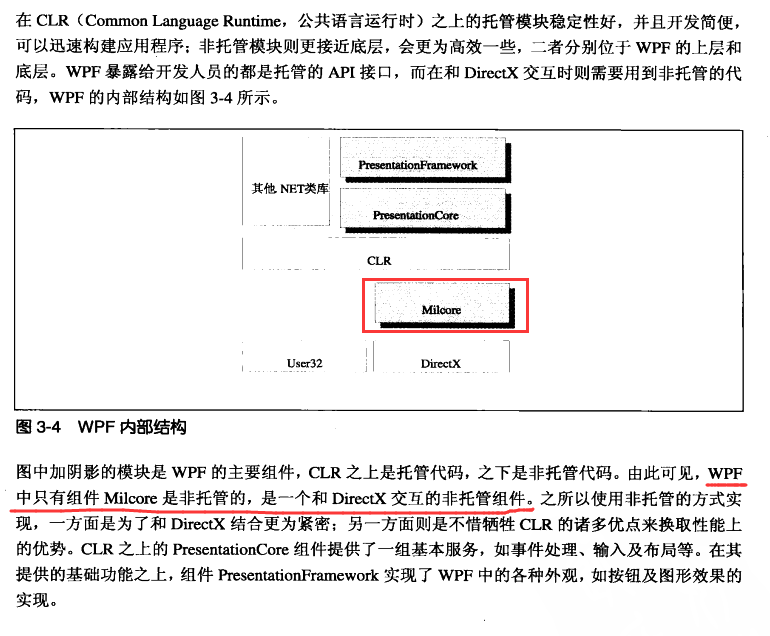
一、WPF内部结构

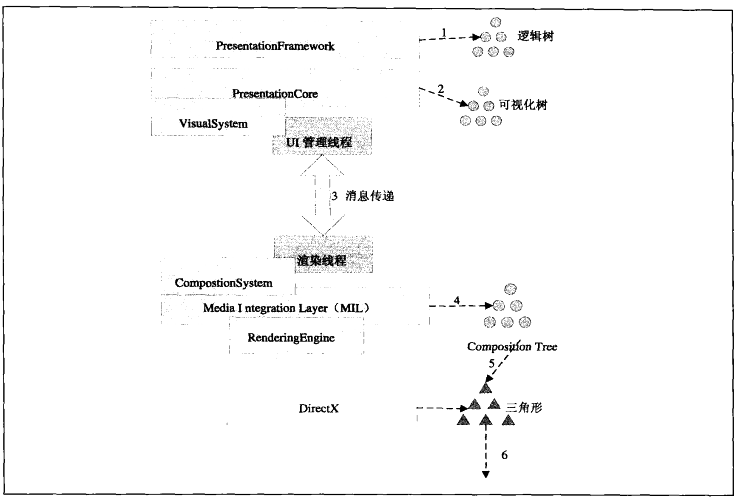


二、WPF如何实现绘制

（1）绘制所需数据结构（逻辑树）:WPF将每一个可视化单元都看做是一个Visual，类似早期程序中的窗口，每个Visual也是自顶向下的树（可视化树，Visual Tree），每个节点都会保存绘制自己所需的命令。PS：构成3D图形的基本图元是一个个的三角形

（2）绘制过程

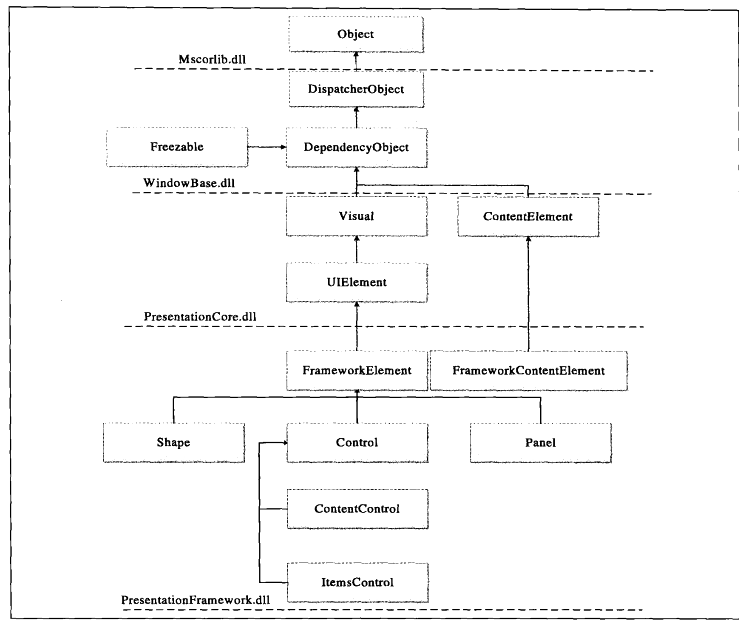
WPF程序从两个线程开始，分别负责UI管理（托管代码，逻辑树、可视化树）和渲染（非托管代码，Composition Tree），二者通过消息传递数据



三、层次结构

（1）WPF的类主要集中在WindowsBase.dll、PresentationCore.dll、PresentationFramework.dll

（2）主要类结构



Object

DispatcherObject（System.Treading.DispatcherObject），多线程时可能用到

DependencyObject（System.Windows.DependencyObject），支持依赖属性的基类

Freezable（System.Windows.Freezable），Changed事件，Freeze方法可使自己只读，通常在两个线程共享或提高效率时，需要将对象“冻结”

Visual（System.Windows.Media.Visual），可视化树的一个节点，为WPF提供可视化支持，包括输出显示、透明度、坐标转换及区域剪切等

UIElement（System.Windows.UIElement），增加了布局、输入和事件

FrameworkElement（System.Windows. FrameworkElement），增加了数据绑定、样式及资源等WPF核心功能

Shape（System.Windows.Shapes.Shape），基本形状的基类，继承自FrameworkElement，因此具备一些元素的特性，如响应鼠标和键盘消息等

Control（System.Windows.Controls.Control），控件的基类。WPF的控件可以支持控件模板，使得控件的外观可以随心所欲的变化。WPF中可视化内容被称为“元素”，对控件模板的支持是判别控件和元素一个重要标准

ContentControl（System.Windows.Controls.ContentControl），单一内容控件，文字、图片及视频等任何类型数据

ItemsControl（System.Windows.Control.ItemsControl），多条目显示控件的基类（列表、树形视图），如菜单、工具栏及状态栏

Panel（System.Windows.Controls.Panel），可用作所有布局的容器，是WPF布局系统的基础，合理使用之，是灵活布局界面内容的关键