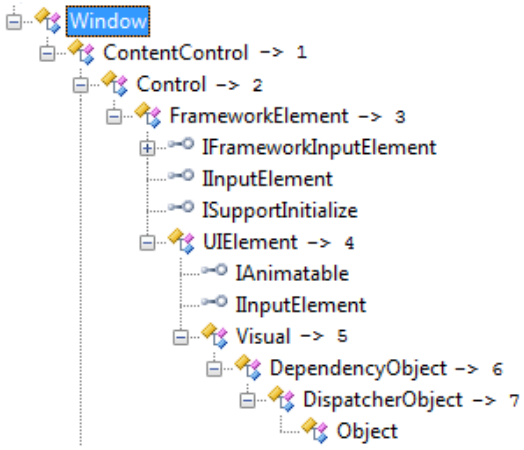
# Summary of WPF Class Hierarchy




1. The **ContentControl** parent class allows derived types to host *content*. WPF content can be composed of any sort of UI elements. Many WPF controls also have ContentControl in their inheritance chain, like Frame, GroupItem, *HeaderedContentControl*, Label, ListBoxItem, ButtonBase, StatusBarItem, ScrollViewer, ToolTip, UserControl and Window. The WPF content model allows us to radically change the composition of a control with minimal fuss (e.g. a Button could maintain an inner ellipse, rectangle and/or text as *content*).
2. The **Control** parent class defines a number of members which give derived types (including Window) their *core look and feel*, properties like *opacity*, *tab order* logic, *background color*, and so on. The Control type also provides the infrastructure to *apply templates* and *styles* to a UI widget.
3. **FrameworkElement** is another key parent class to many UI widgets, in that it provides members to control *size*, *tooltips*, *mouse cursor* and other settings. This class also provides support for WPF *animation* services.
4. **UIElement** provides the greatest amount of functionality: events to process *mouse and input focus* and properties to control *focus*, *visibility*, and *geometric transformation*.
5. **Visual** provides derived types the core infrastructure to *render* their *UI output*. It provides *hit-testing support* and *coordinates transformation*. It is also the *connection* between *WPF* and the *DirectX* subsystem. This shim is represented by an unmanaged *.dll named *milcore.dll* (stands for *Media Integration Layer*)
6. **DependencyObject** is the parent which provides derived types the ability to work with the WPF dependency property model. Dependency property makes it possible for a property to *receive input* from *multiple locations* and is key part of WPF's animation/data binding services.
7. Finally, **DispatcherObject** provides access to the WPF application's lower level *event queue* via the Dispatcher property.

More details can be found here (which is also the inspiration for this document):
https://docs.microsoft.com/en-us/dotnet/framework/wpf/advanced/wpf-architecture