

## WPF Single Instance Uygulama Yapmak

Yapmış olduğumuz WPF uygulamasının birden fazla kopyasını aynı anda çalıştırabiliriz. Fakat bazı durumlarda birden fazla çalışması doğru olmaz. Örneğin document tabanlı uygulamanın tek kopyasının çalışması daha mantıklı olacaktır. Buna en somut örnek Office Word uygulamasıdır. Ne kadar dosya açarsak açalım tek bir tane Office Word uygulaması çalışır. Bu tarz uygulamalara **single-instance** uygulamalar denir.

WPF'te single-instance uygulama yapmak biraz karmaşıktır. Bundan dolayı yapılması gereken işlem adımlarını sırasıyla inceleyelim.

Visual Basic ve Windows Form uygulamalarının application sınıfını WPF uygulamasında kullanmamız gerekmektedir. Çünkü WPF'in application sınıfı single-instance işlemlerini halledemez. Bunun için bizim Visual Basic ve Windows Form uygulamalarında kullanılan application sınıfını kullanmamız gerekmektedir.

Bunun için ilk olarak uygulamamıza **Microsoft.VisualBasic.dll** assembly eklenir ve **Microsoft.VisualBasic.ApplicationServices.WindowsFormsApplicationBase** sınıftan türeyen bir sınıf yaratılır. Bu sınıfın üç önemli üyesi bulunmaktadır:

**IsSingleInstance:** Bu property single-instance uygulama yapılmasını sağlar. Constructor'da bu property true yapılır.

**OnStartup():** Uygulama başladığı zaman çalışır. Bu metodu override edilir ve bu metod içinde WPF application nesnesi yaratılır.

**OnStartupNextInstance():** Bu metod uygulamanın bir başka instance'ı çalıştırılmaya çalışıldığı zaman tetiklenir. Bu metod komut-satırı argümanlarına erişmeyi sağlar. Bu metod ile yeni bir application nesnesi yaratılmasını engelleyebiliriz. Bunun için, yeni bir application nesnesi yaratmak yerine, var olan application nesnesinin başka bir penceresini gösterebiliriz.

### Örnek:

```
1 public class SingleInstanceApplicationWrapper :
2 Microsoft.VisualBasic.ApplicationServices.WindowsFormsApplicationBase
3 {
4     public SingleInstanceApplicationWrapper()
5     {
6         // Enable single-instance mode.
7         this.IsSingleInstance= true;
8     }
9     // Create the WPF application class.
10    private WpfApp app;
11    protected override bool OnStartup(
12 Microsoft.VisualBasic.ApplicationServices.StartupEventArgs e)
13    {
14        //WpfApp sınıfı System.Windows.Application sınıfından türetilmiştir.
15        app = new WpfApp()
16        app.Run();
17        return false;
18    }
19    // Direct multiple instances.
20    protected override void OnStartupNextInstance(
21 Microsoft.VisualBasic.ApplicationServices.StartupNextInstanceEventArgs e)
22    {
23        if (e.CommandLine.Count > 0)
24        {
25            app.ShowDocument(e.CommandLine[0]);
26        }
27    }
28 }
```

### WpfApp sınıfı

```
1 public class WpfApp : System.Windows.Application
2 {
3     protected override void OnStartup(System.Windows.StartupEventArgs e)
4     {
5         base.OnStartup(e);
6         WpfApp.current= this;
7         // Load the main window.
8         DocumentList list = new DocumentList();
```

```

9         this.MainWindow= list;
10        list.Show();
11        // Load the document that was specified as an argument.
12        if (e.Args.Length > 0) ShowDocument(e.Args[0]);
13    }
14    public void ShowDocument(string filename)
15    {
16        try
17        {
18            Document doc = new Document();
19            doc.LoadFile(filename);
20            doc.Owner= this.MainWindow;
21            doc.Show();
22            // If the application is already loaded, it may not be visible.
23            // This attempts to give focus to the new window.
24            doc.Activate();
25        }
26        catch
27        {
28            MessageBox.Show("Could not load document.");
29        }
30    }
31 }

```

Bu sınıfları oluşturduktan sonra, uygulama çalıştırıldığı zaman **App** sınıfından önce, **SingleInstanceApplicationWrapper** sınıfının çalıştırılmasını sağlamalıyız. Bunu sağlayabilmek için de, uygulamanın App.xaml dosyasını değil, klasik **Main** metodu kullanılmalıdır:

```

1  public class Startup
2  {
3      [STAThread]
4      public static void Main(string[] args)
5      {
6          SingleInstanceApplicationWrapper wrapper =
7          new SingleInstanceApplicationWrapper();
8          wrapper.Run(args);
9      }
10 }

```