附加事件也是一种路由事件。附加事件宿主不是从UIElement类派生而来的，而一般的路由事件宿主则是由UIElement类派生而来的。

例：程序attached\_event\_test1

MainWindow.xaml

<Window x:Class="attached\_event\_test1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="MainWindow" Height="200" Width="200">

<Grid x:Name="gridMain">

<Button x:Name="button" Content="OK" Width="80" Height="80"

Click="button\_Click" />

</Grid>

</Window>

Student.cs

using *System*;

using *System*.*Collections*.*Generic*;

using *System*.*Linq*;

using *System*.*Text*;

using *System*.*Threading*.*Tasks*;

using *System*.*Windows*;

namespace attached\_event\_test1

{

class Student

{

// 声明并定义路由事件（附加事件）

public static readonly *RoutedEvent* NameChangedEvent = *EventManager*

.*RegisterRoutedEvent*("NameChanged", *RoutingStrategy*.*Bubble*,

typeof(*RoutedEventHandler*), typeof(Student));

// 为界面元素添加路由事件侦听包装器

public static void AddNameChangedHandler(*DependencyObject* d, *RoutedEventHandler* h)

{

*UIElement* e = d as *UIElement*;

if (e != null)

{ // Student本身不是UIElement类

// 通过别的UIElement对象侦听路由事件

e.*AddHandler*(Student .NameChangedEvent, h);

}

}

// 移除侦听

public static void RemoveNameChangedHandler(*DependencyObject* d, *RoutedEventHandler* h)

{

*UIElement* e = d as *UIElement*;

if (e != null)

{

e.*RemoveHandler*(Student.NameChangedEvent, h);

}

}

public int Id { get; set; }

public string Name { get; set; }

}

}

MainWindow.xaml.cs

using *System*;

using *System*.*Collections*.*Generic*;

using *System*.*Linq*;

using *System*.*Text*;

using *System*.*Threading*.*Tasks*;

using *System*.*Windows*;

using *System*.*Windows*.*Controls*;

using *System*.*Windows*.*Data*;

using *System*.*Windows*.*Documents*;

using *System*.*Windows*.*Input*;

using *System*.*Windows*.*Media*;

using *System*.*Windows*.*Media*.*Imaging*;

using *System*.*Windows*.*Navigation*;

using *System*.*Windows*.*Shapes*;

namespace attached\_event\_test1

{

/// <summary>

/// MainWindow.xaml 的交互逻辑

/// </summary>

public partial class MainWindow : *Window*

{

public MainWindow()

{

*InitializeComponent*();

// 为外层Grid添加路由（附加）事件侦听器

Student.AddNameChangedHandler(

this.gridMain, new *RoutedEventHandler*(StudentNameChangedHandler));

}

private void button\_Click(object sender, *RoutedEventArgs* e)

{

Student student = new Student() { Id = 101, Name = "Tim" };

student.Name = "Tom";

// 准备事件消息并发送路由事件

*RoutedEventArgs* arg = new *RoutedEventArgs*(Student.NameChangedEvent, student);

// Student本身不是UIElement类派生类

// 通过Button来激发路由事件

button.*RaiseEvent*(arg);

}

// Grid捕捉到NameChangedEvent后的处理器

private void StudentNameChangedHandler(object sender, *RoutedEventArgs* e)

{

*MessageBox*.*Show*((e.*OriginalSource* as Student).Id.*ToString*());

}

}

}