




# Silverlight 2.0 and WPF

TM

What's the Same  
What's Different



Joseph Cooney, Paradigm Logic  
[joseph@learnwpf.com](mailto:joseph@learnwpf.com)  
[josephcooney@paradigmlogic.com](mailto:josephcooney@paradigmlogic.com)

# What is Silverlight 2.0?

- Cross-platform browser plug-in
- Enhance/augment HTML/CSS
- Sandboxed execution environment (CLR), Base Class Libraries, small parts of WCF, LINQ, UI framework<sup>TM</sup>
- Targeting a 4MB download and ~5 second install time for plug-in



**A 4MB Download!**

See:

`System.Windows.Media.Colors`



# What is WPF?

- Next-Generation UI Platform for Windows (no size constraints on scope of API – everything + kitchen sink)
- Successor to GDI/GDI+ and User subsystems
- Rich Composition and Customization
- Hardware accelerated
- Deployed as part of .NET framework 3.0

# Core Features of Both Platforms

- Vector-based
- Broad integration of different content types (Text, Controls, Video + graphics)
- Customizable look for controls -> designers + developers living together in harmony
- Declarative programming model with XAML
- Imperative managed programming model
- Platform for delivering great experiences

# What we're Comparing (and why)

- The UI parts of Silverlight to WPF (not the BCL, WCF parts etc)
- **Not** trying to say one is better than the other
- Give people familiar with one a better understanding of the other
- Help people make informed choices about which one is right for them

# Something Technical

## Similarities

- Foo
- Bar
- Baz

TM

## Differences

- Spam
- Spam
- Spam, Spam
- Spammity Spam!

# Class Hierarchy

## ASP.NET

Button

WebControl

Control

Object

## Windows Forms

Button

ButtonBase

Control

Component

MarshalByRefObject

Object

## Silverlight 2.0 Beta 1

Button

ButtonBase

ContentControl

Control

FrameworkElement

UIElement

DependencyObject

Object

## WPF

Button

ButtonBase

ContentControl

Control

FrameworkElement

UIElement

Visual

DependencyObject

DispatcherObject

Object

# Control Set

ReconversionMenuItem ContentPresenter\* TabItem CalendarButtonBase  
Selector ScrollViewer Calendar  
Month Thumb ReaderTwoPageViewer  
WatermarkedTextBox ListViewItem ResizeGrip  
ComboBox GroupItem HeaderedItemsControl DocumentViewer  
DataGridRowHeader MenuBase PasswordBox Separator GroupBox  
Control ScrollBar TreeView ContentControl ProgressBar  
ListView ItemsControl ReaderPageViewer UserControl DayButton  
GridViewColumnHeader DataGrid ListBox TreeViewItem HeaderedContentControl  
Frame RootBrowserWindow RichTextBox FlowDocumentScrollViewer Slider  
ComboBoxItem ReadOnlyCheckBox Window ScrollContentPresenter ContextMenu  
ReaderScrollViewer Menu DatePicker Expander RepeatButton ButtonBase  
PreviewControl FlowDocumentReader ListBoxItem StatusBarItem DataGridRow  
TextBoxBase ToolTip NavigationWindow EmptyGroupItem StatusBarItem  
RadioButton EditorContextMenu ViewerContextMenu DocumentViewerBase  
HyperlinkButton GridSplitter MonthButton Label DataGridColumnHeader  
MenuItem EditorMenuItem TabControl Toolbar TextBox StickyNoteControl  
FlowDocumentPageViewer ToggleButton RangeBase EditorMenuItem



# Control Set



# SL2 + 3D != WPF

1577 public types  
3592 classes

WPF

SL 2.0

376 public types  
335 classes

# API Size – Binding object in Reflector

```
public class Binding
{
    // Fields
    private IValueConverter _converter;
    private CultureInfo _converterCulture;
    private object _converterParameter;
    internal bool _isSealed;
    private BindingMode _mode;
    internal string _path;
    internal string[] _paths;
    private object _source;

    // Methods
    public Binding(string path);
    private void CheckSealed();
    internal BindingExpression CreateBindingExpression();
    internal static string[] ParsePaths(string path);

    // Properties
    public IValueConverter Converter { get; set; }
    public CultureInfo ConverterCulture { get; set; }
    public object ConverterParameter { get; set; }
    public BindingMode Mode { get; set; }
    public object Source { get; set; }
}
Expand Methods
```

```
public class Binding : BindingBase
{
    // Fields
    private object _asyncState;
    private int _attachedPropertyChangedPath;
    private bool _bindDirectlyToSource;
    private IValueConverter _converter;
    private object _converterParameter;
    private CultureInfo _culture;
    private bool _dontTransferDefaultValue;
    private string _elementSource;
    private UpdateSourceExceptionFilterCallback _exceptionFilterCallback;
    private bool _isAsync;
    private WeakReference<object> _objectSource;
    private string _propertyPath;
    private RelativeSource _relativeSource;
    private object[] _source;
    private SourceProperties _sourceReference;
    private ValidationRuleCollection _validationRules;
    private object _workerData;
    private string _xpath;
    public static readonly object DoNothing;
    public const string BindName = "Bind";
    public static readonly RoutedEventArgs SourceUpdatedEvent;
    public static readonly RoutedEventArgs TargetUpdatedEvent;
    private static readonly Object[] UserSource;
    public static readonly DependencyProperty XmlNamespaceManagerProperty;

    // Methods
    static Binding();
    public Binding();
    public Binding(string path);
    public static void AddSourceUpdatedHandler(DependencyObject element, EventHandler<DataTransferEventArgs> handler);
    public static void addTargetUpdatedHandler(DependencyObject element, EventHandler<DataTransferEventArgs> handler);
    internal override BindingExpressionBase CreateBindingExpressionOverride(DependencyObject target, DependencyProperty dp, BindingExpressionBase owner);
    private void DetermineSource();
    internal object DoNothingException(object bindArg, Exception exception);
    public static XmlNamespaceManager GetXmlNamespaceManager(DependencyObject target);
    private static bool IsValidXmlNamespaceManager(object value);
    private static bool IsValidXmlNamespaceManagerHelper(object value);
    internal override ValidationRuleCollection ValidationRules { type: type; }
    public static void RemoveSourceUpdatedHandler(DependencyObject element, EventHandler<DataTransferEventArgs> handler);
    public static void SetXmlNamespaceManagerOverride(DependencyObject target, XmlNamespaceManager value);
    [EditorBrowsable(EditorBrowsableState.Never)]
    public bool ShouldSerializePath();
    [EditorBrowsable(EditorBrowsableState.Never)]
    public bool ShouldSerializeSource();
    [EditorBrowsable(EditorBrowsableState.Never)]
    public bool ShouldSerializeValidationRules();
    internal void UsePath(string path);

    // Properties
    [DefaultValue(string, null)]
    public object AsyncState { get; set; }
    [DefaultValue(int)]
    public int AttachedPropertyChangedPath { get; set; }
    public bool BindDirectlyToSource { get; set; }
    [DefaultValue(string, null)]
    public IValueConverter Converter { get; set; }
    [TypeConverter(typeof(CultureInfoOfIdetLanguageTagConverter), DefaultValue(string, null))]
    public CultureInfo ConverterCulture { get; set; }
    internal override CultureInfo ConverterCultureInternal { get; }
    [DefaultValue(string, null)]
    public object ConverterParameter { get; set; }
    [DefaultValue(string, null)]
    public string ElementName { get; set; }
    [DefaultValue(bool)]
    public bool IsAsync { get; set; }
    [DefaultValue(int)]
    public BindingMode Mode { get; set; }
    [DefaultValue(bool)]
    public bool NotifyOnSourceUpdated { get; set; }
    [DefaultValue(bool)]
    public bool NotifyOnTargetUpdated { get; set; }
    [DefaultValue(bool)]
    public bool NotifyOnValidationError { get; set; }
    public string PropertyPath { get; set; }
    [DefaultValue(string, null)]
    public object RelativeSource { get; set; }
    public object Source { get; set; }
    internal ObjectReference SourceReference { get; set; }
    internal bool TransferDefaultValue { get; set; }
    internal bool TreeContextIsRequired { get; }
    [DesignerSerializationVisibility(DesignerSerializationVisibility.Hidden)]
    public UpdateSourceExceptionFilterCallback UpdateSourceExceptionFilter { get; set; }
    [DefaultValue(0)]
    public UpdateSourceTrigger UpdateSourceTrigger { get; set; }
    [DefaultValue(bool)]
    public bool ValidatesOnDataErrors { get; set; }
    [DefaultValue(bool)]
    public bool ValidatesOnExceptions { get; set; }
    public ICollection<ValidationRule> ValidationRules { get; }
    internal ICollection<ValidationRule> ValidationRulesInternal { get; }
    internal object WorkerData { get; set; }
    [DefaultValue(string, null)]
    public string XPath { get; set; }

    // Nested Types
    private enum SourceProperties : byte
    {
        ElementName = 2,
        InternalSource = 4,
        Mode = 8,
        RelativeSource = 1,
        Source = 3
    }
}
Expand Methods
```

# Graphics

## Similarities

- Basic shape classes – Ellipse, Line, Path, Polygon, Polyline, Rectangle
- Brushes – Gradient and Solid Color Brushes, Video and Image brush
- RenderTransform

## Differences

- SL has no DrawingBrush or VisualBrush
- SL has no adorning layer
- SL has no OnRender() method to over-ride!
- SL has not BitmapEffects (blur, glow, emboss)
- SL – no LayoutTransform

# Layout

## Similarities

- Panels – Canvas, Grid & StackPanel
- Margin Property
- Extensibility – write your own panels. TM
- Vertical and Horizontal Alignment on elements
- Width, Height, ActualWidth, ActualHeight, MaxWidth, MaxHeight, MinWidth, MinHeight

## Differences

- SL is **missing**: WrapPanel, DockPanel & UniformGrid
- SL: Partial implementation of Padding Property (on Control and descendants)
- SL Panels are slimmed down versions of their WPF counterparts: eg. Canvas.

\*\*\* STOP: 0x00000019 (0x00000000,0xC00E0FF0,0xFFFFEFD4,0xC0000000)  
BAD\_POOL\_HEADER

CPUID:GenuineIntel 5.2.c irq1:1f SYSVER 0xf0000565

Dll	Base	DateStmp	-	Name	Dll	Base	DateStmp	-	Name
80100000	3202c07e		-	ntoskrnl.exe	80010000	31ee6c52		-	hal.dll
80001000	31ed06b4		-	atapi.sys	80006000	31ec6c74		-	SCSIPTORT.SYS
802c6000	31ed06bf		-	aic78xx.sys	802cd000	31ed237c		-	Disk.sys
802d1000	31ec6c7a		-	CLASS2.SYS	8037c000	31eed0a7		-	Ntfs.sys
fc698000	31ec6c7d		-	Floppy.SYS	fc6a8000	31ec6ca1		-	Cdrom.SYS
fc90a000	31ec6df7		-	Fs_Rec.SYS	fc9c9000	31ec6c99		-	Null.SYS
fc864000	31ed868b		-	KSecDD.SYS	fc9ca000	31ec6c78		-	Beep.SYS
fc6d8000	31ec6c90		-	i8042prt.sys	fc86c000	31ec6c97		-	mouclass.sys
fc874000	31ec6c94		-	kbdclass.sys	fc6f0000	31f50722		-	VIDEOPTORT.SYS
feffa000	31ec6c62		-	mga_mil.sys	fc890000	31ec6c6d		-	vga.sys
fc708000	31ec6ccb		-	MsfS.SYS	fc4b0000	31ec6cc7		-	Npfs.SYS
fefbc000	31eed262		-	NDIS.SYS	a0000000	31f954f7		-	win32k.sys
feffa4000	31f91a51		-	mga.dll	fec31000	31eedd07		-	Fastfat.SYS
feb8c000	31ec6e6c		-	TDI.SYS	feaf0000	31ed0754		-	nbfs.sys
feacf000	31f130a7		-	tcpip.sys	feab3000	31f50a65		-	netbt.sys
fc550000	31601a30		-	e159x.sys	fc560000	31f8f864		-	afd.sys
fc718000	31ec6e7a		-	netbios.sys	fc858000	31ec6c9b		-	Parport.sys
fc870000	31ec6c9b		-	Parallel.SYS	fc954000	31ec6c9d		-	ParUdm.SYS
fc5b0000	31ec6cb1		-	Serial.SYS	fea4c000	31f5003b		-	rdr.sys
fea3b000	31f7a1ba		-	mup.sys	fe9da000	32031abe		-	srv.sys

Address	dword	dump	Build	[1381]	-	Name
fec32d84	80143e00	80143e00	80144000	ffdf0000	00070b02	- KSecDD.SYS
801471c8	80144000	80144000	ffdf0000	c03000b0	00000001	- ntoskrnl.exe
801471dc	80122000	f0003fe0	f030eee0	e133c4b4	e133cd40	- ntoskrnl.exe
80147304	803023f0	0000023c	00000034	00000000	00000000	- ntoskrnl.exe

Restart and set the recovery options in the system control panel  
or the /CRASHDEBUG system start option.

Everyone still awake? - GREAT!

# Data Binding

## Similarities

- Binding object
- Change notification thru `INotifyPropertyChanged` + `INotifyCollectionChanged`
- Data Context<sup>TM</sup>
- Converters
- Data Templates
- `ObservableCollection`

## Differences

- Binding object – pared down (only 5 properties)
- `{Binding}` markup extension
- Element Binding
- Data Triggers
- No `ObjectDataProvider`
- `INotifyCollection changed` only raises single item changes + no move event
- No `IDataErrorInfo`

# Styling + Templating

- SL - Missing Event Triggers and Property Triggers in Styles
- SL - No Data Triggers
- Can be worked around, but requires code which...  
TM

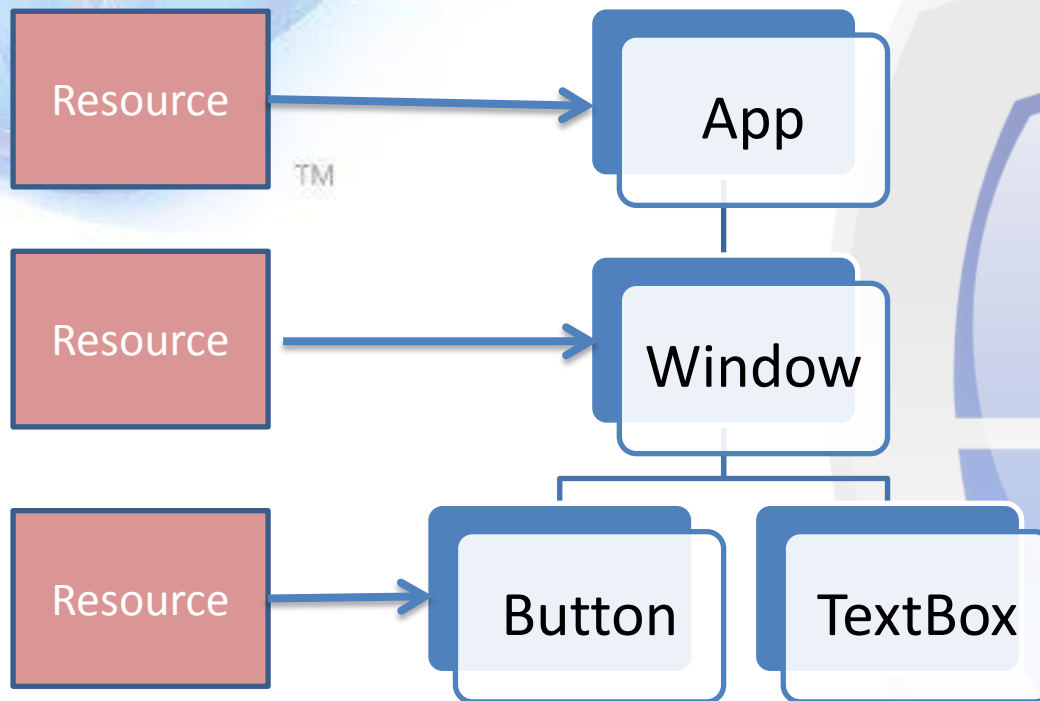


**Breaks the Designer – Developer  
Workflow!**

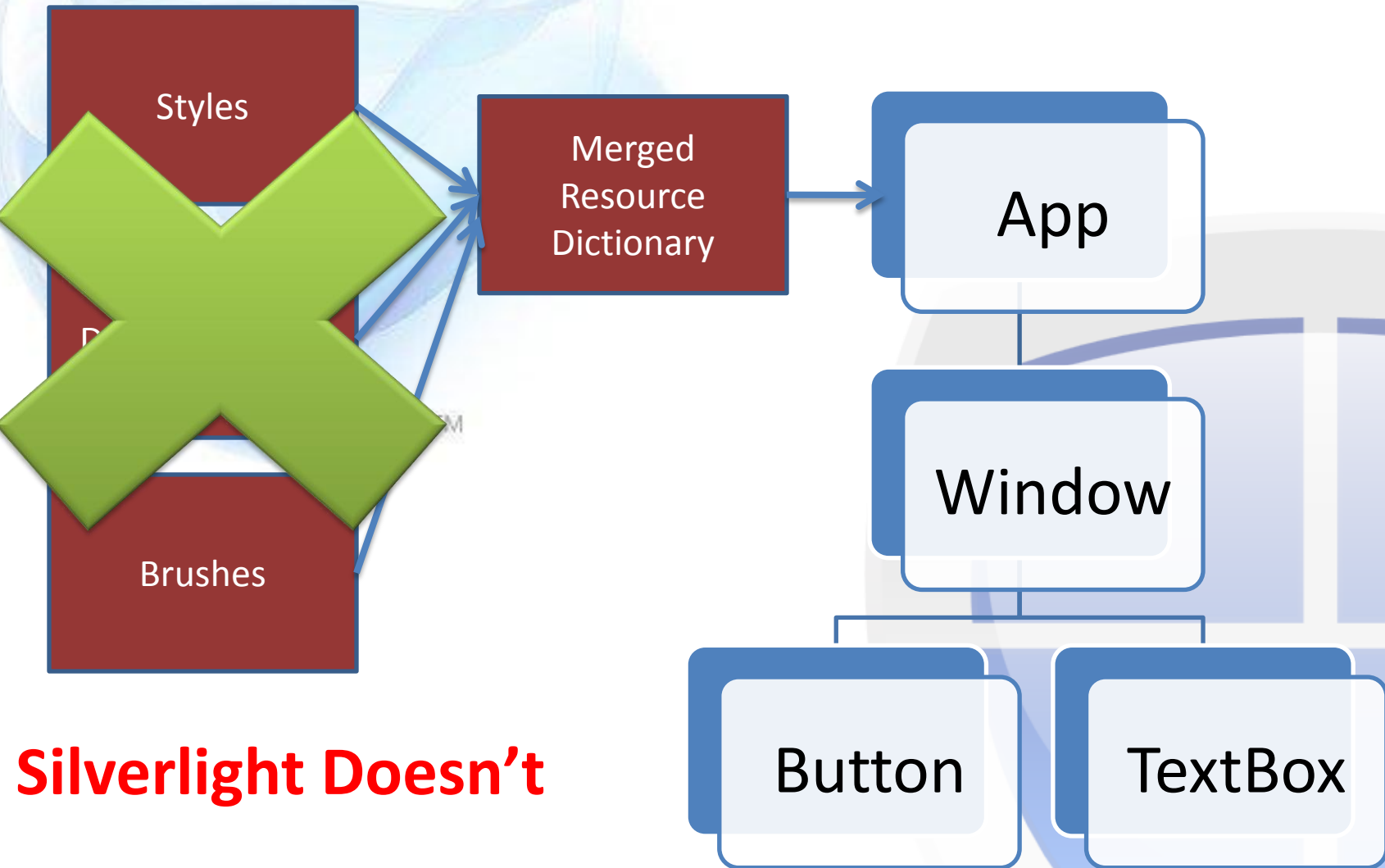
TM

# Managing Resources

- Resources -> styles, templates, brushes etc
- Key-value pairs



# WPF allows this



**Silverlight Doesn't**

# Threading

- Silverlight 2.0 is truly multi-threaded
- Like WPF, SL 2.0 Controls are “owned” by the creating thread.
- Breaking this rule will cause an exception, but SL is more forgiving than WPF.
- Dispatcher object – use it to jump back on to the UI thread (no exceptions).
- Blocking the UI thread – still bad.

# Security Sandbox

- SL sandbox analogous to WPF XBAP
- No private member reflection
- File I/O to isolated storage and via OpenFileDialog
- XBAP – no cross domain calls. SL – policy based cross domain. Full WPF – full trust, anything goes