



Windows Presentation Foundation

Part Contents

■ Introduction to Xaml	459
Extensible Application Markup Language - XAML	462
Why the Windows Presentation Foundation?	464
Creating a WPF Application - Visual Studio Blend	466
WPF Application References and Code-behind	469
Getting Nosey with ILDASM and ILSpy	472
<i>A Sample Silverlight Application</i>	473
Slimline WPF	477
■ F# and WPF	479
Displaying a WPF Window	479
<i>A Guaranteed Failure!</i>	484
Parsing the Embedded BAML Resource	485
<i>Using a System.Resources.ResourceReader</i>	486
<i>Rendering the BAML</i>	488
<i>BAML Element Selection</i>	489
What about non-Window Xaml Elements?	495
Displaying Non-Window Controls	496
What about Silverlight?	502
■ An F# WPF Host Control Utility	507
Units of Measure	507
<i>A Unit of Measure for Point sizes</i>	511
<i>Units of Measure for Percentages and Scaling</i>	514

<i>A Unit of Measure for Pixels</i>	515
Defining your own Attributes	521
Another use for Unions	524
A Strongly Typed Disk File Value Type	528
<i>Implementing and Using IDisposable</i>	530
<i>The Strongly Typed Disk File</i>	532
Framework Element Extensions	537
<i>Displaying Error Text</i>	538
<i>A Scalable, Empty Window</i>	540
<i>An Error Display Window</i>	541
<i>Constructing a Host Window</i>	543
A WPF Library Record Type	545
Testing the Utility	551
Summary	554
■ Enumerating the Control Hierarchy	557
The WPF Visual and Logical Trees	559
<i>The Visual Tree</i>	560
<i>The Logical Tree</i>	561
<i>A Recursive Element Enumerator</i>	563
Recursive Types	563
<i>Modelling a Tree</i>	564
<i>A Type to represent a Control Node</i>	567
<i>A Recursive Union to model the Control Tree</i>	568
<i>A Rich WPF Sample Window</i>	571
<i>A Recursive List versus a Sequence</i>	574
<i>Asynchronously Evaluating the Node Index</i>	582
<i>Control Tree List or Sequence - revisited</i>	588
Refining our Control Tree	589
<i>Refining our Position Key</i>	591
Structural Equality & Comparison	592
Custom Equality & Comparison	594

Operator Overloads & New Operators	595
<i>A Composite WPF Map & Tree</i>	596
The Graph Populate Member	598
Normalising Graph Member Arguments	602
Tree Traversal	603
A Nodes Children and Descendants	605
A Nodes Parent and Ancestors	608
Selecting a Sequence of Typed Controls	609
Picking a Single Node	610
Testing Node Selection	610
■ Assembly Load Context	613
The Xceed Extended WPF Toolkit	613
<i>Creating a WPF Window to Display the Sample Chart</i>	616
<i>Trying to Show the Sample Chart from F#</i>	618
The Application Domain	619
Dynamic Load Context	621
Assembly Resolution Errors	626
Handling Default Constructors & Static Members	632
<i>Invoking a Default Constructor via Reflection</i>	637
<i>Invoking a Static Member via Reflection</i>	640
<i>Finishing off the MVVM Window</i>	642
■ WPF Image Resources	645
Demonstrating Image Referencing in WPF/Blend	646
<i>Methods of Image Referencing</i>	649
F# and WPF Image Resources - Preamble	655
<i>Loading a BAML App as an Application Instance</i>	657
<i>Inspecting the Runtime Image Source Property</i>	658
Deserialising a WPF Window	658
Inspecting Runtime Resources	661
Inspecting the Window BAML	664
Suppressing Dynamic Image Loading	667
<i>Image Load Strategy</i>	670

■ Manifest Resources	677
The Variety of Manifest Resources	678
Building a Manifest Resource Module	682
<i>Manifest Module Layout</i>	683
<i>Supporting types and functions</i>	684
<i>Evaluating a Resources Mime Type</i>	687
The Function bitmapMimeType	688
The Function evalMimeType	688
The Function evalResourceType	690
The function parseResourceValue	693
<i>Manifest.ForAssembly Private Fields</i>	693
Implementing IDisposable	694
Verifying the Target Assembly for Manifest Resource Access	694
The Target Assembly Short Name	695
Instantiation Evaluation of the Manifest Resource Map	695
Assigning Resource Managers for each RESX Repository	697
<i>Manifest.ForAssembly Public Properties</i>	697
<i>Fetching BAML and Xaml WPF Resources</i>	698
<i>Fetching Non-WPF Resource Streams/Objects</i>	700
WPF Resource Repository	702
Embedded Resource Repository	703
Resx Resource Repository	703
<i>Fetching Strongly Typed Resource Values</i>	704
Method GetIcon	704
Method GetBitmap	705
Method GetXml	706
Method GetText	706
<i>Fetching a Temporary File of a Resource Value</i>	707
Testing the Manifest Module	708
■ Modifying the WPF Image Source	715
Program Structure Overview	716
Replacing Null WPF Sources	718

Exposing GDI+ Graphics as Streams	720
Iterating the Graph Framework Elements	722
Dynamically Evaluating a BitmapImage	724
Running the Code	728
■ Summary	735

Accompanying Videos

130. A first look at Blend for Visual Studio 2015.	468
131. Basic WPF application structure.	470
132. Inspecting WPF & Silverlight assemblies - Parts I & II.	476
133. Slimline WPF.	477
134. Creating a convenience tool-bar with save and build buttons.	478
135. Displaying a WPF Window in F# - Parts I through III	485
136. Correctly parsing a single Xaml element from BAML - Parts I & II.	487
137. Cleaning up WPF Assembly access - Parts I & II.	495
138. Creating a Host Window for non-Window Elements - Parts I & II.	501
139. Hosting Silverlight in a WPF Window.	506
140. Setting up the Utility Source Code.	509
141. Units of Measure - Parts I through IV.	520
142. Attributes.	523
143. The Union Messages.	525
144. The Union Constants.	527
145. Simplistic testing of strongly-typed attributes.	528
146. Dealing with temporary files.	532
147. A strongly-type disk file type.	536
148. A TextBlock extension.	540
149. Windows Extensions - Parts I & II.	543
150. A WPF Library Record Type - Parts I & II.	549
151. Running the host control utility.	554
152. The WPF Visual Tree Helper.	560

153. The WPF Logical Tree Helper.	563
154. Recursive enumeration through the Logical Tree Helper.	563
155. An overview of tree structures.	565
156. A Simple control tree of our hosted Silverlight user control.	570
157. Making the MSDN WPF Control Sample Gallery F# compliant.	573
158. Enumerating the WPF Gallery controls.	574
159. Control Node Numbering - Parts I & II.	583
160. Using a Mailbox Processor to evaluate the Index - Parts I through III.	588
161. Custom Equality, Comparison and Operator Overloading - Parts I & II.	595
162. An overview of our Control Graph	598
163. The Graph Populate member.	601
164. Validation of our normalised member argument.	602
165. ControlTree GetNode and GetBranch.	604
166. A Graph nodes Children and Descendants.	606
167. A Graph nodes Parent and Ancestors.	609
168. Selecting nodes by framework element type.	610
169. Picking a strongly typed single node.	610
170. Running some Graph examples.	612
171. Using a third-party toolkit - Parts I & II.	618
172. Displaying a WPF Window in F# - Parts I through III.	620
173. Assembly Load Context - Parts I through III.	626
174. Dealing with Assembly Resolution exceptions - Parts I through III.	631
175. Building the MVVM Sample - Parts I & II.	636
176. Using reflection to instantiate a class via a default constructor - Parts I & II.	640
177. Using reflection to invoke a static member.	641
178. Successfully running the MVVM Sample.	643
179. Specifying a Window Icon - Parts I & II.	646
180. Defining Image Resources - Parts I & II.	649
181. Setting up the F# WPF Stub.	655
182. Using app.baml as an Application instance.	658

183. Deserialising a WPF Window to Xaml - Parts I & II.	659
184. Inspecting the BAML stream - Parts I & II.	667
185. Resource keys and suppressing Image loading - Parts I & II.	669
186. Adding sample resources to our projects.	682
187. Overview of the Manifest module - Parts I & II.	687
188. Evaluating the mime type - Parts I & II.	693
189. ForAssembly Private Fields.	697
190. ForAssembly Public Properties and utility methods.	698
191. Extracting BAML and Xaml from a Wpf repository.	700
192. Fetching non-WPF resources from the manifest - Parts I & II.	708
193. Testing Manifest.ForAssembly - Parts I & II.	713
194. Overview of layout for re-assigning image sources.	717
195. Changes to the Manifest module - Parts I & II.	722
196. Changes to the Wpf module - Parts I & II.	728
197. Running the WpfImageSource code - Parts I & II.	733