



OCX Controls & software technologies
<http://www.precisionocx.com>

Precision Bubble Knob

Overview

Purpose

Precision Bubble Knob is a powerful OCX control for you applications which displays an interactive knob in form of a 3D bubble accepting input from the user. The user rotates it in order to choose a value in a predefined range. The control can cover an angle up to 360°.

Optimization

Precision Bubble Knob was written entirely in C++ and is composed of strongly optimized code. The only required libraries are the MFC and MSVCRT DLLs, that are shipped with every version of Microsoft Windows. You do not need to add any additional component to your application distribution kit.

Small size

Precision Bubble Knob is only 48k in size! This means that it is very fastly loaded at runtime into your application and it does not significantly increase your final application size.

High customisability

Precision Bubble Knob has 22 parameters you can set. For example it is possible to set the color of every single graphical element, the font of the text, the number of divisions, the geometry and the layout of the components, the line thickness and so on.

Graphics

Precision Bubble Knob makes no use of fixed-size bitmaps ! Every component you see, is dynamically scaled according to the container window size and to user's choices !

Flicker-free refresh

Precision Bubble Knob makes use of a triple-buffering technology, which does not only avoid unpleasant flickering effects, but it stores partial drawing results in order to minimize redrawing time. Every graphic manipulation is performed in memory, using when possible the results of the previous cycles, and only eventually the final result is transferred to the video buffer, using all the available hardware optimizations that the graphic card provides for the BitBlt.

Any special requirement ?

If you are developing an application and found Precision Bubble Knob nearly perfect for you needs except for small details, please feel free to contact us: let's talk about it !

Documentation

All the documentation you may need on Precision Bubble Knob is on line !

Quick start

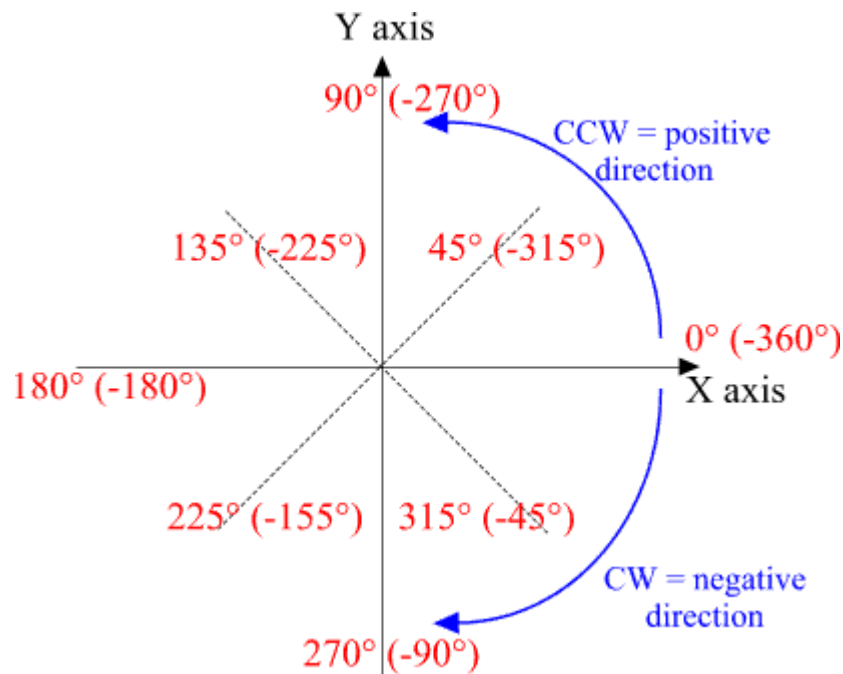
You can start using Precision Bubble Knob incredibly fast: the control default factory parameters are already sensible for a basic use, you have only got to:

- | If you are using the full version (NOT the trial one) you must call the *SetLicense* method passing your license number as argument and make sure it returns *True*. Simply ignore this step if you are testing the trial version.
- | Set the *MinimumValue* and the *MaximumValue* properties to the desired values.
- | Set the *StartAngle* and *EndAngle* properties to the appropriate values (you may want to read the following paragraph to make sure you do it right).
- | Set your current *Value* over and over again.

Now you can take a more accurate look at the properties to fully exploit the power of Precision Bubble Knob.

A note on angles

Since you can customize the knob in order to cover whichever angle you want, it is required that you correctly specify the correct *StartAngle* and *EndAngle*. By definition the start angle is the angle in which the *MinimumValue* will be positioned, while the end angle is the one in which there will be the *MaximumValue*. The angles are measured in the reference system portrayed in the following picture (everyone should remember the standard trigonometry reference system, as studied at the college):



In many cases it is natural to choose a *StartAngle* and an *EndAngle* such that the first is greater than the second, since the values increase clockwise (as in any real-world knob) whereas in the trigonometry reference system, the positive direction is the counterclockwise.

Nevertheless, it can happen the case in which you have a *StartAngle* which is less than the *EndAngle*, as in the following example: your graduated zone will cover exactly the right-hand part of the knob, in this case your start angle is 90° and your end angle is 270° .

In cases like these ones, in which the *StartAngle* is less than the *EndAngle*, subtract 360° to the *EndAngle*. Please see the example configurations in the [art gallery](#): pay attention to the angles they cover: the first example extends from 180 to -90 degrees (thus covering three quadrants out of four), the second one from 90 to -270 (thus covering the whole circle), the third one from 70 to -70 (thus covering a region on the right which is symmetrical with respect to the X axis).

Properties

BackColor

Domain: any color

Default value: 0x00C0C0C0 (light gray)

Meaning: this is the color used to fill the back plane of the control and the bubble of the control. This value can be dynamically varied to convey easy-readable status information to the user.

DivisionLineThickness

Domain: any integer number (small)

Default value: 2

Meaning: this is the thickness of the small segments which represent the division measuring marks. These marks are usually accompanied by text labels which indicates the value there assumed by the quantity you want to represent.

DotColor

Domain: any color

Default value: 0x000000FF (red)

Meaning: this is the color used to fill the little embossed dot which represent the current value chosen by the user or set in to the knob.

EndAngle

Domain: a meaningful real number

Default value: -45

Meaning: this is the angle, expressed in degrees, with respect to the instrument X axis (as indicated on the diagram; positive if counterclockwise, negative clockwise) in which the instrument graduated bar ends. The instrument graduated bar extends from the *StartAngle* to the *EndAngle* along a clockwise direction.

Font

Domain: a font

Default value: MS Sans Serif, 8

Meaning: this is the font used to draw the control's text labels associated with the division marks.

GradientFactor

Domain: a real number greater than 0 and suggested less than 2

Default value: 0.7

Meaning: this number describes how sharps are the color shades used to render the 3D aspect of the bubble. Please refer to the last part of the [art gallery](#) to see how changing this parameter impacts on the control's look.

LineColor

Domain: any color

Default value: 0x00808080 (gray)

Meaning: this is the color used to draw the division and subdivision marking lines.

MaximumValue

Domain: any real number

Default value: 100

Meaning: this is the maximum value you will allow the *Value* to assume.

MinimumValue

Domain: any real number

Default value: 0

Meaning: this is the minimum value you will allow the *Value* to assume.

NumberOfDivisions

Domain: any integer number (small)

Default value: 20

Meaning: this is the number of divisions in which the full angular range between *MinimumValue* and *MaximumValue* is divided, with the use of measuring marks. Division marks are usually accompanied by text labels, which indicate the corresponding values there assumed by the quantity you want to represent.

NumberOfSubdivisions

Domain: any integer number (small)

Default value: 2

Meaning: this is the number of subdivisions in which every division is furtherly divided. Subdivision marks are not accompanied by text labels.

RelativeBubbleRadius

Domain: any real number (suggested between 0.5 and 2)

Default value: 0.7

Meaning: this is the ratio between the radius of the bubble radius and the half of the minimum of the two edges of the rectangle which encloses the control.

RelativeDotPositionRadius

Domain: any real number (suggested between 0.5 and 2)

Default value: 0.75

Meaning: this is the ratio between the radius of the point in which is located the center of the little embossed dot and the radius of the bubble.

RelativeDotRadius

Domain: any real number (suggested between 0.5 and 2)

Default value: 0.11

Meaning: this is the ratio between the radius of the little embossed dot and the radius of the bubble.

RelativeExternalRadius

Domain: any real number (suggested between 0.5 and 2)

Default value: 1.2

Meaning: this is the ratio between the radius of the external circle of the division marking lines and the bubble radius.

RelativeInternalRadius

Domain: any real number (suggested between 0.5 and 2)

Default value: 1.04

Meaning: this is the ratio between radius of the internal circle of the division marking lines and the bubble radius.

RelativeTextRadius

Domain: any real number (suggested between 0.5 and 2)

Default value: 1.15

Meaning: this is the ratio between the text radius and the bubble radius. The text radius is the radius of the circle on which centers of the text labels associated to division marks are positioned.

StartAngle

Domain: a meaningful real number

Default value: 225

Meaning: this is the angle, expressed in degrees, with respect to the instrument X axis (as indicated on the diagram; positive if counterclockwise, negative clockwise) in which the instrument graduated bar start. The instrument graduated bar extends from the *StartAngle* to the *EndAngle* along a clockwise direction.

SubdivisionLineThickness

Domain: any integer number (small)

Default value: 1

Meaning: this is the thickness of the small segments which represent the subdivision measuring marks. These marks are not usually accompanied by text labels, whereas division ones do.

TextColor

Domain: any color

Default value: 0x00606060 (dark gray)

Meaning: this is the color used to draw the text associated with the division marks.

Value

Domain: any real number (recommended between *MaximumValue* and *MinimumValue*).

Default value: 0

Meaning: this is the actual value set by the user, or the value you want to represent with the control. The user is allowed to choose any value between *MinimumValue* and *MaximumValue*, the control will not permit him to move the knob out of the preset area.

This boundary is not enforced on the programmer's side, you will be able to set a value that **can** be greater than *MaximumValue* or less than *MinimumValue*: in this cases the instrument will behave exactly like real instruments, which can be brought out of range (eventually damaging them): the dot on the knob will move on, reaching zones out of the graduated bar and possibly finishing in an *unexpected* position, possibly after completing some revolutions.

ValueFormatString

Domain: any appropriate format string (see below)

Default value: `%.0f`

Meaning: : this string tells to the control how to format the text inside the labels located near the division measuring marks; this is a standard C "*printf()*-style" format string, but do not worry if you do not know the C programming language, every information you will need is hereby given.

- i If you do not want text labels on your control, you can simply set this property to an empty string.
- i If you want to specify simple text strings (constant), set this property to the desired string (strings are replicated many times unchanged).
- i Otherwise, you can use a format string composed by a preceding text, then the placeholder for the value and a following text. The placeholder itself is composed of
 - i a leading percent sign, followed by
 - i an (optional) padding size indicator (the number of total characters which compose the number representation, including digits, spaces and punctuation); if you put a zero "0" before this indicator, the padding character is the zero, otherwise the space is used;
 - i an (optional) precision indicator, preceded by a dot (the number of digits after the decimal point; set it to 0 if you want integer values);
 - i and an "f" character, which stands for "floating point value".

Examples are provided in the following table. Since the percent character is used to indicate the placeholder, if you want to represent the percent sign itself, you will have to use two percent signs.

Format string	Result
<code>%f</code>	50.000000
<code>value=%f</code>	value=50.000000
<code>value=%.1f</code>	value=50.0
<code>%.2f percent</code>	50.00 percent
<code>%06.2f</code>	050.00
<code>%.0f %%</code>	50 %
<code>blah %3.0f blah</code>	blah 50 blah

Methods

SetLicense(long LicenseNumber)

Parameters: *LicenseNumber*, a long integer number which represent your license number.

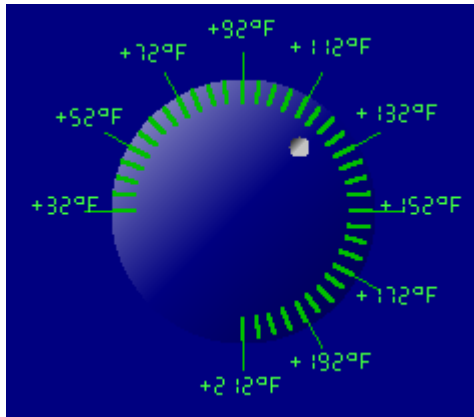
Return value: a boolean value that represents the result of the authentication.

Meaning: this method is used to activate the full-version control. You should call this method before calling any other method, passing the license number you obtained at purchase time as *LicenseNumber* parameter, otherwise the control will not draw itself on the screen.

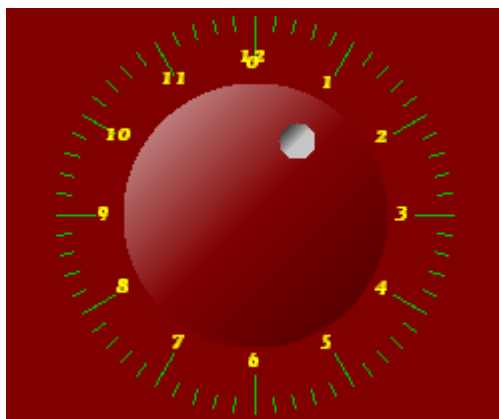
Attention: you do not need to use this method using the trial version of the control.

Art Gallery

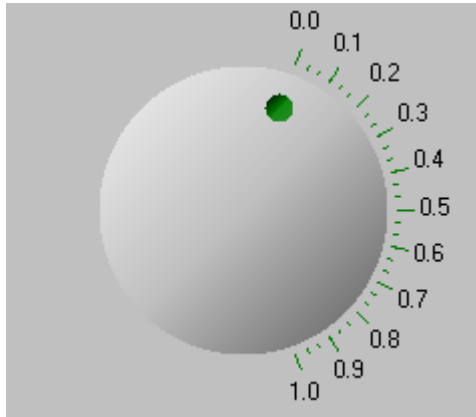
Some sample configurations



BackGroundColor	0x00800000
DivisionLineThickness	1
DotColor	0x00C0C0C0
EndAngle	-90
Font	DigitSmall
GradientFactor	0.7
LineColor	0x0000C000
MaximumValue	212
MinimumValue	32
NumberOfDivisions	9
NumberOfSubdivisions	4
RelativeBubbleRadius	0.65
RelativeDotPositionRadius	0.65
RelativeDotRadius	0.08
RelativeExternalRadius	1.22
RelativeInternalRadius	0.83
RelativeTextRadius	1.02
StartAngle	180
SubdivisionLineThickness	2
TextColor	0x0040FF40
Value	360
ValueFormatString	%+.0°F

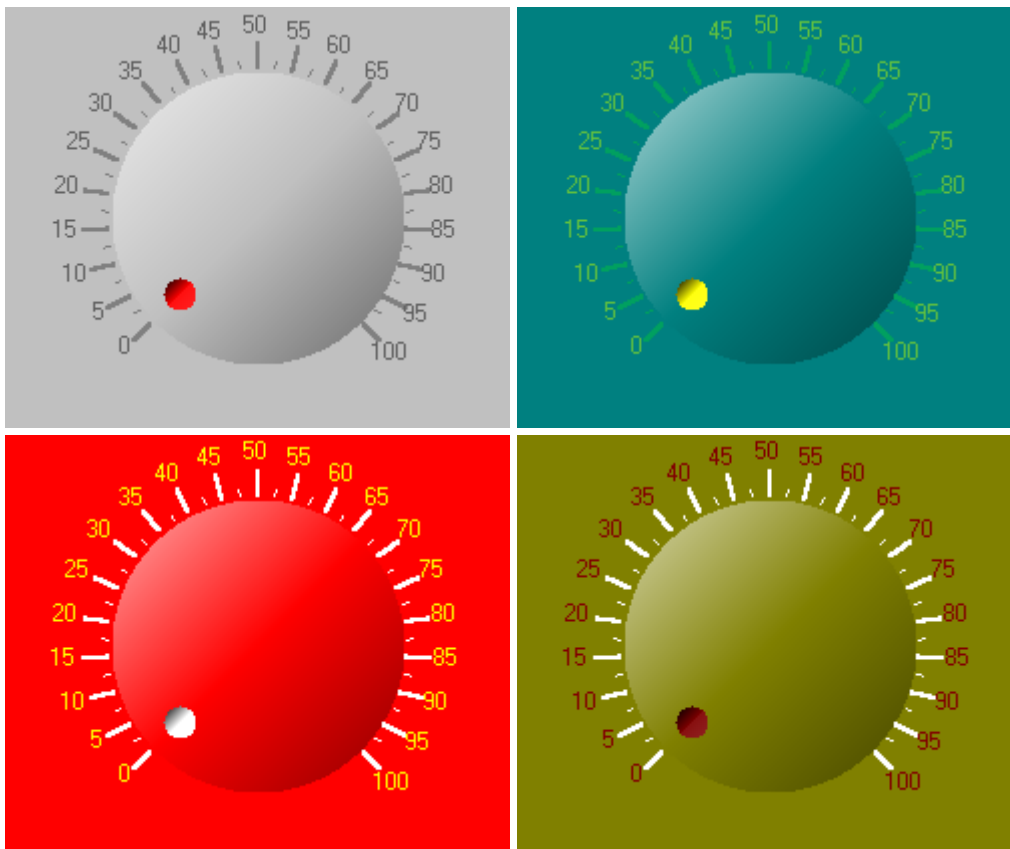


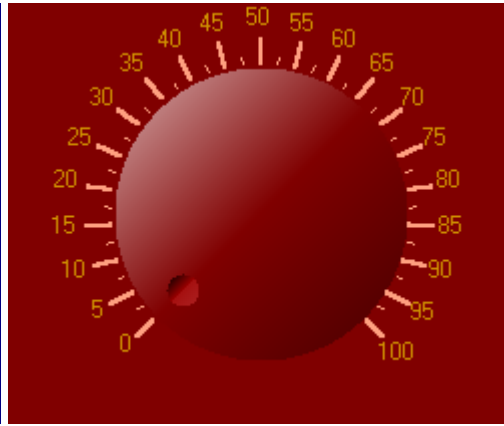
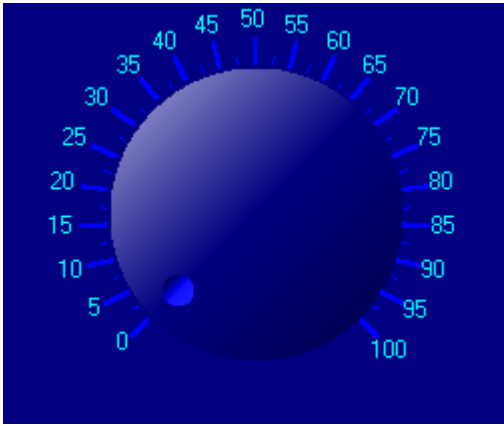
BackGroundColor	0x00000080
DivisionLineThickness	1
DotColor	0x00C0C0C0
EndAngle	-270
Font	Cooper Md BT
GradientFactor	0.7
LineColor	0x0000C000
MaximumValue	12
MinimumValue	0
NumberOfDivisions	12
NumberOfSubdivisions	5
RelativeBubbleRadius	0.65
RelativeDotPositionRadius	0.65
RelativeDotRadius	0.15
RelativeExternalRadius	1.2
RelativeInternalRadius	1.5
RelativeTextRadius	1.01
StartAngle	90
SubdivisionLineThickness	1
TextColor	0x0000FFFF
Value	1
ValueFormatString	%.0f



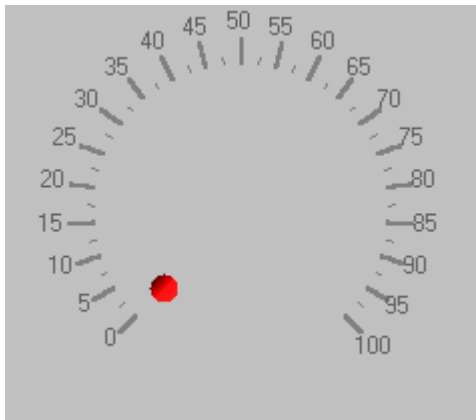
BackgroundColor	0x00C0C0C0
DivisionLineThickness	1
DotColor	0x00008000
EndAngle	-70
Font	MS Sans Serif
GradientFactor	0.8
LineColor	0x00008000
MaximumValue	1
MinimumValue	0
NumberOfDivisions	10
NumberOfSubdivisions	3
RelativeBubbleRadius	0.7
RelativeDotPositionRadius	0.75
RelativeDotRadius	0.11
RelativeExternalRadius	1.2
RelativeInternalRadius	1.07
RelativeTextRadius	1.2
StartAngle	70
SubdivisionLineThickness	1
TextColor	0x00000000
Value	0
ValueFormatString	%1f

Some color schemes

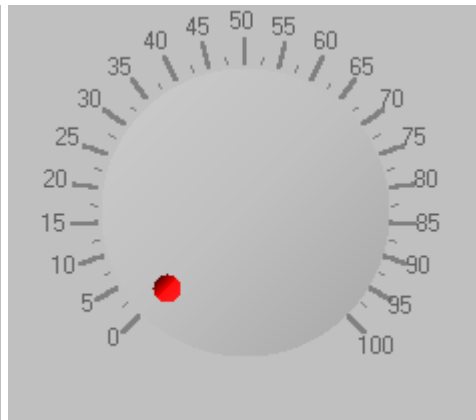




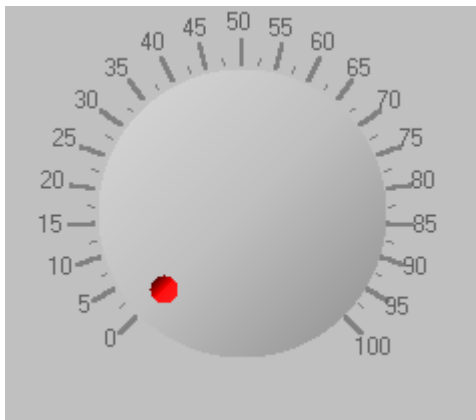
Changing Gradient factor



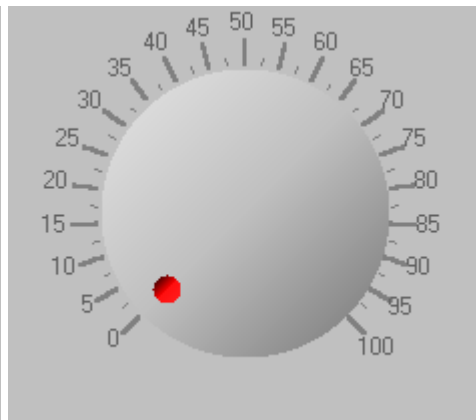
Gradient factor = 0



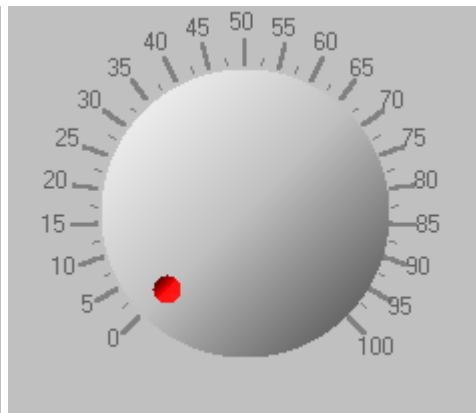
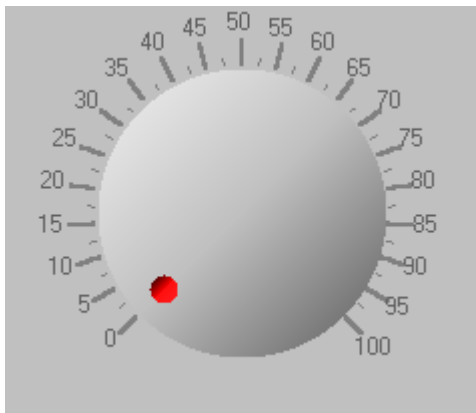
Gradient factor = 0.2



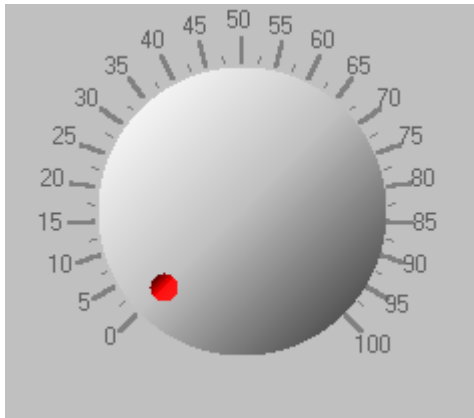
Gradient factor = 0.4



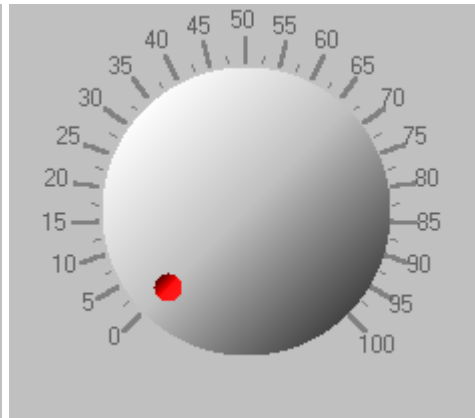
Gradient factor = 0.6



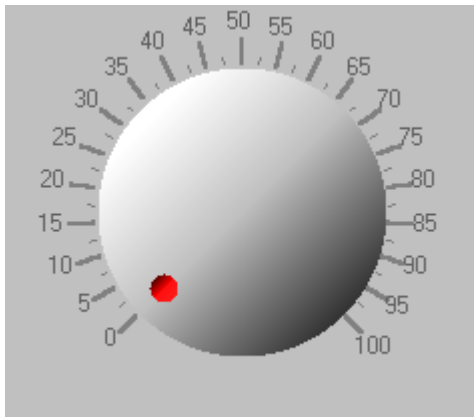
Gradient factor = 0.8



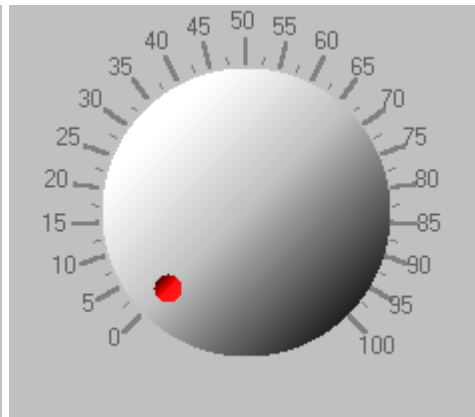
Gradient factor = 1.0



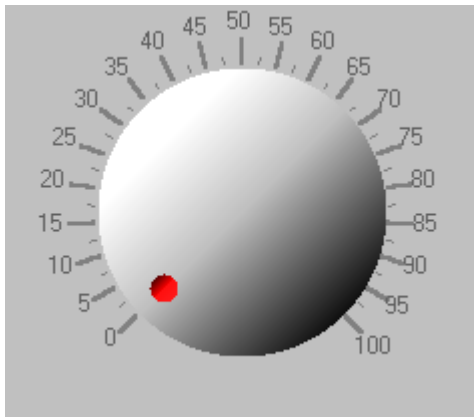
Gradient factor = 1.2



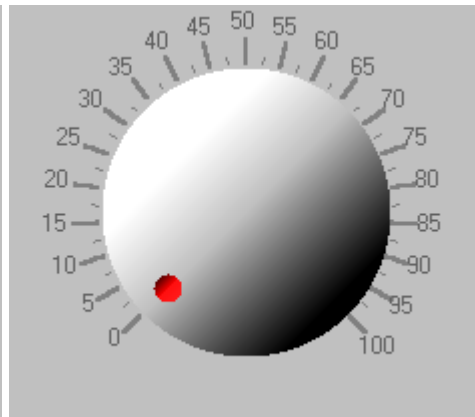
Gradient factor = 1.4



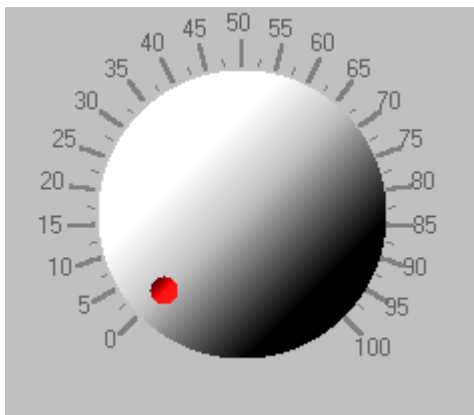
Gradient factor = 1.6



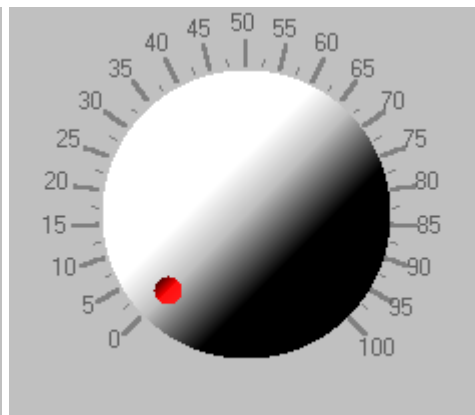
Gradient factor = 1.8



Gradient factor = 2.0

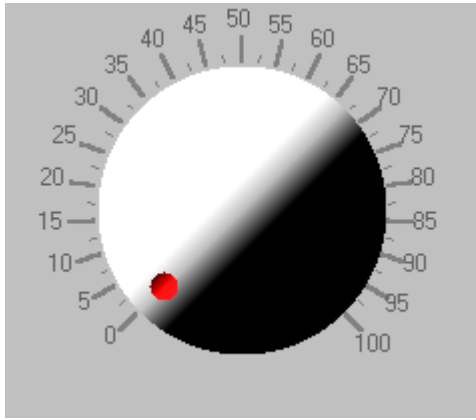


Gradient factor = 2.5

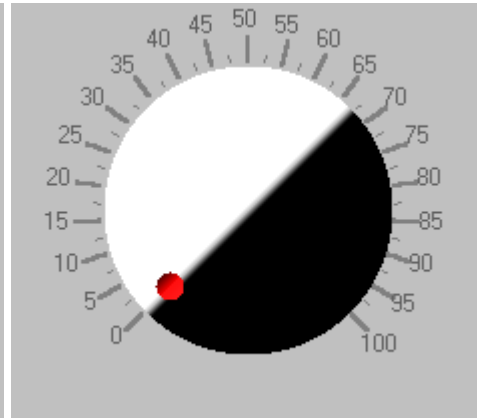


Gradient factor = 3.0

Gradient factor = 5.0



Gradient factor = 10.0



Gradient factor = 50.0