

Namespace HarpsBOOSEapp

Classes

[HarpsAppCanvas](#)

[HarpsCircle](#)

Draws a circle on the canvas with the specified radius

[HarpsClear](#)

Command to clear the canvas

[HarpsCommandFactory](#)

Factory class to create Harps command objects based on command type

[HarpsDrawTo](#)

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

[HarpsMoveTo](#)

Moves the current position to the specified (x, y) coordinates without drawing

[HarpsPen](#)

Sets the pen color for drawing on the canvas

[HarpsRect](#)

Draws a rectangle on the canvas with the specified width and height

[mainForm](#)

Class HarpsAppCanvas

Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

```
public class HarpsAppCanvas : ICanvas
```

Inheritance

[object](#) ← HarpsAppCanvas

Implements

ICanvas

Inherited Members

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

Constructors

HarpsAppCanvas(int, int)

```
public HarpsAppCanvas(int xsize, int ysize)
```

Parameters

xsize [int](#)

ysize [int](#)

Properties

PenColour

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

Property Value

[object](#)

Xpos

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

Property Value

[int](#)

Ypos

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

Property Value

[int](#)

Methods

Circle(int, bool)

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

Parameters

radius [int](#)

filled [bool](#)

Clear()

```
public void Clear()
```

DrawTo(int, int)

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

Parameters

x [int](#)

y [int](#)

MoveTo(int, int)

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

Parameters

x [int](#)

y [int](#)

Rect(int, int, bool)

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

Parameters

width [int](#)

height [int](#)

filled [bool](#)

Reset()

```
public void Reset()
```

Set(int, int)

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

Parameters

width [int](#)

height [int](#)

SetColour(int, int, int)

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

Parameters

red [int](#)

green [int](#)

blue [int](#)

Tri(int, int)

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

Parameters

width [int](#)

height [int](#)

WriteText(string)

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

Parameters

text [string](#)

getBitmap()

```
public object getBitmap()
```

Returns

[object](#)

setPenWidth(float)

```
public void setPenWidth(float width)
```

Parameters

width [float](#)

Class HarpsCircle

Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

Draws a circle on the canvas with the specified radius

```
public class HarpsCircle : CommandOneParameter, ICommand
```









Inheritance

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

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

HarpsCircle()

initial new instance of the [HarpsCircle](#) class

```
public HarpsCircle()
```

HarpsCircle(Canvas, int)

initial new instance of the [HarpsCircle](#) class with specified canvas and radius

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

Parameters

c Canvas

radius [int](#)

Fields

radius

```
public int radius
```

Field Value

[int](#)

Methods

CheckParameters(string[])

Check the parameters for the Circle command

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

Parameters

parameterList [string](#)[]

The list of parameters to check

Exceptions

CommandException

Thrown when the parameters are invalid

Execute()

Execute the Circle command

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

Exceptions

[ArgumentOutOfRangeException](#) ↗

Thrown when the radius is less than 1

Class HarpsClear


Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

Command to clear the canvas

```
public class HarpsClear : CommandOneParameter, ICommand
```









Inheritance

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

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\(object, object\)](#) 

Remarks

the [HarpsClear](#) class implements the clear command for the Harps drawing application.

Constructors

HarpsClear()

initial new instance of the [HarpsClear](#) class

```
public HarpsClear()
```

HarpsClear(Canvas)

initial new instance of the [HarpsClear](#) class with canvas parameter

```
public HarpsClear(Canvas c)
```

Parameters

c Canvas

The canvas to clear

Methods

CheckParameters(string[])

Check that no parameters are passed to the clear command

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

Parameters

parameterList [string](#)[]

The list of parameters to check

Exceptions

CommandException

Thrown when the parameters are invalid

Execute()

execute the Clear command

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

Class HarpsCommandFactory


Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

Factory class to create Harps command objects based on command type

```
public class HarpsCommandFactory : CommandFactory, ICommandFactory
```








Inheritance

[object](#)  ← CommandFactory ← HarpsCommandFactory

Implements

ICommandFactory

Inherited Members

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

Remarks

the [HarpsCommandFactory](#) class extends the CommandFactory to create Harps specific commands for the drawing application such as DrawTo, MoveTo, Pen, Circle, Rect, and Clear.

Methods

MakeCommand(string)

Creates and returns an ICommand object based on the provided command type string.

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

Parameters

commandType [string](#) 

The name of the command to create

Returns

ICommand

An instance of the command called

Exceptions

[ArgumentNullException](#) 

Thrown if `commandType` is null or empty

Class HarpsDrawTo


Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

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

```
public class HarpsDrawTo : CommandTwoParameters, ICommand
```









Inheritance

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

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,
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

HarpsDrawTo()

initial new instance of the [HarpsDrawTo](#) class

```
public HarpsDrawTo()
```

HarpsDrawTo(Canvas, int, int)

initial new instance of the [HarpsDrawTo](#) class with specified canvas and target coordinates

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

Parameters

c Canvas

x [int](#)

y [int](#)

Fields

x

```
public int x
```

Field Value

[int](#)

y

```
public int y
```

Field Value

[int](#)

Methods

CheckParameters(string[])

Check the parameters for the DrawTo command

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

Parameters

```
parameterList string[]
```

The list of parameters to check

Exceptions

CommandException

Thrown when the parameters are invalid

Execute()

Execute the DrawTo command, drawing a line to the specified (x, y) coordinates

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


Class HarpsMoveTo


Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

Moves the current position to the specified (x, y) coordinates without drawing

```
public class HarpsMoveTo : CommandTwoParameters, ICommand
```









Inheritance

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

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,
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

HarpsMoveTo()

initial new instance of the [HarpsMoveTo](#) class

```
public HarpsMoveTo()
```

HarpsMoveTo(Canvas, int, int)

Initializes a new instance of the [HarpsMoveTo](#) class with the specified canvas and target coordinates.

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

Parameters

c Canvas

x [int](#)

y [int](#)

Fields

X

```
public int x
```

Field Value

[int](#)

y

```
public int y
```

Field Value

[int](#)

Methods

CheckParameters(string[])

Check the parameters for the MoveTo command

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

Parameters

```
parameterList string[]
```

The list of parameters to check

Exceptions

CommandException

Thrown when the parameters are invalid

Execute()

Execute the MoveTo command, moving to the specified (x, y) coordinates without drawing

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

Class HarpsPen


Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

Sets the pen color for drawing on the canvas

```
public class HarpsPen : CommandThreeParameters, ICommand
```









Inheritance

[object](#)  ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← CommandThreeParameters ← HarpsPen

Implements

ICommand

Inherited Members

CommandThreeParameters.param3 , CommandThreeParameters.param3unprocessed ,
CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,
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

HarpsPen()

initial new instance of the [HarpsPen](#) class

```
public HarpsPen()
```

HarpsPen(Canvas, int, int, int)

initial new instance of the [HarpsPen](#) class with specified canvas and color components

```
public HarpsPen(Canvas c, int red, int green, int blue)
```

Parameters

c Canvas

red [int](#)

green [int](#)

blue [int](#)

Methods

CheckParameters(string[])

Check if the parameters for the Pen command are valid

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

Parameters

parameterList [string](#)[]

The list of parameters to check

Exceptions

CommandException

Thrown when the parameters are invalid

Execute()

execute the Pen command

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

Class HarpsRect

Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

Draws a rectangle on the canvas with the specified width and height

```
public class HarpsRect : CommandOneParameter, ICommand
```









Inheritance

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

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

HarpsRect()

initial new instance of the [HarpsRect](#) class

```
public HarpsRect()
```

HarpsRect(Canvas, int, int)

initial new instance of the [HarpsRect](#) class with specified canvas, width and height

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

Parameters

c Canvas

width [int](#)

height [int](#)

Fields

height

```
public int height
```

Field Value

[int](#)

width

```
public int width
```

Field Value

[int](#)

Methods

CheckParameters(string[])

Check the parameters for the Rect command

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


Parameters

`parameterList` [string](#)[]

The list of parameters to check

Exceptions

CommandException

Thrown when the parameters are invalid

Execute()

Execute the Rect command

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

Class mainForm

Namespace: [HarpsBOOSEapp](#)

Assembly: HarpsBOOSEapp.dll

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

Inheritance

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

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

mainForm()

```
public mainForm()
```

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.