XAML^3

Greg Lutz
Product Manager
gregoryl@componentone.com

Agenda

- State of XAML
  - Where it's been?
  - Where it's now?
  - Where it's going?
- Multi-targeting
  - Why develop cross-platform?
  - How XAML makes this possible
- 10 Tips
- XAML on Windows 8

What is “XAML”?

- “XAML” = WPF/Silverlight/WP7
- eXtensible Application Markup Language
- XML based
- Defines the user interface
- Conceptually similar to HTML
  - + Better tools (today)
  - + More controls
  - + Data binding
  - = Higher productivity for business apps

WPF

- 2006 – Windows Presentation Foundation
- Considered replacement for Windows Forms
- Clearer separation between UI and logic
- Flexible UI
- Key Features:
  - Templating
  - Data binding
  - Styling
  - Animation
  - 3D

Silverlight

- 2007 – WPF/Everywhere
- Lightweight version of WPF for the Web
- New opportunities for business apps
- Cross-Browser plug-in
- Key Features:
  - Templating
  - Data binding
  - Styling
  - Animation

Windows Phone (WP7)

- 2010
- Smart phone contender
- Replacement for Windows Mobile 6
- Supports subset version of Silverlight for business/general development
- XNA for game development

Today

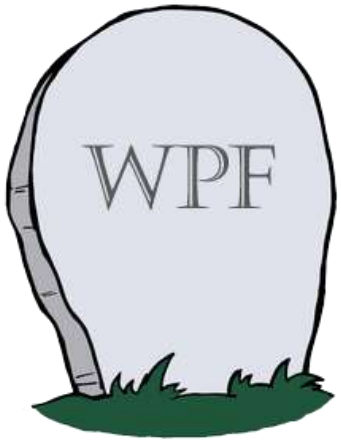
- Silverlight and WPF are still viable solutions for business applications today
- Mature and proven for enterprise development
- ComponentOne seeing continuous growth
  - Studio for Silverlight booming (#1 download)
  - Studio for WPF on the rise

Word on the street

- “Silverlight is now the development platform for the Windows Phone”
- “Microsoft is shifting focus to HTML”
- “Desktop computing is dead”
- “Silverlight is dead”

Today

- Trick-or-treat?



Tomorrow

- .NET 4.5/Silverlight 5 coming soon
- No end of support in sight
- Windows Phone 7+
- Microsoft LightSwitch
- New XAML platform for Windows 8
  - Metro-style WinRT
  - Consumer-based apps

Overview

- Problem: write code today (Win7) that can still be used tomorrow (Win8)
- Solution: leverage the cross-development power of “XAML” to be successful today and tomorrow

Multi-targeting

- Writing code that targets two or more different platforms with largely the same code-base
- Targeting multiple platforms (desktop, web, mobile) is becoming common
- Microsoft's solution today is XAML
  - WPF (desktop)
  - Silverlight (web)
  - Windows Phone (mobile)

ComponentOne Studios

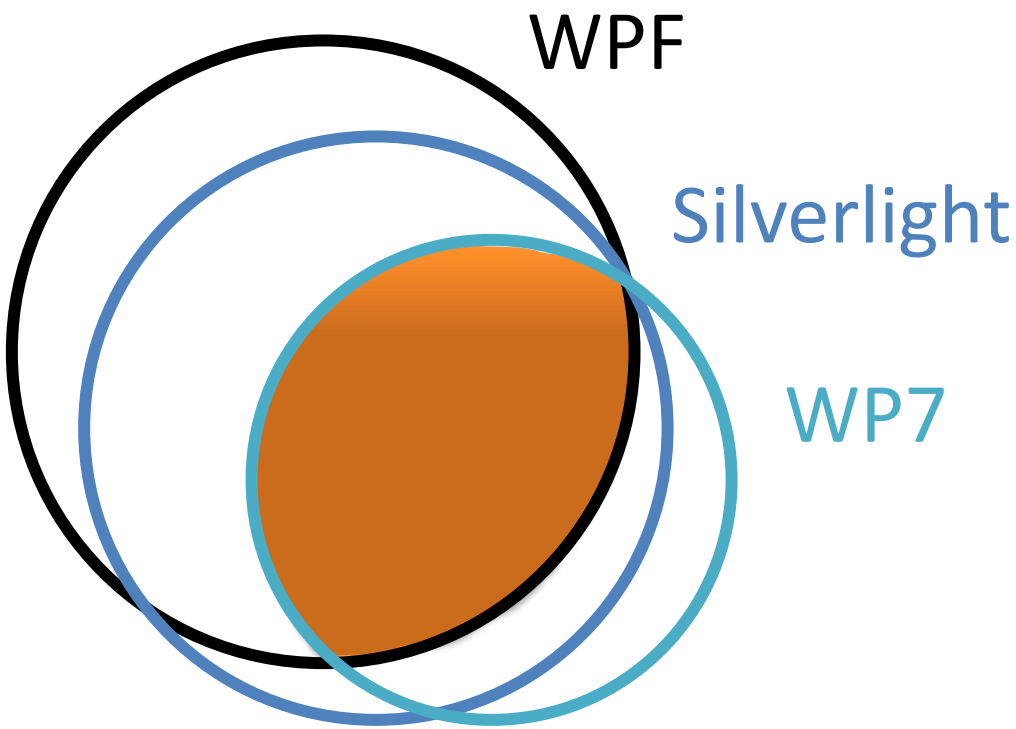
Control	WPF	Silverlight	Windows Phone
Accordion	X	X	--
Chart	X	X	X
DataGrid	X	X	X
Docking	X	X	--
Gauges	X	X	X
Maps	X	X	X
PdfViewer	X	X	X
RichTextBox	X	X	X
Scheduler	X	X	--

Multi-targeting

- Silverlight, WPF and WP7 are “similar”, but
  - Not binary compatible
  - API/XAML have minor differences
- Sharing code is hard
- Sharing UI is harder
- Simplicity and factoring is the key to success

Multi-targeting

- Silverlight, WPF and WP7 are have some overlap but each contain unique features



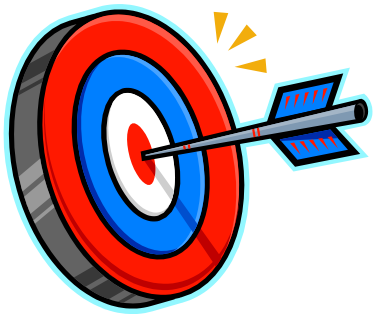
Multi-targeting

- Strategy: share what you can, specialize each platform where it makes sense
  - WP7 UI considerably different (form-factor, touch)
  - WPF can undock floating windows

- Next,
  - 10 tips for cross-platform development and code sharing for WPF, Silverlight and WP7

Tip #1: Identify your Targets

- Understand differences and benefits of each platform
- Evaluate application requirements versus platform capabilities
- Decide which platforms to target

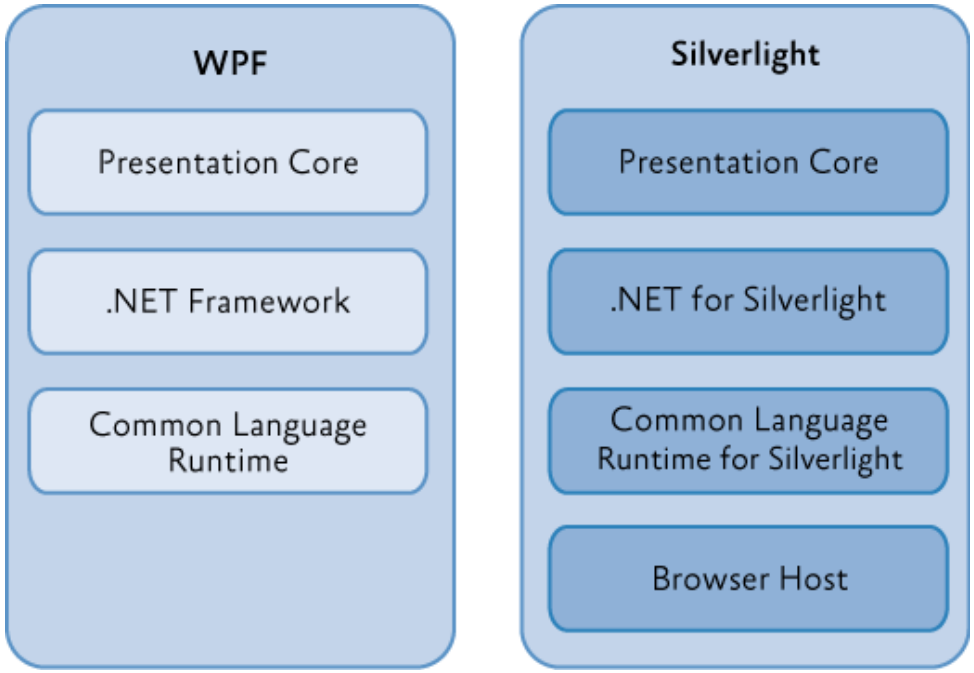


Tip #1: Identify your Targets

- WPF is desktop and sometimes Web (XBAP)
- Silverlight is Web and sometimes desktop (OOB)
- Trust Model
  - WPF: Full Trust
  - In browser Silverlight: Sandboxed
  - Out of browser Silverlight: Can be elevated
  - Windows Phone: Sandboxed

Tip #1: Identify your Targets

- WPF leverages the full .NET framework
- Silverlight uses a subset of .NET and executes on different CLR (Common Language Runtime)



Tip #1: Identify your Targets

- Silverlight
  - Cross-platform
  - Requires ASYNC connections
  - Transparent installers
- WPF
  - Can work offline
  - Interacts with peripherals
  - Full access to hard-drive
- WP7
  - Performance concerns

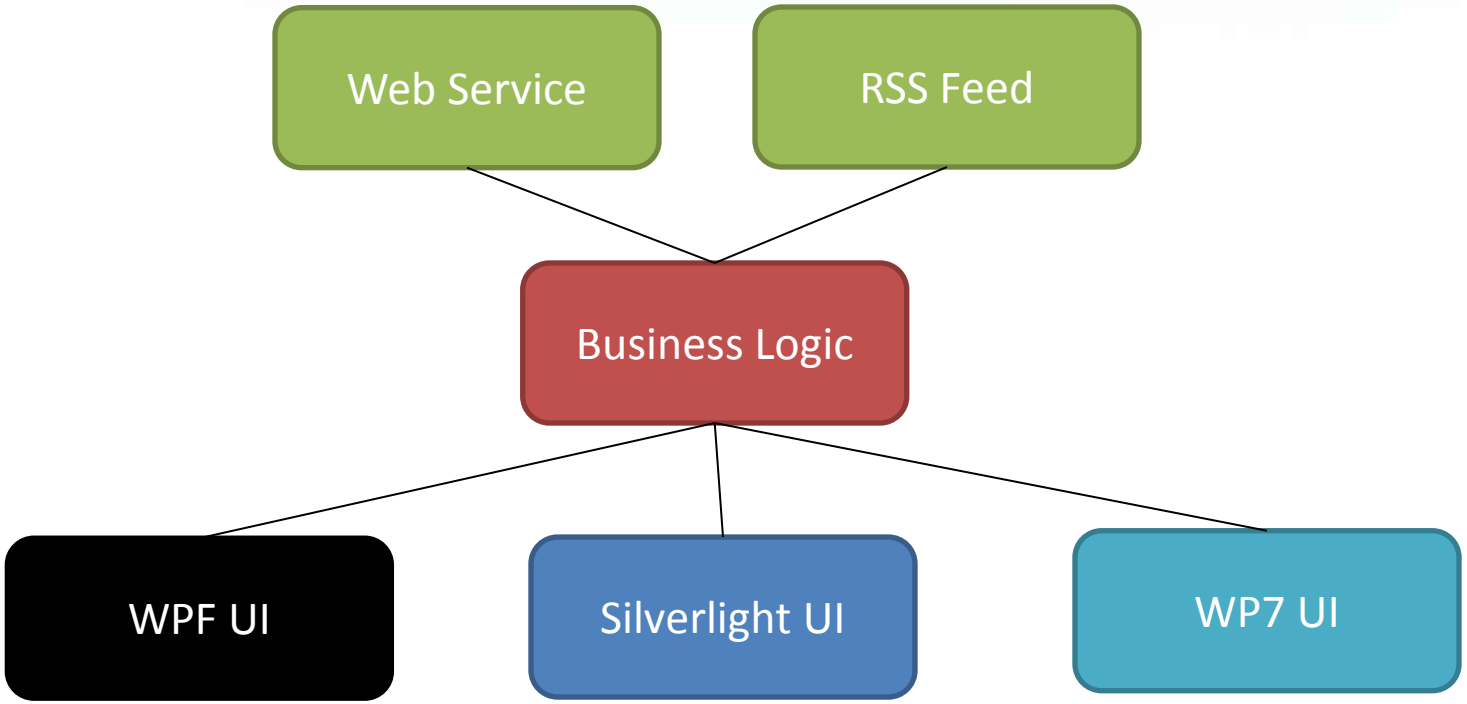
Tip #1: Identify your Targets

- Identify what can be shared
- Separation of concerns
  - Data layer
  - Business logic
  - User interface

Tip #1: Identify your Targets

- Easier to share
  - Presentation logic
  - Business logic and rules
  - Business entities
- Harder to share
  - Visual elements (views)
  - Configuration settings
  - Data access
  - Interop
  - Logging and tracing

Tip #1: Identify your Targets



Tip #2: Lowest Common Denominator

- Start with the lowest common denominator
- WP7 is a subset of Silverlight
- Silverlight is a subset of WPF

- Start with WP7 -> Silverlight -> WPF

Tip #2: Lowest Common Denominator

- Class Libraries
  - .NET class libraries cannot be referenced from Silverlight/WP7
  - WP7 class libraries can be referenced by all
  - Silverlight class libraries can be referenced by all, but with warning in WP7*

Tip #3: Portable Class Libraries

- Microsoft Portable Class Library Project
- New in 2011
- Write and build managed assemblies that work on more than one .NET platform
  - NET 4 (WPF, WinForms, ASP.NET)
  - Silverlight 4
  - Windows Phone 7
  - Xbox 360

Tip #3: Portable Class Libraries

- Based on the Class Library Project and .NET 4's assembly portability feature
- Narrows available APIs based on targeted platforms for you (referenced assemblies)
- Easy Separation
  - Portable code goes into a portable library project
  - Platform specific code goes into the application project
- Easy Coding
  - You don't have to know the entire API surface

Tip #3: Portable Class Libraries

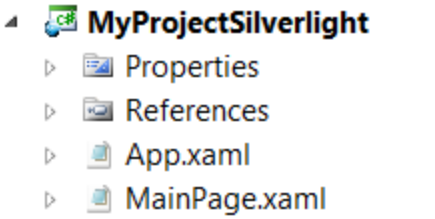
- Write portable code once
  - Business Logic
  - Validation Code
  - File/Protocol Processors
  - Serialization
  - Service Access
  - View Models
  - Authentication Code
  - Basic Networking
  - XML Processing

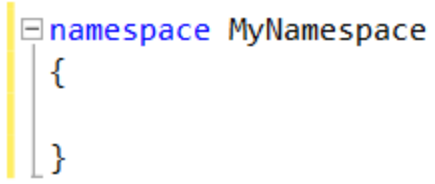
Tip #3: Portable Class Libraries

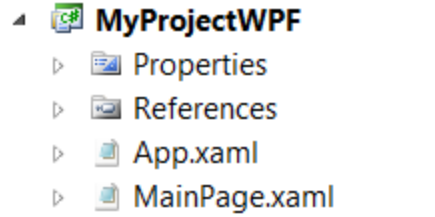
- Portable Core Assemblies
  - mscorlib.dll
  - System.dll
  - System.Core.dll
  - System.Xml.dll
  - System.Net.dll
  - System.Runtime.Serialization.dll
  - System.ServiceModel.dll
  - System.Xml.Serialization.dll
  - System.Windows.dll *(no WPF)*
  - System.ComponentModel.Composition.dll *(no WP7)*

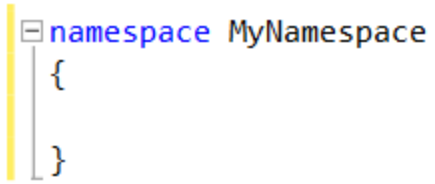
Tip #4: Universal Namespace

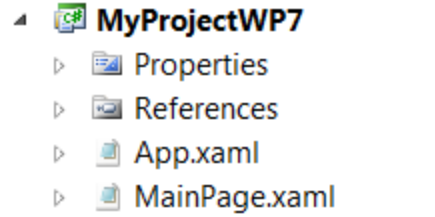
- Use the same namespace in all platforms
- Enables sharing of files
  - CS/VB
  - XAML

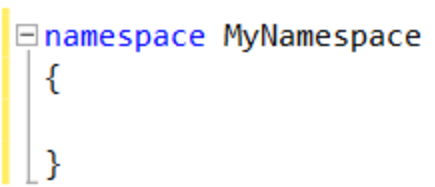












Tip #5: UserControls

- Most basic way to create a control
- UserControl is common to all 3 platforms
- Enables sharing of UI elements (XAML)

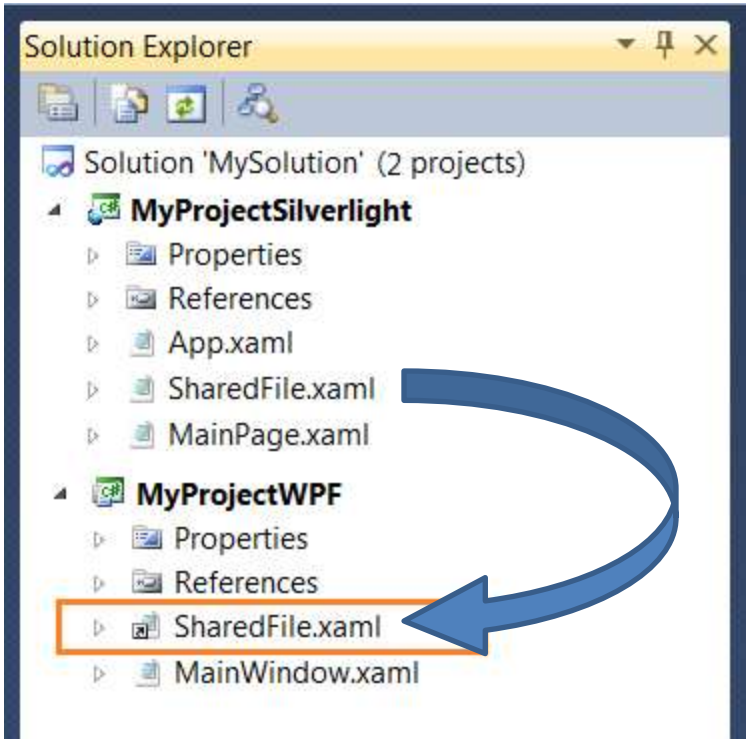
WPF	Silverlight	Windows Phone
Window	*Page*	*PhoneApplicationPage*
UserControl	**UserControl**	**UserControl**

Tip #5: UserControls

- Universal namespace is necessary
- Sharing XAML with Windows Phone is not as easy or ideal as it is between WPF and Silverlight
- In many cases, it might not make sense to share any XAML between projects

Tip #6: Project Linking

- Share files across multiple projects
- In Visual Studio: Add Existing Item as Link



Tip #6: Project Linking

- Project Linker Tool
  - Automatically creates and maintains links from a source project to a target project
  - VS2010 Extension
  - Developed by Prism team

Tip #6: Project Linking

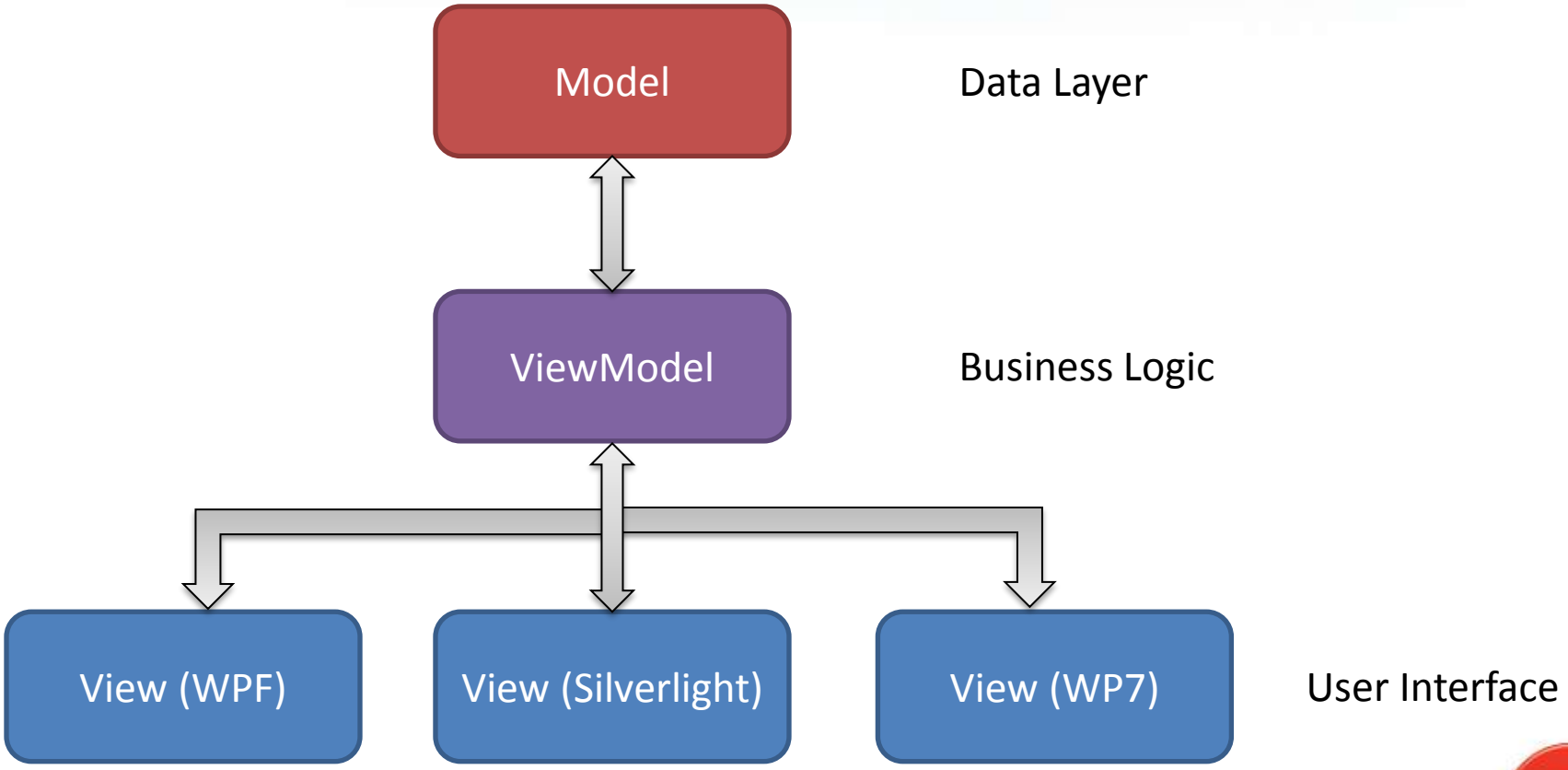
- Prism provides guidance designed to help you more easily design and build rich, flexible, and easy-to-maintain WPF, Silverlight and WP7 applications
- Important concepts in Prism
  - XAML
  - Data binding
  - Resources
  - Commands
  - UserControls
  - Dependency Properties

Tip #7: MVVM

- Separation of concerns and solid design patterns are essential
- Separation of concerns
  - Data layer
  - Business logic
  - User interface
- The Model-View-ViewModel design pattern helps separate code-base

Tip #7: MVVM

- Model-View-ViewModel



Tip #7: MVVM

- You can write in “MVVM” without any special tools or libraries
- Popular MVVM frameworks which help
  - Caliburn
  - MVVM Light
  - WAF
  - Cinch
  - nRoute

Tip #8: Conditional Compilation

- Compile incompatible parts of code using #if/#elif preprocessors with conditional compilation symbols
  - WINDOWS, SILVERLIGHT, WINDOWS_PHONE

```
#if WINDOWS
// Execute code that is specific to Windows
#elif SILVERLIGHT
// Execute code that is specific to Silverlight
#elif WINDOWS_PHONE
// Execute code that is specific to Windows Phone
#else
// Print a compile-time error message
#endif
```

Tip #9: Partial Classes

- Split class definitions over two or more source files to extend platform-specific functionality
- Enables a class to contain both shared and unique parts

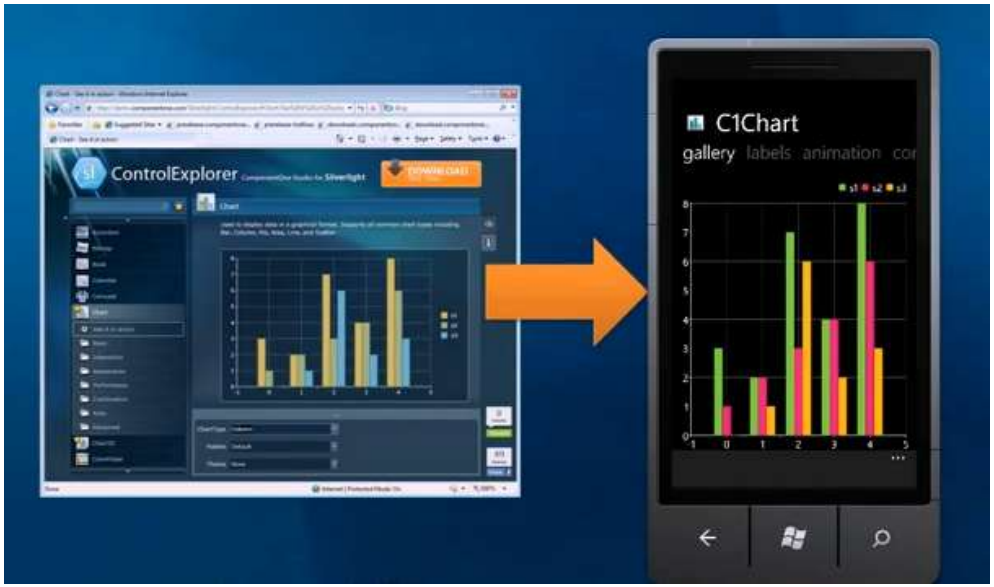
C#

```
public partial class Employee
{
    public void DoWork()
    {
    }
}

public partial class Employee
{
    public void GoToLunch()
    {
    }
}
```

Tip #10: Cross-Platform Control Libraries

- Use cross-platform compatible control libraries
- Not all control vendors ship cross-platform XAML controls

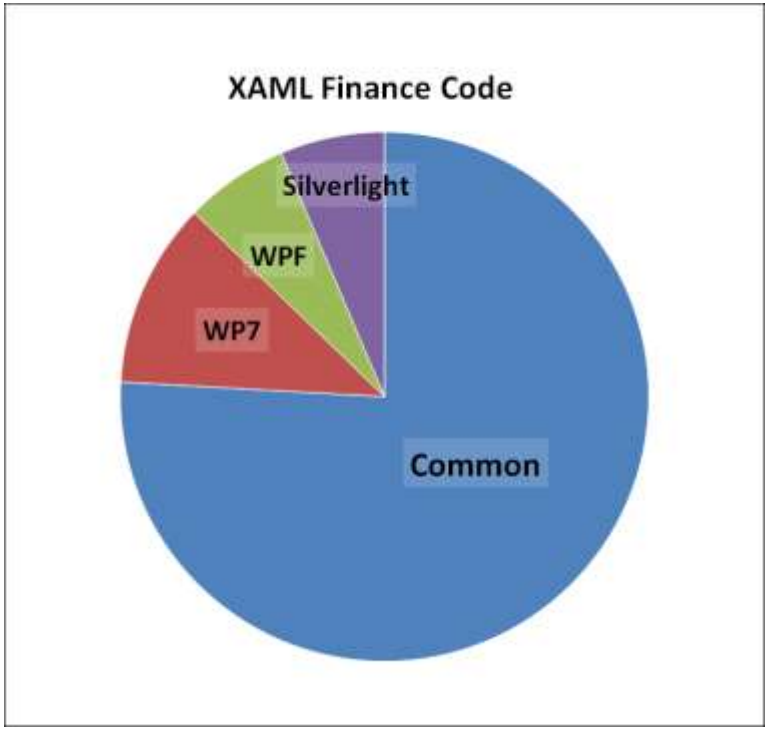


Other Things to Watch Out For

- DynamicResource is WPF only; StaticResource is common
- Triggers not supported in Silverlight
- Visual State Manager added in WPF 4
- WP7 has no pixel shader effects
- No routed commands in Silverlight
- No DataTemplateSelector in Silverlight

XAML Finance Sample

- Written by Colin Eberhardt @ ScottLogic
  - http://www.codeproject.com/KB/windows-phone-7/XAMLFinance.aspx

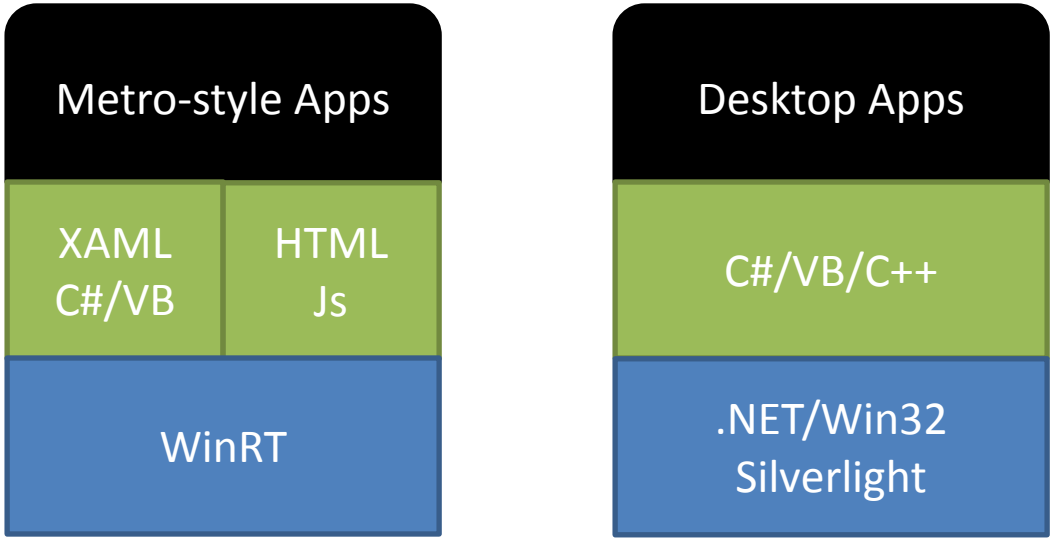


XAML[4]: Windows 8

- WinRT Metro-style apps
  - Tablet-inspired
  - Consumer-based
  - Touch-first

XAML[4]: Windows 8

- Write in HTML+Js or XAML+C#/VB



XAML[4]: Windows 8

- Cross-platform solutions with WinRT is entirely possible using the same techniques
- Share code by linking files to WPF/SL/WP7
- Some classes are in different libraries for WinRT (easy to workaround with #if WINRT)
- Makes no sense to share XAML with WinRT

Conclusion

- Multi-targeting is sharing code to target two or more platforms
  - Portable assemblies
  - Linking source files
- Sharing code is not always easy or practical but it is possible
  - Silverlight, WPF and WP7 are have some overlap and each contain unique features
  - There are many tips and guidelines to follow
  - Decrease development time and cost while offering new business opportunities

10 Tips Recap

1. Identify Your Targets
2. Lowest Common Denominator
3. Portable Class Libraries
4. Universal Namespace
5. UserControls
6. Project Linking
7. MVVM
8. Conditional Compilation
9. Partial Classes
10. Cross-Platform Control Libraries

Resources

- XAML Finance sample
  - Colin Eberhardt @ ScottLogic
  - http://www.codeproject.com/KB/windows-phone-7/XAMLFinance.aspx
- Code sharing practices with Prism
  - http://msdn.microsoft.com/en-us/library/ff921109(v=PandP.40).aspx
- Sharing Silverlight assemblies in .NET
  - http://blogs.msdn.com/b/dotnet/archive/2009/12/01/sharing-silverlight-assemblies-with-net-apps.aspx