

# Namespace BOOSEapp

## Classes

[AppCanvas](#)

[AppCircle](#)

Specific implementation of the circle command. This is drawn on the canvas using a single integer parameter (radius)

[AppCommandFactory](#)

[AppMoveTo](#)

Custom MoveTo command for the application Handled through own command factory ( taking two params : x and y ) and moves drawing curser to that position

[AppRect](#)

Custom Rectangle class to be used in the command factory. This must be passed two sides as we are treating this purely as a rectangle. A square class can be made in the future which implements everything a rectangle does - but both side and height lengths will be equal.

[Form1](#)

# Class AppCanvas

Namespace: [BOOSEapp](#)

Assembly: BOOSEapp.dll

```
public class AppCanvas : ICanvas
```

## Inheritance

[object](#) ← AppCanvas

## Implements

ICanvas

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Constructors

## AppCanvas(int, int)

CONSTRUCTOR. Initializes a bitmap of the specified size (provided by two parameters) Asserts a Graphics object from the bitmap for drawing

```
public AppCanvas(int width, int height)
```

## Parameters

**width** [int](#)

horizontal x size of canvas

**height** [int](#)

vertical y size of canvas

# Properties

# PenColour

Gets or sets pens colour for drawing purposes

```
public object PenColour { get; set; }
```

Property Value

[object](#)

## Xpos

Gets or sets the horizontal position of the pen.

```
public int Xpos { get; set; }
```

Property Value

[int](#)

## Ypos

Gets or sets the vertical position of the pen.

```
public int Ypos { get; set; }
```

Property Value

[int](#)

# Methods

## Circle(int, bool)

Draws circle at cursor position of given radius There exists 4 overloaded ways of performing DrawEllipse.

```
public void Circle(int radius, bool filled)
```

## Parameters

**radius** [int](#)

Half the diameter of the circle (width or height)

**filled** [bool](#)

boolean to determine if shape is filled in or not

## Clear()

DOES NOT act as a complete reset Fills the background with the default colour

```
public void Clear()
```

## DrawTo(int, int)

Draws a line from the current pen position to the defined destination position The cursor position is then moved to the destination.

```
public void DrawTo(int x, int y)
```

## Parameters

**x** [int](#)

destinations lateral position of the pen

**y** [int](#)

destinations vertical position of the pen

## MoveTo(int, int)

Reassigns latest pen position to start next drawing

```
public void MoveTo(int x, int y)
```

## Parameters

x [int](#)

lateral location of new pen position

y [int](#)

vertical location of new pen position

## Rect(int, int, bool)

Draws a rectangle at the cursor position

```
public void Rect(int width, int height, bool filled)
```

## Parameters

width [int](#)

lateral size of the shape

height [int](#)

vertical size of the shape

filled [bool](#)

boolean to determine if this shape should be filled

## Reset()

resets cursor position and pen back to default values

```
public void Reset()
```

## Set(int, int)

Sets output display size. Drawing display is created of a specific size.

```
public void Set(int width, int height)
```

Parameters

width [int](#)

height [int](#)

Exceptions

[NotImplementedException](#)

## SetColour(int, int, int)

Sets the pen colour to RGB values

```
public void SetColour(int red, int green, int blue)
```

Parameters

red [int](#)

0 to 255 integer to determine RED weighting of the colour

green [int](#)

0 to 255 integer to determine GREEN weighting of the colour

blue [int](#)

0 to 255 integer to determine BLUE weighting of the colour

## Tri(int, int)

Draws a (close-to) equilateral triangle. Its location starts from the bottom left point.

```
public void Tri(int width, int height)
```

## Parameters

**width** [int](#)

horizontal size of the triangle. This is divided by two and the floor function is applied for an integer value

**height** [int](#)

Vertical size of the triangle

## WriteText(string)

Uses Font and Brush to help populate the parameters for g.drawString Currently not modular enough - static in that the font cannot be changed. The brush can change as this follows the pen colour

```
public void WriteText(string text)
```

## Parameters

**text** [string](#)

Provided string to be displayed on screen at the given cursor position

## getBitmap()

Gets the drawing object of whatever native type. Purpose is to use the return object to display/ output.

```
public object getBitmap()
```

## Returns

[object](#)

CanvasBitmap object to be cast to a native type

# Class AppCircle

Namespace: [BOOSEapp](#)

Assembly: BOOSEapp.dll

Specific implementation of the circle command. This is drawn on the canvas using a single integer parameter (radius)

```
public class AppCircle : CommandOneParameter, ICommand
```

## Inheritance

[object](#) ← Command ← CanvasCommand ← CommandOneParameter ← AppCircle

## Implements

ICommand

## Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
Command.IsDouble , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , Command.Compile() ,  
[Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program , Command.Name ,  
Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

# Constructors

## AppCircle()

Blank signature constructor for the AppCircle class, this inherits the CommandOneParameter clas (base)

```
public AppCircle()
```

## AppCircle(Canvas, int)

Constructor initialising a new instance of the AppCircle class with a specified canvas (for drawing) and radius

```
public AppCircle(Canvas c, int radius)
```

## Parameters

c Canvas

The canvas which the circle will be drawn on

radius int ↴

Radius provided for the circle instance

## Methods

### CheckParameters(string[])

Overrides base method to say radius and not p1

```
public override void CheckParameters(string[] parameterList)
```

## Parameters

parameterList string ↴[]

An array of parameters passed to the command

## Exceptions

CommandException

Thrown if the incorrect number of parameters are provided

## Execute()

Executes the circle command for drawing purposes. Overridden for the circles specific case. Validation of the parameter commences first. This includes checking if the value provided is not a double, and radius

is not less than 1. Exceptions are thrown if these conditions are not met.

```
public override void Execute()
```

## Exceptions

### CanvasException

Thrown if the parameter provided is a double (rather than an int) and if the radius provided is less than 1

# Class AppCommandFactory

Namespace: [BOOSEapp](#)

Assembly: BOOSEapp.dll

```
public class AppCommandFactory : CommandFactory, ICommandFactory
```

## Inheritance

[object](#) ← CommandFactory ← AppCommandFactory

## Implements

ICommandFactory

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Methods

### MakeCommand(string)

Makes a command object from the string provided. Additional commands requires a factory that extends this class - that also creates a MakeCommand for new commands This will call base.MakeCommand for existing commands The entire factory can be replaced by implementing an IFactory interface.

```
public override ICommand MakeCommand(string commandType)
```

#### Parameters

commandType [string](#)

String holding command to be created, may have additional spaces on either side and mixed case.  
This is tidied in the file

#### Returns

ICommand

ICommand Object if successful

## Exceptions

### FactoryException

Thrown if there is no such command 'commandType'

# Class AppMoveTo

Namespace: [BOOSEapp](#)

Assembly: BOOSEapp.dll

Custom MoveTo command for the application Handled through own command factory ( taking two params : x and y ) and moves drawing curser to that position

```
public class AppMoveTo : CommandTwoParameters, ICommand
```

## Inheritance

[object](#) ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← AppMoveTo

## Implements

ICommand

## Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,  
[CommandTwoParameters.CheckParameters\(string\[\]\)](#) , CommandOneParameter.param1 ,  
CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos ,  
CanvasCommand.canvas , CanvasCommand.Canvas , Command.IsFalse , Command.program ,  
Command.parameterList , Command.parameters , Command.paramsint ,  
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [Command.ProcessParameters\(string\)](#) ,  
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,  
Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#)

## Constructors

### AppMoveTo()

Default constructor, inheriting the constructor from CommandTwoParameters

```
public AppMoveTo()
```

## AppMoveTo(Canvas, int, int)

Manual constructor, storing the horizontal and vertical values of the new curser position

```
public AppMoveTo(Canvas c, int x, int y)
```

### Parameters

c Canvas

Canvas in question for the curser to move on

x int ↗

horizontal value of new curser position

y int ↗

vertical value of new curser position

## Methods

### Execute()

Execute command to move the curser position to the provided x,y variables.

```
public override void Execute()
```

### Exceptions

CanvasException

Thrown if a boolean is passed for any parameter, or if negative values are provided for the new curser position

# Class AppRect

Namespace: [BOOSEapp](#)

Assembly: BOOSEapp.dll

Custom Rectangle class to be used in the command factory. This must be passed two sides as we are treating this purely as a rectangle. A square class can be made in the future which implements everything a rectangle does - but both side and height lengths will be equal.

```
public class AppRect : CommandOneParameter, ICommand
```

## Inheritance

[object](#) ← Command ← CanvasCommand ← CommandOneParameter ← AppRect

## Implements

ICommand

## Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
Command.IsDouble , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , Command.Compile() ,  
[Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program , Command.Name ,  
Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

# Constructors

## AppRect()

Default constructor which inherits the constructor provided from CommandOneParameter

```
public AppRect()
```

## AppRect(Canvas, int, int)

Manual constructor for creating a rectangle with a specified size and canvas.

```
public AppRect(Canvas c, int sideLength, int heightLength)
```

## Parameters

c Canvas

Canvas provided for drawing purposes

sideLength int

horizontal size provided for the rectangle

heightLength int

vertical size provided for the rectangle

## Methods

### CheckParameters(string[])

Ensures that the command wasnt given too many parameters for a rectangle

```
public override void CheckParameters(string[] parameterList)
```

## Parameters

parameterList string[]

provides the parameters of the rectangle (excluding the "rect" part of the line)

## Exceptions

CommandException

Thrown if provided with more than 2 parameters, a rectangle can only have a specified width and height

## Execute()

Runs the rectangle command. First checks if parameters provided are double (rather than int) and if edge lengths are less than one. The rectangle is drawn on the canvas if all is well.

```
public override void Execute()
```

## Exceptions

### CanvasException

Thrown if a boolean is provided for parameters, or if the edge lengths are less than 1

# Class Form1

Namespace: [BOOSEapp](#)

Assembly: BOOSEapp.dll

```
public class Form1 : Form, IDropTarget, ISynchronizeInvoke, IWin32Window,  
IBindableComponent, IComponent, IDisposable, IContainerControl
```

## Inheritance

[object](#) ← [MarshalByRefObject](#) ← [Component](#) ← [Control](#) ← [ScrollableControl](#) ←  
[ContainerControl](#) ← [Form](#) ← Form1

## Implements

[IDropTarget](#), [ISynchronizeInvoke](#), [IWin32Window](#), [IBindableComponent](#), [IComponent](#),  
[IDisposable](#), [IContainerControl](#)

## Inherited Members

[Form.SetVisibleCore\(bool\)](#), [Form.Activate\(\)](#), [Form.ActivateMdiChild\(Form\)](#),  
[Form.AddOwnedForm\(Form\)](#), [Form.AdjustFormScrollbars\(bool\)](#), [Form.Close\(\)](#),  
[Form.CreateAccessibilityInstance\(\)](#), [Form.CreateControlsInstance\(\)](#), [Form.CreateHandle\(\)](#),  
[Form.DefWndProc\(ref Message\)](#), [Form.ProcessMnemonic\(char\)](#), [Form.CenterToParent\(\)](#),  
[Form.CenterToScreen\(\)](#), [Form.LayoutMdi\(MdiLayout\)](#), [Form.OnActivated\(EventArgs\)](#),  
[Form.OnBackgroundImageChanged\(EventArgs\)](#),  
[Form.OnBackgroundImageLayoutChanged\(EventArgs\)](#), [Form.OnClosing\(CancelEventArgs\)](#),  
[Form.OnClosed\(EventArgs\)](#), [Form.OnFormClosing\(FormClosingEventArgs\)](#),  
[Form.OnFormClosed\(FormClosedEventArgs\)](#), [Form.OnCreateControl\(\)](#),  
[Form.OnDeactivate\(EventArgs\)](#), [Form.OnEnabledChanged\(EventArgs\)](#), [Form.OnEnter\(EventArgs\)](#),  
[Form.OnFontChanged\(EventArgs\)](#), [Form.OnGotFocus\(EventArgs\)](#),  
[Form.OnHandleCreated\(EventArgs\)](#), [Form.OnHandleDestroyed\(EventArgs\)](#),  
[Form.OnHelpButtonClicked\(CancelEventArgs\)](#), [Form.OnLayout\(LayoutEventArgs\)](#),  
[Form.OnLoad\(EventArgs\)](#), [Form.OnMaximizedBoundsChanged\(EventArgs\)](#),  
[Form.OnMaximumSizeChanged\(EventArgs\)](#), [Form.OnMinimumSizeChanged\(EventArgs\)](#),  
[Form.OnInputLanguageChanged\(InputLanguageChangedEventArgs\)](#),  
[Form.OnInputLanguageChanging\(InputLanguageChangingEventArgs\)](#),  
[Form.OnVisibleChanged\(EventArgs\)](#), [Form.OnMdiChildActivate\(EventArgs\)](#),  
[Form.OnMenuStart\(EventArgs\)](#), [Form.OnMenuComplete\(EventArgs\)](#),  
[Form.OnPaint\(PaintEventArgs\)](#), [Form.OnResize\(EventArgs\)](#),  
[Form.OnDpiChanged\(DpiChangedEventArgs\)](#), [Form.OnGetDpiScaledSize\(int, int, ref Size\)](#),  
[Form.OnRightToLeftLayoutChanged\(EventArgs\)](#), [Form.OnShown\(EventArgs\)](#),

[Form.OnTextChanged\(EventArgs\)](#) , [Form.ProcessCmdKey\(ref Message, Keys\)](#) ,  
[Form.ProcessDialogKey\(Keys\)](#) , [Form.ProcessDialogChar\(char\)](#) ,  
[Form.ProcessKeyPreview\(ref Message\)](#) , [Form.ProcessTabKey\(bool\)](#) ,  
[Form.RemoveOwnedForm\(Form\)](#) , [Form.Select\(bool, bool\)](#) ,  
[Form.ScaleMinAxisSize\(float, float, bool\)](#) ,  
[Form.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#) ,  
[Form.ScaleControl\(SizeF, BoundsSpecified\)](#) , [Form.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#) ,  
[Form.SetClientSizeCore\(int, int\)](#) , [Form.SetDesktopBounds\(int, int, int, int\)](#) ,  
[Form.SetDesktopLocation\(int, int\)](#) , [Form.Show\(IWin32Window\)](#) , [Form.ShowDialog\(\)](#) ,  
[Form.ShowDialog\(IWin32Window\)](#) , [Form.ToString\(\)](#) , [Form.UpdateDefaultButton\(\)](#) ,  
[Form.OnResizeBegin\(EventArgs\)](#) , [Form.OnResizeEnd\(EventArgs\)](#) ,  
[Form.OnStyleChanged\(EventArgs\)](#) , [Form.ValidateChildren\(\)](#) ,  
[Form.ValidateChildren\(ValidationConstraints\)](#) , [Form.WndProc\(ref Message\)](#) , [Form.AcceptButton](#) ,  
[Form.ActiveForm](#) , [Form.ActiveMdiChild](#) , [Form.AllowTransparency](#) , [Form.AutoScroll](#) ,  
[Form.AutoSize](#) , [Form.AutoSizeMode](#) , [Form.AutoValidate](#) , [Form.BackColor](#) ,  
[Form.FormBorderStyle](#) , [Form.CancelButton](#) , [Form.ClientSize](#) , [Form.ControlBox](#) ,  
[Form.CreateParams](#) , [Form.DefaultImeMode](#) , [Form.DefaultSize](#) , [Form.DesktopBounds](#) ,  
[Form/DesktopLocation](#) , [Form/DialogResult](#) , [Form/HelpButton](#) , [Form/Icon](#) , [Form/IsMdiChild](#) ,  
[Form/IsMdiContainer](#) , [Form/IsRestrictedWindow](#) , [Form/KeyPreview](#) , [Form/Location](#) ,  
[Form/MaximizedBounds](#) , [Form/MaximumSize](#) , [Form/MainMenuStrip](#) , [Form/MinimumSize](#) ,  
[Form/MaximizeBox](#) , [Form/MdiChildren](#) , [Form/MdiChildrenMinimizedAnchorBottom](#) ,  
[Form/MdiParent](#) , [Form/MinimizeBox](#) , [Form/Modal](#) , [Form/Opacity](#) , [Form/OwnedForms](#) ,  
[Form/Owner](#) , [Form/RestoreBounds](#) , [Form/RightToLeftLayout](#) , [Form>ShowInTaskbar](#) ,  
[Form>ShowIcon](#) , [Form>ShowWithoutActivation](#) , [Form/Size](#) , [Form/SizeGripStyle](#) ,  
[Form/StartPosition](#) , [Form/Text](#) , [Form/TopLevel](#) , [Form/TopMost](#) , [Form/TransparencyKey](#) ,  
[Form/WindowState](#) , [Form/AutoSizeChanged](#) , [Form/AutoValidateChanged](#) ,  
[Form/HelpButtonClicked](#) , [Form/MaximizedBoundsChanged](#) , [Form/MaximumSizeChanged](#) ,  
[Form/MinimumSizeChanged](#) , [Form/Activated](#) , [Form/Deactivate](#) , [Form/FormClosing](#) ,  
[Form/FormClosed](#) , [Form/Load](#) , [Form/MdiChildActivate](#) , [Form/MenuComplete](#) ,  
[Form/MenuStart](#) , [Form/InputLanguageChanged](#) , [Form/InputLanguageChanging](#) ,  
[Form/RightToLeftLayoutChanged](#) , [Form/Shown](#) , [Form/DpiChanged](#) , [Form/ResizeBegin](#) ,  
[Form/ResizeEnd](#) , [ContainerControl.OnAutoValidateChanged\(EventArgs\)](#) ,  
[ContainerControl.OnMove\(EventArgs\)](#) , [ContainerControl.OnParentChanged\(EventArgs\)](#) ,  
[ContainerControl.PerformLayout\(\)](#) , [ContainerControl.RescaleConstantsForDpi\(int, int\)](#) ,  
[ContainerControl.Validate\(\)](#) , [ContainerControl.Validate\(bool\)](#) ,  
[ContainerControl.AutoScaleDimensions](#) , [ContainerControl.AutoScaleFactor](#) ,  
[ContainerControl.AutoScaleMode](#) , [ContainerControl.BindingContext](#) ,  
[ContainerControl.CanEnableIme](#) , [ContainerControl.ActiveControl](#) ,  
[ContainerControl.CurrentAutoScaleDimensions](#) , [ContainerControl.ParentForm](#) ,  
[ScrollableControl.ScrollStateAutoScrolling](#) , [ScrollableControl.ScrollStateHScrollVisible](#) ,

[ScrollableControl.ScrollStateVScrollVisible](#) , [ScrollableControl.ScrollStateUserHasScrolled](#) ,  
[ScrollableControl.ScrollStateFullDrag](#) , [ScrollableControl.GetScrollState\(int\)](#) ,  
[ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#) ,  
[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,  
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,  
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.SetDisplayRectLocation\(int, int\)](#) ,  
[ScrollableControl.ScrollControlIntoView\(Control\)](#) , [ScrollableControl.ScrollToControl\(Control\)](#) ,  
[ScrollableControl.OnScroll\(ScrollEventArgs\)](#) , [ScrollableControl.SetAutoScrollMargin\(int, int\)](#) ,  
[ScrollableControl.SetScrollState\(int, bool\)](#) , [ScrollableControl.AutoScrollMargin](#) ,  
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,  
[ScrollableControl.DisplayRectangle](#) , [ScrollableControl.HScroll](#) , [ScrollableControl.HorizontalScroll](#) ,  
[ScrollableControl.VScroll](#) , [ScrollableControl.VerticalScroll](#) , [ScrollableControl.Scroll](#) ,  
[Control.GetAccessibilityObjectById\(int\)](#) , [Control.SetAutoSizeMode\(AutoSizeMode\)](#) ,  
[Control.GetAutoSizeMode\(\)](#) , [Control.GetPreferredSize\(Size\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#) , [Control.BeginInvoke\(Delegate\)](#) ,  
[Control.BeginInvoke\(Action\)](#) , [Control.BeginInvoke\(Delegate, params object\[\]\)](#) ,  
[Control.BringToFront\(\)](#) , [Control.Contains\(Control\)](#) , [Control.CreateGraphics\(\)](#) ,  
[Control.CreateControl\(\)](#) , [Control.DestroyHandle\(\)](#) , [Control.DoDragDrop\(object, DragDropEffects\)](#) ,  
[Control.DoDragDrop\(object, DragDropEffects, Bitmap, Point, bool\)](#) ,  
[Control.DrawToBitmap\(Bitmap, Rectangle\)](#) , [Control.EndInvoke\(IAsyncResult\)](#) , [Control.FindForm\(\)](#) ,  
[Control.GetTopLevel\(\)](#) , [Control.RaiseKeyEvent\(object, KeyEventArgs\)](#) ,  
[Control.RaiseMouseEvent\(object, MouseEventArgs\)](#) , [Control.Focus\(\)](#) ,  
[Control.FromChildHandle\(nint\)](#) , [Control.FromHandle\(nint\)](#) ,  
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#) , [Control.GetChildAtPoint\(Point\)](#) ,  
[Control.GetContainerControl\(\)](#) , [Control.GetNextControl\(Control, bool\)](#) ,  
[Control.GetStyle\(ControlStyles\)](#) , [Control.Hide\(\)](#) , [Control.InitLayout\(\)](#) , [Control.Invalidate\(Region\)](#) ,  
[Control.Invalidate\(Region, bool\)](#) , [Control.Invalidate\(\)](#) , [Control.Invalidate\(bool\)](#) ,  
[Control.Invalidate\(Rectangle\)](#) , [Control.Invalidate\(Rectangle, bool\)](#) , [Control.Invoke\(Action\)](#) ,  
[Control.Invoke\(Delegate\)](#) , [Control.Invoke\(Delegate, params object\[\]\)](#) ,  
[Control.Invoke<T>\(Func<T>\)](#) , [Control.InvokePaint\(Control, PaintEventArgs\)](#) ,  
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#) , [Control.IsKeyLocked\(Keys\)](#) ,  
[Control.IsInputChar\(char\)](#) , [Control.IsInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,  
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.LogicalToDeviceUnits\(Size\)](#) ,  
[Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) , [Control.NotifyInvalidate\(Rectangle\)](#) ,  
[Control.InvokeOnClick\(Control, EventArgs\)](#) , [Control.OnAutoSizeChanged\(EventArgs\)](#) ,  
[Control.OnBackColorChanged\(EventArgs\)](#) , [Control.OnBindingContextChanged\(EventArgs\)](#) ,  
[Control.OnCausesValidationChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,  
[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDataContextChanged\(EventArgs\)](#) ,  
[Control.OnDockChanged\(EventArgs\)](#) , [Control.OnForeColorChanged\(EventArgs\)](#) ,

[Control.OnNotifyMessage\(Message\)](#) , [Control.OnParentBackColorChanged\(EventArgs\)](#) ,  
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#) ,  
[Control.OnParentBindingContextChanged\(EventArgs\)](#) , [Control.OnParentCursorChanged\(EventArgs\)](#) ,  
[Control.OnParentDataContextChanged\(EventArgs\)](#) , [Control.OnParentEnabledChanged\(EventArgs\)](#) ,  
[Control.OnParentFontChanged\(EventArgs\)](#) , [Control.OnParentForeColorChanged\(EventArgs\)](#) ,  
[Control.OnParentRightToLeftChanged\(EventArgs\)](#) , [Control.OnParentVisibleChanged\(EventArgs\)](#) ,  
[Control.OnPrint\(PaintEventArgs\)](#) , [Control.OnTabIndexChanged\(EventArgs\)](#) ,  
[Control.OnTabStopChanged\(EventArgs\)](#) , [Control.OnClick\(EventArgs\)](#) ,  
[Control.OnClientSizeChanged\(EventArgs\)](#) , [Control.OnControlAdded\(ControlEventArgs\)](#) ,  
[Control.OnControlRemoved\(ControlEventArgs\)](#) , [Control.OnLocationChanged\(EventArgs\)](#) ,  
[Control.OnDoubleClick\(EventArgs\)](#) , [Control.OnDragEnter\(DragEventArgs\)](#) ,  
[Control.OnDragOver\(DragEventArgs\)](#) , [Control.OnDragLeave\(EventArgs\)](#) ,  
[Control.OnDragDrop\(DragEventArgs\)](#) , [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,  
[Control.InvokeGotFocus\(Control, EventArgs\)](#) , [Control.OnHelpRequested\(HelpEventArgs\)](#) ,  
[Control.OnInvalidate\(InvalidEventArgs\)](#) , [Control.OnKeyDown\(KeyEventArgs\)](#) ,  
[Control.OnKeyPress\(KeyPressEventArgs\)](#) , [Control.OnKeyUp\(KeyEventArgs\)](#) ,  
[Control.OnLeave\(EventArgs\)](#) , [Control.InvokeLostFocus\(Control, EventArgs\)](#) ,  
[Control.OnLostFocus\(EventArgs\)](#) , [Control.OnMarginChanged\(EventArgs\)](#) ,  
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#) , [Control.OnMouseClick\(MouseEventArgs\)](#) ,  
[Control.OnMouseCaptureChanged\(EventArgs\)](#) , [Control.OnMouseDown\(MouseEventArgs\)](#) ,  
[Control.OnMouseEnter\(EventArgs\)](#) , [Control.OnMouseLeave\(EventArgs\)](#) ,  
[Control.OnDpiChangedBeforeParent\(EventArgs\)](#) , [Control.OnDpiChangedAfterParent\(EventArgs\)](#) ,  
[Control.OnMouseHover\(EventArgs\)](#) , [Control.OnMouseMove\(MouseEventArgs\)](#) ,  
[Control.OnMouseUp\(MouseEventArgs\)](#) ,  
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,  
[Control.OnRegionChanged\(EventArgs\)](#) , [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#) ,  
[Control.OnSizeChanged\(EventArgs\)](#) , [Control.OnChangeUICues\(UICuesEventArgs\)](#) ,  
[Control.OnSystemColorsChanged\(EventArgs\)](#) , [Control.OnValidating\(CancelEventArgs\)](#) ,  
[Control.OnValidated\(EventArgs\)](#) , [Control.PerformLayout\(\)](#) , [Control.PerformLayout\(Control, string\)](#) ,  
[Control.PointToClient\(Point\)](#) , [Control.PointToScreen\(Point\)](#) ,  
[Control.PreProcessMessage\(ref Message\)](#) , [Control.PreProcessControlMessage\(ref Message\)](#) ,  
[Control.ProcessKeyEventArgs\(ref Message\)](#) , [Control.ProcessKeyMessage\(ref Message\)](#) ,  
[Control.RaiseDragEvent\(object, DragEventArgs\)](#) , [Control.RaisePaintEvent\(object, PaintEventArgs\)](#) ,  
[Control.RecreateHandle\(\)](#) , [Control.RectangleToClient\(Rectangle\)](#) ,  
[Control.RectangleToScreen\(Rectangle\)](#) , [Control.ReflectMessage\(nint, ref Message\)](#) ,  
[Control.Refresh\(\)](#) , [Control.ResetMouseEventArgs\(\)](#) , [Control.ResetText\(\)](#) , [Control.ResumeLayout\(\)](#) ,  
[Control.ResumeLayout\(bool\)](#) , [Control.Scale\(SizeF\)](#) , [Control.Select\(\)](#) ,  
[Control.SelectNextControl\(Control, bool, bool, bool, bool\)](#) , [Control.SendToBack\(\)](#) ,  
[Control.SetBounds\(int, int, int, int\)](#) , [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#) ,  
[Control.SizeFromClientSize\(Size\)](#) , [Control.SetStyle\(ControlStyles, bool\)](#) , [Control.SetTopLevel\(bool\)](#) ,

[Control.RtlTranslateAlignment\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(LeftRightAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(ContentAlignment\)](#) ,  
[Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#) , [Control.RtlTranslateContent\(ContentAlignment\)](#) ,  
[Control.Show\(\)](#) , [Control.SuspendLayout\(\)](#) , [Control.Update\(\)](#) , [Control.UpdateBounds\(\)](#) ,  
[Control.UpdateBounds\(int, int, int, int\)](#) , [Control.UpdateBounds\(int, int, int, int, int, int\)](#) ,  
[Control.UpdateZOrder\(\)](#) , [Control.UpdateStyles\(\)](#) , [Control.OnImeModeChanged\(EventArgs\)](#) ,  
[Control.AccessibilityObject](#) , [Control.AccessibleDefaultActionDescription](#) ,  
[Control.AccessibleDescription](#) , [Control.AccessibleName](#) , [Control.AccessibleRole](#) ,  
[Control.AllowDrop](#) , [Control.Anchor](#) , [Control.AutoScrollOffset](#) , [Control.LayoutEngine](#) ,  
[Control.DataContext](#) , [Control.BackgroundImage](#) , [Control.BackgroundImageLayout](#) ,  
[Control.Bottom](#) , [Control.Bounds](#) , [Control.CanFocus](#) , [Control.CanRaiseEvents](#) ,  
[Control.CanSelect](#) , [Control.Capture](#) , [Control.CausesValidation](#) ,  
[Control.CheckForIllegalCrossThreadCalls](#) , [Control.ClientRectangle](#) , [Control.CompanyName](#) ,  
[Control.ContainsFocus](#) , [Control.ContextMenuStrip](#) , [Control.Controls](#) , [Control.Created](#) ,  
[Control.Cursor](#) , [Control.DataBindings](#) , [Control.DefaultBackColor](#) , [Control.DefaultCursor](#) ,  
[Control.DefaultFont](#) , [Control.DefaultForeColor](#) , [Control.DefaultMargin](#) ,  
[Control.DefaultMaximumSize](#) , [Control.DefaultMinimumSize](#) , [Control.DefaultPadding](#) ,  
[Control.DeviceDpi](#) , [Control.IsDisposed](#) , [Control.Disposing](#) , [Control.Dock](#) ,  
[Control.DoubleBuffered](#) , [Control.Enabled](#) , [Control.Focused](#) , [Control.Font](#) ,  
[Control.FontHeight](#) , [Control.ForeColor](#) , [Control.Handle](#) , [Control.HasChildren](#) , [Control.Height](#) ,  
[Control.IsHandleCreated](#) , [Control.InvokeRequired](#) , [Control.Accessible](#) ,  
[Control.IsAncestorSiteInDesignMode](#) , [Control.IsMirrored](#) , [Control.Left](#) , [Control.Margin](#) ,  
[Control.ModifierKeys](#) , [Control.MouseButtons](#) , [Control.mousePosition](#) , [Control.Name](#) ,  
[Control.Parent](#) , [Control.ProductName](#) , [Control.ProductVersion](#) , [Control.RecreatingHandle](#) ,  
[Control.Region](#) , [Control.RenderRightToLeft](#) , [Control.ResizeRedraw](#) , [Control.Right](#) ,  
[Control.RightToLeft](#) , [Control.ScaleChildren](#) , [Control.Site](#) , [Control.TabIndex](#) , [Control.TabStop](#) ,  
[Control.Tag](#) , [Control.Top](#) , [Control.TopLevelControl](#) , [Control.ShowKeyboardCues](#) ,  
[Control.ShowFocusCues](#) , [Control.UseWaitCursor](#) , [Control.Visible](#) , [Control.Width](#) ,  
[Control.PreferredSize](#) , [Control.Padding](#) , [Control.ImeMode](#) , [Control.ImeModeBase](#) ,  
[Control.PropagatingImeMode](#) , [Control.BackColorChanged](#) , [Control.BackgroundImageChanged](#) ,  
[Control.BackgroundImageLayoutChanged](#) , [Control.BindingContextChanged](#) ,  
[Control.CausesValidationChanged](#) , [Control.ClientSizeChanged](#) ,  
[Control.ContextMenuStripChanged](#) , [Control.CursorChanged](#) , [Control.DockChanged](#) ,  
[Control.EnabledChanged](#) , [Control.FontChanged](#) , [Control.ForeColorChanged](#) ,  
[Control.LocationChanged](#) , [Control.MarginChanged](#) , [Control.RegionChanged](#) ,  
[Control.RightToLeftChanged](#) , [Control.SizeChanged](#) , [Control.TabIndexChanged](#) ,  
[Control.TabStopChanged](#) , [Control.TextChanged](#) , [Control.VisibleChanged](#) , [Control.Click](#) ,  
[Control.ControlAdded](#) , [Control.ControlRemoved](#) , [Control.DataContextChanged](#) ,

[Control.DragDrop](#) , [Control.DragEnter](#) , [Control.DragOver](#) , [Control.DragLeave](#) ,  
[Control.GiveFeedback](#) , [Control.HandleCreated](#) , [Control.HandleDestroyed](#) ,  
[Control.HelpRequested](#) , [Control.Invalidated](#) , [Control.PaddingChanged](#) , [Control.Paint](#) ,  
[Control.QueryContinueDrag](#) , [Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) ,  
[Control.Enter](#) , [Control.GotFocus](#) , [Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) ,  
[Control.Layout](#) , [Control.Leave](#) , [Control.LostFocus](#) , [Control.MouseClick](#) ,  
[Control.MouseDoubleClick](#) , [Control.MouseCaptureChanged](#) , [Control.MouseDown](#) ,  
[Control.MouseEnter](#) , [Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) ,  
[Control.DpiChangedAfterParent](#) , [Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) ,  
[Control.MouseWheel](#) , [Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) ,  
[Control.ChangeUICues](#) , [Control.StyleChanged](#) , [Control.SystemColorsChanged](#) ,  
[Control.Validating](#) , [Control.Validated](#) , [Control.ParentChanged](#) , [Control.ImeModeChanged](#) ,  
[Component.Dispose\(\)](#) , [Component.GetService\(Type\)](#) , [Component.Container](#) ,  
[Component.DesignMode](#) , [Component.Events](#) , [Component.Disposed](#) ,  
[MarshalByRefObject.GetLifetimeService\(\)](#) , [MarshalByRefObject.InitializeLifetimeService\(\)](#) ,  
[MarshalByRefObject.MemberwiseClone\(bool\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#)

## Constructors

### Form1()

Initializes the form and the canvas. Checks if BOOSE library is working properly through the AboutBOOSE class.

```
public Form1()
```

## Methods

### Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

### Parameters

## **disposing** bool ↗

true if managed resources should be disposed; otherwise, false.