Essential Studio 2013 Volume 4 - v.11.4.0.26

Essential Gauge for Windows Forms

Contents

1 Overview 3

1.1 Introduction to Gauge for Windows Forms 3

1.2 Prerequisites and Compatibility 3

1.2.1 Prerequisites 3

1.2.2 Compatibility 3

2 Installation and Deployment 5

2.1 Installation 5

2.2 Samples and Location 5

2.2.1 Samples Installation Location 5

2.2.2 Viewing Samples 5

2.2.3 Source Code Location 8

2.3 Deployment Requirements 9

2.3.1 DLL List 9

3 Radial Gauge 11

3.1 Real World Scenarios 11

3.2 Getting Started 11

3.2.1.1 Creating a Radial Gauge 12

3.2.1.2 Elaborate Structure of the Control 14

3.2.2 Concepts and Features 14

3.2.2.1 Radial Gauge Frame 14

3.2.2.2 Scales 17

3.2.2.3 Ticks 19

3.2.2.4 Needles 22

3.2.2.5 Ranges 23

3.2.2.6 Scaling Divisions 25

3.3 Visual Styles for all the Gauges 26

3.4 Data Binding 29

1 Overview

This section covers information on the Gauge control for Windows Forms, its key features, prerequisites to use the control, compatibility with various operating systems, and other documentation details for the product. It comprises the following topics:

- Introduction
- Installation and Deployment
- Getting Started
- Gauge for Windows Forms controls

1.1 Introduction to Gauge for Windows Forms

The Gauge control for Windows Forms is a data visualization tool that can be used to display several data points or data ranges in a precise and compact area. The data displayed in the control can be quickly understood by the user. Syncfusion's Windows Forms library enables users to configure the Gauge control through C# codes. The Gauge control comes with sophisticated customization support.

The Gauge control is intended for developers who want to use gauges in their Windows Forms applications. It is a very useful control to indicate the current value in a range of values.

1.2 Prerequisites and Compatibility

This section covers the requirements for using Gauge control. It also lists operating systems and browsers compatible with the product.

1.2.1 Prerequisites

The prerequisites details are listed below:

Table 1: Prerequisites Table

Development Environments	• Visual Studio 2012 (Ultimate, Premium, Professional and Express) • Visual Studio 2010 (Ultimate, Premium, Professional and Express) • Visual Studio 2008 with SP1 (Team, Professional, Standard and Express)
.NET Framework versions	• .NET 4.5 • .NET 4.0 • .NET 3.5 SP1

1.2.2 Compatibility

The compatibility details are as follows:

Table 2: Compatibility

Operating Systems	• Windows Server 2008 (32 bit and 64 bit) • Windows 7 (32 bit and 64 bit) • Windows Vista (32 bit and 64 bit) • Windows XP • Windows 2003

2 Installation and Deployment

This section covers information on installation, the process of viewing samples through the sample browser, and the locations of samples and source code.

2.1 Installation

For step-by-step installation procedure for the installation of Essential Studio, refer to the **Installation** topic under **Installation and Deployment** in the **Common UG**.

See Also

For licensing, patches, and information on adding or removing selective components, refer to the following topics in **Common UG** under **Installation and Deployment**:

- Licensing
- Patches
- Add/Remove Components

2.2 Samples and Location

This section covers the location of the installed samples and describes the procedure to run the samples through the Sample Browser and online. It also provides the location of the source code.

2.2.1 Samples Installation Location

The Gauge samples are installed in the following location locally on the disk:

Windows XP:
C:\Syncfusion\Essential Studio<version number>\Windows\Gauge.Windows\Samples

Windows 7/Vista:

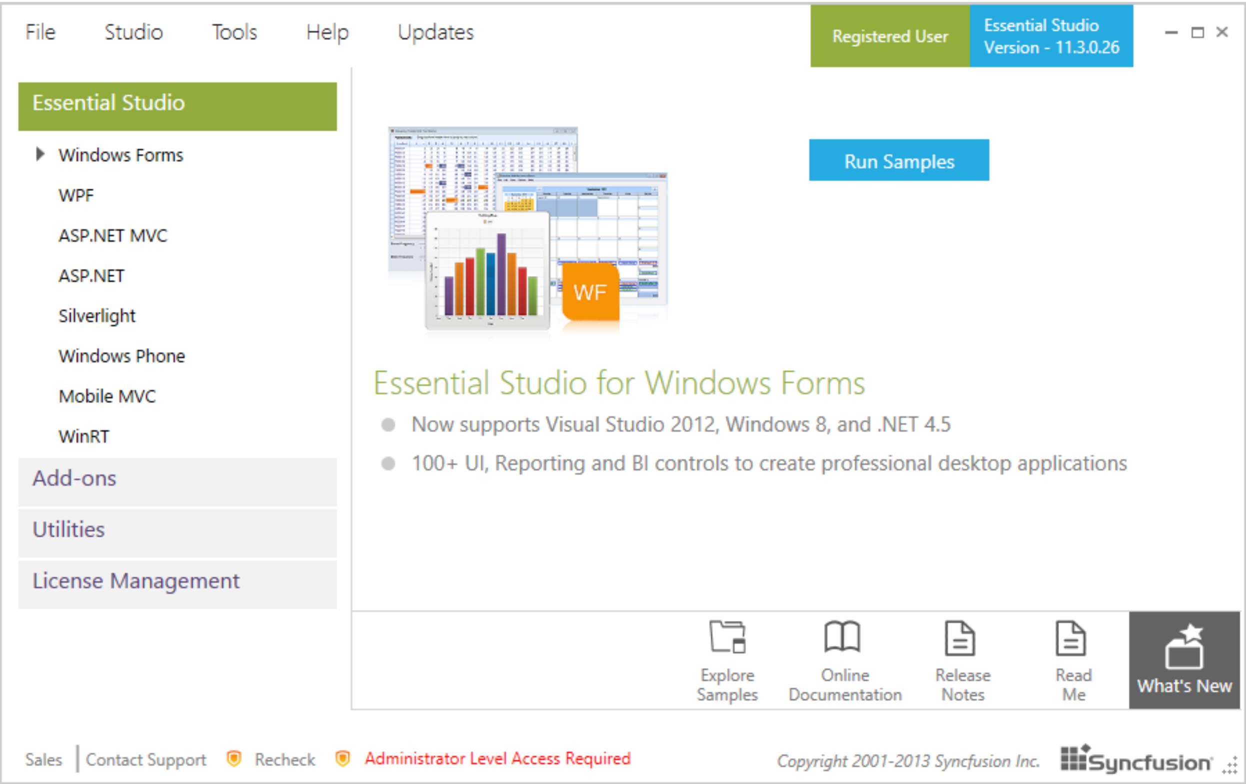
C:\Users\<user name>\AppData\Local\Syncfusion\EssentialStudio\<version number>\Windows\Gauge.Windows\Samples

2.2.2 Viewing Samples

Use the following steps to view the samples:

1. Click Start > All Programs > Syncfusion > Essential Studio <version number> >Dashboard

The Essential Studio Enterprise Edition window will be displayed.



Essential Studio for Windows Forms

- Now supports Visual Studio 2012, Windows 8, and .NET 4.5
- 100+ UI, Reporting and BI controls to create professional desktop applications

Administrator Level Access Required

Figure 1: Syncfusion Essential Studio Dashboard

2. In the Dashboard window, click **Run Samples** for **Windows Forms** under **UI Edition**. The UI **Windows Forms Sample Browser** window will be displayed.

Note: You can view the samples in any of the following three ways:

●Run Samples - Click to view the locally installed samples.

●Online Samples - Click to view online samples.

●Explore Samples - Explore the UI for Windows Forms on disk.

The User Interface Edition panel is displayed by default.

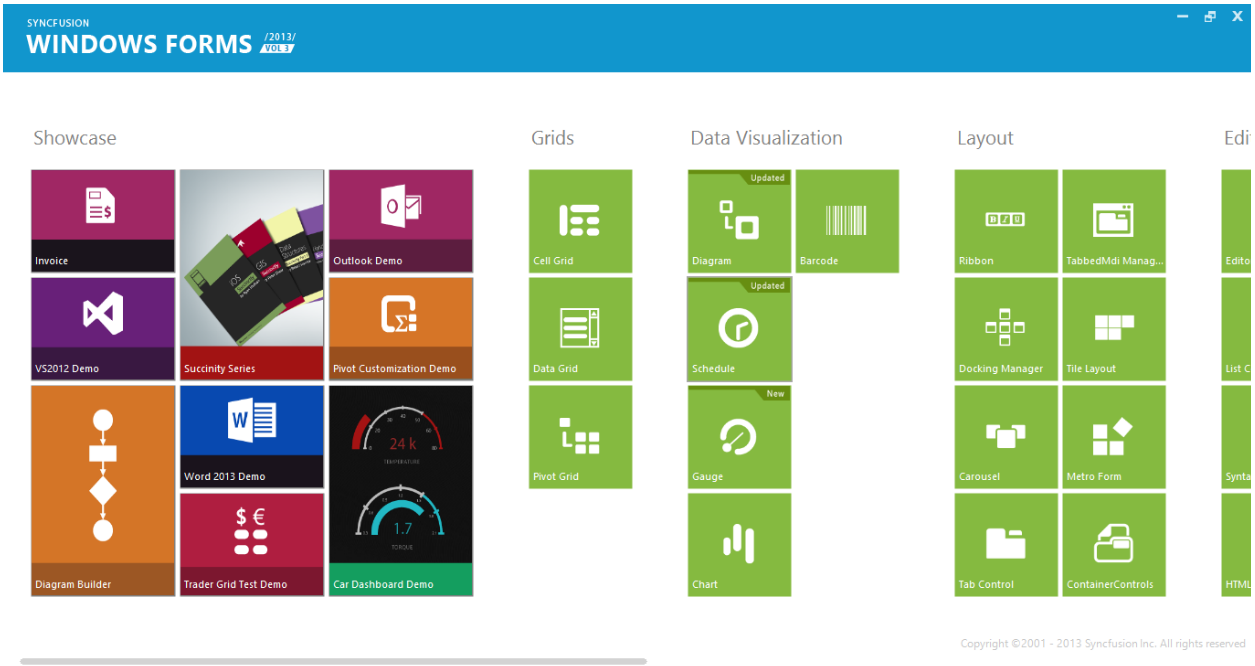


Figure 2: UI Windows Forms Sample Browser

3. Click the Gauge tile under Data Visualization. The Gauge samples will be displayed.

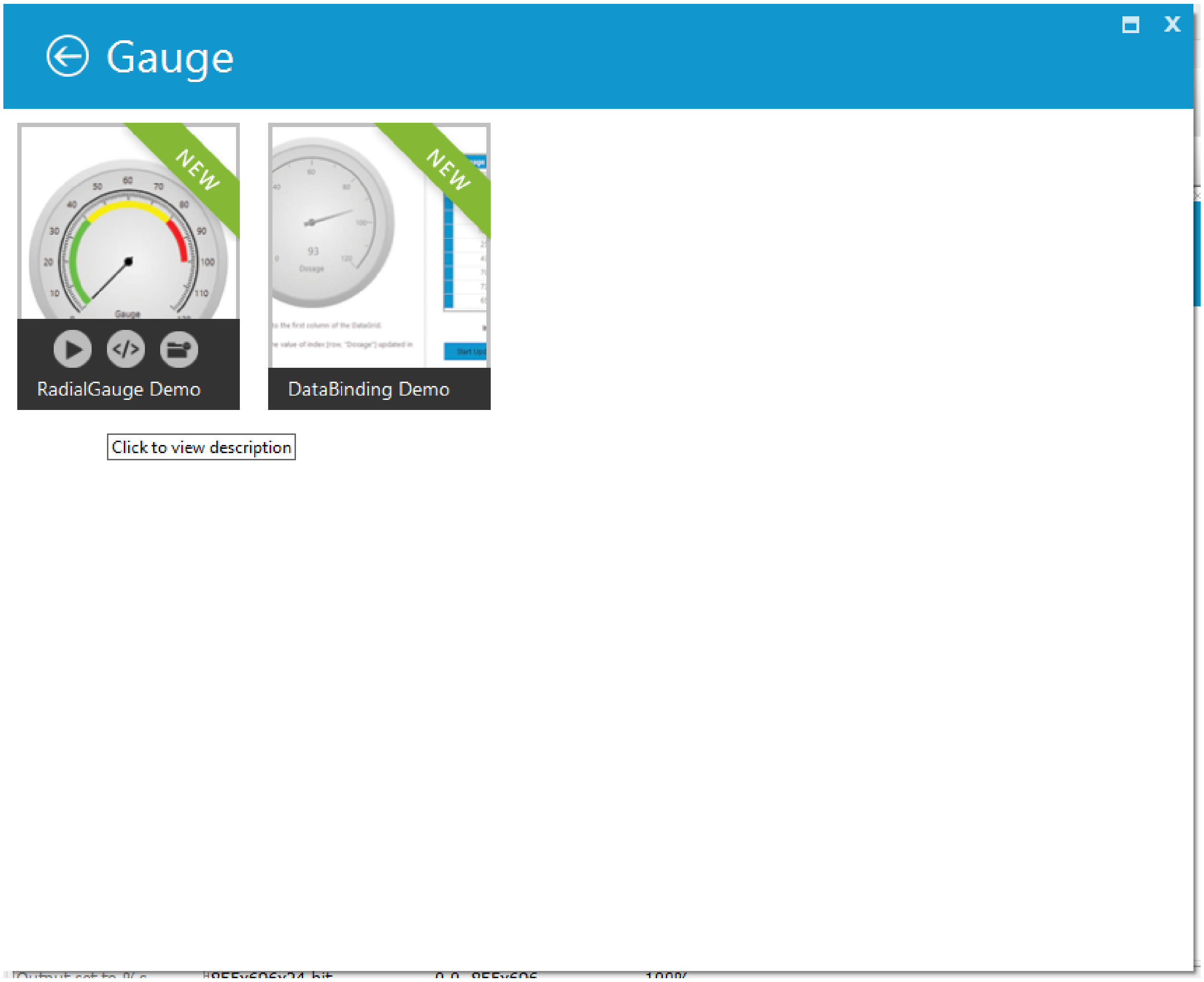


Figure 3: Essential Gauge WF Samples

4. Select any sample and browse through the features.

2.2.3 Source Code Location

The default location of the Windows Forms Gauge control source code is:

C:\Program Files\Syncfusion\Essential Studio\[VersionNumber]\Windows\Gauge.Windows\Src

2.3 Deployment Requirements

2.3.1 DLL List

While deploying an application that references a Syncfusion Windows Forms Gauge control assembly, the following dependencies must be included in the distribution:

- Syncfusion.Core.dll
- Syncfusion.Gauge.Windows.dll
- Syncfusion.Shared.base.dll

3 Radial Gauge

The **RadialGauge** control can be used for representing a range of values in circular form. It can be used to create sophisticated dashboards, clocks, industrial equipment, medical equipment, more.

3.1 Real World Scenarios

A radial gauge evaluates the values of scales and presents them in a radial manner. Radial Gauge enables you to quickly build high quality dashboard, process control, gadget and clocks. Radial gauges consist of important internal domains to present data in a more sophisticated way.

The best example of a radial gauge is a speedometer. The speedometer can be designed to be placed in a racing game application, denoting the speed of a vehicle.

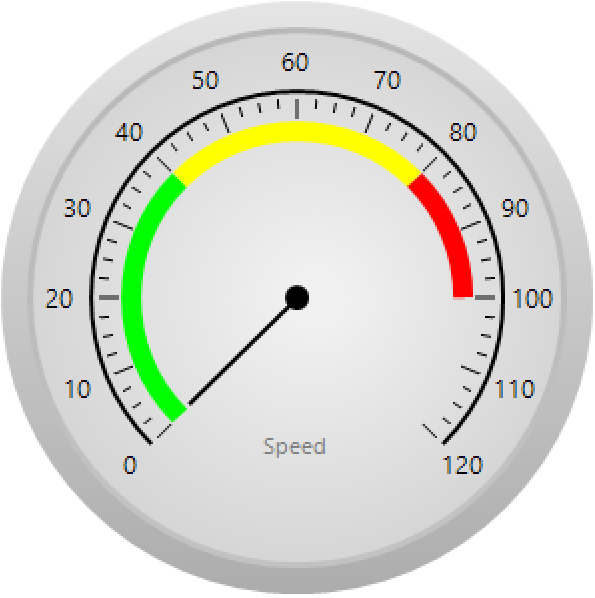


Figure 4: Speedometer

3.2 Getting Started

This section provides information about radial gauges for developers who are new to the Gauge control.

The Gauge control is built on Windows Forms programming, which is extended to support a broad set of dashboard development features including resources, controls, graphics, layout, and data binding.

3.2.1.1 Creating a Radial Gauge

Radial gauges can be enhanced with a circle frame or semi-circle frame. This section covers how to include a radial gauge in an application.

Through Designer:

Drag the RadialGauge control from the toolbox onto the form.

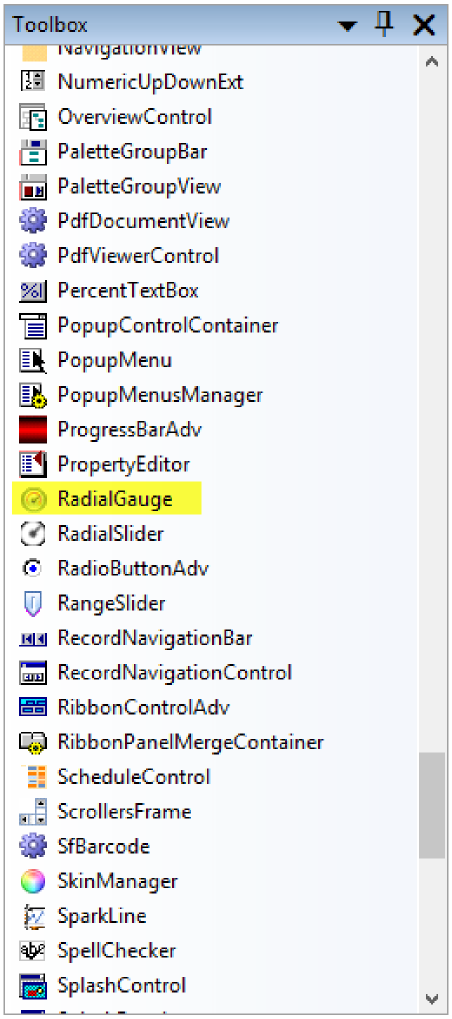


Figure 5: RadialGauge control selection

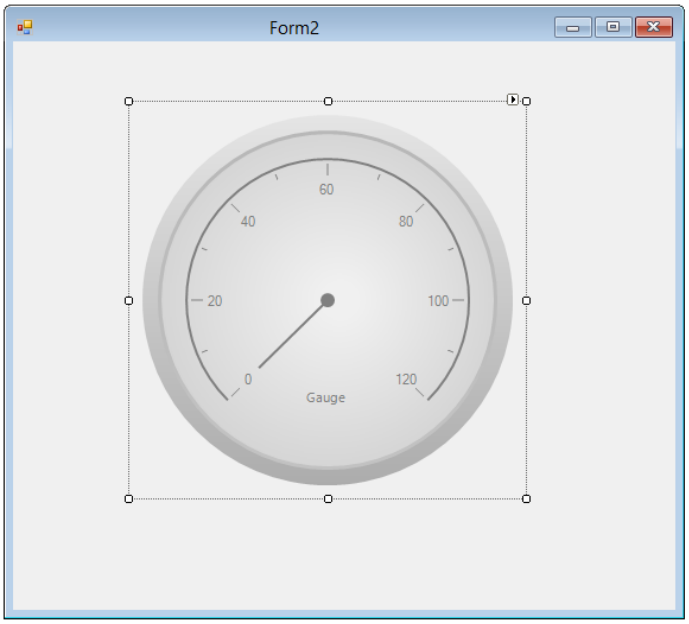


Figure 6: Default Radial Gauge

As soon as the control is dropped, it will be loaded with its default style.

Through Code:

```
[C#]

 private Syncfusion.Windows.Forms.Gauge.RadialGauge radialGauge1;
 this.radialGauge1 = new Syncfusion.Windows.Forms.Gauge.RadialGauge();
 this.radialGauge1.Name = "radialGauge1";
 this.radialGauge1.Size = new System.Drawing.Size(230, 230);
this.Controls.Add(this.radialGauge1);
```

```
 [VB]


 Private radialGauge1 As Syncfusion.Windows.Forms.Gauge.RadialGauge
 Me.radialGauge1 = New Syncfusion.Windows.Forms.Gauge.RadialGauge()
 Me.radialGauge1.Name = "radialGauge1"
 Me.radialGauge1.Size = New System.Drawing.Size(230, 230)
 Me.Controls.Add(Me.radialGauge1)
```

3.2.1.2 Elaborate Structure of the Control

The RadialGauge control includes the following elements: Scale Label, Needle, Gauge Value, and Background Frame. All the elements are optional for displaying the Gauge control.

Feature Summary:

Here is a brief overview of various features of the RadialGauge control.

- **Scales—**The scale value can be customized to be displayed within defined limits.
- **Frame types—**Allows you to specify the radial built-in frame style, such as full circle or half circle.
- **Pointers**—Provide complete support for pointers to point the value.
- **Range**—Full-fledged range support.
- **Image and Labels**—Used to customize the appearance of gauge pointer values with user-specified colors and alignment.

3.2.2 Concepts and Features

This section illustrates the features of Radial Gauge with images and sample code. It contains the following topics:

- Radial Gauge Frames
- Scales
- Ticks
- Needles
- Ranges
- Scaling Divisions

3.2.2.1 Radial Gauge Frame

The frame defines the frame types of radial gauges. Frames can be applied using the FrameType property. The RadialGauge control contains two frame types:

- Full Circle
- Half Circle

Property	Type	Description
FrameType	Enum	Gets or sets the frame type.
ShowBackgroundFrame	Boolean	Gets or sets the visibility of the background frame.

BackgroundGradientStartColor	Color	Gets or sets the gradient start color for the gauge background.
BackgroundGradientEndColor	Color	Gets or sets the gradient end color for the gauge background.
InnerFrameGradientStartColor	Color	Gets or sets the gradient start color for the inner frame.
InnerFrameGradientEndColor	Color	Gets or sets the gradient end color for the inner frame.
OuterFrameGradientStartColor	Color	Gets or sets the gradient end color for the outer frame.
OuterFrameGradientEndColor	Color	Gets or sets the gradient end color for the outer frame.
GaugeArcColor	Color	Gets or sets the arc color of the gauge.
GaugeLableColor	Color	Gets or sets the gauge label color.
GaugeValueColor	Color	Gets or sets the gauge value color.
ShowGaugeValue	Boolean	Gets or sets the gauge value visibility.

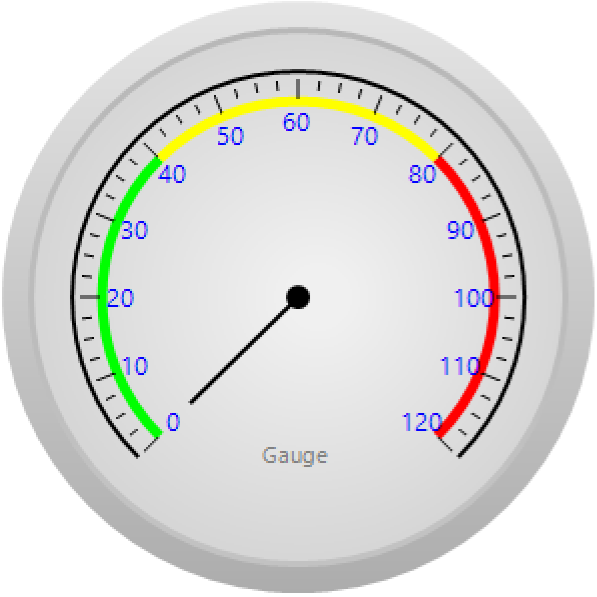


Figure 7: Full Circle Gauge

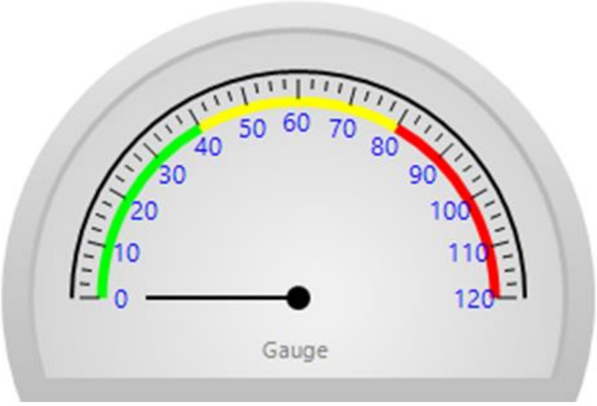


Figure 8: Half circle gauge

Code Sample:

```
[C#]

this.radialGauge1.FrameType =
Syncfusion.Windows.Forms.Gauge.FrameType.HalfCircle;
```

```
[VB]
```

```
Me.radialGauge1.FrameType =
Syncfusion.Windows.Forms.Gauge.FrameType.HalfCircle
```

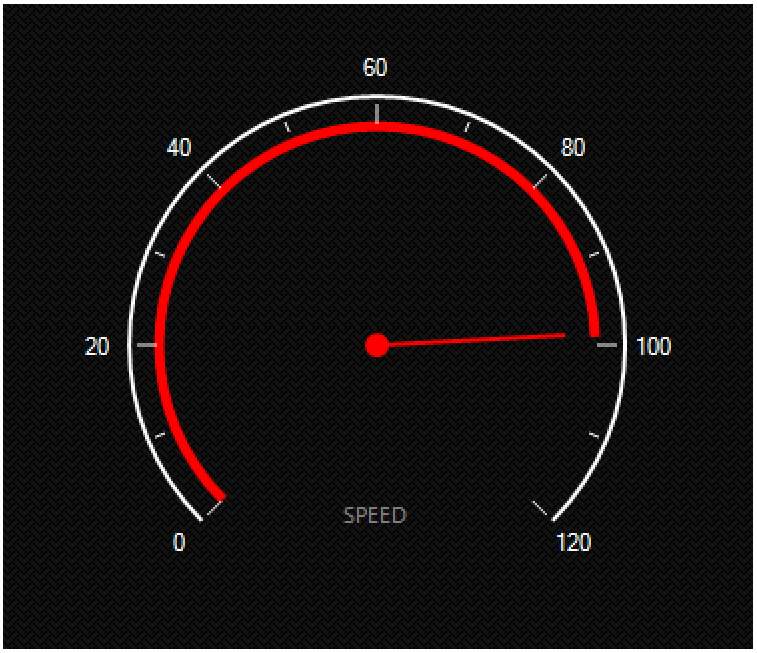


Figure 9: ShowBackgroundFrame as False(Transparency)

```
[C#]

 this.radialGauge1.ShowBackgroundFrame = false;
```

```
[VB]


Me.radialGauge1.ShowBackgroundFrame = false;
```

3.2.2.2 Scales

Scales are used to control element placement and value ranges.

Customizing Scales

You can customize scales added to the Radial Gauge using the properties listed in the following table:

Table 3: Property Table

Property	Type	Description
ShowScaleLabel	Boolean	Gets or sets the scale label visibility.
ScaleLabelColor	Color	Gets or sets the scale label color of the gauge.
LabelPlacement	Enum	Gets or sets whether to place the ticks inside or outside the arc.
TextOrientation	Enum	Gets or sets the text orientation layout.

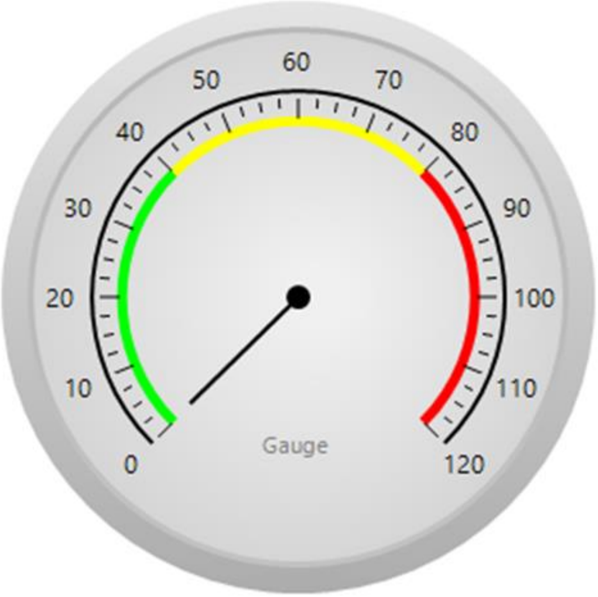


Figure 10: Scale outside the arc

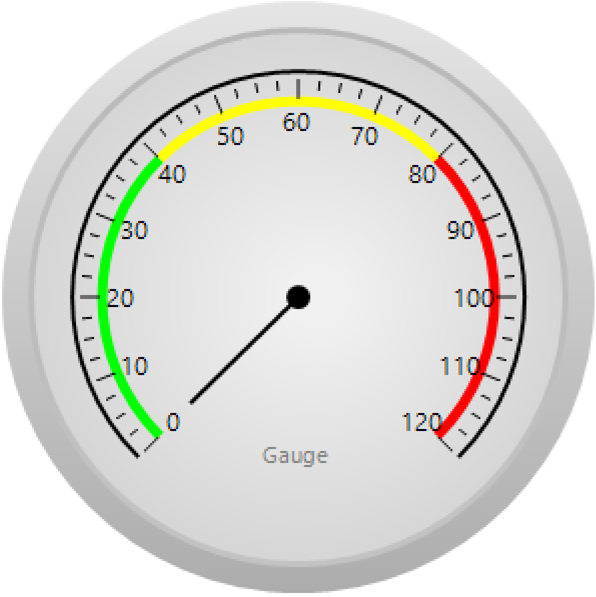


Figure 11: Scale inside the arc

The following code sample illustrates how to place labels in the RadialGauge control:

```
[C#]

this.radialGauge1.LabelPlacement =
Syncfusion.Windows.Forms.Gauge.LabelPlacement.Outside;

this.radialGauge1.TextOrientation =
Syncfusion.Windows.Forms.Gauge.TextOrientation.SlideOver;
```

```
[VB]

Me.radialGauge1.LabelPlacement =
Syncfusion.Windows.Forms.Gauge.LabelPlacement.Outside

Me.radialGauge1.TextOrientation =
Syncfusion.Windows.Forms.Gauge.TextOrientation.SlideOver
```

3.2.2.3 Ticks

Two types of ticks can be added to the RadialGauge control scale. Major tick marks are the primary scale indicators. Minor tick marks and Inter tick marks are the secondary scale indicators that fall between the major ticks. The ticks can be placed by setting the TickPlacement property. Ticks can be placed inside or outside the arc.

The following table lists the important properties that can be used to customize the radial tick marks. This is done to represent the scale with meaningful markers and labels.

Table 4: Property Table

Property	Type	Description
TickPlacement	Enum	Gets or sets whether to place the tickmarks inside or outside the arc.
MajorTickMarkColor	Color	Gets or sets the color of the major tickmarks.
MajorTickMarkHeight	Integer	Gets or sets the height of the major tickmarks.
MinorTickMarkColor	Color	Gets or sets the color of the minor tickmarks.
MinorTickMarkHeight	Integer	Gets or sets the height of the minor tickmarks.
InterLinesColor	Color	Gets or sets the color of the InterLines of the gauge.
MinorInnerLinesHeight	Integer	Gets or sets the line height of the minor inner lines.

The following code example illustrates how to add major and minor ticks to the radial scale.

```
[C#]

this.radialGauge1.TickPlacement =
Syncfusion.Windows.Forms.Gauge.TickPlacement.OutSide;

this.radialGauge1.MajorTickMarkColor = System.Drawing.Color.White;

this.radialGauge1.MinorTickMarkColor = System.Drawing.Color.White;

this.radialGauge1.InterLinesColor = System.Drawing.Color.White;
this.radialGauge1.MinorTickMarkHeight = 6;

this.radialGauge1.MajorTickMarkHeight = 12;

this.radialGauge1.MinorInnerLinesHeight = 6;
```

```
[VB]

Me.radialGauge1.TickPlacement =
Syncfusion.Windows.Forms.Gauge.TickPlacement.OutSide

Me.radialGauge1.MajorTickMarkColor = System.Drawing.Color.White

Me.radialGauge1.MinorTickMarkColor = System.Drawing.Color.White

Me.radialGauge1.InterLinesColor = System.Drawing.Color.White

Me.radialGauge1.MinorTickMarkHeight = 6

Me.radialGauge1.MajorTickMarkHeight = 12

Me.radialGauge1.MinorInnerLinesHeight = 6
```

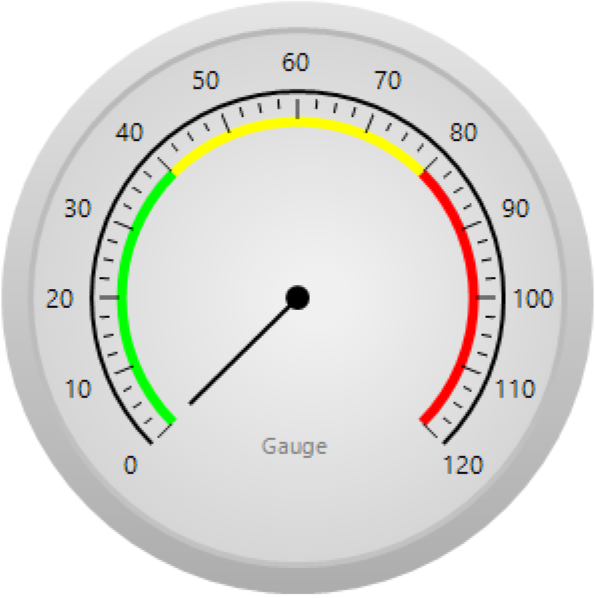


Figure 12: TickPlacement Inside

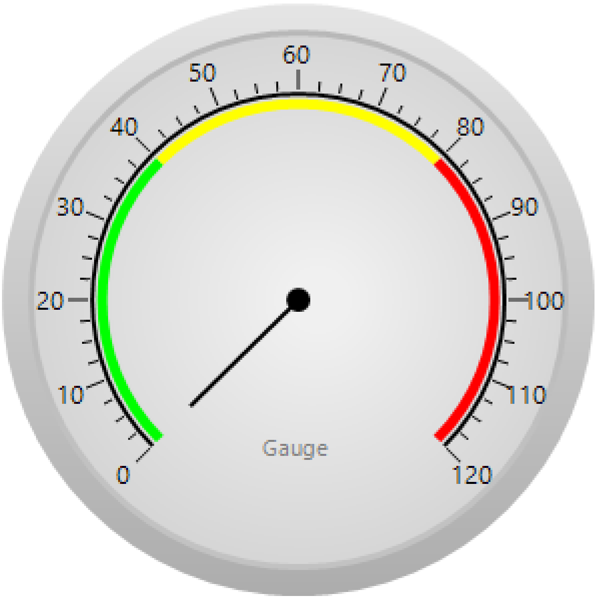


Figure 13: TickPlacement Outside

3.2.2.4 Needles

You can easily customize the style of the needle and the positions of the sub elements within the RadialGauge through the following enumerations

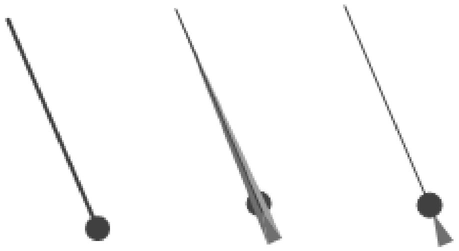


Figure 14: Default and Advanced pointer styles

Property	Type	Description
NeedleColor	Color	Gets or sets the gauge needle color.

NeedleStyle	Enum	Gets or sets the needle style.
ShowNeedle	Boolean	Gets or sets the needle visibility.

```
[C#]

      this.radialGauge1.ShowNeedle = true;

      this.radialGauge1.NeedleColor = System.Drawing.Color.Black;

      this.radialGauge1.NeedleStyle =
      Syncfusion.Windows.Forms.Gauge.NeedleStyle.Advanced;
```

```
[VB]

      Me.radialGauge1.ShowNeedle = True

      Me.radialGauge1.NeedleColor = System.Drawing.Color.Black

      Me.radialGauge1.NeedleStyle =
      Syncfusion.Windows.Forms.Gauge.NeedleStyle.Advanced
```

3.2.2.5 Ranges

Ranges are objects that highlight a range of values and can display different ranges in different colors. Ranges can be customized using various attributes such as range placement, height, color of the range, and so on. All the available attributes are listed in the following table:

Property	**Type**	**Description**
Startvalue	Integer	Specify the start value of the range. Default value is set to 0.
Endvalue	Integer	Specify the end value of the range. Default value is set to 0.
RangePlacement	Enum	Using this attribute, the range can be positioned in two areas along the radial scale. It includes the following options: • Inside • Outside The default value is Inside.

Height	Integer	Specify the height of the range. Default value is set to 5.
Color	Color	Gets or sets the color of the range.

The following code sample illustrates how to add ranges to the radial gauge:

```
[C#]

Syncfusion.Windows.Forms.Gauge.Range range1 = new
Syncfusion.Windows.Forms.Gauge.Range();

range1.Color = System.Drawing.Color.FromArgb(((int)(((byte)(225)))),
((int)(((byte)(128)))), ((int)(((byte)(128)))));

            range1.EndValue = 0F;

            range1.Height = 5;

            range1.InRange = false;

            range1.Name = "GaugeRange1";

            range1.RangePlacement =
Syncfusion.Windows.Forms.Gauge.TickPlacement.Inside;

            range1.StartValue = 0F;

            this.radialGauge1.Ranges.Add(range1);
```

```
[VB]

      Dim range1 As New Syncfusion.Windows.Forms.Gauge.Range()

      range1.Color = System.Drawing.Color.FromArgb(CInt(CByte(225)),
      CInt(CByte(128)), CInt(CByte(128)))

      range1.EndValue = 0F

      range1.Height = 5

      range1.InRange = False

      range1.Name = "GaugeRange1"

      range1.RangePlacement =
      Syncfusion.Windows.Forms.Gauge.TickPlacement.Inside

      range1.StartValue = 0F

      Me.radialGauge1.Ranges.Add(range1)
```

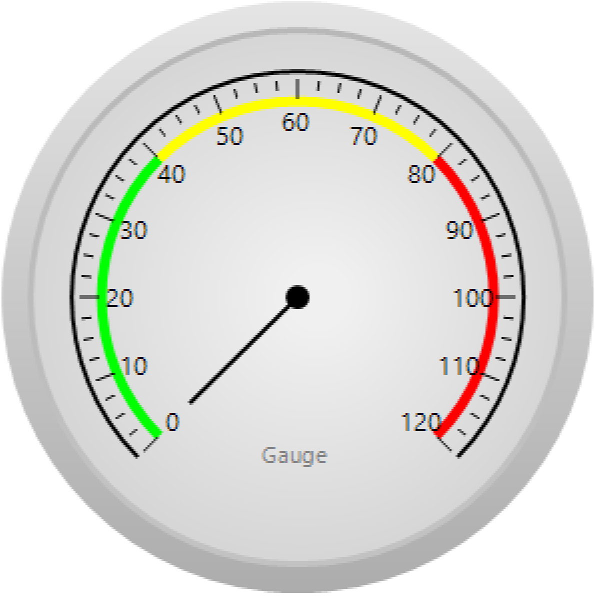


Figure 15: RangePlacement Inside

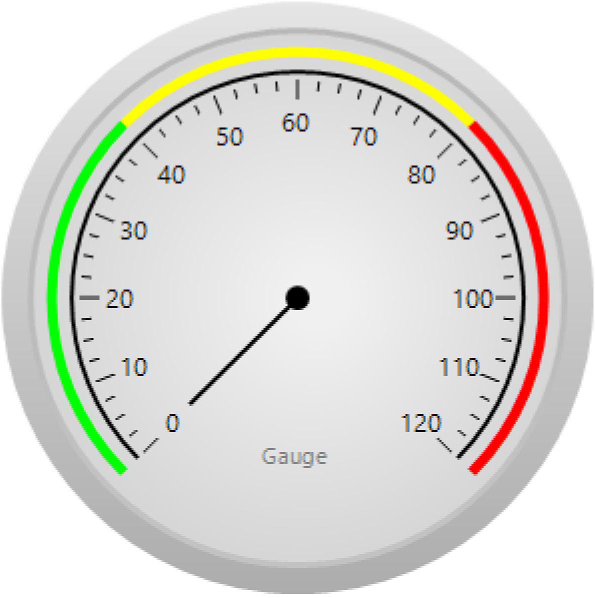


Figure 16: RangePlacement Outside

3.2.2.6 Scaling Divisions

The Gauge control for Windows Forms includes support for customizing the number of major tick lines and number of minor tick lines using the **Major Difference** and **Minor Difference**. It also provides support to customize the number of tick lines using **MaximumValue and MinimunValue.**

Properties

Property	Type	Description
Minimum	Float	Gets or sets the minimum value for the radial scale. Default value is set to 0.
Maximum	Float	Gets or sets the maximum value for the radial scale. Default value is set to 120.
MajorDifference	Float	Gets or sets the major difference value.
MinorDifference	Integer	Gets or sets the minor difference value.

[C#]

```
            this.radialGauge1.MajorDifference = 20F;
            this.radialGauge1.MaximumValue = 120F;
            this.radialGauge1.MinimumValue = 0F;
            this.radialGauge1.MinorDifference = 1;
```

[VB]

```
             Me.radialGauge1.MajorDifference = 20F
             Me.radialGauge1.MaximumValue = 120F
             Me.radialGauge1.MinimumValue = 0F
             Me.radialGauge1.MinorDifference = 1
```

3.3 Visual Styles for all the Gauges

The Gauge control for Windows Forms includes four stunning skins for professional representation of gauges. You can easily modify the look and feel of the gauge component using the built-in visual styles and color schemes.

The styles are built in for all of the gauges. Using SkinManager, these four styles can be assigned to the gauge. The following skins are available:

- Blue
- Black
- Silver
- Metro

Code sample:

```
[C#]


this.radialGauge1.VisualStyle =
Syncfusion.Windows.Forms.Gauge.ThemeStyle.Black;
```

```
[VB]


Me.radialGauge1.VisualStyle =

Syncfusion.Windows.Forms.Gauge.ThemeStyle.Black
```

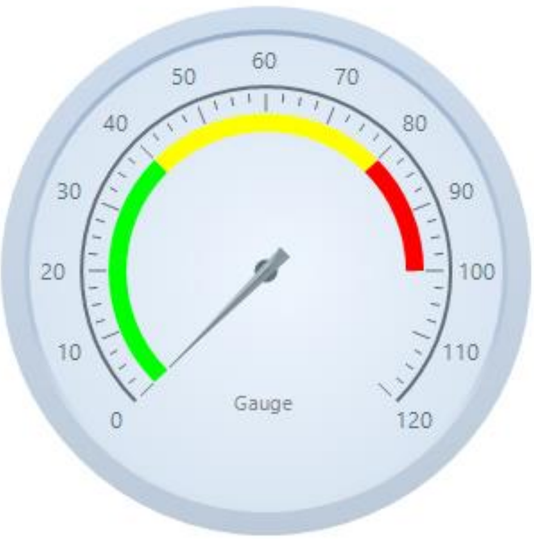


Figure 17: Blue

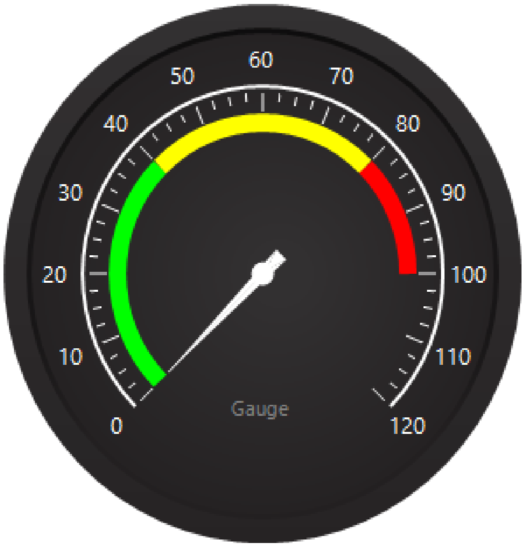


Figure 18: Black

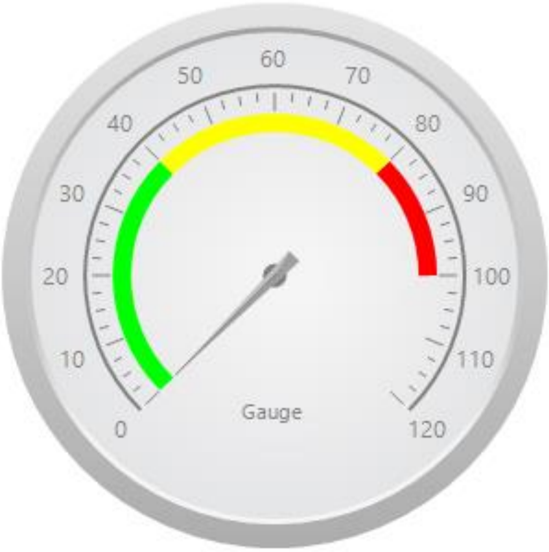


Figure 19: Silver

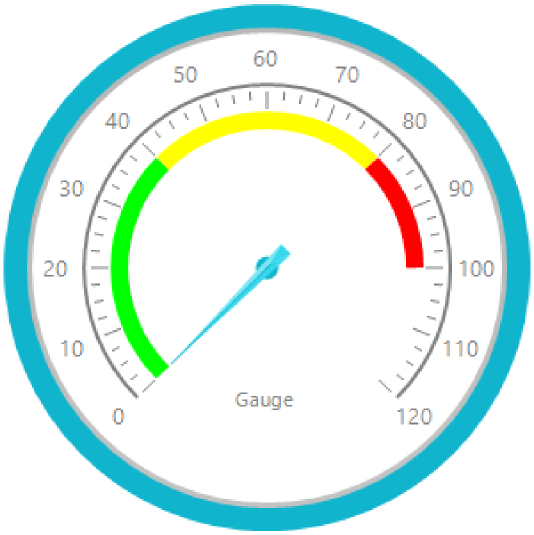


Figure 20: Metro

3.4 Data Binding

You can bind any data source to the RadialGauge and map an index of a record to represent the actual value in RadialGauge. The DisplayMember and DisplayRecordIndex properties will map the DataColumn and DataRow of the binding source respectively to the Gauge control, which will then support high frequency data updates.

Example:

```
this.radialGauge1.DataSource = dataTable;

this.radialGauge1.DisplayRecordIndex = [Row Index];

this.radialGauge1.DisplayMember = [column name];
```

```
[ASPX]
        <input type="text" value="" id="SetDropDowntext" />
         <input type="button" value="Set Text" id="setText" />

      <script type="text/javascript">
             $("#setText").bind('click', function () {
                 var multiDD = $find("MultiColumnDropdown");
                 multiDD.setText($('#SetDropDowntext').val());
             });
```

```
  </script>
```

Index

C

Compatibility 3

Concepts and Features 14

Creating a Radial Gauge 12

D

Data Binding 29

Deployment Requirements 9

DLL List 9

E

Elaborate Structure of the Control 14

G

Getting Started 11

I

Installation 5

Installation and Deployment 5

Introduction to Gauge for Windows Forms 3

N

Needles 22

O

Overview 3

P

Prerequisites 3

Prerequisites and Compatibility 3

R

Radial Gauge 11

Radial Gauge Frame 14

Ranges 23

Real World Scenarios 11

S

Samples and Location 5

Samples Installation Location 5

Scales 17

Scaling Divisions 25

Source Code Location 8

T

Ticks 19

V

Viewing Samples 5

Visual Styles for all the Gauges 26