Yes" && IsarenaReadyRight)

{

//Теперь переносим ее.

// Добавим линию, и назначим ее в созданную область.

Chart1.Series[SerName].ChartArea = Chart1.ChartAreas[chartareaName].Name + "2a";

}

//ПРи удалении единственной линии на оси дополнительной, мы ее крыли. Откроем

if (ISsecond.Contains("Yes"))

{

if (lrAxis == "left")

{

Chart1.ChartAreas["Area# 11b"].AxisY.Enabled = AxisEnabled.True;

}

//ПРавая ось

else

{

Chart1.ChartAreas["Area# 12b"].AxisY2.Enabled = AxisEnabled.True;

}

}

///// <summary>

///// Выровнять графики

///// </summary>

//public void SetAlignment(Chart Chart1)

//{

// for (int i = 1; i < Chart1.ChartAreas.Count; i++)

// {

// Chart1.ChartAreas[i].AlignWithChartArea = "Area" + "# " + 1;//1 площадка рисования

// Chart1.ChartAreas[i].AlignmentOrientation = AreaAlignmentOrientations.Vertical;

// //ChartFromForm.ChartAreas[i].AlignmentStyle = AreaAlignmentStyles.PlotPosition;

// }

//}

/// <summary>

/// Создать ось СЛЕВА. (Создавая 2 доп. арены, 1 линию доп. невидимую, 1 перенося)

/// </summary>

/// <param name="chart">График</param>

/// <param name="area">Исходная арена (для определения позиции)</param>

/// <param name="series">Серия данных передаваема</param>

/// <param name="axisOffset">Сдвиг оси</param>

// Set Chart Area position

chart1.ChartAreas["Default"].Position.Auto = false;

chart1.ChartAreas["Default"].Position.X = 10;

chart1.ChartAreas["Default"].Position.Y = 10;

chart1.ChartAreas["Default"].Position.Width = 80;

chart1.ChartAreas["Default"].Position.Height= 80;

// Set the plotting area position. Coordinates of a plotting

// area are relative to a chart area position.

chart1.ChartAreas["Default"].InnerPlotPosition.Auto = false;

chart1.ChartAreas["Default"].InnerPlotPosition.X = 10;

chart1.ChartAreas["Default"].InnerPlotPosition.Y = 10;

chart1.ChartAreas["Default"].InnerPlotPosition.Width = 80;

chart1.ChartAreas["Default"].InnerPlotPosition.Height= 80;

/// <summary>

/// Создать дополнительную ось.

/// </summary>

/// <param name="chart">График</param>

/// <param name="area">Исходная арена (для определения позиции)</param>

/// <param name="series">Серия данных передаваема</param>

/// <param name="axisOffset">Сдвиг оси</param>

public void CreateSecondYAxis(string LeftRight, Chart chart, ChartArea area, Series series, float axisOffset, float labelsSize)

{

// СОздаем площадку для графика. Здесь русем график.

ChartArea areaForSeria = new ChartArea();

if (LeftRight=="L")

{

areaForSeria = chart.ChartAreas.Add(area.Name + "1a");

}

else

{

areaForSeria = chart.ChartAreas.Add(area.Name + "2a");

}

areaForSeria.BackColor = Color.Transparent;

areaForSeria.BorderColor = Color.Transparent;

//Выровнять. Площадку, где рисуем и начальную.

areaForSeria.Position.FromRectangleF(area.Position.ToRectangleF());

areaForSeria.InnerPlotPosition.FromRectangleF(area.InnerPlotPosition.ToRectangleF());

//Осей нет.

areaForSeria.AxisX.Enabled = AxisEnabled.False;

areaForSeria.AxisY.Enabled = AxisEnabled.False;

areaForSeria.AxisY2.Enabled = AxisEnabled.False;

//Разрешение пользоваться курсором

areaForSeria.CursorX.IsUserEnabled = true;

areaForSeria.CursorX.IsUserSelectionEnabled = true;

areaForSeria.CursorY.IsUserEnabled = true;

areaForSeria.CursorY.IsUserSelectionEnabled = true;

//Выбор интервала для курсора. Ноль даст возможность выделять прямоугольные области.

areaForSeria.CursorX.Interval = 0;

areaForSeria.CursorY.Interval = 0;

// Добавим линию, и назначим ее в созданную область.

series.ChartArea = areaForSeria.Name;

//Легенда линии привязывается к первой арене

series.Legend = area.Name;

// Создаем арену для размещения в ней оси.

// СОздаем площадку для графика. Здесь русем график.

ChartArea areaForAxis = new ChartArea();

if (LeftRight == "L")

{

areaForAxis = chart.ChartAreas.Add(area.Name + "1b");

areaForAxis.AxisY.Name = "Y1" + "b";

}

else

{

areaForAxis = chart.ChartAreas.Add(area.Name + "2b");

areaForAxis.AxisY2.Name = "Y2" + "b";

}

areaForAxis.BackColor = Color.Transparent;

areaForAxis.BorderColor = Color.Transparent;

areaForAxis.Position.FromRectangleF(chart.ChartAreas[series.ChartArea].Position.ToRectangleF());

areaForAxis.InnerPlotPosition.FromRectangleF(chart.ChartAreas[series.ChartArea].Position.ToRectangleF());

// Create a copy of specified series. Копия невидимая, так как надо ось настроить видимую.

Series seriesCopy = chart.Series.Add(series.Name + "\_Copy");

seriesCopy.ChartType = series.ChartType;

foreach (DataPoint point in series.Points)

{

seriesCopy.Points.AddXY(point.XValue, point.YValues[0]);

}

// Hide copied series

seriesCopy.IsVisibleInLegend = false;

seriesCopy.Color = Color.Transparent;

seriesCopy.BorderColor = Color.Transparent;

seriesCopy.ChartArea = areaForAxis.Name;

// Настроим оси

areaForAxis.AxisX.Enabled = AxisEnabled.False;

if (LeftRight == "L")

{

areaForAxis.AxisY.Enabled = AxisEnabled.True;

areaForAxis.AxisY2.Enabled = AxisEnabled.False;

areaForAxis.AxisY.MajorGrid.Enabled = false;

}

else

{

areaForAxis.AxisY.Enabled = AxisEnabled.False;

areaForAxis.AxisY2.Enabled = AxisEnabled.True;

areaForAxis.AxisY2.MajorGrid.Enabled = false;

}

//Разрешение пользоваться курсором

areaForAxis.CursorX.IsUserEnabled = true;

areaForAxis.CursorX.IsUserSelectionEnabled = true;

areaForAxis.CursorY.IsUserEnabled = true;

areaForAxis.CursorY.IsUserSelectionEnabled = true;

//Выбор интервала для курсора. Ноль даст возможность выделять прямоугольные области.

areaForAxis.CursorX.Interval = 0;

areaForAxis.CursorY.Interval = 0;

// Adjust area position

if (LeftRight == "L")

{

areaForAxis.Position.X -= axisOffset;

areaForAxis.InnerPlotPosition.X -= labelsSize;

}

else

{

areaForAxis.Position.X += axisOffset;

areaForAxis.InnerPlotPosition.X += labelsSize;

}

}