

# Namespace Session\_1

## Classes

### [AsCanvas](#)

A class that implements the abstract commands of ICanvas.

### [AsCommandFactory](#)

A Class of BOOSE Commands

### [Clear](#)

Class to implement the Clear() command from the AsCommandFactory.

### [Form1](#)

Form1

### [Rect](#)

Class to implement the Rect() command from the AsCommandFactory.

### [Reset](#)

A class that implements the Reset() command from the AsCommandFactory.

### [Tri](#)

A class that implements the Tri() command from the AsCommandFactory.

### [WriteText](#)

A class that implements the WriteText() command from the AsCommandFactory.

# Class AsCanvas

Namespace: [Session 1](#)

Assembly: Session 1.dll

A class that implements the abstract commands of ICanvas.

```
public class AsCanvas : ICanvas
```

## Inheritance

[object](#) ← AsCanvas

## Implements

ICanvas

## Inherited Members

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

# Constructors

## AsCanvas()

Constructor that calls Set().

```
public AsCanvas()
```

# Properties

## PenColour

Gets or sets the color of the pen used for drawing.

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

## Property Value

[object](#)

## Xpos

Gets or sets the X position on the canvas.

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

Property Value

[int](#)

## Ypos

Gets or sets the Y position on the canvas.

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

Property Value

[int](#)

## Methods

### Circle(int, bool)

Draws a circle on the canvas/bitmap.

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

Parameters

**radius** [int](#)

The radius of the circle.

**filled** [bool](#)

Indicates whether the circle is filled or not.

## Exceptions

### CanvasException

Thrown if the radius is invalid.

## Clear()

Clears the canvas/bitmap

```
public void Clear()
```

## DrawTo(int, int)

Draws a line from the current position to a specified (x, y) position.

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

## Parameters

x [int](#)

The x-coordinate.

y [int](#)

The y-coordinate.

## Exceptions

### CanvasException

Thrown if the position is invalid.

## MoveTo(int, int)

Moves the current position on the canvas to the specified (x, y) coordinates.

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

## Parameters

**x** [int](#)

The x-coordinate.

**y** [int](#)

The y-coordinate.

## Exceptions

### CanvasException

Thrown if the position is invalid.

## Rect(int, int, bool)

Draws a rectangle on the canvas/bitmap.

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

## Parameters

**width** [int](#)

The width of the rectangle.

**height** [int](#)

The height of the rectangle.

**filled** [bool](#)

Indicates whether the rectangle is filled or not.

## Exceptions

## CanvasException

Thrown if the dimensions are invalid.

## Reset()

Resets the position on the canvas/bitmap

```
public void Reset()
```

## Set(int, int)

Initializes the canvas with specified dimensions.

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

## Parameters

width [int](#)

The width of the PictureBox.

height [int](#)

The height of the PictureBox.

## SetColour(int, int, int)

Sets the color of the pen using RGB values.

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

## Parameters

red [int](#)

The red component (0-255).

green [int](#)

The green component (0-255).

blue [int](#)

The blue component (0-255).

## Exceptions

### CanvasException

Thrown if RGB values are invalid.

## Tri(int, int)

Draws a triangle on the canvas/bitmap.

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

## Parameters

width [int](#)

The width of the triangle.

height [int](#)

The height of the triangle.

## Exceptions

### CanvasException

Thrown if the values are not valid.

## WriteText(string)

Writes text onto the canvas/bitmap.

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

## Parameters

text [string](#) 

The text to write.

## getBitmap()

Returns the bitmap containing the graphics.

```
public object getBitmap()
```

## Returns

[object](#) 

The Bitmap



# Class AsCommandFactory

Namespace: [Session 1](#)

Assembly: Session 1.dll

A Class of BOOSE Commands

```
public class AsCommandFactory : CommandFactory, ICommandFactory
```








## Inheritance

[object](#)  ← CommandFactory ← AsCommandFactory

## 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)

This is a class off all the commands

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

## Parameters

commandType [string](#) 

## Returns

ICommand

## Exceptions

FactoryException

# Class Clear


Namespace: [Session 1](#)

Assembly: Session 1.dll

Class to implement the Clear() command from the AsCommandFactory.

```
public class Clear : CommandOneParameter, ICommand
```










## Inheritance

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

## Implements

ICommand

## Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , [CommandOneParameter.CheckParameters\(string\[\]\)](#)  , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , 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

## Clear()

Default constructor that is called from AsCommandFactory.

```
public Clear()
```

## Clear(Canvas)

Constructor that initializes the Clear command with a specific canvas.

```
public Clear(Canvas c)
```

## Parameters

**c** Canvas

The canvas on which the clear operation will be performed.

## Methods

### Execute()

Executes the Clear command by calling the Clear() method of the AsCanvas class.

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

# Class Form1

Namespace: [Session 1](#)

Assembly: Session 1.dll

Form1

```
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.ScaleMinMaxSize\(float, float, bool\)](#),  
[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.PerformAutoScale\(\)](#), [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.OnInvalidated\(InvalidateEventArgs\)](#), [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.IsAccessible](#),  
[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()

```
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.

# Class Rect


Namespace: [Session 1](#)

Assembly: Session 1.dll

Class to implement the Rect() command from the AsCommandFactory.

```
public class Rect : CommandTwoParameters, ICommand
```









## Inheritance

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

## Implements

ICommand

## Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,  
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
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

### Rect()

Default constructor.

```
public Rect()
```

### Rect(Canvas, int, int)

Constructor that initializes the Rect command with a specific canvas and dimensions.

```
public Rect(Canvas c, int width, int height)
```

## Parameters

**c** Canvas

The canvas on which the rectangle will be drawn.

**width** [int](#)

The width of the rectangle.

**height** [int](#)

The height of the rectangle.

## Methods

### CheckParameters(string[])

Checks the number of parameters provided for the Rect command.

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

## Parameters

**parameterList** [string](#)[]

The list of parameters to check.

## Exceptions

### CommandException

Thrown if the number of parameters is invalid.

## Execute()

Executes the Rect command by drawing a rectangle on the canvas.

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

## Exceptions

### RestrictionException

Thrown if the rectangle size exceeds allowed limits.

# Class Reset


Namespace: [Session 1](#)

Assembly: Session 1.dll

A class that implements the Reset() command from the AsCommandFactory.

```
public class Reset : CommandOneParameter, ICommand
```










## Inheritance

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

## Implements

ICommand

## Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , [CommandOneParameter.CheckParameters\(string\[\]\)](#)  , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , 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

## Reset()

Default constructor.

```
public Reset()
```

## Reset(Canvas)

Constructor that initializes the Reset command with a specific canvas.

```
public Reset(Canvas c)
```

## Parameters

**c** Canvas

The canvas to be reset.

## Methods

### Execute()

Executes the Reset command by calling the Reset method on the canvas.

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

# Class Tri


Namespace: [Session 1](#)

Assembly: Session 1.dll

A class that implements the Tri() command from the AsCommandFactory.

```
public class Tri : CommandTwoParameters, ICommand
```









## Inheritance

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

## Implements

ICommand

## Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,  
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
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

### Tri()

Default constructor.

```
public Tri()
```

### Tri(Canvas, int, int)

Constructor that initializes the Tri command with a specific canvas and dimensions.



```
public Tri(Canvas c, int width, int height)
```

## Parameters

**c** Canvas

The canvas on which the triangle will be drawn.

**width** [int](#)

The width of the triangle.

**height** [int](#)

The height of the triangle.

## Methods

### CheckParameters(string[])

Checks the number of parameters provided for the Tri command.

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

## Parameters

**parameterList** [string](#)[]

The list of parameters to check.

## Exceptions

### CommandException

Thrown if the number of parameters is invalid.

## Execute()

Executes the Tri command by calling the Tri method on the canvas.

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

## Exceptions

### RestrictionException

Thrown if the dimensions exceed allowed limits.

# Class WriteText


Namespace: [Session 1](#)

Assembly: Session 1.dll

A class that implements the WriteText() command from the AsCommandFactory.

```
public class WriteText : CommandOneParameter, ICommand
```









## Inheritance

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

## Implements

ICommand

## Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
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

### WriteText()

Default constructor.

```
public WriteText()
```

### WriteText(Canvas, string)

Constructor that initializes the WriteText command with a specific canvas and text.

```
public WriteText(Canvas c, string text)
```

## Parameters

**c** Canvas

The canvas where text will be written.

**text** [string](#)

The text to be written.

## Methods

### CheckParameters(string[])

Checks the number of parameters provided for the WriteText command.

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

## Parameters

**parameterList** [string](#)[]

The list of parameters to check.

## Exceptions

### CommandException

Thrown if the number of parameters is invalid.

### Execute()

Executes the WriteText command by calling the WriteText method on the canvas.

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

# Exceptions

## RestrictionException

Thrown if the text is empty.