MODULE 06

LAYOUTS

MODULE TOPICS

WPF's Layout Philosophy

Measure and Arrange Stages

Alignment and Margin

Border Component

Layout Containers

Nesting Layout Containers

WPF'S LAYOUT PHILOSOPHY

KEY WPF LAYOUT PRINCIPLES

- Elements should not be explicitly sized
  - They should grow to fit their content
  - Use maximum and minimum size instead to influence this

WPF'S LAYOUT PHILOSOPHY

KEY WPF LAYOUT PRINCIPLES

- Elements do not indicate their position
  - Instead they are arranged by their container based on size, order, and other info
  - To add whitespace between elements use the Margin property

WPF'S LAYOUT PHILOSOPHY

KEY WPF LAYOUT PRINCIPLES

- Layout containers share available space among their children
    - They attempt to give each element its preferred size if space is available
    - They can also distribute extra space to one or more children

WPF'S LAYOUT PHILOSOPHY

KEY WPF LAYOUT PRINCIPLES

- Layout containers can be nested
  - A Window can only hold a single element, so usually a Layout Container
  - A typical UI begins with the Grid that contains other containers to arrange smaller groups of elements

THE LAYOUT PROCESS

WPF LAYOUT TAKES PLACE IN 2 STAGES

- Measure Stage - The container loops through its child elements and asks for their preferred size
- Arrange Stage - The container places the child elements in the appropriate position
  - If preferred size is not available the container must truncate the element to fit the visible area
  - Setting minimum window size can help with this
  - ScrollViewer can also help (Controls module)

LAYOUT CONTAINERS

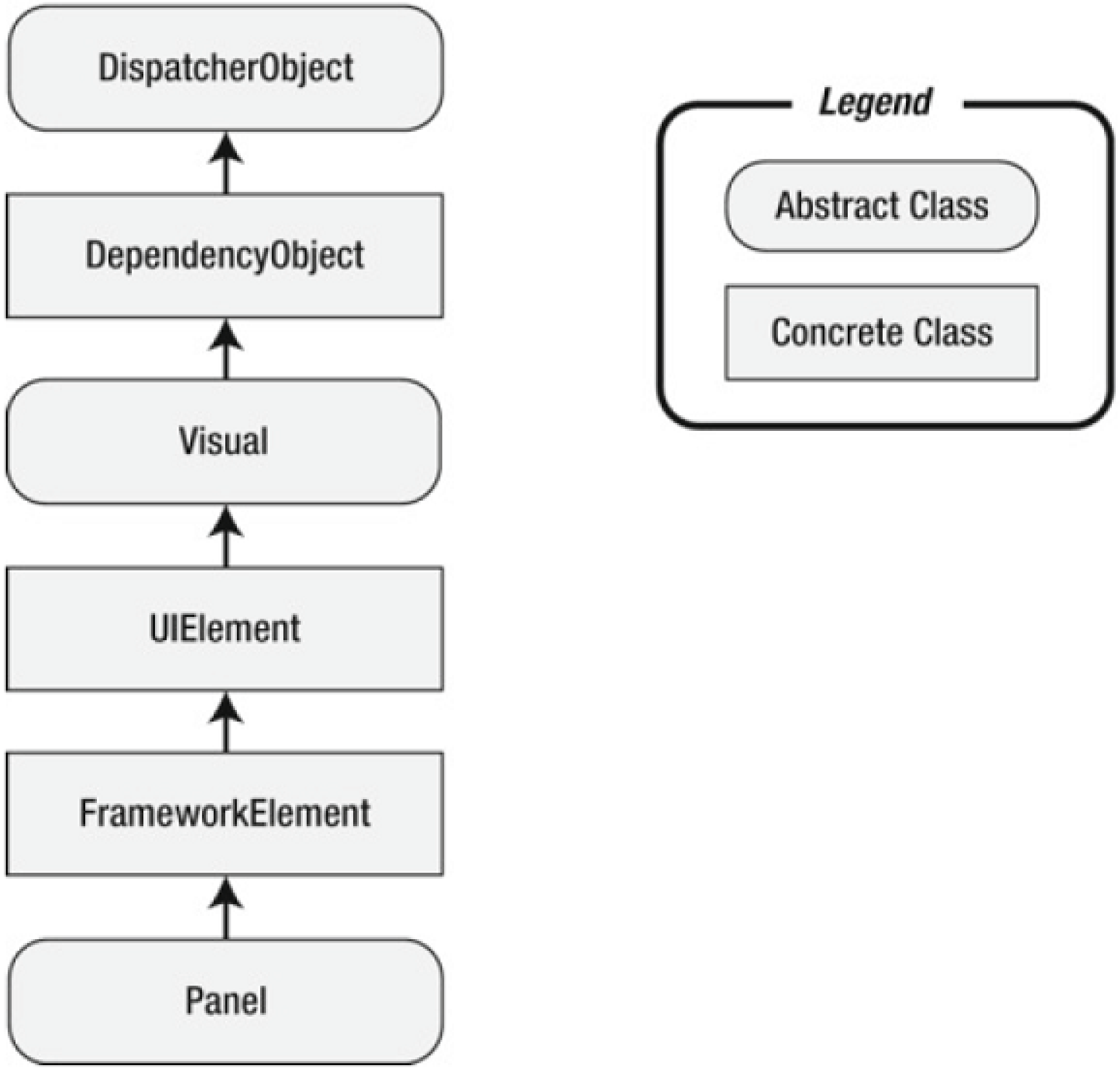

Figure 3-1. *The hierarchy of the Panel class*

CORE LAYOUT PANELS

- **StackPanel** - Places elements in a horizontal or vertical stack
- **WrapPanel** - Places elements in a series of wrapped lines.
  - Horizontal orientation layouts elements left to right then wraps to next row
  - Vertical orientation layouts elements top to bottom then wraps to next column
- **DockPanel** - Aligns elements against the entire edge of the container

CORE LAYOUT PANELS

- **Grid** - Arrange elements in rows and columns according to an invisible table
  - Most flexible and commonly used layout containers
- **UniformGrid** - Places elements in an invisible table but forces all cells to have the same size
- **Canvas** - Allows elements to be positioned absolutely by using fixed coordinates

WALKTHRUS - LAYOUT PANELS

ANY QUESTIONS?