

## Microsoft Knowledge Base Article - 308825

# HOWTO: Handle PowerPoint Events With Visual C# .NET

[View products that this article applies to.](#)

This article was previously published under Q308825

For a Microsoft Visual Basic .NET version of this article, see [308330](#).

For a Microsoft Visual C++ .NET version of this article, see [309309](#).

## SUMMARY

This article demonstrates how to automate PowerPoint and handle events using Visual C# .NET.

## MORE INFORMATION

With Microsoft Visual Studio .NET, you cannot use delegates to sink events with PowerPoint. PowerPoint uses the **IDispatch** interface to fire events. To correctly sink the events, the Visual C# .NET application must use the **IConnectionPointContainer** and **IConnectionPoint** interface. The receiving application must also know the DISPIDs of the events to sink. These DISPIDs are not listed in PowerPoint's type library, but are listed in the sample code below for reference.

### Create the Sample C# Automation Client

1. Start Microsoft Visual Studio .NET. On the **File** menu, click **New** and then click **Project**. Under **Project types** click **Visual C# Projects**, then click **Windows Application** under **Templates**. Form1 is created by default.
2. Add a reference to the **Microsoft PowerPoint Object Library**. To do this, follow these steps:
  - a. On the **Project** menu, click **Add Reference**.
  - b. On the **COM** tab, locate **Microsoft Powerpoint 10.0 Object Library**, and then click **Select**. **NOTE:** If you have not already done so, it is recommended that you download, and then install the Microsoft Office XP Primary Interop Assemblies (PIAs). For additional information about Office XP PIAs, click the article number below to view the article in the Microsoft Knowledge Base:
 

[328912](#) INFO: Microsoft Office XP PIAs Are Available for Download
  - c. Click **OK** in the **Add References** dialog box to accept your selection.
3. On the **View** menu, click **Toolbox** to display the Toolbox and add two buttons and a list box to Form1.
4. In sequence, double-click **Button1**, **Button2**, and **Form1**.
5. In the code window, replace the following code

```
private void button1_Click(object sender, System.EventArgs e)
{

}

private void button2_Click(object sender, System.EventArgs e)
{

}

private void Form1_Load(object sender, System.EventArgs e)
{

}
```

with:

```
private UCOMIConnectionPoint m_oConnectionPoint;
private int m_Cookie;
private PowerPoint.ApplicationClass oPPT;
```

```
private void button1_Click(object sender, System.EventArgs e)
{
    // QI for IConnectionPointContainer.
    UCOMIConnectionPointContainer oConnPointContainer = (UCOMIConnectionPointContainer) oPPT
    // Get the GUID of the EApplication interface.
    Guid guid=typeof(PowerPoint.EApplication).GUID;

    // Find the connection point.
    oConnPointContainer.FindConnectionPoint(ref guid,out m_oConnectionPoint);
    // Call Advise to sink up the connection.
    m_oConnectionPoint.Advise(this,out m_Cookie);
}

[DispId(2001)]
public void WindowSelectionChange(PowerPoint.Selection Sel)
{
    this.listBox1.Items.Add("WindowSelectionChange");
}

[DispId(2002)]
public void WindowBeforeRightClick(PowerPoint.Selection Sel,bool Cancel)
{
    this.listBox1.Items.Add("WindowBeforeRightClick");
}

[DispId(2003)]
public void WindowBeforeDoubleClick(PowerPoint.Selection Sel,bool Cancel)
{
    this.listBox1.Items.Add("WindowBeforeDoubleClick");
}

[DispId(2004)]
public void PresentationClose(PowerPoint.Presentation Pres)
{
    this.listBox1.Items.Add("PresentationClose");
}

[DispId(2005)]
public void PresentationSave(PowerPoint.Presentation Pres)
{
    this.listBox1.Items.Add("PresentationSave");
}

[DispId(2006)]
public void PresentationOpen(PowerPoint.Presentation Pres)
{
    this.listBox1.Items.Add("PresentationOpen");
}

[DispId(2007)]
public void NewPresentation(PowerPoint.Presentation Pres)
{
    this.listBox1.Items.Add("NewPresentation");
}

[DispId(2008)]
public void PresentationNewSlide(PowerPoint.Slide Sld)
{
    this.listBox1.Items.Add("PresentationNewSlide");
}

[DispId(2009)]
```

```
public void WindowActivate(PowerPoint.Presentation Pres,PowerPoint.DocumentWindow Wn)
{
    this.listBox1.Items.Add("WindowActivate");
}

[DispId(2010)]
public void WindowDeactivate(PowerPoint.Presentation Pres,PowerPoint.DocumentWindow Wn)
{
    this.listBox1.Items.Add("WindowDeactivate");
}

[DispId(2011)]
public void SlideShowBegin(PowerPoint.SlideShowWindow Wn)
{
    this.listBox1.Items.Add("SlideShowBegin");
}

[DispId(2012)]
public void SlideShowNextBuild(PowerPoint.SlideShowWindow Wn)
{
    this.listBox1.Items.Add("SlideShowNextBuild");
}

[DispId(2013)]
public void SlideShowNextSlide(PowerPoint.SlideShowWindow Wn)
{
    this.listBox1.Items.Add("SlideShowNextSlide");
}

[DispId(2014)]
public void SlideShowEnd(PowerPoint.Presentation Pres)
{
    this.listBox1.Items.Add("SlideShowEnd");
}

[DispId(2015)]
public void PresentationPrint(PowerPoint.Presentation Pres)
{
    this.listBox1.Items.Add("PresentationPrint");
}

[DispId(2016)]
public void SlideSelectionChanged(PowerPoint.SlideRange SldRange)
{
    this.listBox1.Items.Add("SlideSelectionChanged");
}

[DispId(2017)]
public void ColorSchemeChanged(PowerPoint.SlideRange SldRange)
{
    this.listBox1.Items.Add("ColorSchemeChanged");
}

[DispId(2018)]
public void PresentationBeforeSave(PowerPoint.Presentation Pres,bool Cancel)
{
    this.listBox1.Items.Add("PresentationBeforeSave");
}

[DispId(2019)]
public void SlideShowNextClick(PowerPoint.SlideShowWindow Wn,PowerPoint.Effect nEffect)
{
    this.listBox1.Items.Add("SlideShowNextClick");
}
```

```

    }

    private void button2_Click(object sender, System.EventArgs e)
    {
        m_oConnectionPoint.Unadvise(m_Cookie);
        System.Runtime.InteropServices.Marshal.ReleaseComObject(oPPT);
        GC.Collect();
    }

    private void Form1_Load(object sender, System.EventArgs e)
    {
        //Create an instance of PowerPoint.
        oPPT = new PowerPoint.ApplicationClass();

        // Show PowerPoint to the user.
        oPPT.Visible = Microsoft.Office.Core.MsoTriState.msoTrue;
    }

```

6. Add the following to the **using** section at the top of the code window:

```

using System.Runtime.InteropServices;
using PowerPoint = Microsoft.Office.Interop.PowerPoint;

```

7. Test the program. To do this, follow these steps:
- Press F5 to build and run the program. PowerPoint is started.
  - Click **Button1** to set up the event sinks.
  - Create a new presentation in PowerPoint.

The **WindowActivate**, **NewPresentation**, **PresentationNewSlide** and **WindowSelectionChange** events fire.

- Save the presentation.

The **PresentationSave** event fires.

- Close the presentation.

The **PresentationClose** event fires.

- Activate **Form1** in the program. The events that were triggered by PowerPoint and handled by the program appear in the list box.
- Click **Button2** to disconnect the event sinks.
- Close **Form1**.

## REFERENCES

For additional information, click the article number below to view the article in the Microsoft Knowledge Base:

[254009](#) INFO: PowerPoint 2000 Event Demonstration Available for Download

For more information on Office Automation, see the following Microsoft Office Development support site:

FAQs and Highlights for Office Development  
<http://support.microsoft.com/support/officedev>

---

## The information in this article applies to:

- Microsoft Visual C# .NET (2002)
- Microsoft PowerPoint 2002

**Last Reviewed:** 12/12/2003 (3.0)

**Keywords:** kbAutomation kbConnPts kbhowto KB308825

**Contact Us**

© 2004 Microsoft Corporation. All rights reserved. [Terms of use](#) [Security & Privacy](#) [Accessibility](#)