

Creating a Full Executable Program



Jeff Adkin

PLURALSIGHT AUTHOR

@JeffAdkin www.JAdkin.com



Introduction

We will be going over the following components

Using Tabs

Layout
Main GUI

Layout
Secondary
GUI's

Setting up
the
Controls

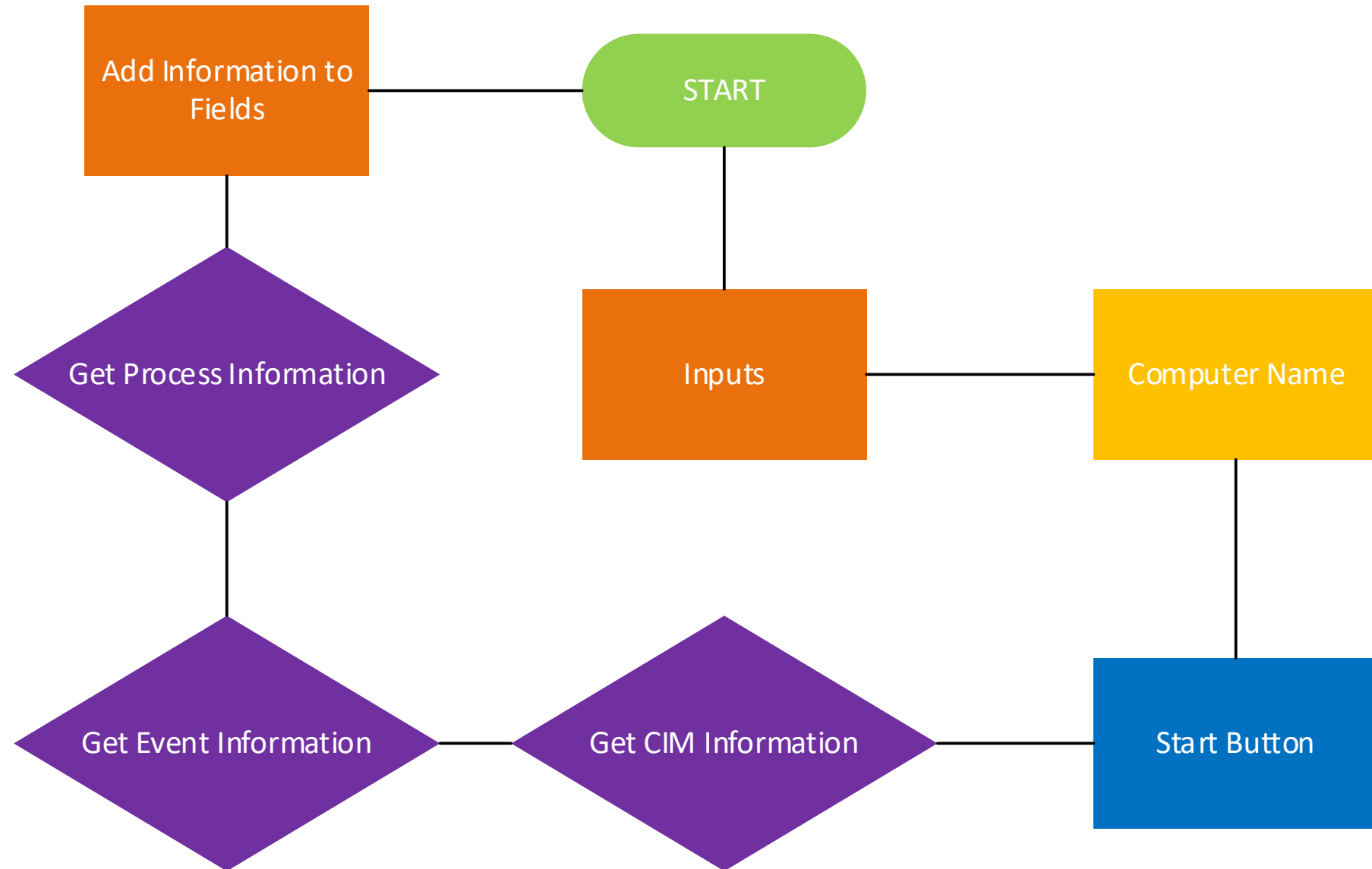
Creating
the
Functions

Creating an
Executable
Program

Running
the Code



Program Flow Chart



Using Tabs



Tab Items function like mini Windows. Each Tab has its own Layout and Controls that are placed on it.

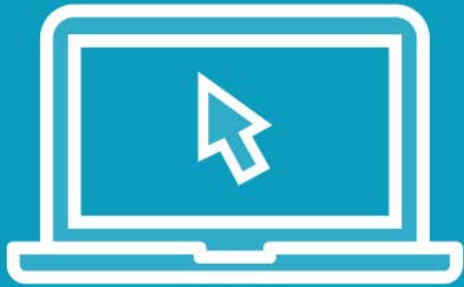


Tab Control XAML

```
<Window>  
  <Grid>  
    <TabControl>  
      <TabItem>  
        <Grid>  
          <CONTROLS>  
        </Grid>  
      </TabItem>  
    </TabControl>  
  </Grid>  
</Window>
```



Demo



Add Tab Control

Add Tab Items

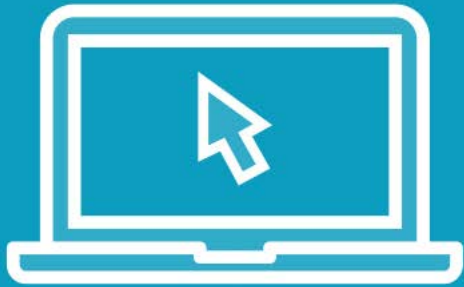
Set Tab Item Headers



Layout Main Tab GUI



Demo



Layout our first Tab Item with:

Image

Labels

Textbox

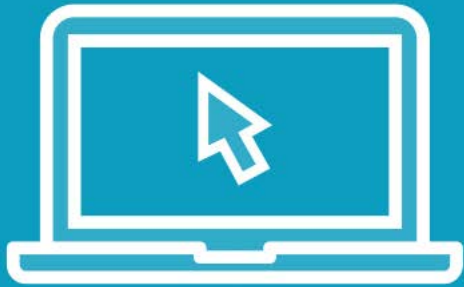
Start Button



Layout Secondary Tab GUI's



Demo



Layout the 3 remaining tabs:

System

Events

Processes



Setting up our Controls



```
[xml]$Form = Get-Content ".\app.xaml"
```

```
$NR=(New-Object System.Xml.XmlNodeReader $Form)
```

```
$Win=[Windows.Markup.XamlReader]::Load( $NR )
```

Import the XAML

Like we have done before we need to import the XAML from the Visual Studio Project.



```
$computer = $win.FindName("ComputerName")  
$start = $win.FindName("Start")  
$os = $win.FindName("OS")  
$Inst = $win.FindName("InstallDate")  
$sp = $win.FindName("ServicePack")  
$edg = $win.FindName("EVdataGrid")  
$pdg = $win.FindName("ProcdataGrid")
```

Add Controls to Variables

Each control we want to manipulate has to be in a variable we can call.



```
$arrev = New-Object System.Collections.ArrayList
```

```
$arrproc = New-Object System.Collections.ArrayList
```

Array Lists for our Data Grids

Using Array Lists is a simple way to get information into a nice DataGrid quickly and easily.



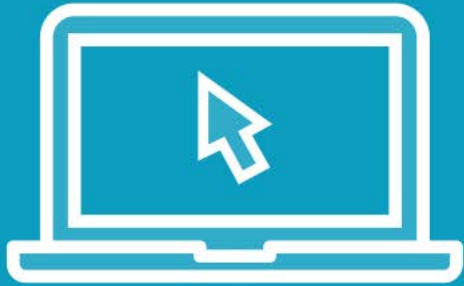
```
$start.add_click({  
$comp = $computer.Text  
Get-Systab $comp  
$events= Get-EventTab $comp  
$arrev.addrange($events)  
$edg.ItemsSource=@($arrev)  
$Procs = Get-ProcTab $comp  
$arrproc.addrange($Procs)  
$pdg.ItemsSource=@($arrproc)})
```

Add_Click

In this program we only have one button, so only one On_Click, but it does a lot of work.



Demo



Create Variables for our controls

Create Array Lists for our Data Grids

Create the Add_Click function



Creating the Functions



```
Function Get-SysTab($computer){  
$sys = Get-CimInstance -ComputerName $Computer win32_operatingsystem |  
select-object Caption, installdate, Servicepackmajorversion  
$os.content = $sys.caption  
$Inst.content = $sys.installdate  
$sp.content = $sys.Servicepackmajorversion  
}
```

System Information

To get the System information we will use the CIM (Common Information Model) and select the pieces of system information we would like. Each of these pieces will be attached to the corresponding Label.



```
Function Get-EventTab($computer){
```

```
$ev = get-eventlog application -ComputerName $computer -newest 100 |  
select TimeGenerated, EntryType, Source, InstanceID | sort -property Time
```

```
Return $ev
```

```
}
```

Events Function

In this function we simply get the newest 100 events and select only the columns we want and then return those to the Add_Click function.



```
Function Get-ProcTab($computer){
```

```
$proc = Get-Process -ComputerName $computer| select ID,Name,CPU | Sort -  
Property CPU -Descending
```

```
Return $proc
```

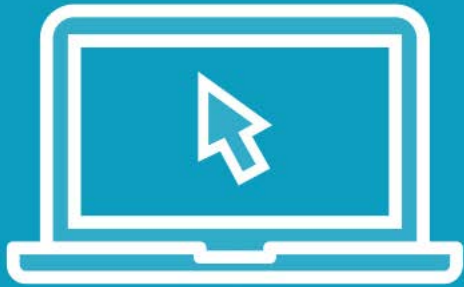
```
}
```

Processes Function

In this Function we are grabbing all of the running processes on the computer, selecting the columns we would like and then sorting by CPU usage from highest to lowest. We then return this set back to the On_Click function.



Demo



Create our 3 functions:

Get-SysTab

Get-EventTab

Get-ProcTab



Creating an Executable Program



Tools for Creating a PowerShell Executable

Power GUI

A free tool that can wrap the PS1 as an executable.

<https://software.dell.com/products/powergui-freeware/>

PS2EXE Script

A wonderful script that can turn your PowerShell scripts into EXE's.

<https://ps2exe.codeplex.com>

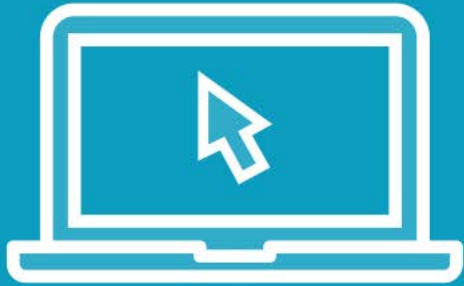
Admin Script Editor

A (now) free tool that allow you to script and change scripts to EXE.

<http://www.itninja.com/community/admin-script-editor>



Demo



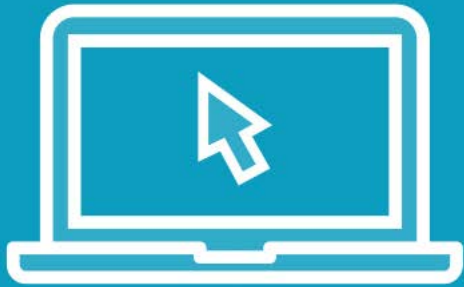
Using ASE to create Executable



Running the Code



Demo



Lets give it a Run!

