

```

using System;

using System.Diagnostics;

using System.Windows;

using System.Windows.Controls;

namespace Executable_Dependency
{

    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

        }

        private void Listener(object sender, RoutedEventArgs e)
        {
            Button btn = (Button)sender;

            String ProcessName = null;

            ProcessModuleCollection myProcessModuleCollection = null;

            ProcessInformation obj = null;

            ProcessName = ProcessTextBox.Text;

            ProcessTextBox.Text = "";
        }
    }
}

```

```
listbox.Items.Clear();
```

```
obj = new ProcessInformation(ProcessName);
```

```
switch (btn.Uid)
```

```
{
```

```
    case "get info":
```

```
        try
```

```
        {
```

```
            myProcessModuleCollection = obj.GetProcessInfo();
```

```
        }
```

```
        catch (Exception exp)
```

```
        {
```

```
            MessageBox.Show(exp.Message);
```

```
            return;
```

```
        }
```

```
        foreach (ProcessModule p in myProcessModuleCollection)
```

```
        {
```

```
            listbox.Items.Add(p.ModuleName);
```

```
        }
```

```
            break;
```

```
    case "reset":
```

```

        ProcessTextBox.Text = "";

        listbox.Items.Clear();

        break;
    }

}

}

}

}

////////////////////////////////////

using System;
using System.Diagnostics;
using System.Windows;

namespace Executable_Dependency
{
    class ProcessException : Exception
    {
        public ProcessException(String exception)
        {
            MessageBox.Show(exception);
        }
    }
}

```

```

class ProcessInformation
{
    String ProcessName;

    public ProcessInformation()
    {
        ProcessName = null;
    }

    public ProcessInformation(String ProcessName)
    {
        this.ProcessName = ProcessName;
    }

    public ProcessModuleCollection GetProcessInfo()
    {
        ProcessModuleCollection myProcessModuleCollection = null;

        Process myProcess = new Process();

        ProcessStartInfo myProcessStartInfo = new
ProcessStartInfo(ProcessName);

        try
        {
            myProcess.StartInfo = myProcessStartInfo;

            myProcess.Start();

            System.Threading.Thread.Sleep(1000);

            myProcessModuleCollection = myProcess.Modules;

```

```

    }

    catch (Exception e)
    {
        throw new ProcessException(e.Message);
    }

    finally
    {
        // myProcess.CloseMainWindow();
        myProcess.Kill();
        // myProcess.Close();
        // myProcess.Dispose();
    }

    return myProcessModuleCollection;
}

}

}

```