Creating Custom Controls

Mark Heath
http://markheath.net
@mark_heath

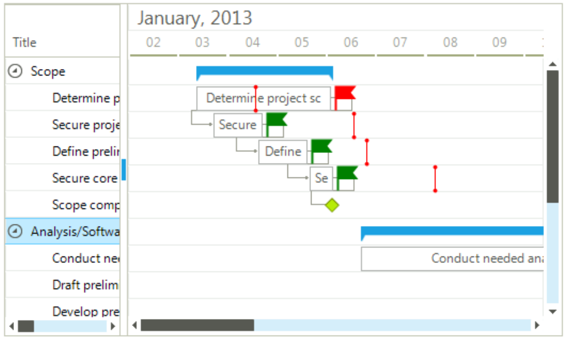
pluralsight
hardcore dev and IT training

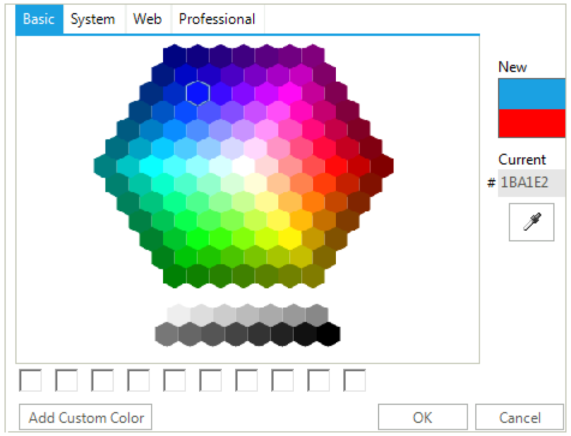
Windows Forms Control Options

Standard Controls

- Button
- CheckBox
- CheckedListBox
- ColorDialog
- ComboBox
- ContextMenuStrip
- DataGridView
- DataSet
- DateTimePicker
- DirectoryEntry
- DirectorySearcher
- DomainUpDown
- ErrorProvider
- EventLog
- FileSystemWatcher
- FlowLayoutPanel
- FolderBrowserDialog
- FontDialog
- GroupBox
- HelpProvider
- HScrollBar
- ImageList

Third Party Controls





Telerik®

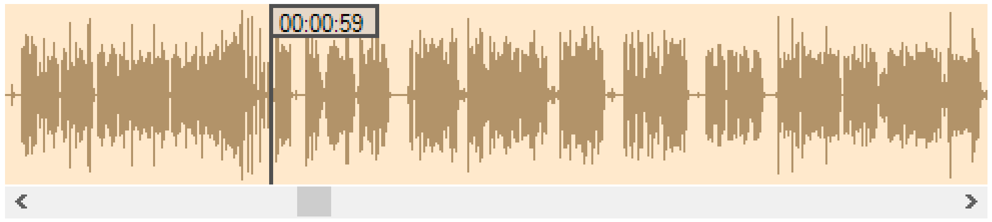
Custom Controls

Appearance

Mouse Handling

Keyboard Handling

Overview



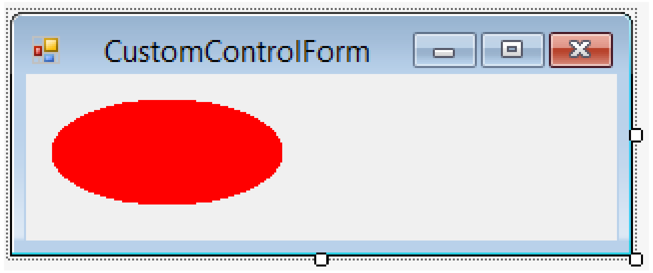
- Choosing a starting point for your control
- Rendering your control using GDI+ (Graphics)
- Scrolling and Invalidation
- Handling Mouse Interactions
- Drop-Down Panels

Choosing a Base Class

- **Inheriting System.Windows.Forms.Control**
  - No built-in appearance or behaviour
  - Access to Paint, Mouse and Keyboard events

```
class CustomControl1 : Control
{
    protected override void OnPaint(PaintEventArgs e)
    {
        base.OnPaint(e);
        e.Graphics.FillEllipse(Brushes.Red, this.ClientRectangle);
    }
}
```

  - Can host child controls
  - No designer support for child controls

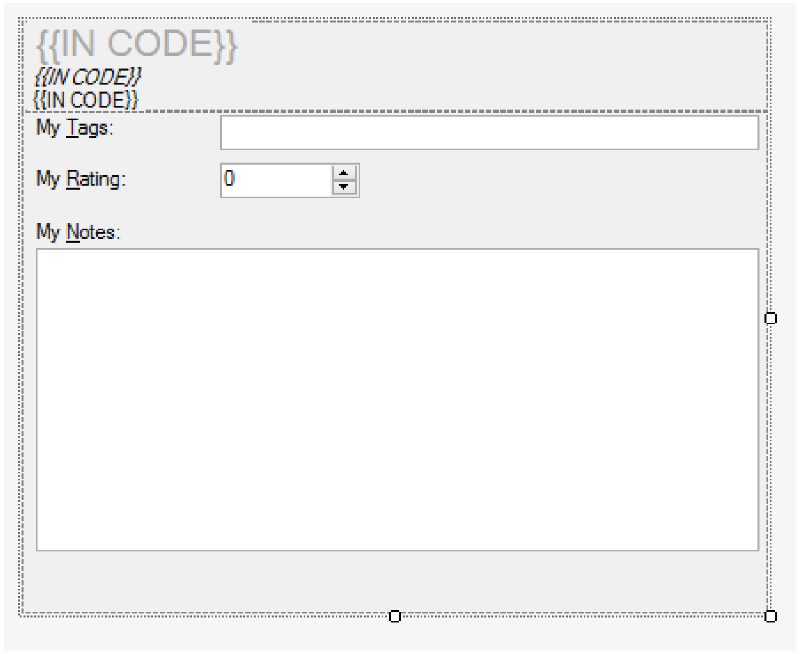


Extending Existing Controls

- **Can inherit from any existing control**
  - Override **behaviour**, reuse **appearance**
  - Can't always fully customize appearance
- **Some controls offer "Owner Drawn" mode**
  - Override **appearance** but reuse **behaviour**
  - e.g. ListView, ComboBox, ToolTip
- **Some controls designed to be base clases**
  - **ButtonBase** inherited by Button, CheckBox, RadioButton
  - **TextBoxBase** inherited by TextBox, MaskedTextBox, RichTextBox

Inheriting From UserControl

- **Intended for custom controls containing child controls**
- **Allows you to use designer for child controls**
- **Inherits from ScrollableControl**
  - Easily support auto-scrolling contents
- **Appears automatically in Toolbox**



GDI+ (Graphics Device Interface)

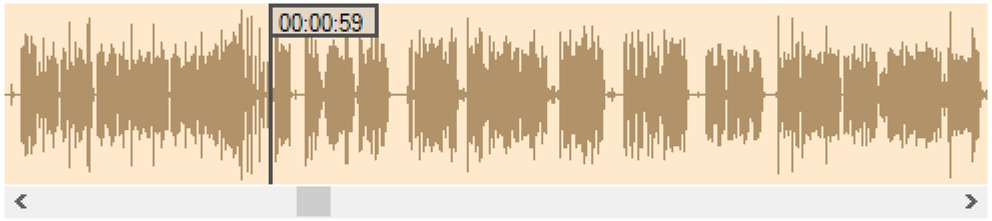
- **System.Drawing.Graphics**
- **Draw shapes and lines**
  - Including Bezier curves, polygons
- **Pens and brushes**
  - Including gradient and texture brushes
- **Draw text and images**
  - Using any fonts
- **Advanced techniques**
  - Anti-aliasing, transparency, transforms and clipping regions

Painting With GDI

- **Handle the Paint event or override OnPaint**
- **PaintEventArgs**
  - Graphics object
  - Clip Rectangle
- **Recommendations**
  - Perform all painting in Paint event handler
  - Use double buffering
  - Reuse Brush, Pen, Image and Font resources rather than create every time

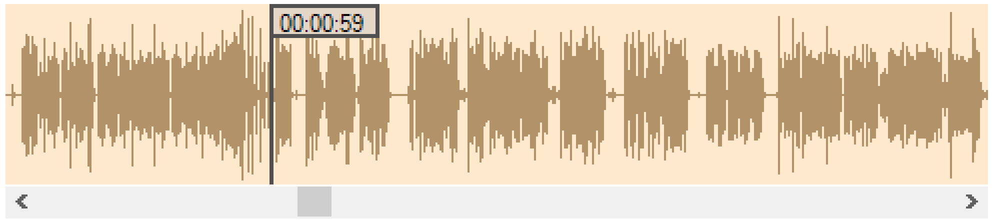
```
public CustomControl1()
{
    DoubleBuffered = true;
    myBrush = new SolidBrush(Color.FromArgb(22, 202, 101));
}
```

Module Summary



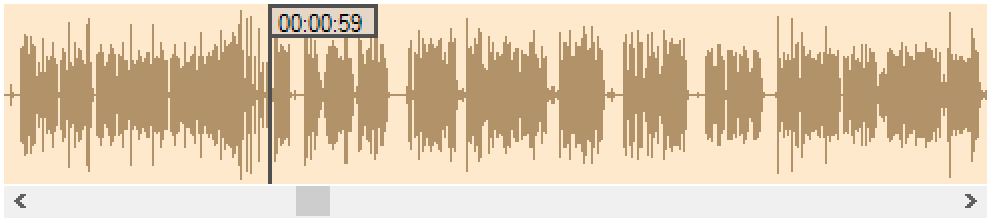
- **Choose the most appropriate starting point for your custom control**
  - Inherit from Control or UserControl
  - Inherit from TextBoxBase, ButtonBase or existing controls
  - Owner draw controls

Module Summary



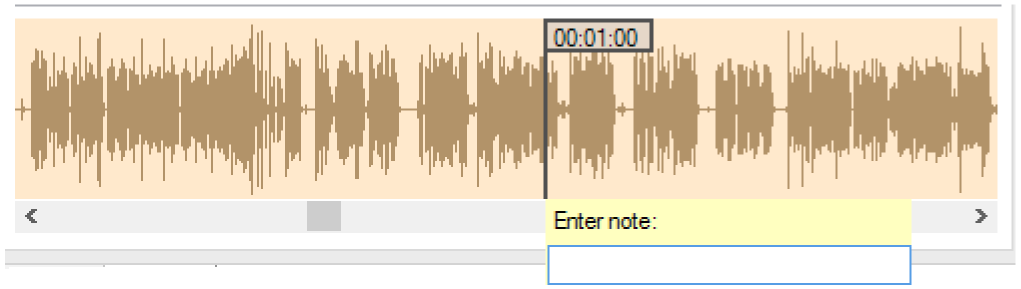
- **Use the power of GDI+ to control appearance**
  - Perform all painting in Paint event
  - Use double buffering
  - Draw only what you know is visible
  - Remember to Invalidate whenever properties are changed
  - Use MeasureString to calculate string size in pixels
- **Combine with existing controls**
  - Scroll bars

Module Summary



- **Handle mouse events to support user interaction**
  - MouseClick, MouseDown & MouseUp
  - Remember to check mouse button
  - MouseMove
  - Set cursor to give hints
  - Raise events to report user actions

Module Summary



- **Use Forms for drop-down panels**
  - FormBorderStyle.None
  - FormStartPosition.Manual
  - PointToScreen
  - Close on Deactivate