GUI PROGRAMMING

WINDOWS PRESENTATION FOUNDATION

WPF basics

- Can be the best tool to build rich standalone and XBAP applications.
- Runs on newer Windows platforms beginning Vista and accompanying Server versions.
- It can also be made to run on ancient Windows XP computers.
  - Still found in many businesses even up to these days.
  - To do it target .NET Framework 4.

WPF innovative features

- Hardware acceleration
- Resolution independence

Graphics rendering tiers

- Video RAM The amount of video memory on the graphics hardware determines the size and number of buffers that can be used for compositing graphics.
- Pixel Shader A pixel shader is a graphics processing function that calculates effects on a per-pixel basis. Depending on the resolution of the displayed graphics, there could be several million pixels that need to be processed for each display frame.
- Vertex Shader A vertex shader is a graphics processing function that performs mathematical operations on the vertex data of the object.
- Multitexture Support Multitexture support refers to the ability to apply two or more distinct textures during a blending operation on a 3D graphics object. The degree of multitexture support is determined by the number of multitexture units on the graphics hardware.

Earlier Windows graphics[1]

- User32, provides the traditional Windows look and feel for elements such as windows, buttons, text boxes and so on.

- GDI/GDI+, provides drawing support for rendering shapes, text, images, at the cost of the additional complexity (and often lackluster performance)
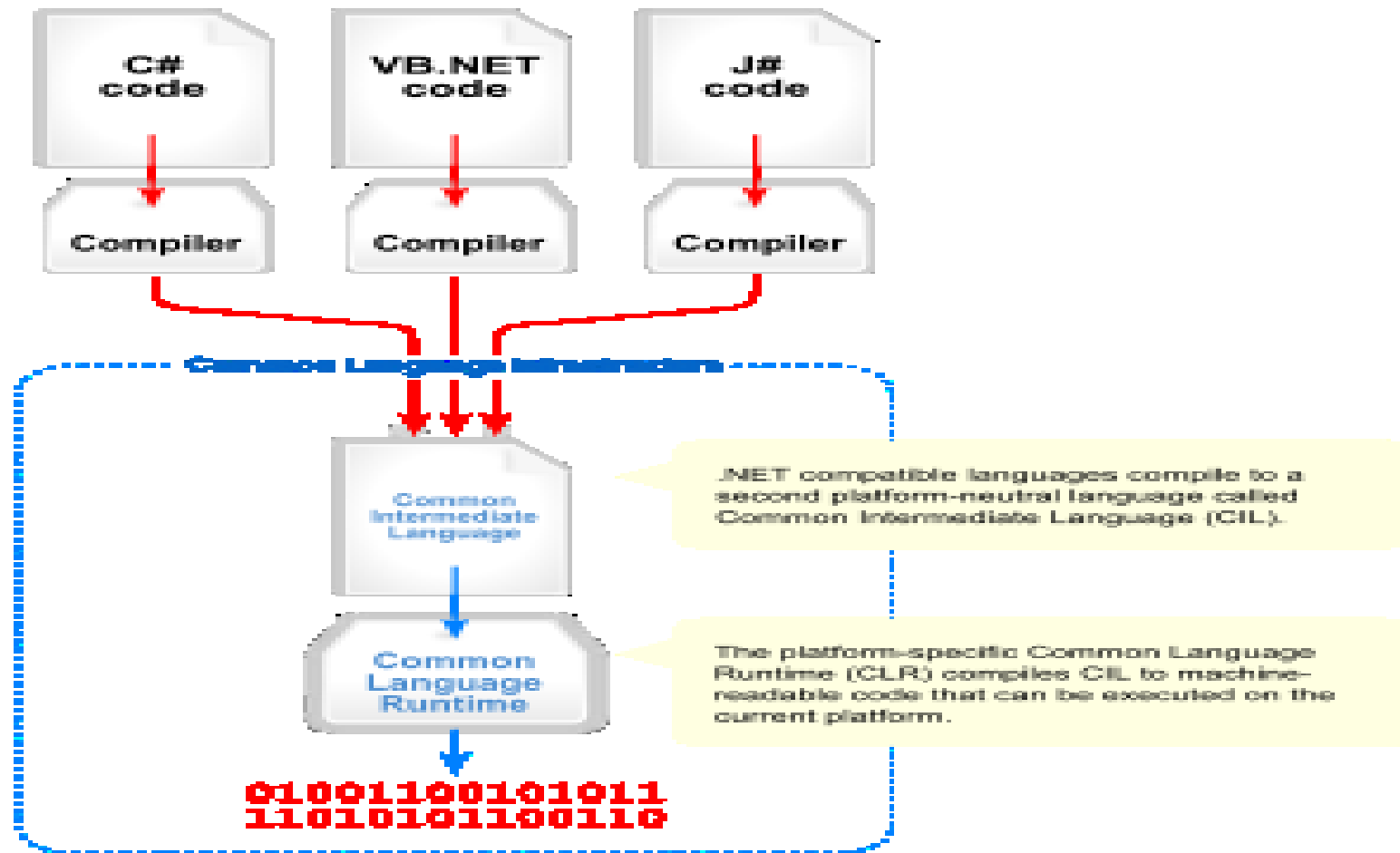
.NET Framework Stack

The .NET Framework Stack

.NET Framework releases

Version number	CLR version	Release date	Included in Development tool	Windows	Windows Server	Replaces
1.0	1.0	2002-02-13	Visual Studio .NET	N/A	N/A	N/A
1.1	1.1	2003-04-24	Visual Studio .NET 2003	N/A	2003	1.0
2.0	2.0	2005-11-07	Visual Studio 2005	N/A	2003, 2008 SP2, 2008 R2 SP1	N/A
3.0	2.0	2006-11-06	Expression Blend	Vista	2008 SP2, 2008 R2 SP1	2.0
3.5	2.0	2007-11-19	Visual Studio 2008	7, 8, 8.1	2008 R2 SP1	2.0, 3.0
4.0	4	2010-04-12	Visual Studio 2010	N/A	N/A	N/A
4.5	4	2012-08-15	Visual Studio 2012	8	2012	4.0
4.5.1	4	2013-10-17	Visual Studio 2013	8.1	2012 R2	4.0, 4.5
4.5.2	4	2014-05-05	N/A	10 (technical preview)	N/A	4.0, 4.5, 4.5.1

Common Language Infrastructure

CLI

- To provide a language-neutral platform for application development and execution, including
  - Functions for exception handling,
  - Garbage collection,
  - Security, and
  - Interoperability.
- By implementing the core aspects of .NET Framework within the scope of CLI, this functionality will not be tied to a single language but will be available across the many languages supported by the framework.

CLI assemblies

- A compiled code library used for
  - Deployment,
  - Versioning, and
  - Security.
- Types
  - Process assemblies (EXE)
  - Library assemblies (DLL)
- Contain code in CIL, which is usually generated from a CLI language, and then compiled into machine language at run time by the just-in-time compiler. In the .NET Framework implementation, this compiler is part of the Common Language Runtime (CLR).

CLI process assemblies

- Represents a process that will use classes defined in library assemblies.

Assembly

- Consist of one or more files.
- Code files are called modules.
- Can contain more than one code module.
- It is possible to use different languages to create code modules,
- It is technically possible to use several different languages to create an assembly.
- Visual Studio however does not support using different languages in one assembly.

Assembly name, consist of four parts

- The short name. On Windows this is the name of the Portable Executable (PE, a file format for executables, object code, DLLs, Font files, and others used in 32-bit and 64-bit versions of Windows operating systems. The PE format is a data structure that encapsulates the information necessary for the Windows OS loader to manage the wrapped executable code) file without the extension.

- The culture. This is an RFC 1766 identifier of the locale for the assembly. In general, library and process assemblies should be culture neutral; the culture should only be used for satellite assemblies.

- The version. This is a dotted number made up of four values — major, minor, build and revision.

- A public key token. This is a 64-bit hash of the public key that corresponds to the private key used to sign the assembly. A signed assembly is said to have a strong name.

Assembly, other important things

- Assembly versions
- Assemblies and CLI security
- Satellite assemblies
- Referencing assemblies
- Delay-signing of an assembly
- Language of an assembly

CIL

- Its code is housed in CLI assemblies.

- As mandated by the specification, assemblies are stored in Portable Executable (PE) format, common on Windows platform for all DLL and EXE files.

- The assembly consists of one or more files, one of which must contain the manifest, which has the metadata for the assembly.

CLR

- Serves as the execution engine of .NET Framework. All .NET programs execute under the supervision of CLR, guaranteeing certain properties and behaviors in the areas of
  - Memory management,
  - Security, and
  - Exception handling.
- Is MS implementation of CLI.

.NET Framework design and architecture (various components)

- Interoperability

- Language independence

- Framework Class Library (using .NET Framework itself)

- Simplified deployment

- Security

- Portability (ISO/IEC 23271:2012 and ISO/IEC 23270:2006, makes it possible to be platform agnostic)

- Memory management

WPF

- A graphical subsystem for rendering user interfaces in Windows-based applications by Microsoft.

- Previously known as Avalon.

- Initially release as part of .NET Framework 3.

- Uses DirectX, rather than older GDI subsystem.
  - DirectX a collection of API for handling tasks related to multimedia, especially game programming and video, on Microsoft platforms.
  - GDI an MS API and core operating system component responsible for representing graphical objects and transmitting them to output devices such as monitors and printers. responsible for tasks such as drawing lines and curves, rendering fonts and handling palettes. It is not directly responsible for drawing windows, menus, etc.; that task is reserved for the user subsystem, which resides in user32.dll and is built atop GDI.
    - Other systems have components that are similar to GDI, for example Mac OS X's Quartz and X Window System's Xlib/XCB.

WPF continuation

- Attempts to provide a consistent programming model for building applications and separates the user interface from business logic.

- It resembles similar XML-oriented object models, such as those implemented in XML User Interface Language (XUL) and Scalable Vector Graphic (SVG).

XUL

- A user interface markup language that is developed by Mozilla. Implemented as an XML dialect; allows for graphical user interfaces to be written in a similar manner to Web pages.

- XUL can be used to write cross-platform applications such as Mozilla Firefox, where it is interpreted by the layout engine known as Gecko which renders Firefox's user interface and Web page display.

SVG

- An XML-based vector image format for two-dimensional graphics with support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C) since 1999.

- Images and their behaviors are defined in XML text files. This means that they can be searched, indexed, scripted, and compressed. As XML files, SVG images can be created and edited with any text editor, but are more often created with drawing software.

- All major modern web browsers—including Mozilla Firefox, Internet Explorer, Google Chrome, Opera, and Safari—have at least some degree of SVG rendering support.

WPF features

- Direct3D
- Data binding
- Media services
- Templates
- Animations
- Imaging
- Effects
- Documents
- Texts
- Interoperability
- Alternative input
- Accessibility

WPF Direct3D

- Graphics, including desktop items like windows, are rendered using Direct3D.

- This allows the display of more complex graphics and custom themes, at the cost of GDI's wider range of support and uniform control theming.

- Allows Windows to offload some graphics tasks to the GPU. This reduces the workload on the computer's CPU.

- GPUs are optimized for parallel pixel computations. This tends to speed up screen refreshes at the cost of decreased compatibility in markets where GPUs are not necessarily as powerful, such as the netbook market.

- WPF's emphasis on vector graphics allows most controls and elements to be scaled without loss in quality or pixelization, thus increasing accessibility.

WPF data binding

- Provides built-in set of data services to enable application developers to bind and manipulate data within applications. It supports four types of data binding:
  - one time: where the client ignores updates on the server.
  - one way: where the client has read-only access to data.
  - two way: where client can read from and write data to the server
  - one way to source: where the client has write-only access to data

WPF data binding continuation

- LINQ queries, including *LINQ to XML*, can also act as data sources for data binding.
- Binding of data has no bearing on its presentation. WPF provides data templates to control presentation of data.
- A set of built-in controls is provided as part of WPF, containing items such as button, menu, grids, and list box.
- A powerful concept in the WPF is the logical separation of a control from its appearance.
  - A control's template can be overridden to completely change its visual appearance.
  - A control can contain any other control or layout, allowing for a high degree of control over composition.
- Features retained mode graphics. Repainting the display isn't always necessary.

WPF media services

- Provides an integrated system for building user interfaces with common media elements like vector and raster images, audio, and video.
- Also provides an animation system and a 2D/3D rendering system.
- Provides shape primitives for 2D graphics along with a built-in set of brushes, pens, geometries, and transforms.
- The 3D capabilities in WPF are a subset of the full-feature set provided by Direct3D. However, WPF provides tighter integration with other features like user interfaces, documents, and media. This makes it possible to have 3D user interfaces, 3D documents, or 3D media.
- There is support for most common image formats: BMP, JPEG, PNG, TIFF, Windows Media Photo, GIF, and ICON.
- WPF supports the video formats WMV, MPEG and some AVI files by default, but since it has Windows Media Player running beneath, WPF can use all the codecs installed for it.

WPF templates

- You can define the look of an element directly, via its properties, or indirectly with a Template or Style. At its simplest a style is a combination of property settings that can be applied to a UI element with a single property attribute. Templates are a mechanism for defining alternate UI for portions of your WPF application.

- There are several template types available in WPF
  - ControlTemplate,
  - DataTemplate,
  - HierarchicalDataTemplate and
  - ItemsPanelTemplate.

WPF control template

- Underlying all UI controls in WPF is a new composition model. Every control is composed of one or more 'visuals'. These visual sub-elements are turned into a hierarchical Visual Tree by WPF and eventually rendered by the GPU. Because WPF controls are not wrappers for standard Windows controls their UI can be radically changed without affecting the normal behavior of the control.

- Every control in WPF has a default 'template' that defines its visual tree. The default template is created by the control author and is replaceable by other developers and designers. The substitute UI is placed within a ControlTemplate.

WPF data template

- Has a flexible data binding system. UI elements can be populated and synchronized with data from an underlying data model. Rather than showing simple text for the bound data WPF can apply a Data Template (replaceable UI for .NET types) before rendering to the Visual Tree.

WPF animations

- Considered the 1st class programming concept.
- WPF supports time-based animations, in contrast to the frame-based approach. This decouples the speed of the animation from how the system is performing.
- WPF supports low level animation via timers and higher level abstractions of animations via the Animation classes.
  - Any WPF element property can be animated as long as it is registered as a Dependency Property.
  - Animation classes are based on the .NET type of property to be animated. For instance, changing the color of an element is done with the `ColorAnimation` class and animating the Width of an element (which is typed as a Double) is done with the `DoubleAnimation` class.

WPF animations continuation

- Animations can be grouped into Storyboards.
    - Storyboards are the primary way to start, stop, pause and otherwise manipulate the animations.
- Animations can be triggered by external events, including user action.
- Scene redraws are time triggered.
- Presentation timers are initialized and managed by WPF.
- Animation effects can be defined on a per-object basis, which can be accessed directly from XAML markup.

WPF imaging

- WPF can natively access Windows Imaging Component (WIC) code and APIs allowing developers to write image codecs for their specific image file formats.

WPF effects

- WPF 3.0 provides for Bitmap effects (BitmapEffect class), which are raster effects applied to a Visual. These raster effects are written in unmanaged code and force rendering of the Visual to be performed on the CPU and not hardware accelerated by the GPU. BitmapEffects were deprecated in .NET 3.5 SP 1.

- .NET Framework 3.5 SP1 adds the Effect class, which is a Pixel-Shader 2.0 effect that can be applied to a visual, which allows all rendering to remain on the GPU.

- The Effect class is extensible, allowing applications to specify their own shader effects.

- The Effect class, in .NET 3.5 SP1 and 4.0, ships with two built-in effects, BlurEffect and DropShadowEffect. There are no direct replacements for OuterGlowBitmapEffect, EmbossBitmapEffect and BevelBitmapEffect, previously provided by the deprecated BitmapEffect class. However there are other ways of achieving the same results with the Effect class. For example, developers can get an outer glow effect by using the DropShadowEffect with its ShadowDepth set to 0.

- Although the BitmapEffect class was marked as deprecated in .Net Framework 3.5 SP1, its use was still allowed and these effects would still render correctly. In .Net Framework 4.0 the BitmapEffect class was effectively deprecated - code referencing BitmapEffect still builds without errors, but no effect gets actually rendered.

WPF documents

- WPF natively supports paginated documents. It provides the DocumentViewer class, which is for reading fixed layout documents. The FlowDocumentReader class offers different view modes such as per-page or scrollable and also reflows text if the viewing area is resized.

- Natively supports XML Paper Specification documents.

- Supports reading and writing paginated documents using Open Packaging Convention.

WPF texts

- WPF includes a number of text rendering features that were not available in GDI. This is the first Windows programming interface to expose OpenType features to software developers, supporting OpenType, TrueType, and OpenType CFF (Compact Font Format) fonts.
- Support for OpenType typographic features includes:
  - Ligatures
  - Old-style numerals (for example, parts of the glyph hang below the text baseline)
  - Swash variants
  - Fractions
  - Superscript and subscript
  - Small caps
  - Line-level justification
  - Ruby characters
  - Glyph substitution
  - Multiple baselines
  - Contextual and Stylistic Alternates
  - Kerning

WPF texts continuation

- Handles texts in Unicode, and handles texts independent of global settings, such as system locale.

- Supports built-in spell checking.

- Supports such features as automatic line spacing, enhanced international text, language-guided line breaking, hyphenation, and justification, bitmap effects, transforms, and text effects such as shadows, blur, glow, rotation etc. Animated text is also supported; this refers to animated glyphs, as well as real-time changes in position, size, color, and opacity of the text.

- takes advantage of advances in ClearType technology, such as sub-pixel positioning, natural advance widths, Y-direction anti-aliasing, hardware-accelerated text rendering, as well as aggressive caching of pre-rendered text in video memory.

WPF interoperability

- Windows Forms is also possible through the use of the `ElementHost` and `WindowsFormsHost` classes.

WPF alternative input

- WPF supports digital ink-related functionality.
- WPF 4.0 supports multi-touch input on Windows 7 computers (but suffers a documented critical bug rendering it inoperable

WPF accessibility

- WPF supports Microsoft UI Automation to allow developers to create accessible interfaces.
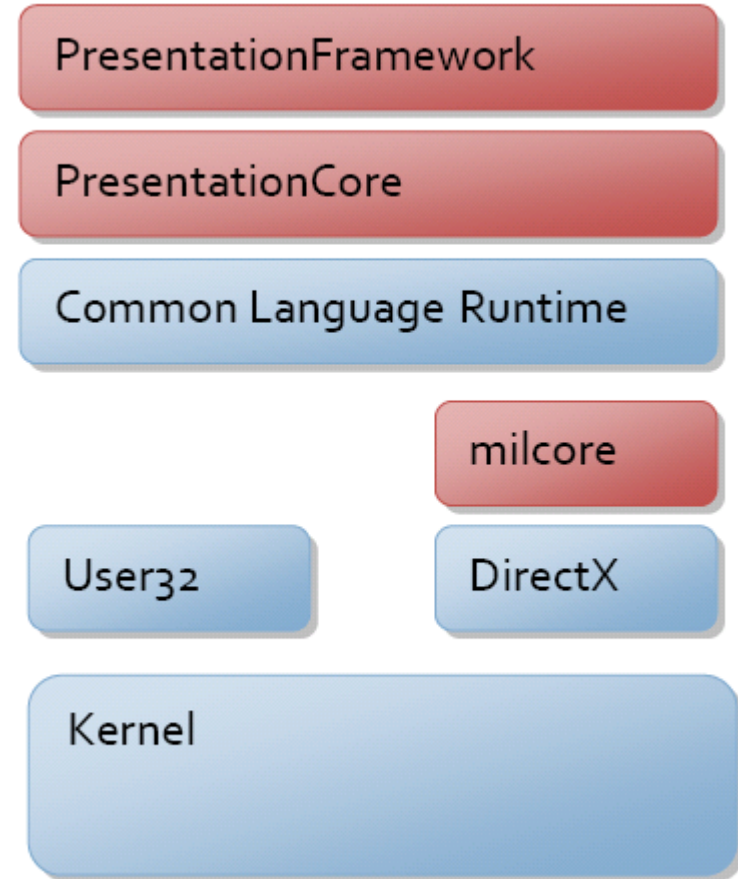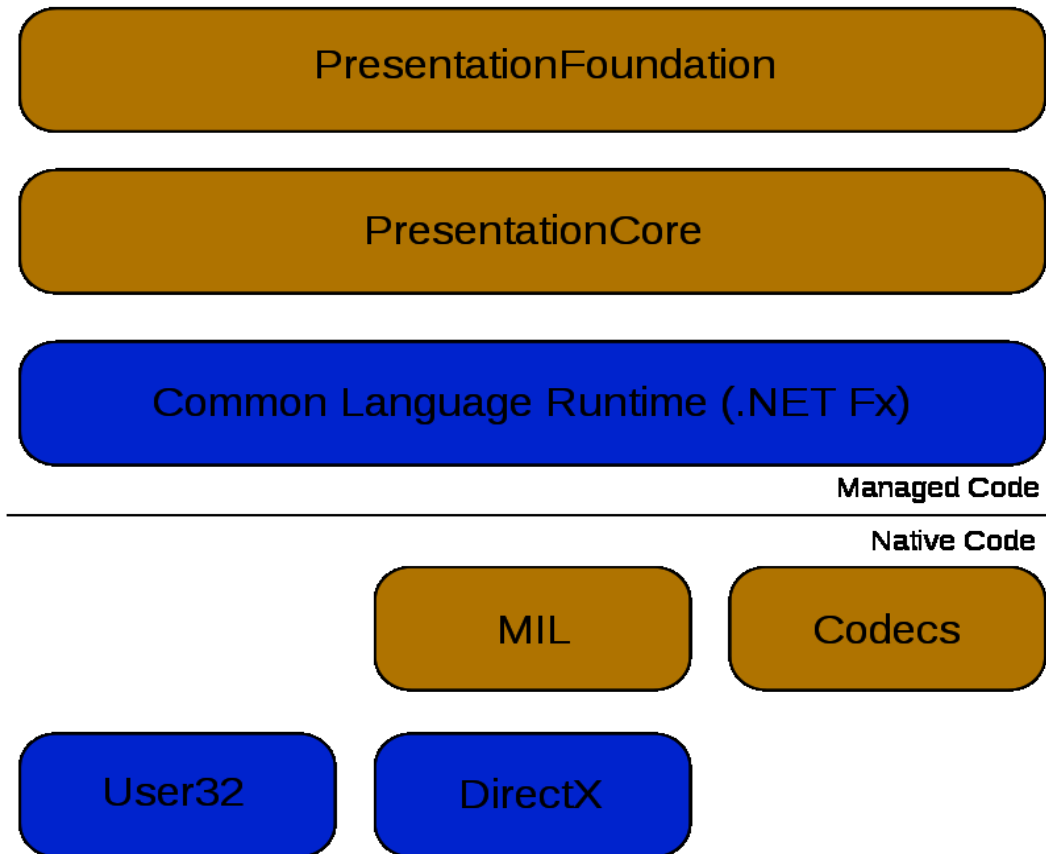- This API also allows automated test scripts to interact with the UI.

WPF employs XAML

- An XML-based language, to define and link various interface elements. WPF applications can also be deployed as standalone desktop programs, or hosted as an embedded object in a website.

- Aims to unify a number of common user interface elements, such as 2D/3D rendering, fixed and adaptive documents, typography, vector graphics, runtime animation, and pre-rendered media. These elements can then be linked and manipulated based on various events, user interactions, and data bindings.

XAML

- A specific advantage it brings to WPF is that XAML is completely declarative language, allowing the developer (or designer) to describe the behavior and integration of components without the use of procedural programming.

- Allows for separation of model and view, which is considered a good architectural principle.

- Elements and attributes map to classes and properties in the underlying APIs.

- Not everything in programming application in its entirety.

WPF Architecture

PresentationFoundation

PresentationCore

Common Language Runtime (.NET Fx)

Managed Code

Native Code

MIL

Codecs

User32

DirectX

PresentationFramework

PresentationCore

Common Language Runtime

milcore

User32

DirectX

Kernel

WPF architecture continuation

- The architecture of WPF spans across both managed code and native code components.
  - Managed code cause the public APIs to be exposed.
  - Native code cause the composition engine which renders WPF application.

WPF architecture directives

- System.Object
- System.Threading.DispatcherObject
- System.Windows.DependencyObject
- System.Windows.Media.Visual
- System.Windows.UIElement
- System.Windows.FrameworkElement
- System.Windows.Controls.Control

WPF architecture, managed

- `PresentationCore` (`presentationcore.dll`) provides a managed wrapper for MIL and implements the core services for WPF, including a property system that is aware of the dependencies between the setters and consumers of the property, a message dispatching system by means of a Dispatcher object to implement a specialized event system and services which can implement a layout system such as measurement for UI elements.

- `PresentationFramework` (`presentationframework.dll`) implements the end-user presentational features, including layouts, time-dependent, story-board based animations, and data binding.

WPF architecture, native

- Media Integration Layer (MIL) and resides in `milcore.dll`.
- It interfaces directly with DirectX and provides basic support for 2D and 3D surfaces, timer-controlled manipulation of contents of a surface with a view to exposing animation constructs at a higher level, and compositing the individual elements of a WPF application into a final 3D "scene" that represents the UI of the application and renders it to the screen.
- The media codecs are also implemented in unmanaged code, and are shipped as `windowscodecs.dll`

WPF deployment

- Offers both **standalone** and **XAML Browser Applications** (XBAP, pronounced "ex-bap") flavors.
- The programming model for building either type of application is similar.

WPF standalone deployment

- Standalone applications are those that have been locally installed on the computer using software such as ClickOnce or Windows Installer (MSI) and which run on the desktop. Standalone applications are considered *full trust* and have full access to a computer's resources.

WPF XBAP deployment

- These programs are hosted inside a web browser.
- Pre-.NET4 XBAP applications run in a *partial trust* sandbox environment, and are not given full access to the computer's resources and not all WPF functionality is available.
- The hosted environment is intended to protect the computer from malicious applications.
- Can run as fully trusted applications in .NET 4, with full access to computer resources. Starting an XBAP from an HTML page or vice versa is seamless (there is no security or installation prompt). Although one gets the perception of the application running in the browser, it actually runs in an out-of-process executable different from the browser.

WPF development tools

- Microsoft VS, Blend, Expresssion Design, XAMLPad
- Third party
  - SharpDevelop, open source
  - PowerBuilder.NET, Sybase