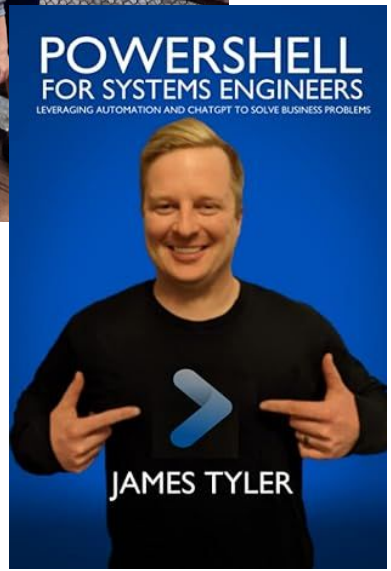
Building Graphical Applications with PowerShell 2.0

James Tyler

Overview

- Introduction
- PowerShell Pro Tools
- PoshGUI.com
- Overview of WinForm Controls
- Build a User Add App
- Converting PowerShell to .exe



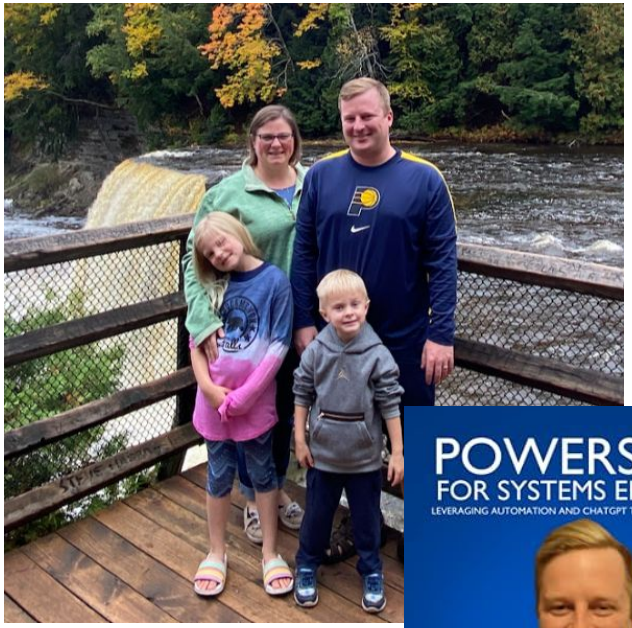
About Me

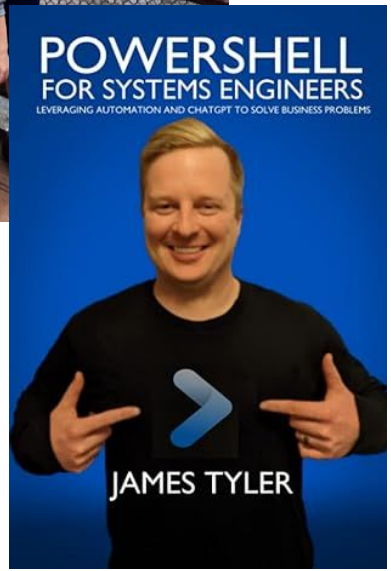
Director of Technology
Niles Community Schools &
River Valley School District

Author of
PowerShell for Systems Engineers

100K+ Views on YouTube
@PowerShellEngineer

President of Watervliet
Baseball/Softball Rec Council





About Me

Presenter at MAEDS 2022

- PowerShell GUIs
- PowerShell Storage Automation with AWS, Azure & GCP

Brief Presenter at MAEDS 2019

- Chris Thomas/Eric Krebill PowerShell Session

SoapBox Time…

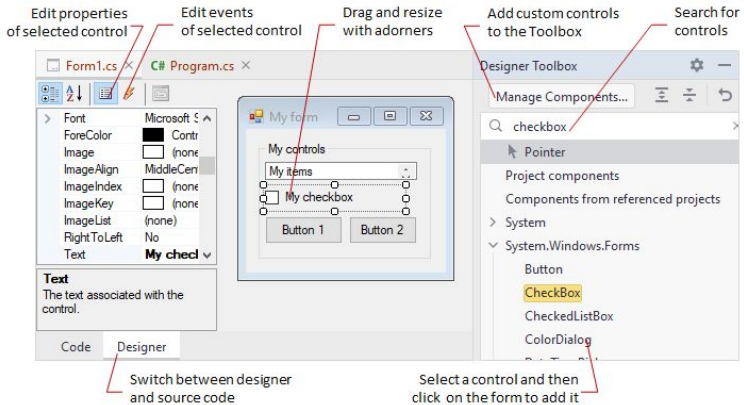
- Don’t fall victim to imposter syndrome!
- I’m going to put on my Chris Thomas hat…
- Everyone has something valuable to share
- Present next year!

PowerShell Resources

- PowerShell Podcast
  https://www.youtube.com/playlist?list=PL1mL90yFExsjUS8DRkzfLUcHds7vlxqgM
- New York PowerShell Meetup
  https://www.meetup.com/NycPowershellMeetup/
- Steve Lee, Principal Software Engineering Manager of PowerShell
  https://twitter.com/steve_msft
- Jeff Hicks Newsletter - “Behind the Pipeline”
  https://jeffhicks.substack.com/

Why use graphical apps?

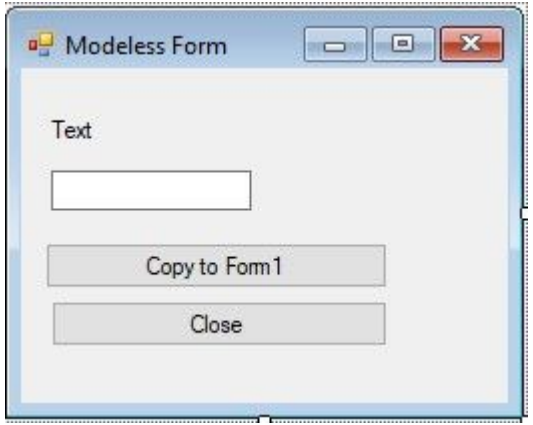
- Make repeated tasks easier
- Make repeated tasks faster
- Make repeated tasks more accurate
- Make tools portable for frontline technical staff



Windows Forms

What are Windows Forms?

- Windows Forms is a UI framework for building Windows desktop apps. It provides one of the most productive ways to create desktop apps based on the visual designer provided in Visual Studio. Functionality such as drag-and-drop placement of visual controls makes it easy to build desktop apps.



Why Windows Forms?

Simplicity in Design

Simplicity in Coding

Windows Forms

PoshGUI.com

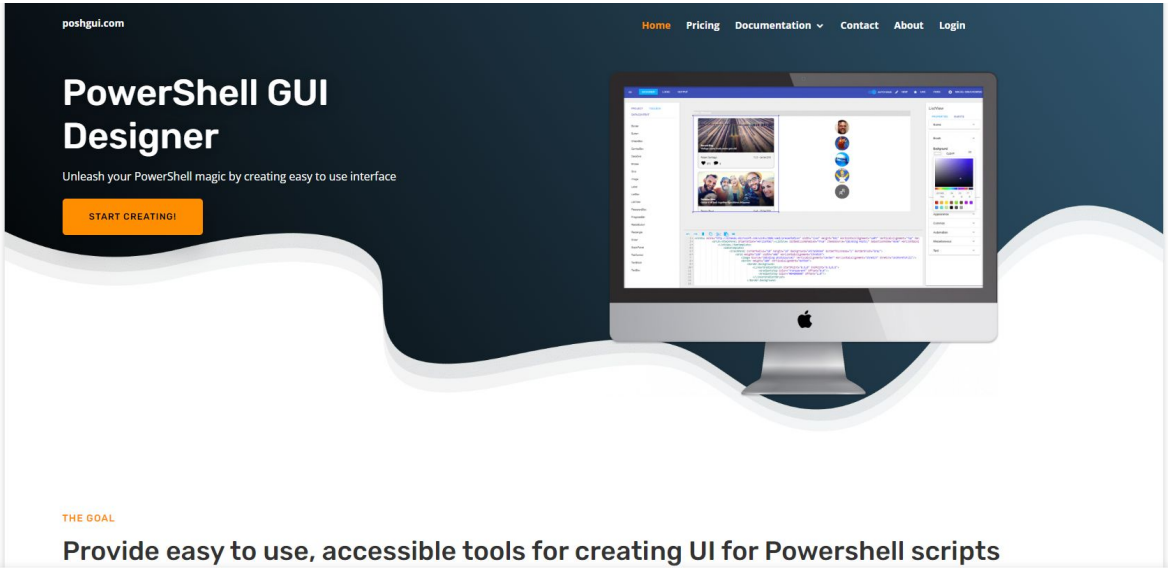
$6.99/m

-WPF Designer

- Winforms Designer

-Cmdlet Builder

-Code Editor



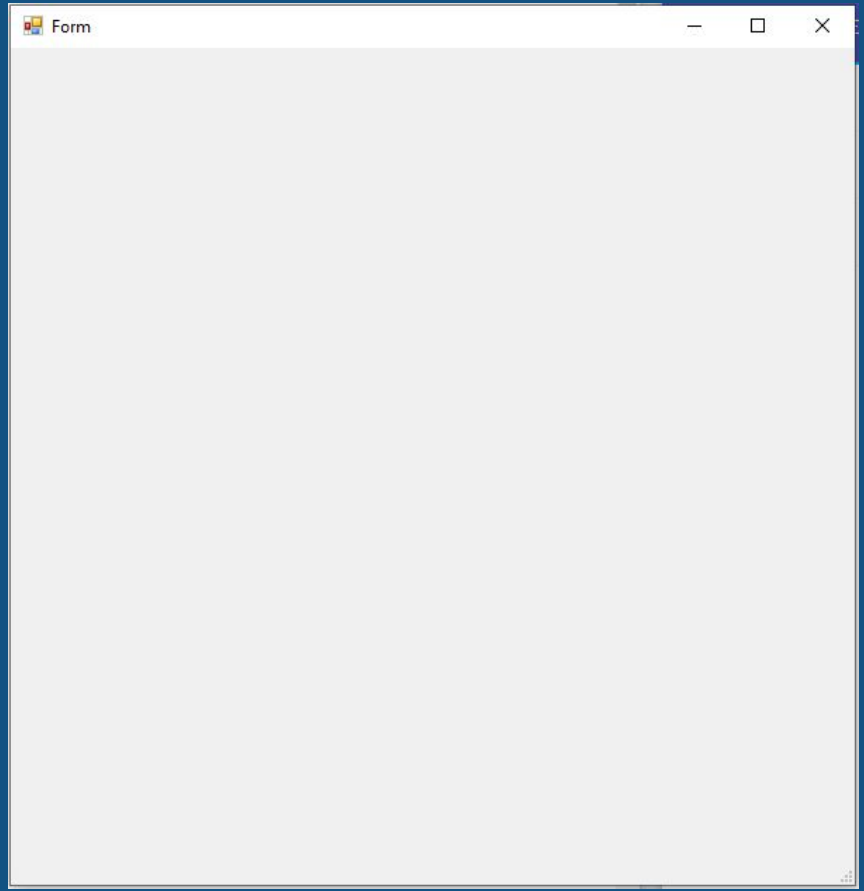
Code Examples

https://github.com/jimrtyler

Initializing an Empty Form

```
Add-Type -AssemblyName System.Windows.Forms

[System.Windows.Forms.Application]::EnableVisualStyles()

$Form = New-Object system.Windows.Forms.Form
$Form.ClientSize = New-Object
System.Drawing.Point(600,600)
$Form.text = "Form"
$Form.TopMost = $false

[void]$Form.ShowDialog()
```

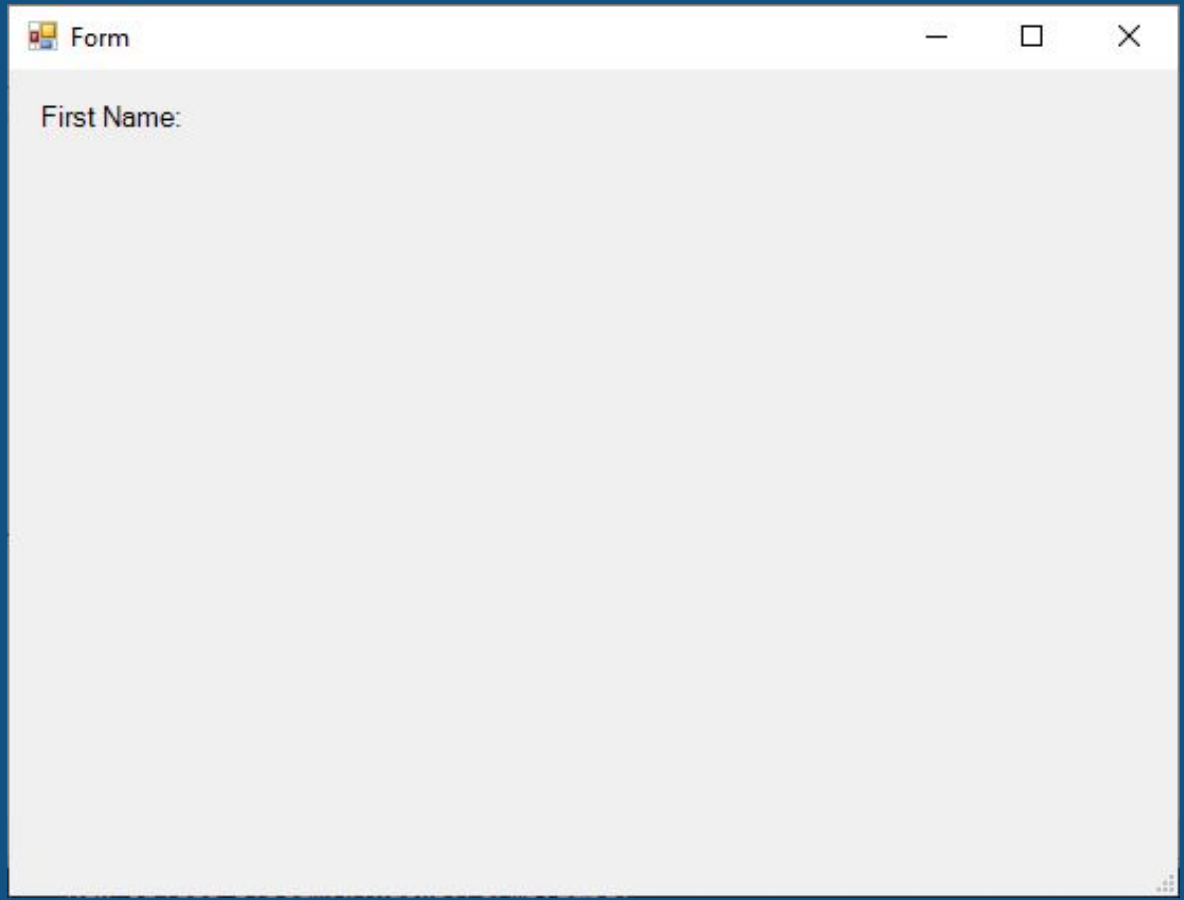
Initializing an Empty Form



Label

```
$FirstNameLabel = New-Object system.Windows.Forms.Label
$FirstNameLabel.text                = "First Name:"
$FirstNameLabel.AutoSize            = $true
$FirstNameLabel.width               = 25
$FirstNameLabel.height              = 10
$FirstNameLabel.location            = New-Object
System.Drawing.Point(13,14)
$FirstNameLabel.Font                = New-Object
System.Drawing.Font('Microsoft Sans Serif',10)

$Form.controls.AddRange(@($FirstNameLabel))


[void]$Form.ShowDialog()
```

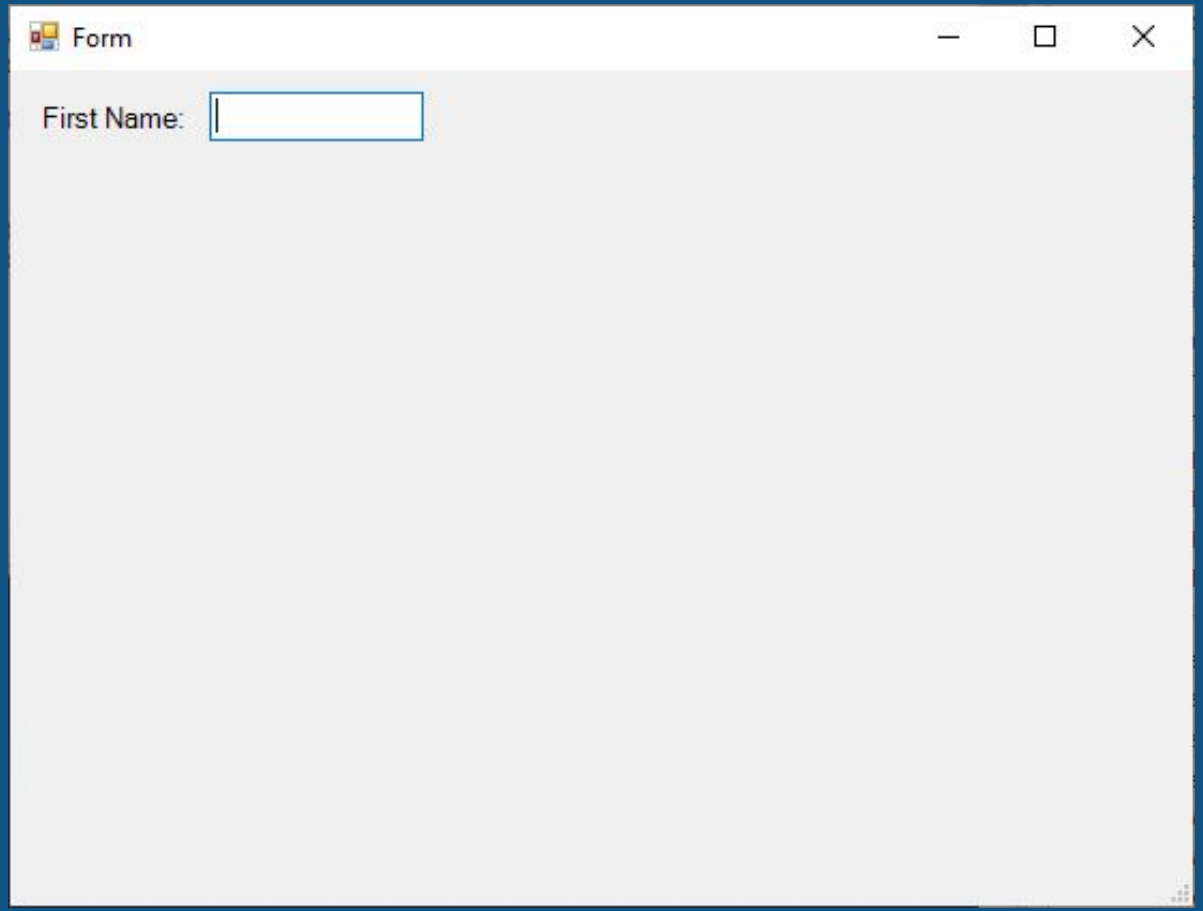
Label



Text Box

```
$FirstNameTextBox = New-Object
system.Windows.Forms.TextBox
$FirstNameTextBox.multiline = $false
$FirstNameTextBox.width = 100
$FirstNameTextBox.height = 20
$FirstNameTextBox.location = New-Object
System.Drawing.Point(93,10)
$FirstNameTextBox.Font = New-Object
System.Drawing.Font('Microsoft Sans Serif',10)

$Form.controls.AddRange(@($FirstNameLabel,$FirstNameTextBo
x))


[void]$Form.ShowDialog()
```

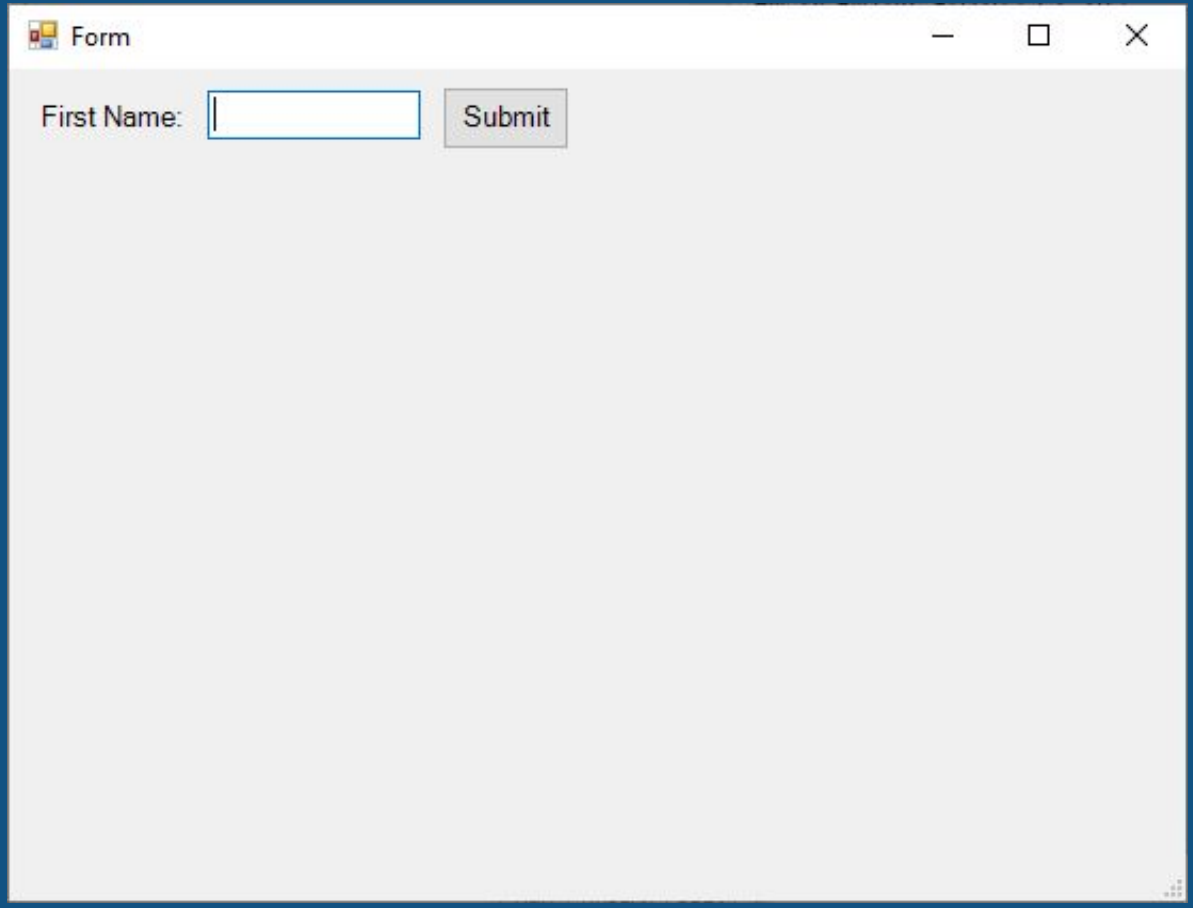
Text Box



Button

```
$SubmitBtn = New-Object system.Windows.Forms.Button
$SubmitBtn.text = "Submit"
$SubmitBtn.width = 60
$SubmitBtn.height = 30
$SubmitBtn.location = New-Object
System.Drawing.Point(203,8)
$SubmitBtn.Font = New-Object
System.Drawing.Font('Microsoft Sans Serif',10)
```

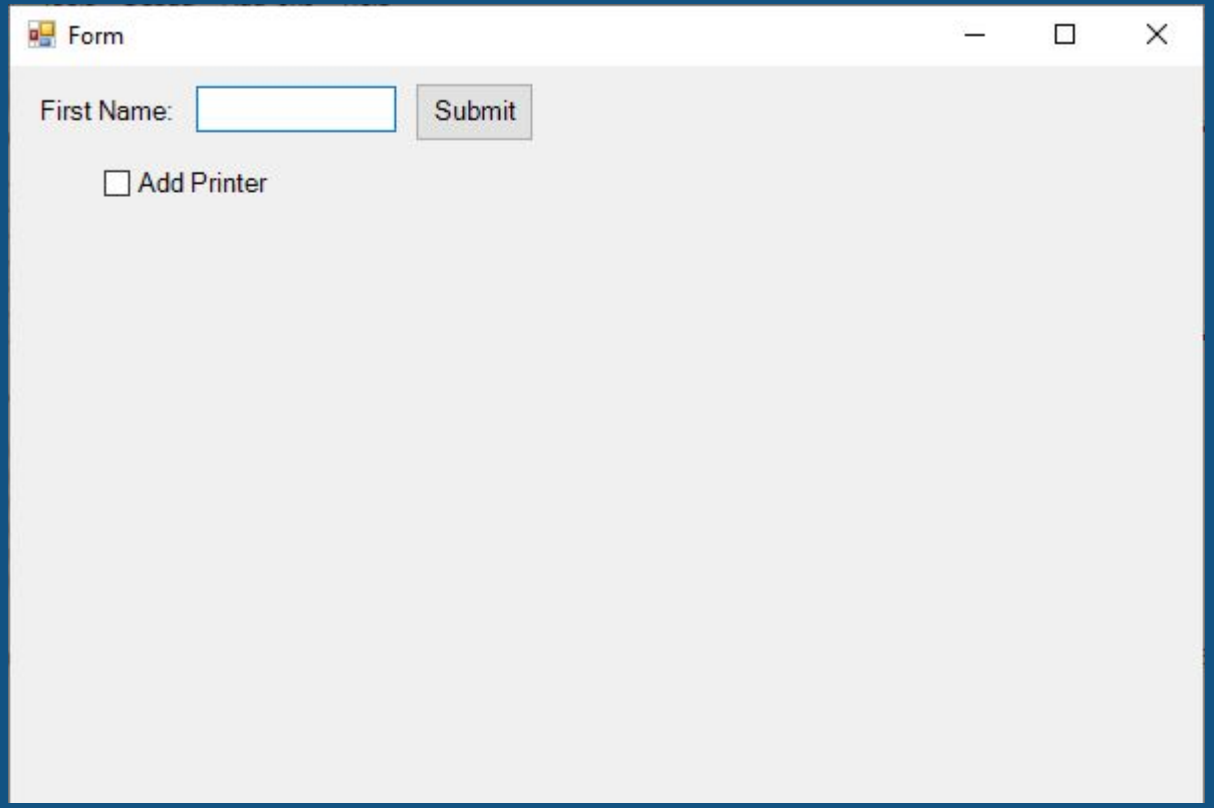
Button



Checkbox

```
$AddPrinterCheckBox = New-Object
system.Windows.Forms.CheckBox
$AddPrinterCheckBox.text          = "Add Printer"
$AddPrinterCheckBox.AutoSize      = $false
$AddPrinterCheckBox.width         = 95
$AddPrinterCheckBox.height        = 20
$AddPrinterCheckBox.location      = New-Object
System.Drawing.Point(47,49)
$AddPrinterCheckBox.Font          = New-Object
System.Drawing.Font('Microsoft Sans Serif',10)
```

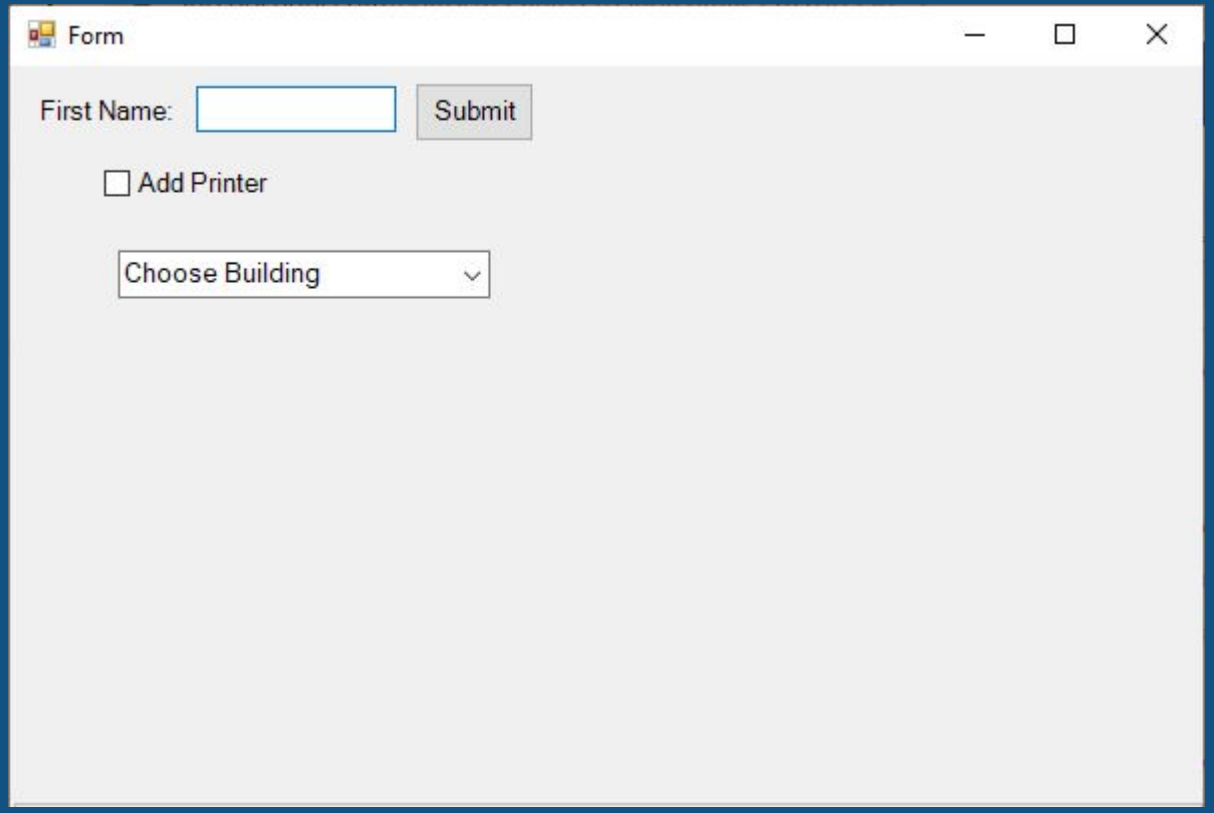
Checkbox



Combo Box

```
$ChooseBuildingComboBox           = New-Object
system.Windows.Forms.ComboBox
$ChooseBuildingComboBox.text      = "Choose Building"
$ChooseBuildingComboBox.width     = 186
$ChooseBuildingComboBox.height    = 20
$ChooseBuildingComboBox.location  = New-Object
System.Drawing.Point(54,92)
$ChooseBuildingComboBox.Font      = New-Object
System.Drawing.Font('Microsoft Sans Serif',10)
```

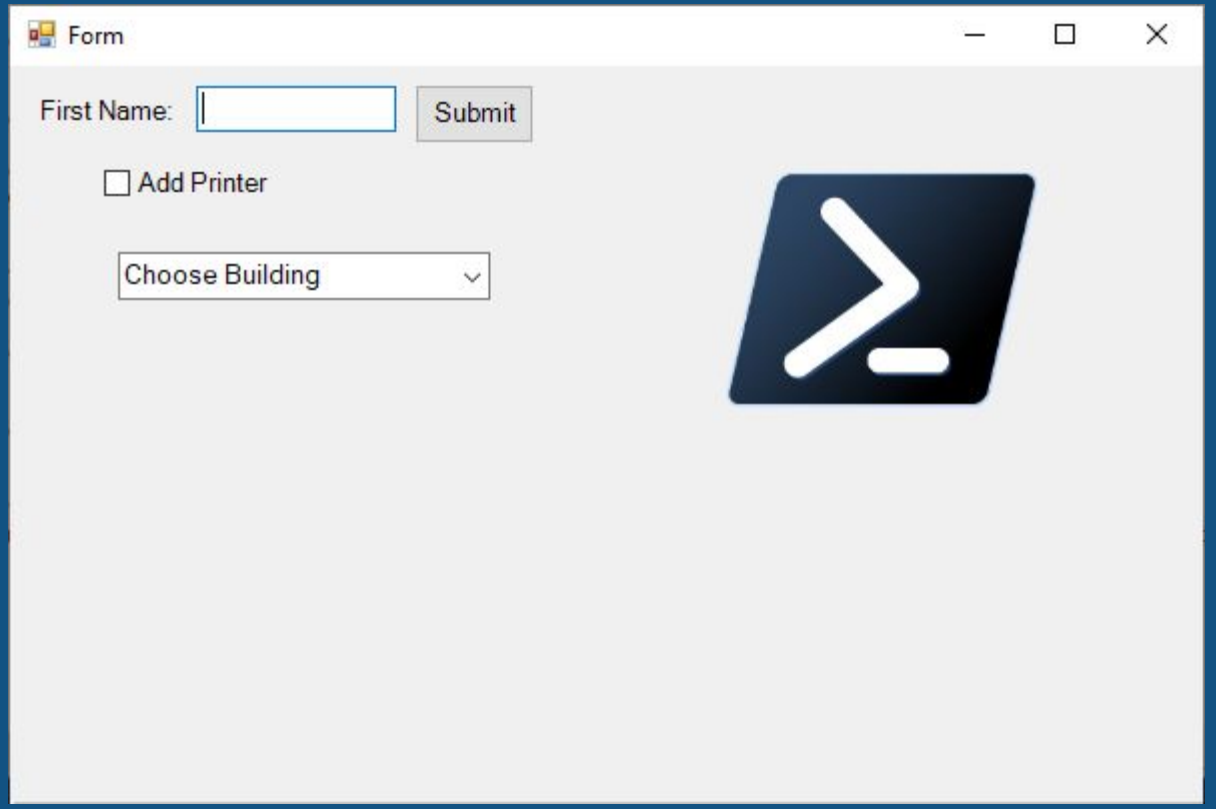
Combo Box



Picture Box

```
$PictureBox1 = New-Object system.Windows.Forms.PictureBox
$PictureBox1.width = 192
$PictureBox1.height = 167
$PictureBox1.location = New-Object
System.Drawing.Point(340,28)
$PictureBox1.imageLocation =
"https://upload.wikimedia.org/wikipedia/commons/a/af/Powe
rShell_Core_6.0_icon.png"
$PictureBox1.SizeMode =
[System.Windows.Forms.PictureBoxSizeMode]::zoom
```

Picture Box



Show/Hide Console

```
# .Net methods for hiding/showing the console in the background

Add-Type -Name Window -Namespace Console -MemberDefinition ' [DllImport("Kernel32.dll")] public static extern IntPtr GetConsoleWindow(); [DllImport("user32.dll")] public static extern bool ShowWindow(IntPtr hWnd, Int32 nCmdShow); '
```

Active Directory User Add Form

Windows Forms

Making Windows Forms Applications with ChatGPT

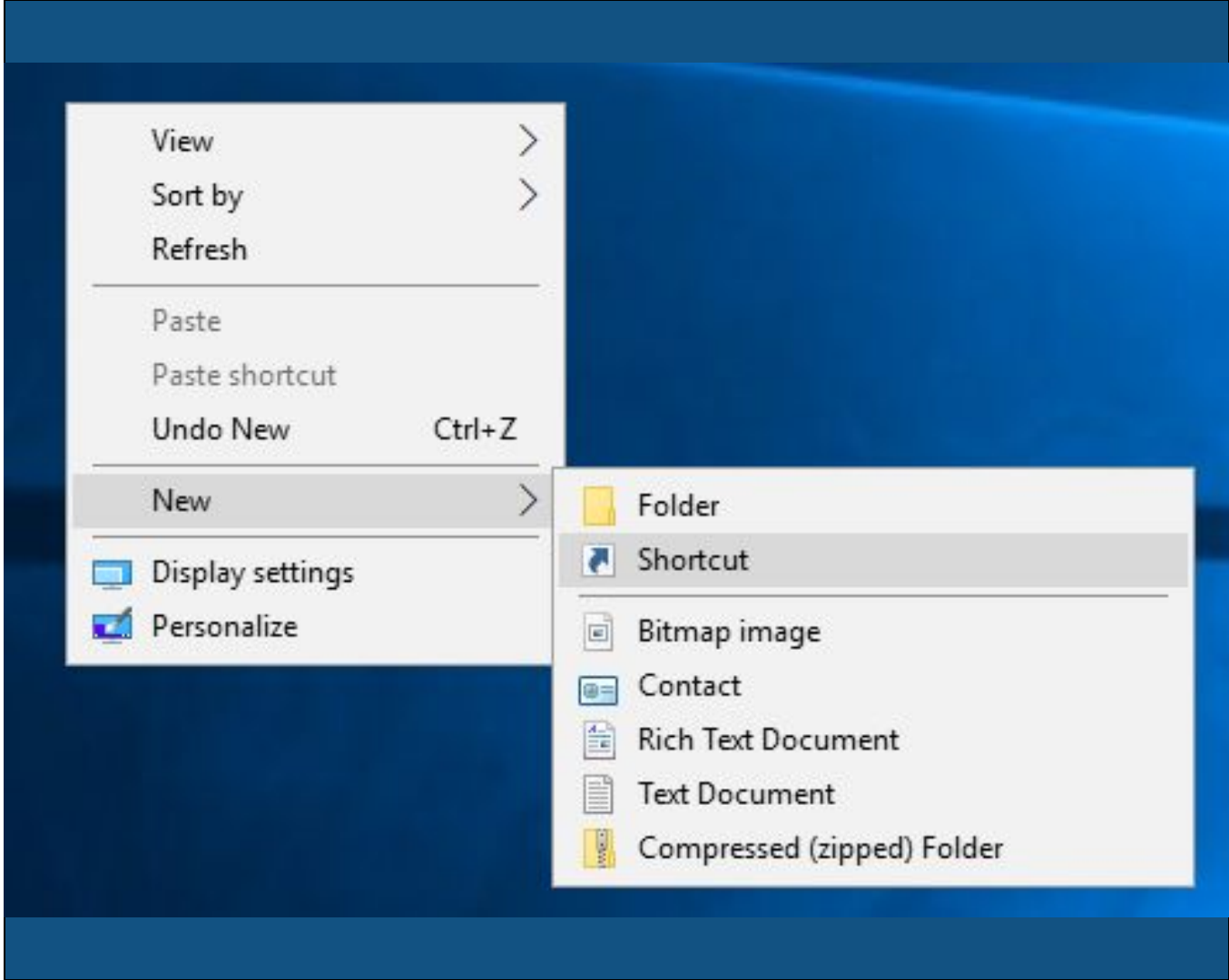
Show/Hide Console

```
function Show-Console {

    $consolePtr = [Console.Window]::GetConsoleWindow()

    # Hide = 0,
    # ShowNormal = 1,
    # ShowMinimized = 2,
    # ShowMaximized = 3,
    # Maximize = 3,
    # ShowNormalNoActivate = 4,
    # Show = 5,
    # Minimize = 6,
    # ShowMinNoActivate = 7,
    # ShowNoActivate = 8,
    # Restore = 9,
    # ShowDefault = 10,
    # ForceMinimized = 11

[Console.Window]::ShowWindow($consolePtr, 4)

    }
```

Show/Hide Console

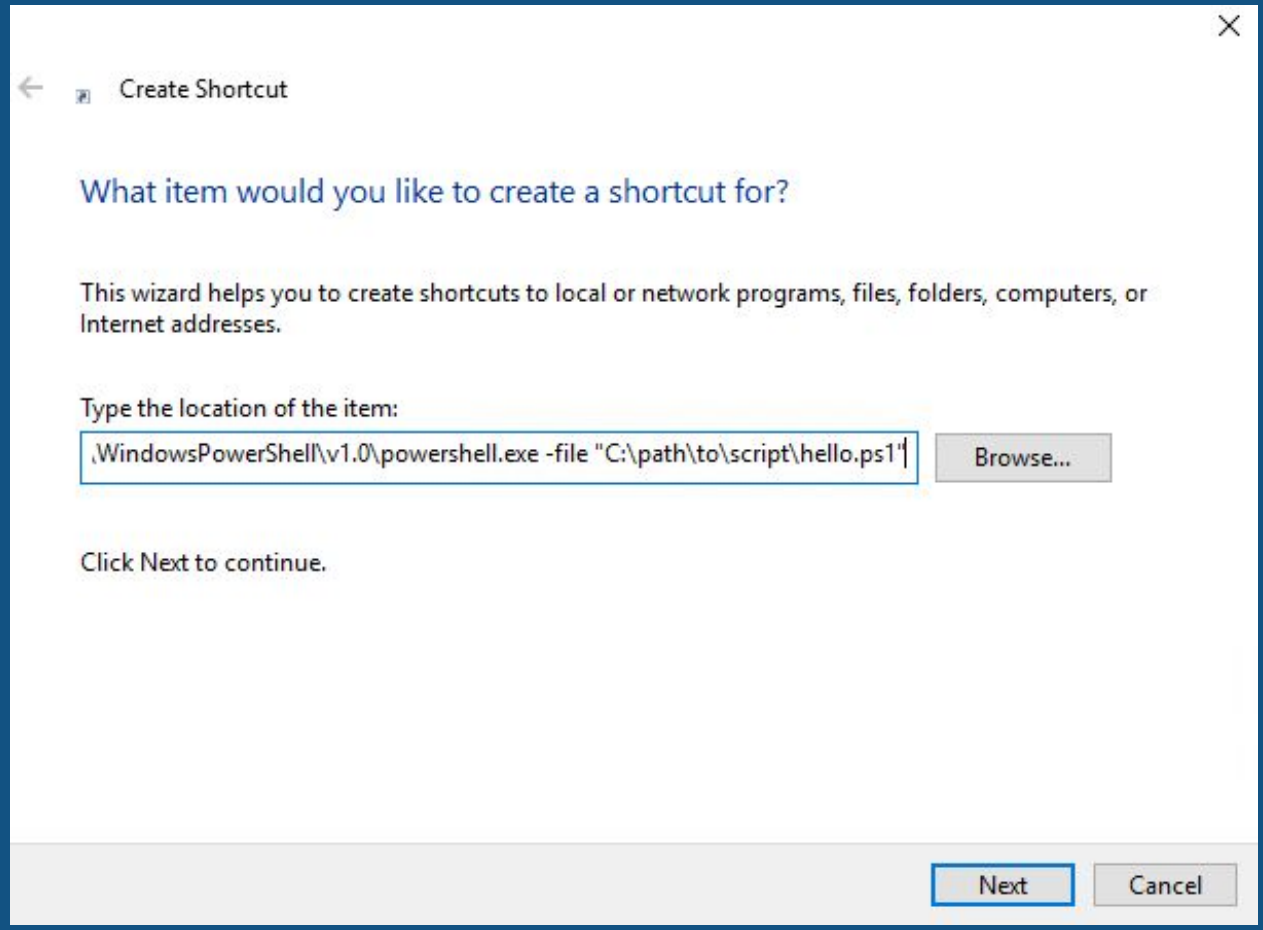
```
function Hide-Console {
    $consolePtr = [Console.Window]::GetConsoleWindow()
    #0 hide
    [Console.Window]::ShowWindow($consolePtr, 0)

}

Hide-Console
```

One-Click Run Script

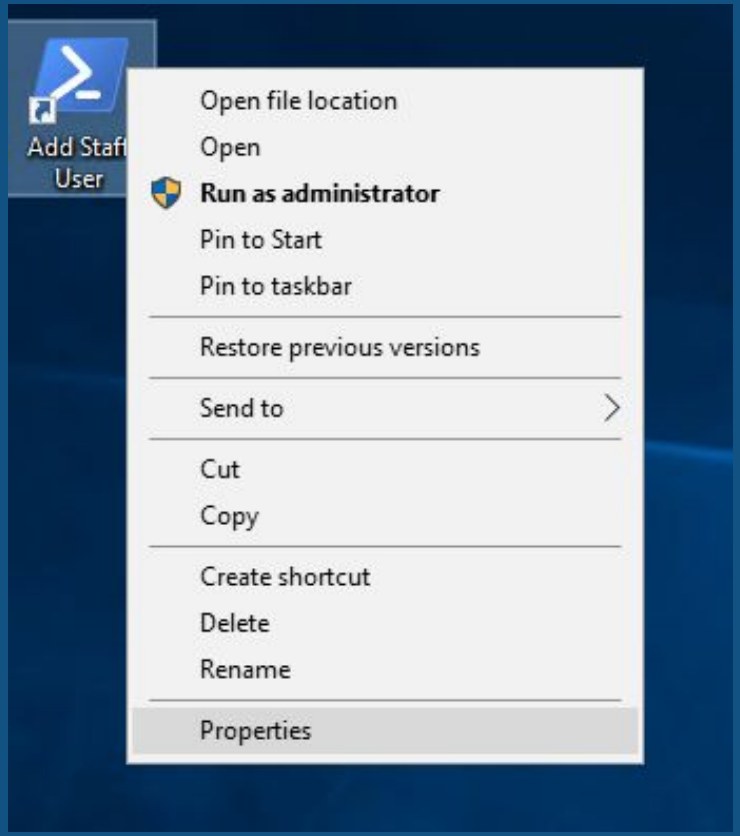


One-Click Run Script

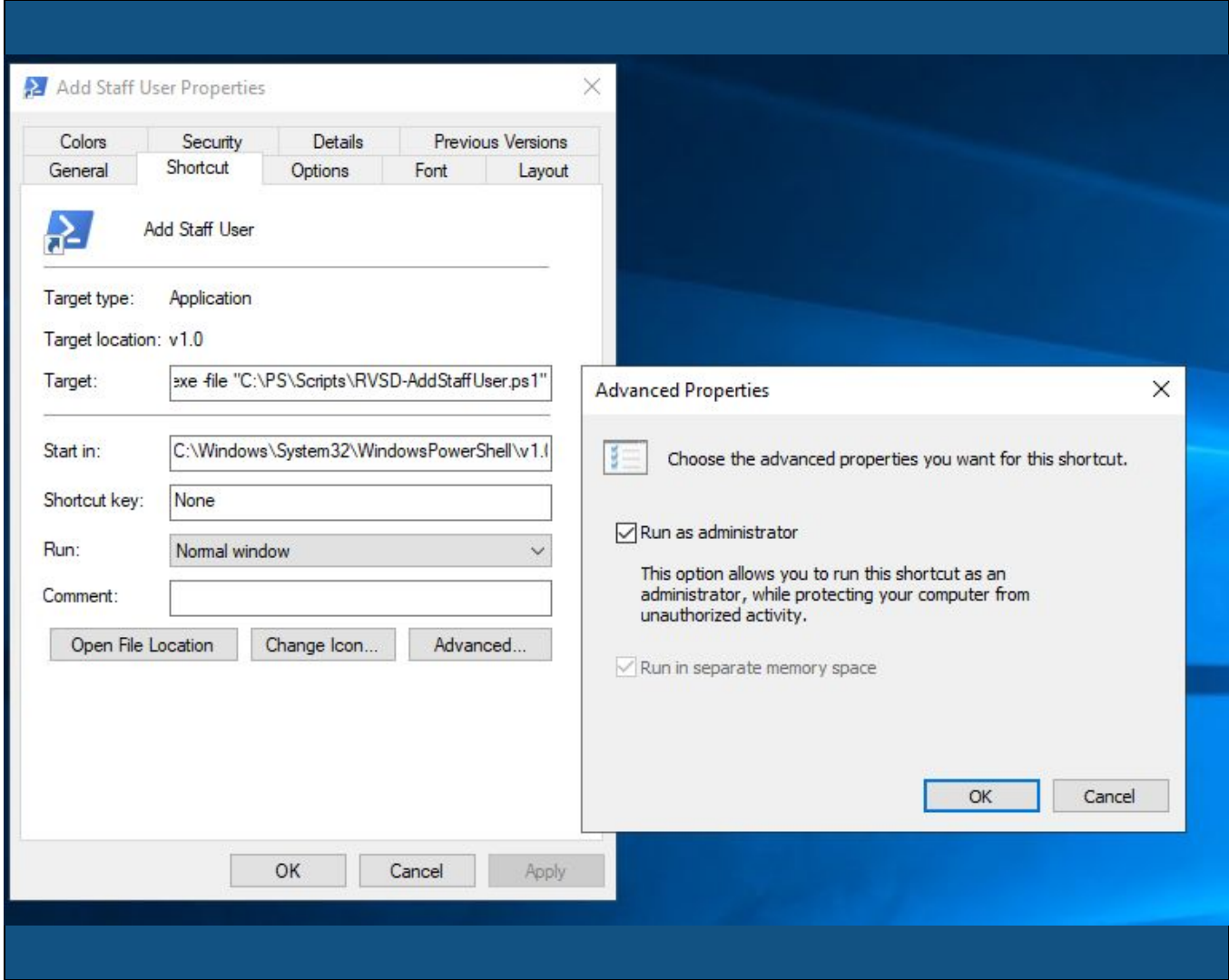
```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
-file "C:\path\to\script\hello.ps1"
```



Run As Administrator



Run As Administrator



Questions?

Windows Forms

Install-Module ps2exe

Auf Wiedersehen

YouTube

https://youtube.com/@PowerShellEngineer

LinkedIn:
https://linkedin.com/in/jamestyler