

Creating an Event Viewer



Jeff Adkin

PLURALSIGHT AUTHOR

@JeffAdkin www.JAdkin.com



Introduction

Create UI

Create
On_Click
Functions

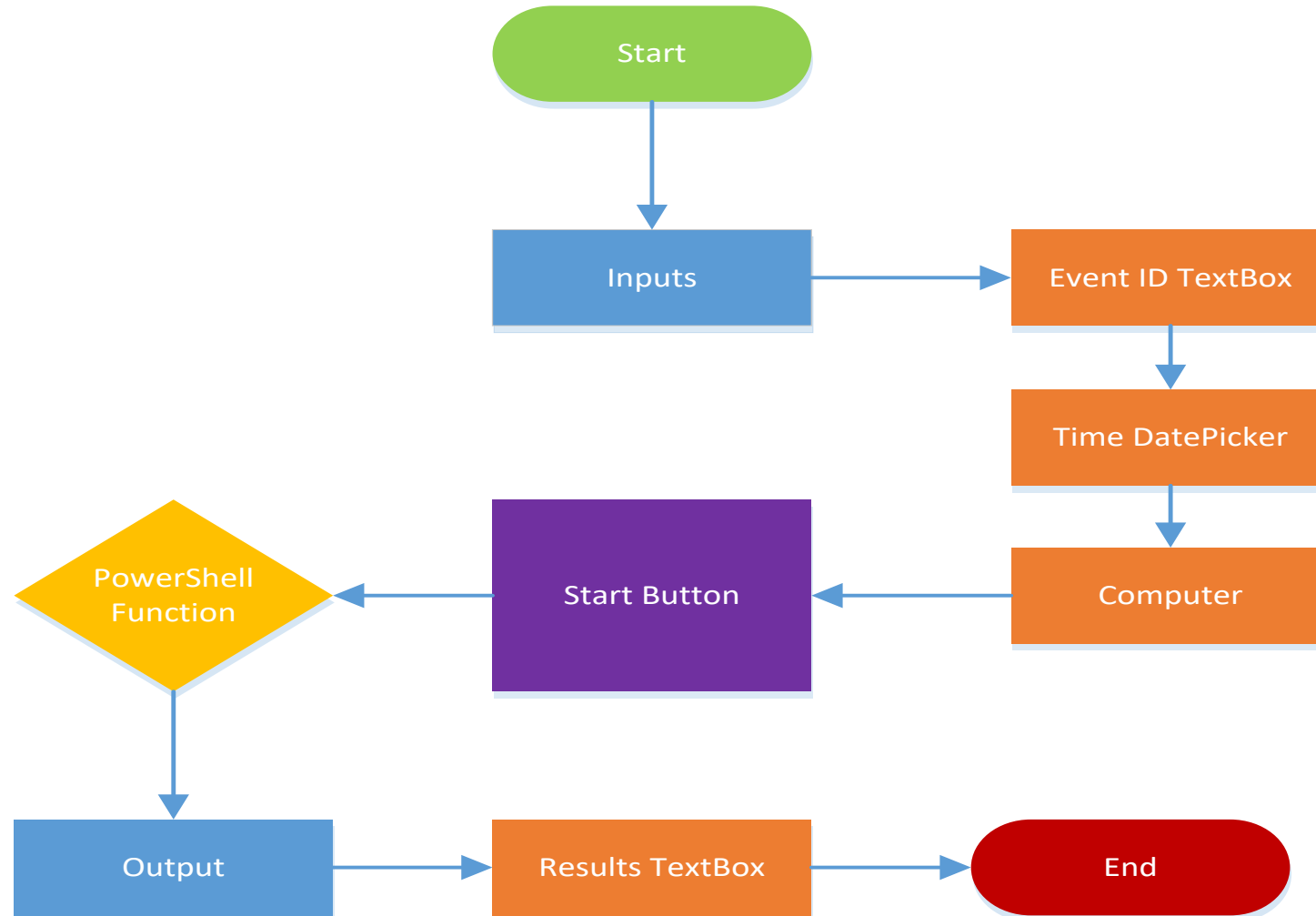
Create
PowerShell
Functions

Running
the Code



Program Flow Chart

Event Log GUI



Create the UI



```
[xml]$Form = @"
```

```
<Window
```

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

```
    Title="Event Viewer" Height="350" Width="525"
```

```
    Background="#FF262626">
```

Set the Window

For this example, we will set the Window to be 350X525 with a Dark Grey background.



<Grid>

Add the Grid

In this example we will use the Grid as one whole section just like the Canvas. To do this we simply put <Grid> with no other information.



```
<TextBox Name="ID" HorizontalAlignment="Left" Height="25" Margin="79,10,0,0" TextWrapping="Wrap"
VerticalAlignment="Top" Width="155" BorderBrush="#FFF96816" />
```

```
<TextBox Name="Comp" HorizontalAlignment="Left" Height="26" Margin="79,41,0,0" TextWrapping="Wrap" Text=" "
VerticalAlignment="Top" Width="155" BorderBrush="#FFF96816" />
```

```
<DatePicker Name="Date" HorizontalAlignment="Left" Height="27" Margin="79,72,0,0" VerticalAlignment="Top" Width="155" />
```

```
<TextBox Name="Results" HorizontalAlignment="Left" Height="156" Margin="10,141,0,0" TextWrapping="Wrap"
VerticalAlignment="Top" Width="495" BorderBrush="#FFF96816" />
```

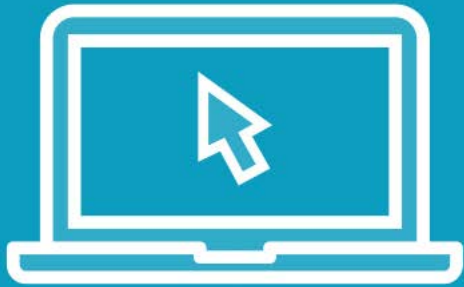
```
<Button Name="Start" Content="START" HorizontalAlignment="Left" Height="89" Margin="279,10,0,0" VerticalAlignment="Top"
Width="226" Background="#FFF96816" Foreground="#FFDFDFD" FontSize="36" FontWeight="Bold" />
```

Add the Controls

For this Program, we are going to need 5 Controls. 3 Inputs, 1 Button and 1 Output.



Demo



Create Our UI



Create On_Click Functions



```
$start = $Win.FindName("Start")
```

```
$eid = $win.FindName("EID")
```

```
$date = $win.FindName("Date")
```

```
$comp = $win.FindName("Comp")
```

```
$res = $win.FindName("Results")
```

Create Variables for Each Input

To create a variable to hold the information of the control we need to find the control by its NAME.



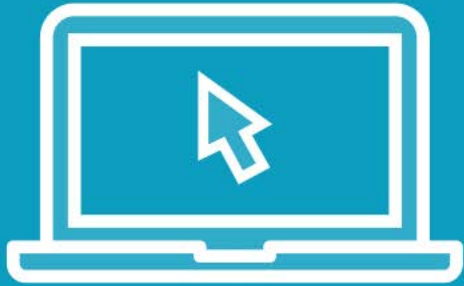
```
$start.Add_Click({  
$event=$eid.text  
$computer = $comp.text  
$da = $date.SelectedDate  
$d = Get-date $da  
Check-Event $event $computer $d})
```

Sending Data to Controls

The On_Click function is only ran when the event is triggered. In this example we are creating variables for the Input information to send to the PowerShell Check-Event function.



Demo



Add in the On_Click Functions



Create PowerShell Functions



```
Function Check-Event($ev,$co,$da){  
  
$comp=$co.trim()  
  
$re = Get-EventLog Application -ComputerName "$comp" -InstanceId $ev -  
After $da  
  
$res.text += "There are "+$re.count+" events of EventID $ev on computer  
$comp after the time $da `r`n"}  

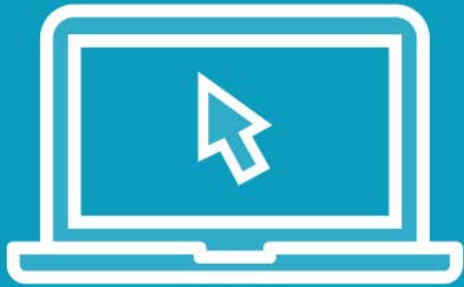
```

Get-EventLogs Function

Our Function has the values sent to it from the Inputs for EventID, Computer and Date. The Computer name is trimmed to make sure we have no white spaces and the Returns are written directly to the Results TextBox.



Demo



Add in the PowerShell Functions



Running the Code



Shoring up our Code

If we run the UI but enter no values we return **Errors**

The Check-Event function has 3 key pieces that are required. For our program to work better we need to ensure that all 3 controls have inputted values.



```
If($event -eq "" -or $computer -eq "" -or $da -eq $null){  
  
}else{  
  
Check-Event $event $computer $d  
  
}
```

Blank/Null Values

In the above code we make sure that none of the controls equal a blank or a null value. If they do not then we Check-Event, but if they do we should add a warning.



```
If($event -eq "" -or $computer -eq "" -or $da -eq $null){  
[System.Windows.MessageBox]::Show("Please make sure all values are entered",  
"Missing Values")  
}Else{Check-Event $event $computer $d}
```

WPF MessageBox

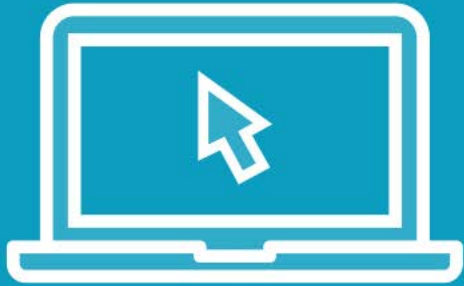
Using WPF we can easily create a MessageBox to popup and give the user more information.



Demo



Demo



Finalize the Code and Run it

