

# Bài: Command trong Lập trình WPF

Xem bài học trên website để ủng hộ Kteam: [Command 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___WPF
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
        #region 1
        //private void NewCommand_CanExecute(object sender, CanExecuteRoutedEventArgs e)
        //{
        //    e.CanExecute = true;
        //}

        //private void NewCommand_Executed(object sender, ExecutedRoutedEventArgs e)
        //{
        //    MessageBox.Show("The New command was invoked");
        //}
        #endregion

        #region 2
        //private void CutCommand_CanExecute(object sender, CanExecuteRoutedEventArgs e)
        //{
        //    e.CanExecute = (txtEditor != null) && (txtEditor.SelectionLength > 0);
        //}

        //private void CutCommand_Executed(object sender, ExecutedRoutedEventArgs e)
        //{
        //    txtEditor.Cut();
        //}

        //private void PasteCommand_CanExecute(object sender, CanExecuteRoutedEventArgs e)
        //{
        //    e.CanExecute = Clipboard.ContainsText();
        //}

        //private void PasteCommand_Executed(object sender, ExecutedRoutedEventArgs e)
        //{
        //    txtEditor.Paste();
        //}
        #endregion
    }
}
```

MainWindow.xaml

xml:

```

<Window x:Class="Command__WPF.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">
    <!--<Window.CommandBindings>
        <CommandBinding Command="ApplicationCommands.New"
                        Executed="NewCommand_Executed"
                        CanExecute="NewCommand_CanExecute" />
    </Window.CommandBindings>

    <StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
        <Button Command="ApplicationCommands.New">New</Button>
    </StackPanel>-->

    <!--<Window.CommandBindings>
        <CommandBinding Command="ApplicationCommands.Cut" CanExecute="CutCommand_CanExecute" Executed="CutCommand_Executed" />
        <CommandBinding Command="ApplicationCommands.Paste" CanExecute="PasteCommand_CanExecute" Executed="PasteCommand_Executed"
    />
</Window.CommandBindings>
<DockPanel>
    <WrapPanel DockPanel.Dock="Top" Margin="3">
        <Button Command="ApplicationCommands.Cut" Width="60">_Cut</Button>
        <Button Command="ApplicationCommands.Paste" Width="60" Margin="3,0">_Paste</Button>
    </WrapPanel>
    <TextBox AcceptsReturn="True" Name="txtEditor" />
</DockPanel>-->

<DockPanel>
    <WrapPanel DockPanel.Dock="Top" Margin="3">
        <Button Command="ApplicationCommands.Cut" CommandTarget="{Binding ElementName=txtEditor}" Width="60">_Cut</Button>
        <Button Command="ApplicationCommands.Paste" CommandTarget="{Binding ElementName=txtEditor}" Width="60"
Margin="3,0">_Paste</Button>
    </WrapPanel>
    <TextBox AcceptsReturn="True" Name="txtEditor" />
</DockPanel>
</Window>

```

### Kết luận

Bài sau chúng ta sẽ cùng nhau tìm hiểu về [Command Implementing 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"**.