

Bài: Command Implementing trong Lập trình WPF

Xem bài học trên website để ủng hộ Kteam: [Command Implementing trong Lập trình WPF](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Sức mạnh của hệ điều hành Window là không thể chối cãi. Và để tạo nên sức mạnh đó không thể thiếu những ứng dụng mạnh mẽ. Vậy để tạo ra những ứng dụng đó, người lập trình viên cần học cái gì? [Lập trình Winform](#) thật sự rất mạnh. Nhưng vẫn còn những giới hạn. Vậy cùng nhau vượt qua các giới hạn đó với [Lập trình WPF](#).

Nội dung

[Tổng quan lập trình WPF](#)

[Download project](#)

MainWindow.xaml.cs

C#:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Command___Implementing
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
        private void ExitCommand_CanExecute(object sender, CanExecuteRoutedEventArgs e)
        {
            e.CanExecute = true;
        }

        private void ExitCommand_Executed(object sender, ExecutedRoutedEventArgs e)
        {
            Application.Current.Shutdown();
        }
    }

    public static class CustomCommands
    {
        public static readonly RoutedUICommand Exit = new RoutedUICommand
            (
                "Exit",
                "Exit",
                typeof(CustomCommands),
                new InputGestureCollection()
                {
                    new KeyGesture(Key.A, ModifierKeys.Alt)
                }
            );

        //Define more commands here, just like the one above
    }
}

```

MainWindow.xaml**xml:**

```
<Window x:Class="Command___Implementing.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:self="clr-namespace:Command___Implementing"
        Title="MainWindow" Height="350" Width="525">
    <Window.CommandBindings>
        <CommandBinding Command="self:CustomCommands.Exit"
                        CanExecute="ExitCommand_CanExecute"
                        Executed="ExitCommand_Executed" />
    </Window.CommandBindings>
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <Menu>
            <MenuItem Header="File">
                <MenuItem Command="self:CustomCommands.Exit" />
            </MenuItem>
        </Menu>

        <StackPanel Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center">
            <Button Command="self:CustomCommands.Exit">Exit</Button>
        </StackPanel>
    </Grid>
</Window>
```

Kết luận

Bài sau chúng ta sẽ cùng nhau tìm hiểu về [UpdateSource Trigger trong lập trình WPF](#) nhé.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.