

عنوان: استفاده از کنترل‌های Active-X در WPF

نویسنده: وحید نصیری

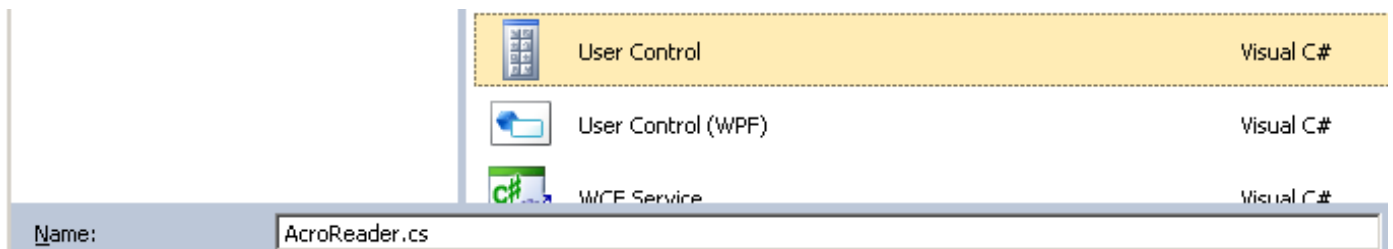
تاریخ: ۱۳۹۰/۰۹/۰۸ ۰۰:۰۰:۰۰

آدرس: www.dotnettips.info

برچسب‌ها: MVVM

گاهی از اوقات شاید نیاز شود تا از یک کنترل Active-X در WPF استفاده شود؛ مثلاً هیچ نمایش دهنده‌ی PDF ایی را در ویندوز نمی‌توان یافت که امکانات و کیفیت آن در حد Acrobat reader و Active-X آن باشد. یک روش استفاده از آن‌را به کمک کنترل WebBrowser در WPF پیشتر در این سایت [مطالعه کرده‌اید](#). روش معرفی شده برای WinForm هم در WPF قابل استفاده است که در ادامه شرح آن خواهد آمد.

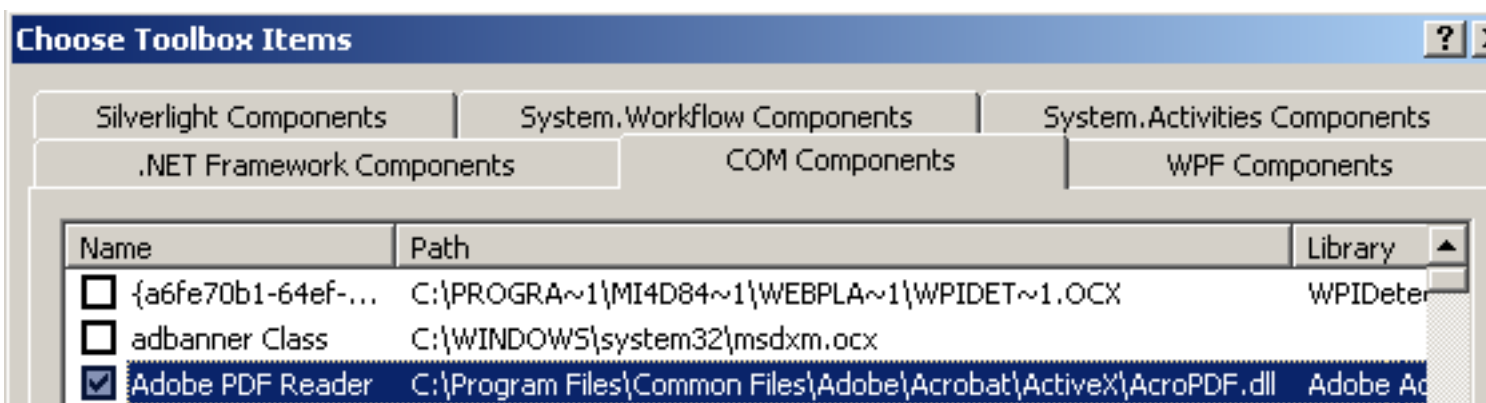
الف) بجای اضافه کردن یک User control مخصوص WPF یک user control از نوع WinForms را به یک پروژه WPF اضافه کنید.



سپس مراحل مشابهی را مانند حالت WinForms، باید طی کرد:

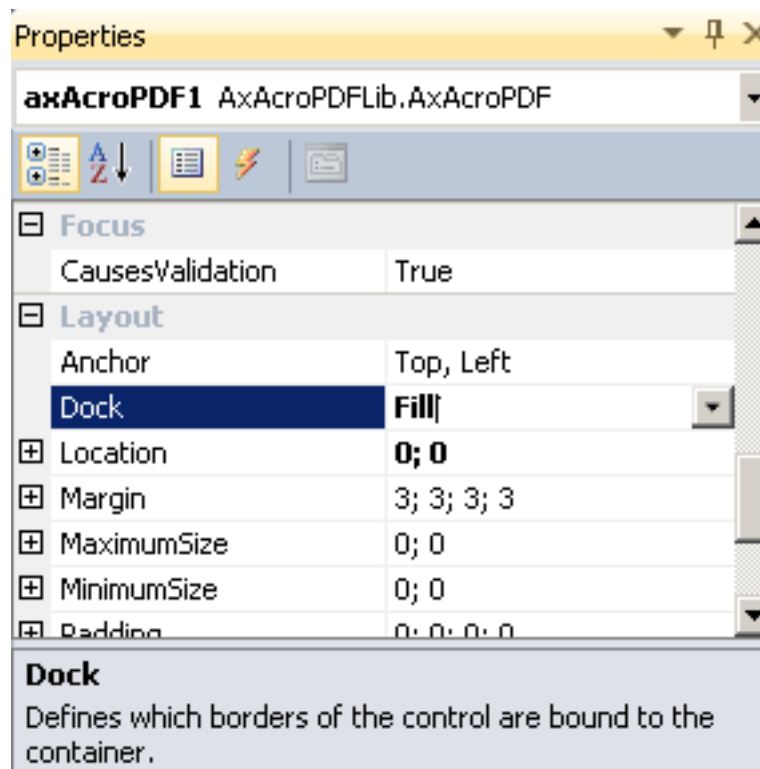
ب) در VS.NET از طریق منوی Tools گزینه‌ی Choose toolbox items، برگه‌ی Com components را انتخاب کنید.

ج) سپس گزینه‌ی Adobe PDF reader را انتخاب نمائید و بر روی دکمه‌ی OK کلیک کنید.



د) اکنون این کنترل جدید را بر روی فرم user control قسمت الف برنامه قرار دهید. به صورت خودکار COMReference های متناظر هم به پروژه اضافه می‌شوند.

پس از اینکه کنترل بر روی فرم قرار گرفت بهتر است به خواص آن مراجعه کرده و خاصیت Dock آن‌را با Fill مقدار دهی کرد تا کنترل به صورت خودکار در هر اندازه‌ای کل ناحیه‌ی متناظر را پوشش دهد.



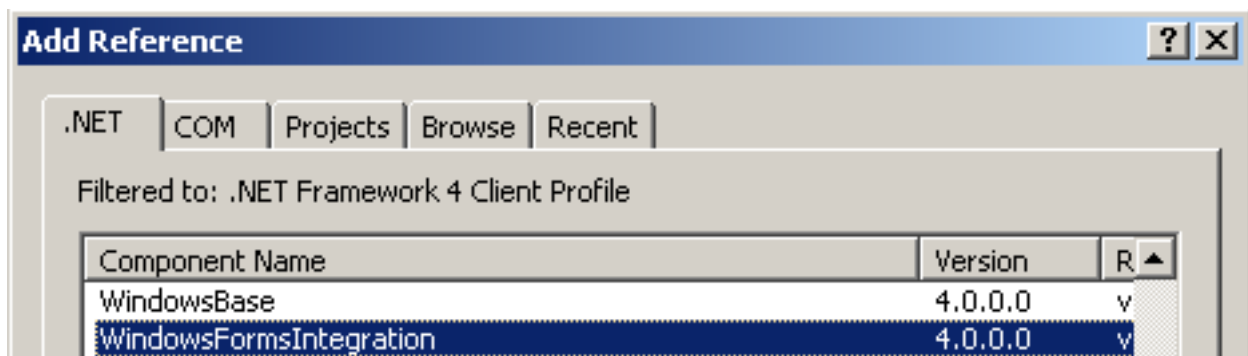
کدهای مرتبط با نمایش فایل PDF این کنترل هم به شرح زیر است:

```
using System.Windows.Forms;

namespace WpfPdfViewer.Controls
{
    public partial class AcroReader : UserControl
    {
        public AcroReader(string fileName)
        {
            InitializeComponent();
            ShowPdf(fileName);
        }

        public void ShowPdf(string fileName)
        {
            if (string.IsNullOrEmpty(fileName)) return;
            axAcroPDF1.LoadFile(fileName);
            axAcroPDF1.setShowToolbar(true);
            axAcroPDF1.Show();
        }
    }
}
```

خوب، ما تا اینجا یک کنترل Active-X را از طریق یک User controls مخصوص WinForms به پروژه‌ی WPF جاری اضافه کرده‌ایم. برای اینکه بتوانیم این کنترل را درون مثلاً یک User control از جنس WPF و XAML نمایش دهیم باید از کنترل WindowsFormsHost استفاده کرد. برای این منظور نیاز است تا ارجاعی را به اسمبلی WindowsFormsIntegration اضافه کنیم. پس از آن کنترل یاد شده قابل استفاده خواهد بود.



برای نمونه کدهای XAML پنجره اصلی برنامه می‌تواند به صورت زیر باشد:

```
<Window x:Class="WpfPdfViewer.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <WindowsFormsHost x:Name="WindowsFormsHost1" />
    </Grid>
</Window>
```

سپس جهت استفاده از کنترل WindowsFormsHost خواهیم داشت:

```
using WpfPdfViewer.Controls;

namespace WpfPdfViewer
{
    public partial class MainWindow
    {
        public MainWindow()
        {
            InitializeComponent();
            WindowsFormsHost1.Child = new AcroReader(@"PageSummary.pdf");
        }
    }
}
```

فقط کافی است شیء Child این کنترل را با وهله‌ای از یوزرکنترل AcroReader اضافه شده به برنامه مقدار دهی کنیم.

سؤال: این روش زیاد MVVM friendly نیست. به عبارتی Child را نمی‌توان از طریق Binding مقدار دهی کرد. آیا راهی برای آن وجود دارد؟

پاسخ: بله. روش متداول برای حل این نوع مشکلات، نوشتن یک DependencyObject و Attached property مناسب می‌باشد که به آن‌ها Behaviors هم می‌گویند. برای مثال یک نمونه از این پیاده سازی را در ذیل مشاهده می‌کنید:

```
using System;
using System.Windows;
using System.Windows.Forms;
using System.Windows.Forms.Integration;

namespace WpfPdfViewer.Behaviors
{
    public class WindowsFormsHostBehavior : DependencyObject
```

```

{
    public static readonly DependencyProperty BindableChildProperty =
        DependencyProperty.RegisterAttached("BindableChild",
            typeof(Control),
            typeof(WindowsFormsHostBehavior),
            new UIPropertyMetadata(null, BindableChildPropertyChanged));

    public static Control GetBindableChild(DependencyObject obj)
    {
        return (Control)obj.GetValue(BindableChildProperty);
    }

    public static void SetBindableChild(DependencyObject obj, Control value)
    {
        obj.SetValue(BindableChildProperty, value);
    }

    public static void BindableChildPropertyChanged(DependencyObject o,
        DependencyPropertyChangedEventArgs e)
    {
        var windowsFormsHost = o as WindowsFormsHost;
        if (windowsFormsHost == null)
            throw new InvalidOperationException("This behavior can only be attached to a
WindowsFormsHost.");

        var control = (Control)e.NewValue;
        windowsFormsHost.Child = control;
    }
}

```

که نهایتاً برای استفاده از آن خواهیم داشت:

```

<WindowsFormsHost
    Behaviors:WindowsFormsHostBehavior.BindableChild="{Binding ...}" />

```

و در ViewModel برنامه هم مانند مثال فوق، فقط کافی است یک وهله از new AcroReader به این خاصیت قابل انقیاد از نوع Control، انتساب داده شود.

یا حتی می‌توان بجای نوشتن یک BindableChild، برای مثال مسیر فایل pdf را به DependencyObject تعریف شده ارسال کرد و سپس در همانجا این وهله سازی و انتسابات صورت گیرد (بجای ViewModel برنامه که اینبار فقط مسیر را تنظیم می‌کند).