

C# グラフ

[C#][Visual Studio]

- [Microsoft Chart Controls for Microsoft .NET Framework 3.5](#)
- [Microsoft Chart Controls Add-on for Microsoft Visual Studio 2008](#)
- [Microsoft Chart Controls for Microsoft .NET Framework 3.5 Language Pack](#)
- [Microsoft Chart Controls for .NET Framework Documentation](#)
- [Samples Environment for Microsoft Chart Controls](#)

Microsoft Chart Controls

- <http://code.msdn.microsoft.com/mschart>

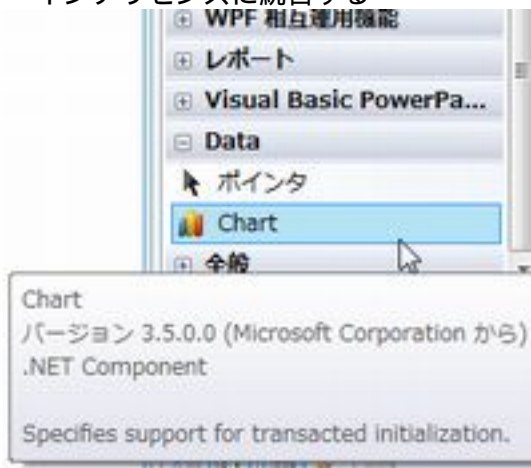
インストール

チャートコントロールのアセンブリ

- [Microsoft Chart Controls for Microsoft .NET Framework 3.5](#)
- ASP.NET および [Windows](#) Forms チャートコントロール を含む新しいアセンブリをインストールします

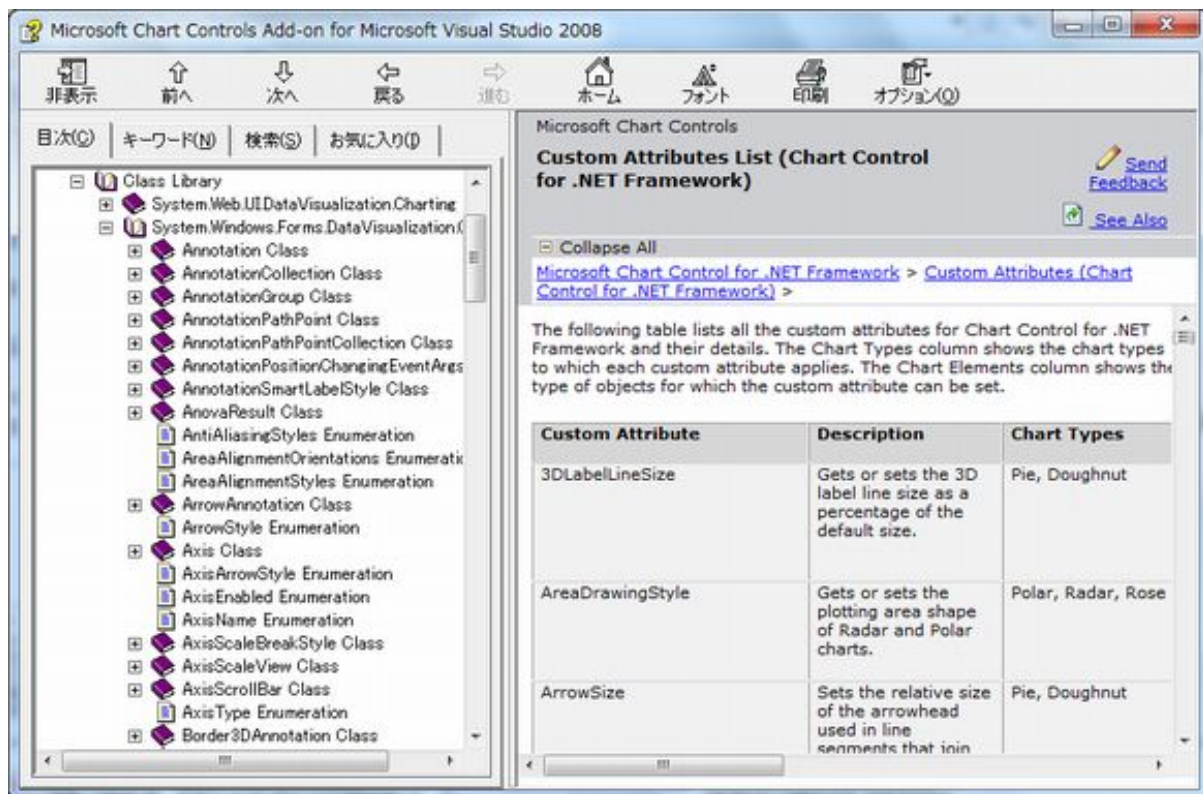
Visual Studio アドオン

- [Microsoft Chart Controls Add-on for Microsoft Visual Studio 2008](#)
- Microsoft Chart Controls for .NET Framework 3.5 が事前に導入されていること
- [Visual Studio](#) の ASP.NET、[Windows](#) Forms チャートコントロールをツールボックスおよびインテリセンスに統合する



ドキュメント

- [Microsoft Chart Controls for .NET Framework Documentation](#)



サンプル

- ・ [MSDN Code Gallery](#)
- ・ [ここから](#)、サンプルソリューションをダウンロードし、ビルドすると、WinFormsChartSamples.exe ができる。

Samples Environment for Microsoft Chart Controls

Home Downloads Discussions People License

0 Votes in Release [\(breakdown\)](#)

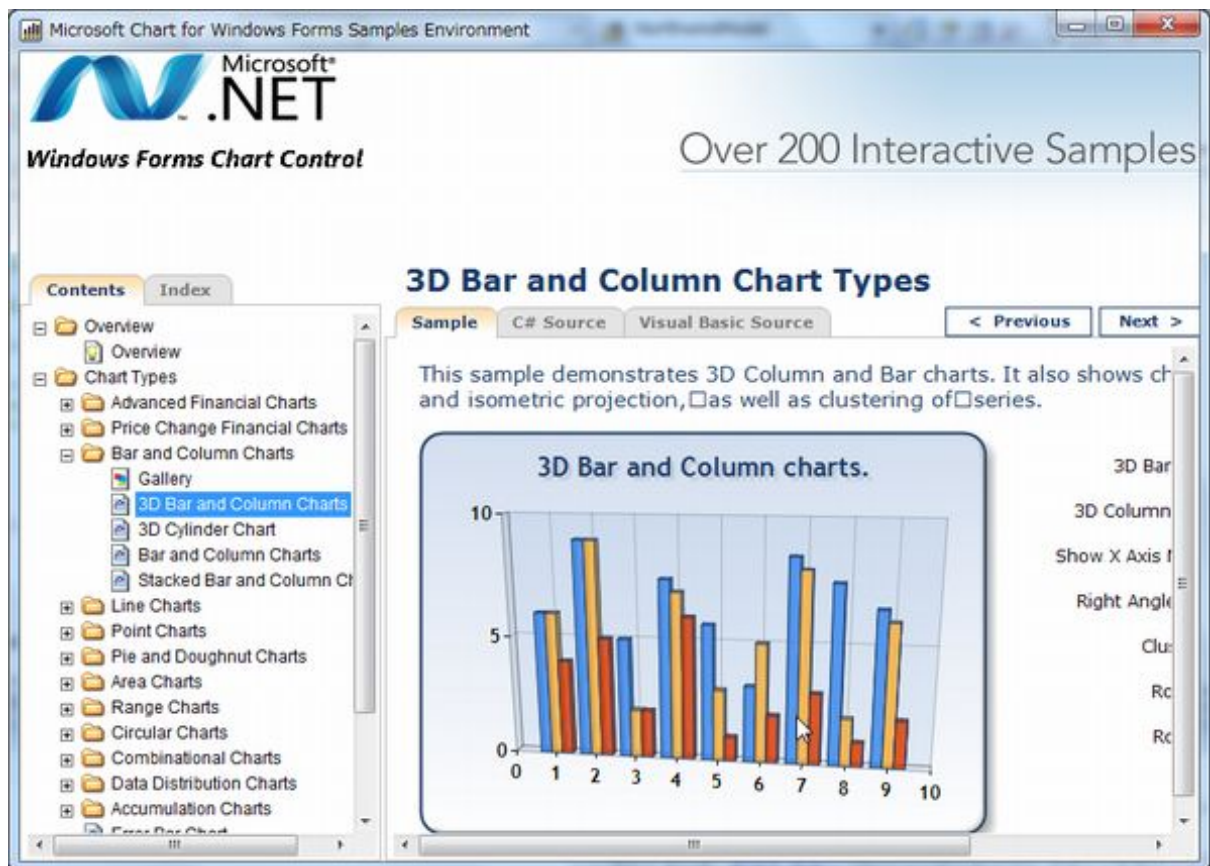
Samples Environment for Chart Controls

Files

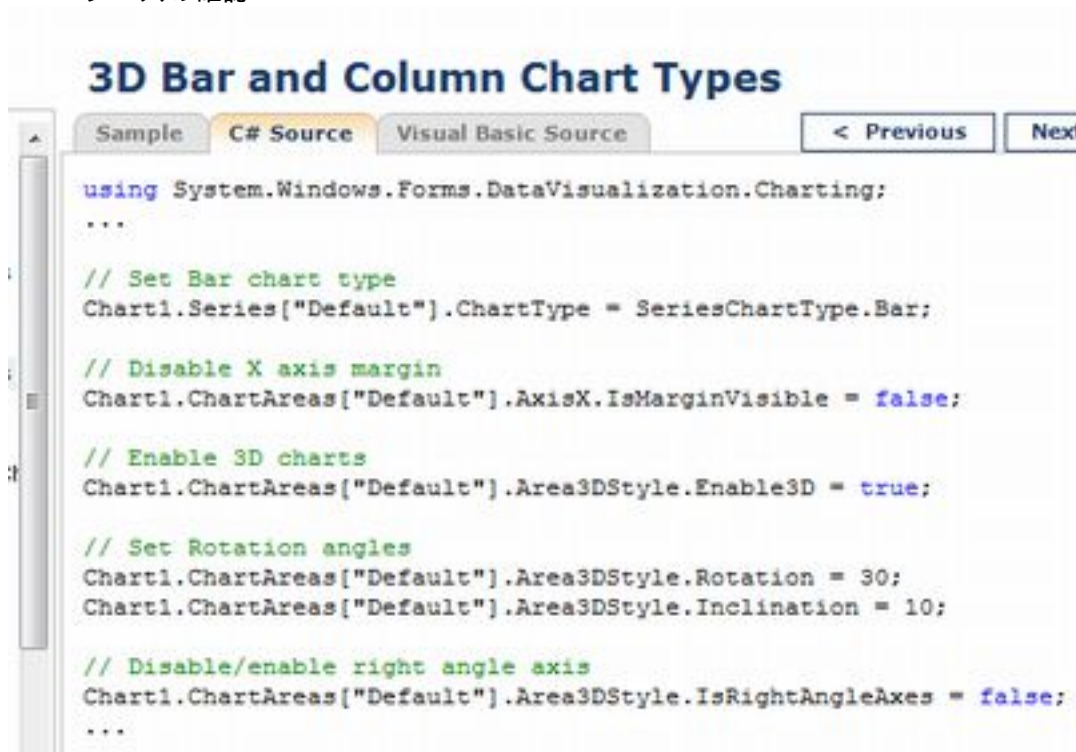
- ASP.NET Samples**
source code, 4125K, uploaded Aug 6 - 120214 downloads
- Windows Forms Samples**
source code, 5381K, uploaded Aug 6 - 83214 downloads

WinFormsChartSamples.exe

- ・ サンプルの例およびソースコードが確認できるプログラム

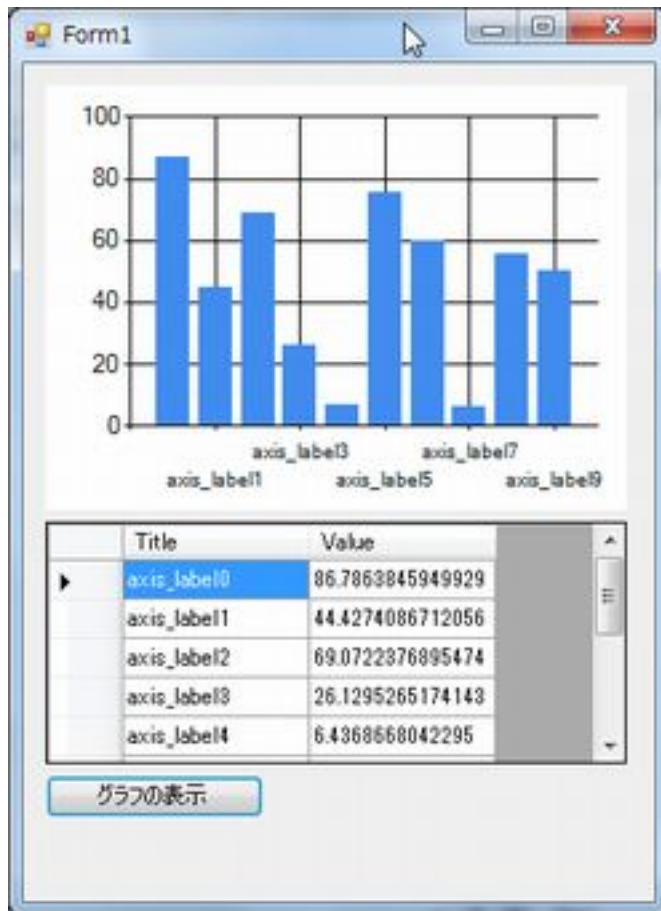


・ソースの確認



グラフの作成

単純なグラフ



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace MsChartCtrlTest
{
    public partial class Form1 : Form
    {
        private Chart chart1 = null;
        public Form1()
        {
            InitializeComponent();

            // 初期表示ダミーデータをデータグリッドに設定
            Random rnd = new Random();
            for (int i = 0; i < 10; i++)
            {
                string[] row = new string[] {
                    "axis_label" + i, (rnd.NextDouble() * 100).ToString()
                };
                dataGridView1.Rows.Add(row);
            }

            // グラフコントロールを動的に配置
            chart1 = new Chart();

            ChartArea chartArea = new ChartArea();
            this.chart1.ChartAreas.Add(chartArea);
            this.chart1.Location = new System.Drawing.Point(12, 12);
            this.chart1.Size = new System.Drawing.Size(312, 228);
            this.Controls.AddRange(new System.Windows.Forms.Control[] { this.chart1 });
        }
    }
}

```

```

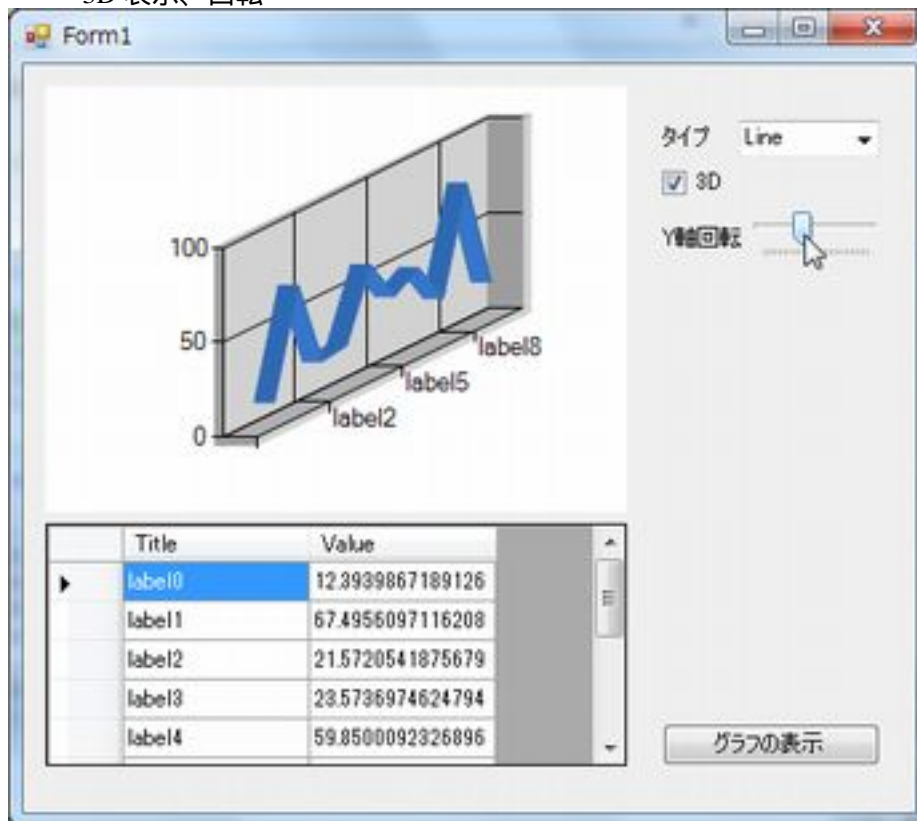
private void button1_Click(object sender, EventArgs e)
{
    // グラフの表示
    this.chart1.Series.Clear();

    Series series = new Series();
    for (int i = 0; i < dataGridView1.Rows.Count-1; i++)
    {
        DataPoint point = series.Points.Add(Convert.ToDouble(dataGridView1.Rows[i].Cells[1
].Value));
        point.AxisLabel = dataGridView1.Rows[i].Cells[0].Value.ToString();
    }
    this.chart1.Series.Add(series);
    this.chart1.ResetAutoValues();
}
}
}

```

機能追加してみる

- ・グラフタイプ変更（棒、折れ線、円）
- ・3D 表示、回転



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace MsChartCtrlTest
{
    public partial class Form1 : Form
    {
        private Chart chart1 = null;
        public Form1()
        {
            InitializeComponent();

```



```

// 初期表示ダミーデータをデータグリッドに設定
Random rnd = new Random();
for (int i = 0; i < 10; i++)
{
    string[] row = new string[] {
        "label" + i, (rnd.NextDouble() * 100).ToString()
    };
    dataGridView1.Rows.Add(row);
}

// グラフのタイプを選択
comboBox1.Items.Add(SeriesChartType.Bar);
comboBox1.Items.Add(SeriesChartType.Line);
comboBox1.Items.Add(SeriesChartType.Pie);
comboBox1.SelectedItem = SeriesChartType.Bar;

// 3D の場合の Y 軸回転
trackBar1.Maximum = 180;
trackBar1.Minimum = -180;
trackBar1.TickFrequency = 10;

// グラフコントロールを動的に配置
chart1 = new Chart();

ChartArea chartArea = new ChartArea();
this.chart1.ChartAreas.Add(chartArea);
this.chart1.Location = new System.Drawing.Point(12, 12);
this.chart1.Size = new System.Drawing.Size(312, 228);
this.Controls.AddRange(new System.Windows.Forms.Control[] { this.chart1 });
}

private void button1_Click(object sender, EventArgs e)
{
    this.chart1.Series.Clear();

    // 3D スタイルの有効無効
    this.chart1.ChartAreas[0].Area3DStyle.Enable3D = checkBox1.Checked;
    Series series = new Series();

    // グラフのタイプを設定
    series.ChartType = (SeriesChartType)comboBox1.SelectedItem;
    for (int i = 0; i < dataGridView1.Rows.Count-1; i++)
    {
        DataPoint point = series.Points.Add(Convert.ToDouble(dataGridView1.Rows[i].Cells[1
].Value));
        point.AxisLabel = dataGridView1.Rows[i].Cells[0].Value.ToString();
    }
    this.chart1.Series.Add(series);
    this.chart1.ResetAutoValues();
}

private void trackBar1_Scroll(object sender, EventArgs e)
{
    if (checkBox1.Checked)
    {
        this.chart1.ChartAreas[0].Area3DStyle.Rotation = trackBar1.Value;
    }
}

private void Form1_Load(object sender, EventArgs e)
{
}
}
}

```