Scripts Tasks and Classes Example

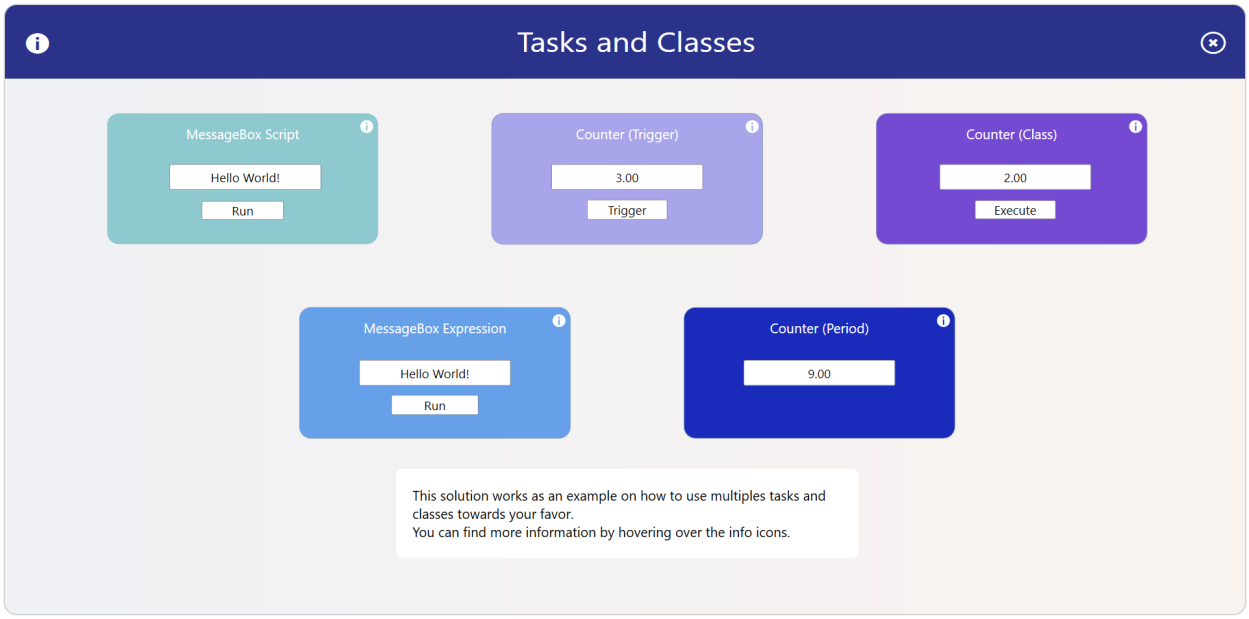
Work with script tasks and classes to create custom logic.

Download the solution Scripts Tasks and Classes Example.dbsln

This example shows how to use Tasks and Classes to create custom logic.

Summary

This solution example shows the basic concepts on using Tasks and Classes to create custom logic in your solution.



Technical Information

The platform is fully programmed using managed C# or Visual Basic code within the Microsoft .NET framework, enabling the creation of scripts that execute on a schedule or in response to specific events. Tasks, written in VB.NET or C#, are triggered either by specific events or at designated intervals, and can be configured under Scripts / Tasks. Additionally, Classes provide a repository for reusable class libraries, methods, and functions, which can be accessed by Tasks, other Classes, or displays, streamlining application-wide functionality.

This solution example demonstrates five different ways to access scripts or tasks.
The first one shows a message box popup using a script.



The button above runs a script in the CodeBehind. The code in the CodeBehind uses the `@Script.Class` method to call a class that was previously set up. The code snippet used in the CodeBehind can be found below:

```
Public Sub MouseLeftButtonDown1(ByVal sender As Object, ByVal e As System.Windows.Input.MouseButtonEventArgs)
        ' You can find the logic related to "MyClass" on Scripts ? Classes ? MyClass
        @Script.Class.MyClass.Hello(@Tag.sayit)
End Sub
```

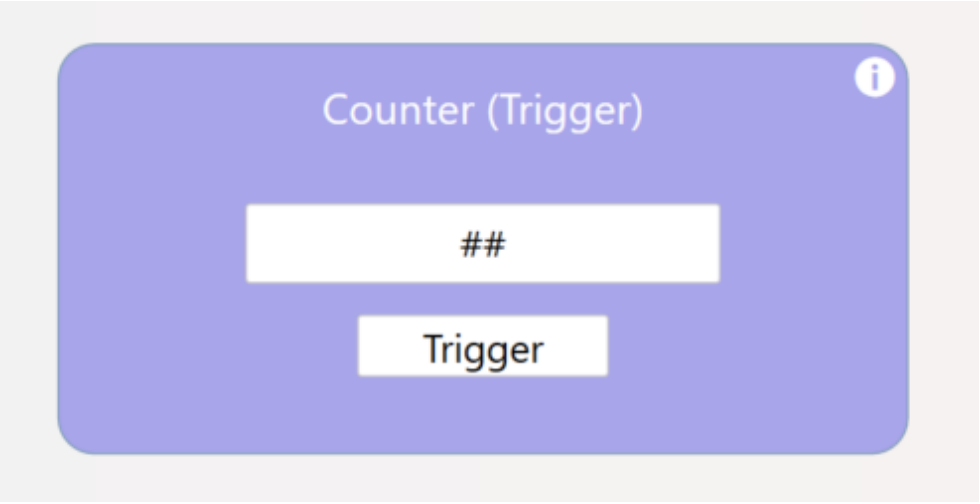
For the second button, it calls a class through expressions written in VB.NET.



The way it works is through “Run Expression” in the Action tab in the Dynamics window, and it calls the function in a similar way as before:

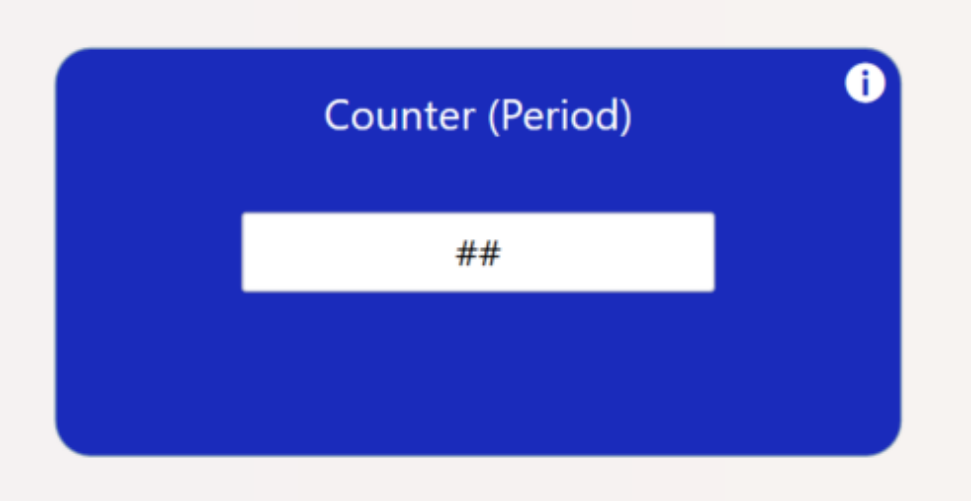
```
Script.Class.MyClass.Hello2(Tag.sayit)
```

The third button works through the use of a trigger.



This trigger is configured in the Tasks page to call a method, and through this intermediary, the counter is incremented by one when the task is called.

The fourth panel does not have a button. However, it uses a trigger that is triggered every second. It is also set up in the Tasks page.



Lastly, the Counter (Class) panel also works as a counter, but its implementation is done through a class that passes the current value as a parameter.



After receiving the parameter, it increments the tag and updates the display.

Along with that information, the solution example also contains information boxes that can lead you to the exact parts of the code where they get triggered. These information boxes can be found in each individual panel, and clicking on them will provide you with the desired information.

In this section...