LẬP TRÌNH WPF

(WINDOWS PRESENTATION FOUNDATION)

Nội dung

1	Tổng quan về WPF
2	Cách tạo một ứng dụng WPF
3	Giới thiệu XAML
4	Layout Panels trong WPF
5	Các controls trong WPF

Tổng quan về WPF

WPF là gì?

- Là hệ thống API tiên tiến được Microsoft phát triển nhằm hỗ trợ việc xây dựng giao diện đồ họa trên nền tảng Windows. Được giới thiệu lần đầu năm 2006 trong .NET Framework 3.0 (ban đầu có tên là Avalon)
- Được xem như thế hệ kế tiếp của WinForms, nâng cao đáng kể khả năng lập trình giao diện bằng cách cung cấp các API tận dụng tối đa lợi thế của đa phương tiện hiện đại
- Là một bộ phận của .NET Framework, được tích hợp trong tất cả các hệ điều hành Windows kể từ Windows Vista và Windows Server 2008
- WPF sử dụng hai thư viện lõi quan trọng là PresentationCore và PresentationFramework để xử lý các điều hướng, ràng buộc dữ liệu, sự kiện và quản lý giao diện
- Nền tảng đồ họa của WPF được xây dựng trên DirectX, cho phép xử lý vector, hỗ trợ gam màu rộng, và tạo hiệu ứng hình ảnh 2D/3D một cách dễ dàng

Tổng quan về WPF

Ba mục tiêu cốt lõi của WPF:

1) Cung cấp một nền tảng thống nhất để xây dựng giao diện người dùng

2) Cho phép người lập trình và người thiết kế giao diện làm việc cùng nhau một cách dễ dàng

3) Cung cấp một công nghệ chung để xây dựng giao diện người dùng trên cả ứng dụng Windows và ứng dụng Web

Tổng quan về WPF

Cung cấp một nền tảng thống nhất để xây dựng giao diện người dùng:

- Trước khi WPF ra đời, việc tạo giao diện ứng dụng phức tạp đòi hỏi lập trình viên phải sử dụng nhiều công nghệ riêng biệt. Để tạo form và các control, họ dùng Windows Forms. Để hiển thị văn bản phức tạp, họ có thể phải dùng thêm các công nghệ như Adobe PDF. Với đồ họa 2D, họ cần GDI+, trong khi đồ họa 3D lại đòi hỏi Direct3D. Để phát âm thanh và video, họ phải tích hợp Windows Media Player.
- WPF đã thay đổi hoàn toàn cách tiếp cận này bằng cách hợp nhất tất cả công nghệ cần thiết vào một nền tảng duy nhất. Điều này giúp giảm đáng kể độ phức tạp trong quá trình phát triển và đơn giản hóa việc tạo ra các giao diện người dùng hiện đại với nhiều thành phần trực quan

Tổng quan về WPF

Cho phép người lập trình và người thiết kế giao diện làm việc cùng nhau một cách dễ dàng:

- Trong thực tế, việc xây dựng giao diện người dùng phức tạp đòi hỏi sự kết hợp giữa kỹ năng của người thiết kế giao diện và lập trình viên. Trước đây, quá trình này thường gặp nhiều khó khăn do thiếu một ngôn ngữ chung.
- WPF giải quyết vấn đề này bằng cách đưa ra ngôn ngữ XAML (eXtensible Application Markup Language). XAML là một ngôn ngữ đánh dấu dựa trên XML, cho phép định nghĩa chính xác diện mạo của giao diện người dùng với các phần tử như Button, TextBox, Label... Mỗi phần tử XAML tương ứng với một lớp WPF, và mỗi thuộc tính của phần tử đó tương ứng với thuộc tính hay sự kiện của lớp này

Tổng quan về WPF

Cung cấp một công nghệ chung để xây dựng giao diện người dùng trên cả ứng dụng Windows và ứng dụng Web:

- WPF còn cho phép lập trình viên tạo ra ứng dụng trình duyệt XBAP (XBAP – XAML Browser Application) có thể chạy trên Internet Explorer. Điều đặc biệt là cùng một codebase có thể được sử dụng để tạo ra cả ứng dụng desktop độc lập và ứng dụng web, giúp tiết kiệm thời gian và nguồn lực phát triển.
- Một ứng dụng XBAP được tải xuống từ web server khi cần thiết, nên nó phải tuân thủ các yêu cầu bảo mật nghiêm ngặt hơn so với ứng dụng desktop. XBAP chạy trong một sandbox bảo mật do hệ thống an ninh của .NET Framework cung cấp, đảm bảo an toàn cho người dùng

Tổng quan về WPF

Các tính năng nổi bật của WPF:

1. Rendering bằng DirectX – Hiệu năng cao

- WPF sử dụng **Direct3D** để render giao diện.
- Tận dụng **GPU**, không phụ thuộc CPU như WinForms.
- Hiển thị mượt hơn, đặc biệt với UI phức tạp, animation, video, 3D.

2. Hỗ trợ đồ họa vector & độc lập DPI

- UI được vẽ bằng **vector**, không phải bitmap.
- Phóng to / thu nhỏ không bị mờ, đặc biệt trên màn hình 2K – 4K.
- Giao diện sắc nét ở mọi độ phân giải

Tổng quan về WPF

Các tính năng nổi bật của WPF:

3. Data Binding mạnh

- Có hệ thống binding mạnh nhất trong các công nghệ .NET UI: OneWay, TwoWay, OneTime, OneWayToSource
- Binding với object, collection, hierarchical data
- Hỗ trợ INotifyPropertyChanged, ObservableCollection

4. Hệ thống Style, Template và Resource rất mạnh

WPF cho phép:

Style: đồng bộ hóa giao diện toàn ứng dụng

ControlTemplate: thay đổi hình dáng control *hoàn toàn*

DataTemplate: tùy biến cách hiển thị dữ liệu

ResourceDictionary: quản lý theme chuyên nghiệp

Tổng quan về WPF

Các tính năng nổi bật của WPF:

5. Animation & Effects được tích hợp sẵn

Gồm:

- Storyboard animation
- Color animations, double animations
- Keyframe animation
- Blur, DropShadow, OpacityMask, BitmapEffects

6. Hỗ trợ 3D Rendering

- Hỗ trợ mô hình 3D: Model3D, MeshGeometry3D
- Kết hợp với ánh sáng, camera
- Dễ chèn 3D vào giao diện 2D

Tổng quan về WPF

Các tính năng nổi bật của WPF:

8. Tích hợp tài liệu (Documents)

Hỗ trợ:

- FlowDocument
- FixedDocument
- XPS Document

9. Tích hợp Multimedia

- Video: MediaElement
- Âm thanh, hình ảnh, overlay video
- Bổ trợ cho ứng dụng trình chiếu và giải trí

Tổng quan về WPF

Các tính năng nổi bật của WPF:

10. Hỗ trợ tốt MVVM (Model – View – ViewModel)

- Binding mạnh
- Command framework
- Update UI tự động khi dữ liệu thay đổi

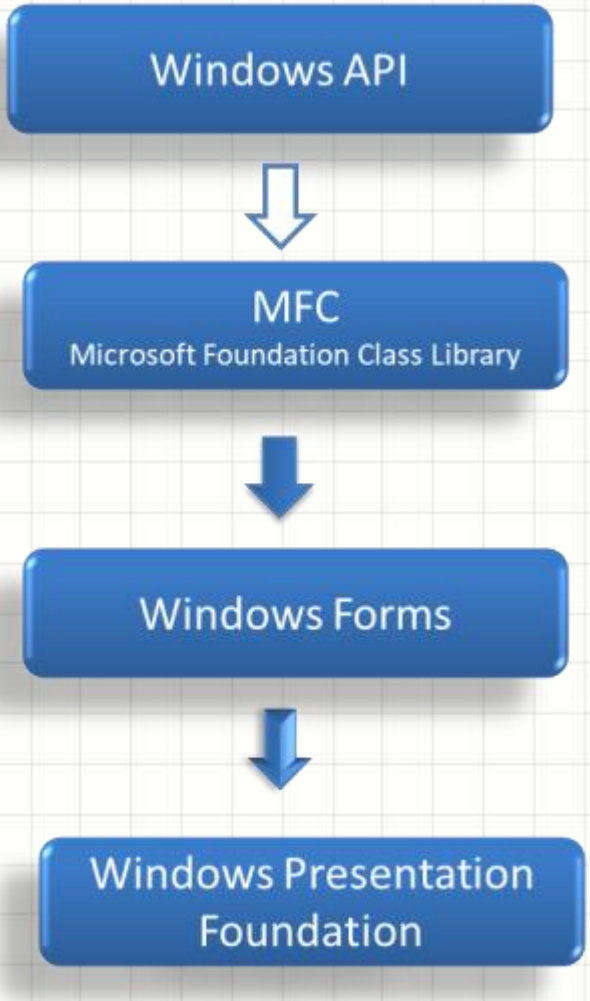
11. Layout System linh hoạt

Không giống WinForms kéo-thả pixel cố định, WPF có:

- Grid
- StackPanel
- WrapPanel
- DockPanel
- Canvas
- UniformGrid

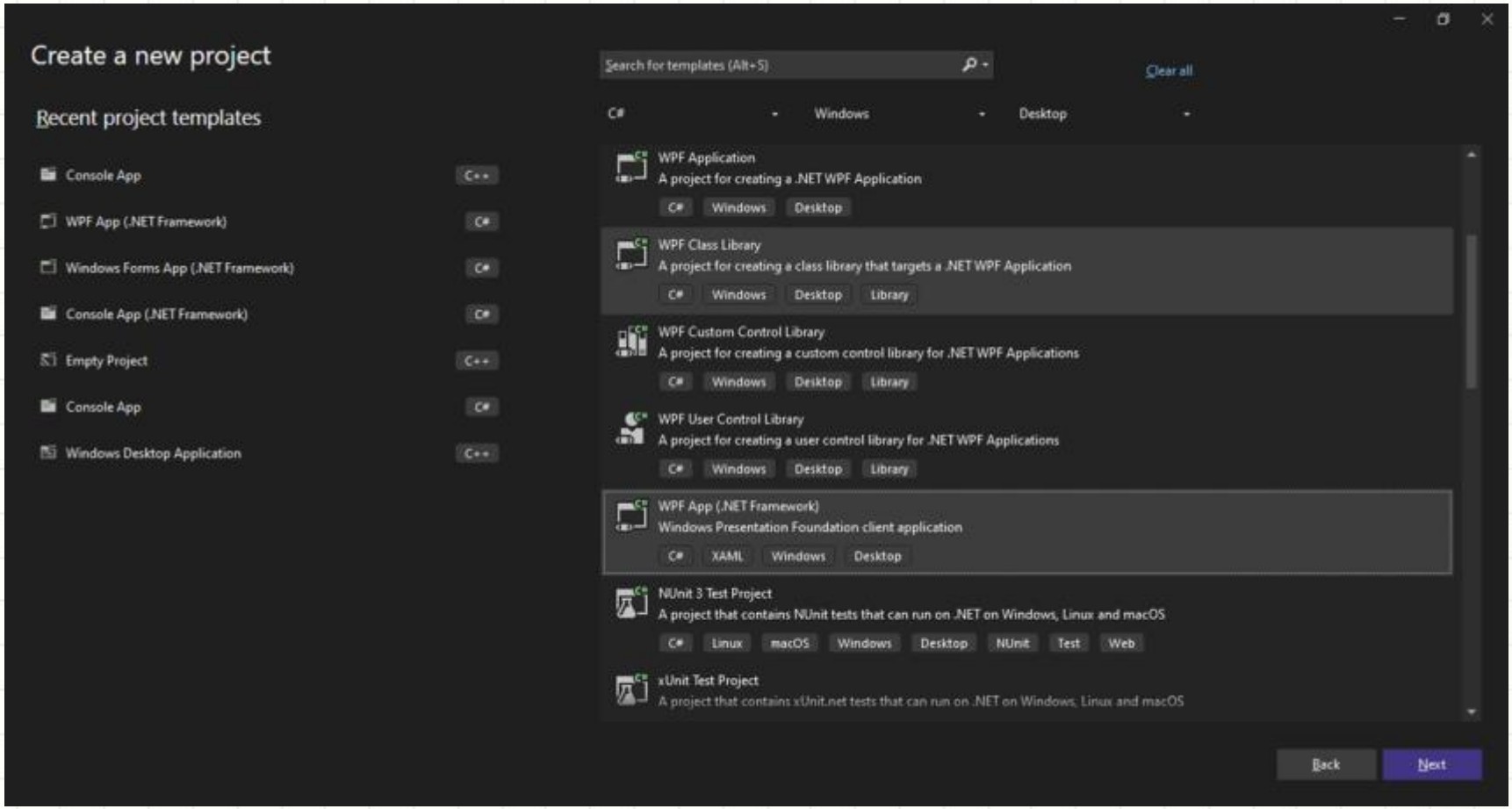
Tổng quan về WPF

Con đường phát triển công nghệ WPF:



Cách tạo ứng dụng WPF

Mở VS và chọn như hình vẽ:



Cách tạo ứng dụng WPF

Có 6 cửa sổ làm việc cần biết:

1) Canvas
2) XAML
3) ToolBox (Controls)
4) Solution Explorer
5) Code-behind
6) Properties

Demo trên máy

XAML

- XAML (Extensible Application Markup Language) là một ngôn ngữ đánh dấu với cú pháp tương tự XML dùng để thể hiện các đối tượng trong .NET.
- Vai trò chính của XAML dùng để xây dựng giao diện người dùng WPF. Nói cách khác, XAML documents sẽ định nghĩa cách sắp xếp, thể hiện các control, buttons trong cửa sổ của một chương trình WPF

XAML

Ưu điểm của XAML:

- XAML có mã ngắn, rõ ràng dễ đọc.
- Tách mã thiết kế và logic.
- Việc dùng XAML cho phép tách tách biệt rõ ràng vai trò của nhà thiết kế và nhà phát triển.

XAML

XAML vs C#

```
<Button Background="Red">
        Click Me!
</Button>
```

```
Button button1 = new Button();
button1.Content = "Click Me!";
button1.Background = new SolidColorBrush(
        Colors.Red);
```

XAML

Cú pháp của XAML:

- Mọi thành phần (elements) trong XAML document đều là một thể hiện của một lớp nào đó trong .NET. Tên của các elements này hoàn toàn giống với tên của các class. Ví dụ như thẻ <Button> trong WPF sẽ tạo ra một đối tượng thuộc lớp Button.
- Cũng như các tài liệu theo chuẩn XML khác, có thể gom (nest) một thẻ đặt bên trong một thẻ khác.
- Có thể thiết lập các property của mỗi class thông qua các attribute (thuộc tính).

XAML

Cú pháp của XAML:

- XML Namespaces <--> .NET Namespaces
- XML Elements <--> .NET Objects
- XML Attributes <--> .NET Object Properties

XAML

Có 3 loại top-level element:

- Window
- Page (tương tự như Window nhưng dùng để chuyển đổi giữa các ứng dụng
- Application (định nghĩa các application resources và thiết lập khởi động)

Ví dụ:

```
<Window x:Class="HelloWord.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
    </Grid>
</Window>
```

XAML

- XML namespace được khai báo như cách khai báo attributes.
- Các attribute này có thể được đặt bên trong bất cứ thẻ bắt đầu nào. Tuy nhiên, quy tắc là tất cả các namespace được sử dụng trong tài liệu nên được khai báo trong thẻ đầu tiên.
- Một khi đã khai báo namespace rồi, có thể sử dụng nó bất kì đâu trong tài liệu.

Ví dụ:

```
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation
" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
```

XAML

- http://schemas.microsoft.com/winfx/2006/xaml/presentation là core namespace của WPF. Nó chứa tất cả các class trong WPF, bao gồm các controls dùng để xây dựng giao diện. Nếu namespace này được khai báo không có prefix, như vậy, nó trở thành default namespace cho toàn bộ tài liệu. Nói cách khác, mọi element đều được tự động đặt trong namespace này, ngoại trừ một vài khai báo đặc biệt khác.
- http://schemas.microsoft.com/winfx/2006/xaml/ là XAML namespace. Nó báo gồm các tính năng khác nhau của XAML cho phép can thiệp vào cách biên dịch tài liệu. Namespace này được khai báo với prefix x. Nghĩa là có thể áp dụng nó bằng cách đặt namespace prefix trước tên của thẻ (<x:ElementName>).