



Windows App SDK



Buy me a coffee

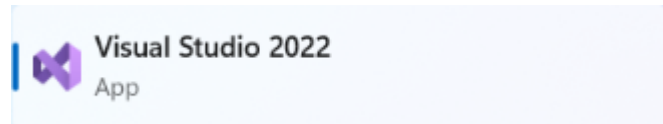
Hello Input

Hello Input shows you how to use a **ContentDialog** with a **TextBox** that you can type into and show that text in another **ContentDialog** using the **Windows App SDK**.

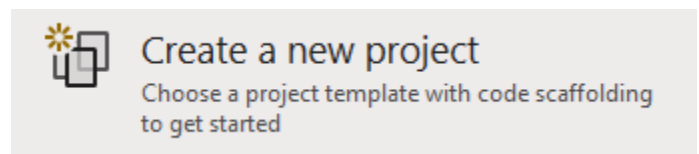
Step 1

Follow **Setup and Start** on how to get **Setup** and **Install** what you need for **Visual Studio 2022** and **Windows App SDK**.

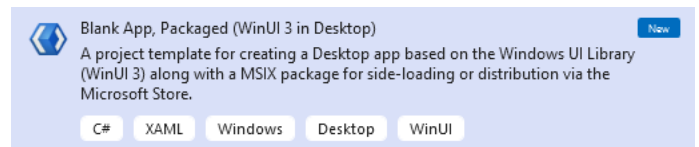
In **Windows 11** choose **Start** and then find or search for **Visual Studio 2022** and then select it.



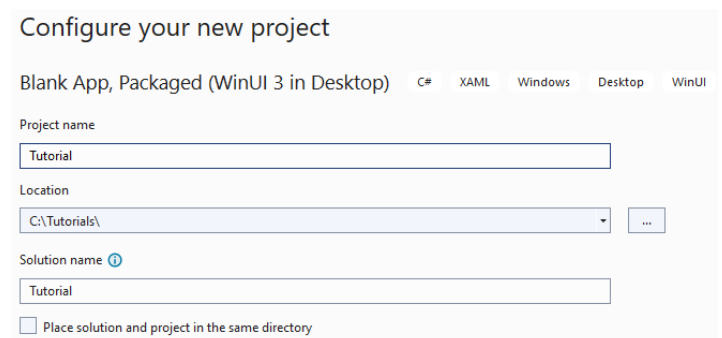
Once **Visual Studio 2022** has started select **Create a new project**.



Then choose the **Blank App, Packages (WinUI in Desktop)** and then select **Next**.

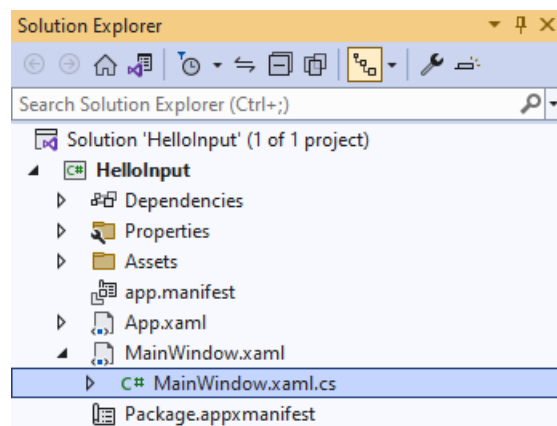


After that in **Configure your new project** type in the **Project name** as *HelloInput*, then select a Location and then select **Create** to start a new **Solution**.



Step 2

Within **Solution Explorer** for the **Solution** select the arrow next to **MainWindow.xaml** then double-click on **MainWindow.xaml.cs** to see the **Code** for the **Main Window**.



Step 3

In the **Code** for **MainWindow.xaml.cs** there will already be a **Method** of **myButton_Click(...)** and within this the following **Line** should be **Removed**:

```
myButton.Content = "Clicked";
```

Step 4

Then in **myButton_Click(...)** where **myButton.Content = "Clicked";** was **Removed** type in the following:

```
TextBox input = new()
{
    PlaceholderText = "Display Text"
};
ContentDialog dialog = new()
{
    XamlRoot = Content.XamlRoot,
    PrimaryButtonText = "Display",
    SecondaryButtonText = "Close",
    Title = "Hello Input",
    Content = input
};
ContentDialogResult result = await dialog.ShowAsync();
if(result == ContentDialogResult.Primary)
{
    dialog.Content = (dialog.Content as TextBox).Text;
    dialog.PrimaryButtonText = string.Empty;
    await dialog.ShowAsync();
}
```

This will create **TextBox** which will be used to type in what should be display, then there is an **Instance** of a **ContentDialog** with the **Content** of set to the **TextBox** it also has a **Title** set to the text of *Hello Input* along with having the **PrimaryButtonText** set to *Display* and the **SecondaryButtonText** set to *Close*, which when selected will **Close** the **ContentDialog**. When *Display* is **Clicked** this will produce the **ContentDialogResult.Primary** value. Should this be the case, the **Content** will be set to the **Property** of **Text** of **TextBox** which will be the contents of what was typed into it. The **PrimaryButtonText** is reset as this option is not needed when the **ContentDialog** is shown again. To show the **ContentDialog** the **Method** for **ShowAsync** is used which uses the **Keyword** for **await** which means it will perform a **Task** that won't happen at the same time, or **Asynchronously** and **XamlRoot** is also set to allow the **ContentDialog** to work correctly. A good concept shown here is of **Code Reuse**, as it is possible to use the same **Instance** of the **ContentDialog** to display both Messages.

Step 5

While still in the **Method** for `myButton_Click(...)` between **private** and **void** type in the following:

```
async
```

Because the **Method** for `ShowAsync` is **Asynchronous** using the **Keyword** of `await` so you need to mark the **Method** it is used in as such, this done with the **Keyword** of `async`.

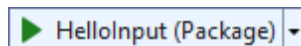
The **Method** for `myButton_Click(...)` should look as follows:

```
private async void myButton_Click(object sender, RoutedEventArgs e)
{
    TextBox input = new()
    {
        PlaceholderText = "Display Text"
    };
    ContentDialog dialog = new()
    {
        XamlRoot = Content.XamlRoot,
        PrimaryButtonText = "Display",
        SecondaryButtonText = "Close",
        Title = "Hello Input",
        Content = input
    };
    ContentDialogResult result = await dialog.ShowAsync();
    if(result == ContentDialogResult.Primary)
    {
        dialog.Content = (dialog.Content as TextBox).Text;
        dialog.PrimaryButtonText = string.Empty;
        await dialog.ShowAsync();
    }
}
```

When the **Button** is **Clicked**, the **Method** of `myButton_Click(...)` will be triggered and this display a **ContentDialog** with the **Content** set to a **TextBox** and the **Title** set to *Hello Input*.

Step 6

That completes the **Windows App SDK** Application. In **Visual Studio 2022** from the **Toolbar** select **HelloInput (Package)** to **Start** the Application.



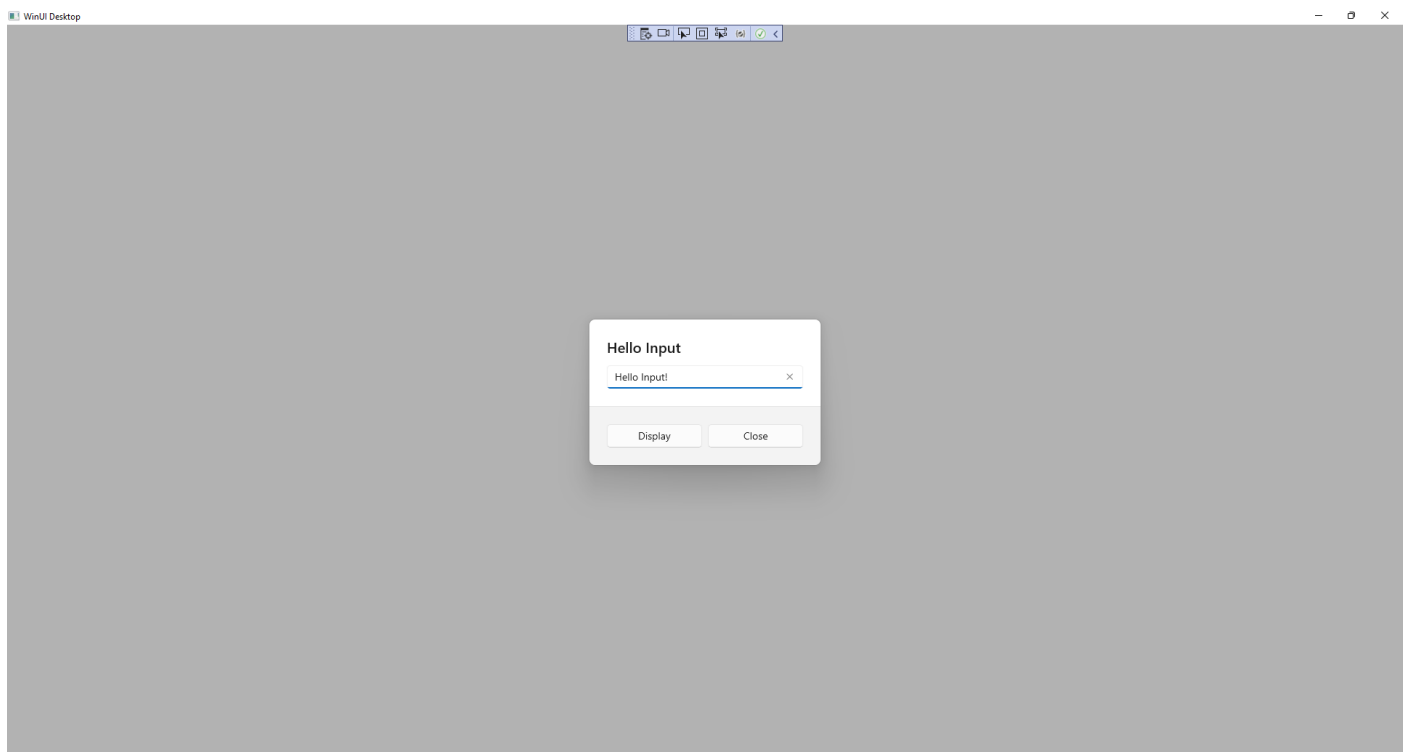
Step 7

Once running you should see the **Button** with the Text *Click Me*

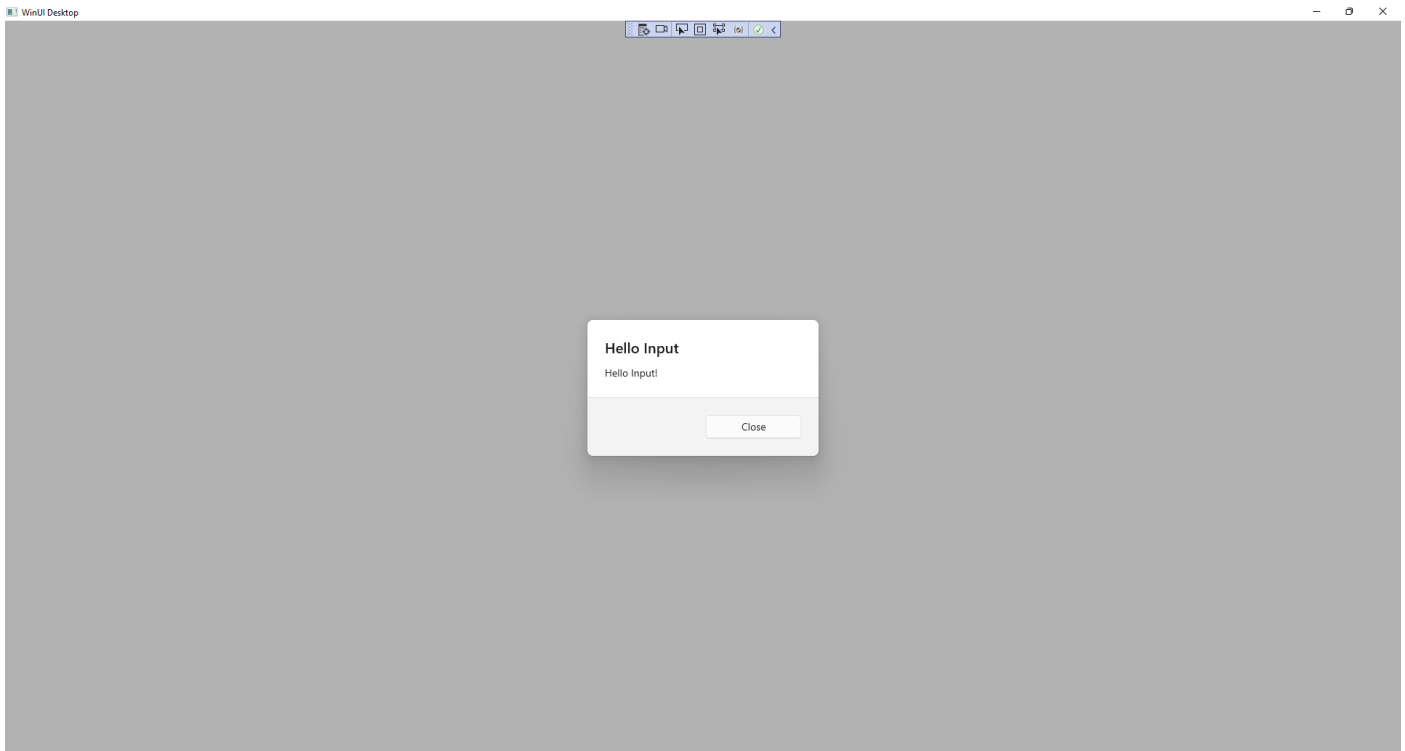


Step 8

If you **Click** on the **Button** with the Text *Click Me* it will display the **ContentDialog** which you can then type anything into the **TextBox** or you can dismiss it with the **Button** for *Close*.



When you **Click** on the **Button** for *Display* whatever you typed in will displayed in a **ContentDialog** which you can then dismiss with the **Button** for *Close*.



Step 9

To **Exit** the **Windows App SDK** Application, select the **Close** button from the top right of the Application as that concludes this **Tutorial** for **Windows App SDK** from tutorialr.com!

