

要查看英语原文，请勾选“英语”复选框。也可将鼠标指针移到文本上，在弹出窗口中显示英语原文。

# ListBoxItem 类

## Silverlight

表示 [ListBox](#) 中的可选项。

## 继承层次结构

```
System.Object
System.Windows.DependencyObject
System.Windows.UIElement
System.Windows.FrameworkElement
System.Windows.Controls.Control
System.Windows.Controls.ContentControl
System.Windows.Controls.ListBoxItem
System.Windows.Controls.ComboBoxItem
```

命名空间：[System.Windows.Controls](#)

程序集：[System.Windows](#) (在 [System.Windows.dll](#) 中)

## 语法

C#

```
[TemplateVisualStateAttribute(Name = "SelectedUnfocused", GroupName =
"SelectionStates")]
[TemplateVisualStateAttribute(Name = "BeforeUnloaded", GroupName = "LayoutStates")]
[TemplateVisualStateAttribute(Name = "Normal", GroupName = "CommonStates")]
[TemplateVisualStateAttribute(Name = "MouseOver", GroupName = "CommonStates")]
[TemplateVisualStateAttribute(Name = "Disabled", GroupName = "CommonStates")]
[TemplateVisualStateAttribute(Name = "Unselected", GroupName = "SelectionStates")]
[TemplateVisualStateAttribute(Name = "Selected", GroupName = "SelectionStates")]
[TemplateVisualStateAttribute(Name = "Unfocused", GroupName = "FocusStates")]
[TemplateVisualStateAttribute(Name = "Focused", GroupName = "FocusStates")]
[TemplateVisualStateAttribute(Name = "BeforeLoaded", GroupName = "LayoutStates")]
[TemplateVisualStateAttribute(Name = "AfterLoaded", GroupName = "LayoutStates")]
public class ListBoxItem : ContentControl
```

### XAML 对象元素用法

```
<ListBoxItem .../>
-or-
<ListBoxItem ...>
  content
</ListBoxItem>
```

XAML 值

内容

一个用作内容的对象元素。

ListBoxItem 类型公开以下成员。

构造函数

显示: ☒ 继承 ☒ 保护

	名称	说明
	<a href="#">ListBoxItem</a>	初始化 <a href="#">ListBoxItem</a> 类的一个新实例。

[页首](#)








属性

显示: ☒ 继承 ☒ 保护

	名称	说明
	<a href="#">ActualHeight</a>	获取 <a href="#">FrameworkElement</a> 的呈现高度。（继承自 <a href="#">FrameworkElement</a> 。）
	<a href="#">ActualWidth</a>	获取 <a href="#">FrameworkElement</a> 的呈现宽度。（继承自 <a href="#">FrameworkElement</a> 。）
	<a href="#">AllowDrop</a>	获取或设置一个值，该值确定此 <a href="#">UIElement</a> 是否可用作 Silverlight 拖放操作的放目标。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">Background</a>	获取或设置一个用于提供控件背景的画笔。（继承自 <a href="#">Control</a> 。）
	<a href="#">BorderBrush</a>	获取或设置一个用于描述控件的边框背景的画笔。（继承自 <a href="#">Control</a> 。）
	<a href="#">BorderThickness</a>	获取或设置控件的边框宽度。（继承自 <a href="#">Control</a> 。）
	<a href="#">CacheMode</a>	获取或设置一个值，该值指示应在可能时高速缓存已呈现内容。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">CharacterSpacing</a>	获取或设置以字体大小的千分之一为单位测量的控件中的文本字符之间的距离。（继承自 <a href="#">Control</a> 。）
	<a href="#">Clip</a>	获取或设置用于定义 <a href="#">UIElement</a> 的内容边框的 <a href="#">Geometry</a> 。（继承自 <a href="#">UIElement</a> 。）

	Content	获取或设置 <a href="#">ContentControl</a> 依赖项属性的值。（继承自 <a href="#">ContentControl</a> 。）
	ContentTemplate	获取或设置用于显示 <a href="#">ContentControl</a> 内容的数据模板。（继承自 <a href="#">ContentControl</a> 。）
	Cursor	获取或设置鼠标指针悬停在 <a href="#">FrameworkElement</a> 上时所显示的光标图像。（继承自 <a href="#">FrameworkElement</a> 。）
	DataContext	获取或设置 <a href="#">FrameworkElement</a> 参与数据绑定时的数据上下文。（继承自 <a href="#">FrameworkElement</a> 。）
	DefaultStyleKey	获取或设置一个键，该键引用控件的默认样式。（继承自 <a href="#">Control</a> 。）
	DesiredSize	获取此 <a href="#">UIElement</a> 在布局过程的测量处理过程中计算的大小。（继承自 <a href="#">UIElement</a> 。）
	Dispatcher	获取与此对象关联的 <a href="#">Dispatcher</a> 。（继承自 <a href="#">DependencyObject</a> 。）
	Effect	获取或设置用于呈现此 <a href="#">UIElement</a> 的像素着色器效果。（继承自 <a href="#">UIElement</a> 。）
	FlowDirection	获取或设置文本和其他用户界面元素在控制其布局的任何父元素中的流动方向。（继承自 <a href="#">FrameworkElement</a> 。）
	FontFamily	获取或设置用于在控件中显示文本的字体。（继承自 <a href="#">Control</a> 。）
	FontSize	获取或设置此控件中文本的大小。（继承自 <a href="#">Control</a> 。）
	FontStretch	获取或设置字体在屏幕上的压缩或扩展程度。（继承自 <a href="#">Control</a> 。）
	FontStyle	获取或设置呈现文本时使用的样式。（继承自 <a href="#">Control</a> 。）
	FontWeight	获取或设置指定字体的粗细。（继承自 <a href="#">Control</a> 。）
	Foreground	获取或设置一个用于描述前景色的画笔。（继承自 <a href="#">Control</a> 。）
	Height	获取或设置 <a href="#">FrameworkElement</a> 的建议高度。（继承自 <a href="#">FrameworkElement</a> 。）
	HorizontalAlignment	获取或设置在布局父级（如面板或项控件）中构成 <a href="#">FrameworkElement</a> 时应用于此元素的水平对齐特征。（继承自 <a href="#">FrameworkElement</a> 。）
	HorizontalContentAlignment	获取或设置控件内容的水平对齐方式。（继承自 <a href="#">Control</a> 。）
	IsEnabled	获取或设置一个值，该值指示用户是否可以与控件交互。（继承自 <a href="#">Control</a> 。）

	IsHitTestVisible	获取或设置此 <a href="#">UIElement</a> 的包含区域是否可为命中测试返回 <b>true</b> 值。（继承自 <a href="#">UIElement</a> 。）
	IsSelected	获取或设置一个指示是否选择 <a href="#">ListBoxItem</a> 的值。
	IsTabStop	获取或设置一个值，该值指示是否将某个控件包含在 Tab 导航中。（继承自 <a href="#">Control</a> 。）
	Language	获取或设置应用于 <a href="#">FrameworkElement</a> 的本地化/全球化语言信息。（继承自 <a href="#">FrameworkElement</a> 。）
	Margin	获取或设置 <a href="#">FrameworkElement</a> 的外边距。（继承自 <a href="#">FrameworkElement</a> 。）
	MaxHeight	获取或设置 <a href="#">FrameworkElement</a> 的最大高度约束。（继承自 <a href="#">FrameworkElement</a> 。）
	MaxWidth	获取或设置 <a href="#">FrameworkElement</a> 的最大宽度约束。（继承自 <a href="#">FrameworkElement</a> 。）
	MinHeight	获取或设置 <a href="#">FrameworkElement</a> 的最小高度约束。（继承自 <a href="#">FrameworkElement</a> 。）
	MinWidth	获取或设置 <a href="#">FrameworkElement</a> 的最小宽度约束。（继承自 <a href="#">FrameworkElement</a> 。）
	Name	获取或设置对象的标识名称。在 XAML 处理器从 XAML 标记创建对象树后，运行时代码可以使用此名称引用 XAML 声明的对象。（继承自 <a href="#">FrameworkElement</a> 。）
	Opacity	获取或设置对象的不透明度。（继承自 <a href="#">UIElement</a> 。）
	OpacityMask	获取或设置用于改变此对象区域的不透明度的画笔。（继承自 <a href="#">UIElement</a> 。）
	Padding	获取或设置控件内的边距。（继承自 <a href="#">Control</a> 。）
	Parent	获取对象树中此 <a href="#">FrameworkElement</a> 的父对象。（继承自 <a href="#">FrameworkElement</a> 。）
	Projection	获取或设置在呈现此 <a href="#">UIElement</a> 时要应用的透视投影（三维效果）。（继承自 <a href="#">UIElement</a> 。）
	RenderSize	获取 <a href="#">UIElement</a> 的最终呈现大小。（继承自 <a href="#">UIElement</a> 。）
	RenderTransform	获取或设置影响 <a href="#">UIElement</a> 的呈现位置的变换信息。（继承自 <a href="#">UIElement</a> 。）
	RenderTransformOrigin	获取或设置由 <a href="#">RenderTransform</a> 声明的任何可能呈现变换的原点，相对于 <a href="#">UIElement</a> 的边界。（继承自 <a href="#">UIElement</a> 。）
	Resources	获取本地定义的资源字典。在 XAML 中，可以通过 XAML 隐式集合语法，将

		资源项建立为 <code>frameworkElement.Resources</code> 属性元素的子对象元素。（继承自 <code>FrameworkElement</code> 。）
	Style	获取或设置呈现过程中应用于此对象的实例 <code>Style</code> 。（继承自 <code>FrameworkElement</code> 。）
	TabIndex	获取或设置一个值，该值决定在用户使用 Tab 键在控件中导航时元素接收焦点的顺序。（继承自 <code>Control</code> 。）
	TabNavigation	获取或设置一个值，该值修改 Tab 键和 <code>TabIndex</code> 针对此控件的工作方式。（继承自 <code>Control</code> 。）
	Tag	获取或设置一个可用于存储有关此对象的自定义信息的任意对象值。（继承自 <code>FrameworkElement</code> 。）
	Template	获取或设置控件模板。（继承自 <code>Control</code> 。）
	Triggers	获取为 <code>FrameworkElement</code> 定义的动画的触发器的集合。（继承自 <code>FrameworkElement</code> 。）
	UseLayoutRounding	获取或设置一个值，该值确定对象及其可视化子树的呈现是否应使用将呈现与整像素对齐的舍入行为。（继承自 <code>UIElement</code> 。）
	VerticalAlignment	获取或设置在父对象（如面板或项控件）中构成 <code>FrameworkElement</code> 时应用于此元素的垂直对齐特征。（继承自 <code>FrameworkElement</code> 。）
	VerticalContentAlignment	获取或设置控件内容的垂直对齐方式。（继承自 <code>Control</code> 。）
	Visibility	获取或设置 <code>UIElement</code> 的可见性。不可见的 <code>UIElement</code> 不呈现，也不将其所需大小告知布局。（继承自 <code>UIElement</code> 。）
	Width	获取或设置 <code>FrameworkElement</code> 的宽度。（继承自 <code>FrameworkElement</code> 。）

页首

方法









显示: ☒ 继承 ☒ 保护

	名称	说明
	AddHandler	为指定的路由事件添加路由事件处理程序，并将该处理程序添加到当前元素的处理程序集合中。将 <code>handledEventsToo</code> 指定为 <code>true</code> 时，可为已标记为由其他元素在事件路由过程中处理的路由事件调用所提供的处理程序。（继承自 <code>UIElement</code> 。）
	ApplyTemplate	加载相关的控件模板，以便可以引用其部件。（继承自 <code>Control</code> 。）

	Arrange	定位子对象并确定 <a href="#">UIElement</a> 的大小。为其子元素实现自定义布局的父对象应从其布局重写实现调用此方法以形成递归布局更新。（继承自 <a href="#">UIElement</a> 。）
	ArrangeOverride	提供 Silverlight 布局的排列处理过程的行为。类可以重写该方法来定义其自己的排列处理过程行为。（继承自 <a href="#">FrameworkElement</a> 。）
	CaptureMouse	将鼠标捕获设置为 <a href="#">UIElement</a> 。（继承自 <a href="#">UIElement</a> 。）
	CheckAccess	确定调用线程是否可以访问此对象。（继承自 <a href="#">DependencyObject</a> 。）
	ClearValue	清除依赖项属性的本地值。（继承自 <a href="#">DependencyObject</a> 。）
	Equals(Object)	确定指定的 <a href="#">Object</a> 是否等于当前的 <a href="#">Object</a> 。（继承自 <a href="#">Object</a> 。）
	Finalize	允许对象在垃圾回收对 <a href="#">Object</a> 进行回收之前尝试释放资源并执行其他清理操作。（继承自 <a href="#">Object</a> 。）
	FindName	检索具有指定标识符名称的对象。（继承自 <a href="#">FrameworkElement</a> 。）
	Focus	尝试设置控件的焦点。（继承自 <a href="#">Control</a> 。）
	GetAnimationBaseValue	返回为 Silverlight 依赖项属性设置的任意基值，在动画不活动时将应用它们。（继承自 <a href="#">DependencyObject</a> 。）
	GetBindingExpression	检索 <a href="#">BindingExpression</a> 中已建立绑定的依赖项属性。（继承自 <a href="#">FrameworkElement</a> 。）
	GetHashCode	用作特定类型的哈希函数。（继承自 <a href="#">Object</a> 。）
	GetTemplateChild	在实例化的 <a href="#">ControlTemplate</a> 可视化树中检索已命名的元素。（继承自 <a href="#">Control</a> 。）
	GetType	获取当前实例的 <a href="#">Type</a> 。（继承自 <a href="#">Object</a> 。）
	GetValue	从 <a href="#">DependencyObject</a> 返回依赖项属性的当前有效值。（继承自 <a href="#">DependencyObject</a> 。）
	InvalidateArrange	使 <a href="#">UIElement</a> 的排列状态（布局）无效。使其无效后，将会更新 <a href="#">UIElement</a> 的布局（以异步方式进行）。（继承自 <a href="#">UIElement</a> 。）
	InvalidateMeasure	使 <a href="#">UIElement</a> 的测量状态（布局）无效。（继承自 <a href="#">UIElement</a> 。）
	Measure	更新 <a href="#">UIElement</a> 的 <a href="#">DesiredSize</a> 。一般地，实现其布局子级的自定义布局的对象从自己的 <a href="#">MeasureOverride</a> 实现调用此方法以形成递归布局更新。（继承自 <a href="#">UIElement</a> 。）
	MeasureOverride	提供 Silverlight 布局的度量处理过程的行为。类可以重写该方法来定义其自



	<code>rride</code>	己的度量处理过程行为。（继承自 <a href="#">FrameworkElement</a> 。）
 	<code>Memberwise Clone</code>	创建当前 <a href="#">Object</a> 的浅表副本。（继承自 <a href="#">Object</a> 。）
 	<code>OnApplyTemplate</code>	当应用新模板时生成 <a href="#">ListBoxItem</a> 控件的可视化树。（重写 <a href="#">FrameworkElement.OnApplyTemplate()</a> 。）
 	<code>OnContentChanged</code>	当 <a href="#">Content</a> 属性的值更改时调用。（继承自 <a href="#">ContentControl</a> 。）
 	<code>OnCreateAutomationPeer</code>	返回一个 <a href="#">ListBoxItemAutomationPeer</a> ，供 Silverlight 自动化基础结构使用。（重写 <a href="#">UIElement.OnCreateAutomationPeer()</a> 。）
 	<code>OnDoubleTap</code>	在 <a href="#">DoubleTap</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnDragEnter</code>	在 <a href="#">DragEnter</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnDragLeave</code>	在 <a href="#">DragLeave</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnDragOver</code>	在 <a href="#">DragOver</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnDrop</code>	在 <a href="#">Drop</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnGotFocus</code>	为 <a href="#">GotFocus</a> 事件提供处理。（重写 <a href="#">Control.OnGotFocus(RoutedEventArgs)</a> 。）
 	<code>OnHold</code>	在 <a href="#">Hold</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnKeyDown</code>	在 <a href="#">KeyDown</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnKeyUp</code>	在 <a href="#">KeyUp</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnLostFocus</code>	为 <a href="#">LostFocus</a> 事件提供处理。（重写 <a href="#">Control.OnLostFocus(RoutedEventArgs)</a> 。）
 	<code>OnLostMouseCapture</code>	在 <a href="#">LostMouseCapture</a> 事件发生之前调用，用于在不附加委托的情况下为派生类中的事件提供处理。（继承自 <a href="#">Control</a> 。）
 	<code>OnManipulationCompleted</code>	在 <a href="#">ManipulationCompleted</a> 事件发生时调用。此成员重写 <a href="#">OnManipulationCompleted</a> 。（重写 <a href="#">Control.OnManipulationCompleted(ManipulationCompletedEventArgs)</a> 。）
 	<code>OnManipulationDelta</code>	在 <a href="#">ManipulationDelta</a> 事件发生之前调用。（继承自 <a href="#">Control</a> 。）
 	<code>OnManipula</code>	在 <a href="#">ManipulationStarted</a> 事件发生时调用。此成员重写

	tionStarted	OnManipulationStarted。 ( 重写 <a href="#">Control.OnManipulationStarted(ManipulationStartedEventArgs)</a> 。 )
	OnMouseEnter	为 <a href="#">MouseEnter</a> 事件提供处理。 ( 重写 <a href="#">Control.OnMouseEnter(MouseEventArgs)</a> 。 )
	OnMouseLeave	为 <a href="#">MouseLeave</a> 事件提供处理。 ( 重写 <a href="#">Control.OnMouseLeave(MouseEventArgs)</a> 。 )
	OnMouseLeftButtonDown	为 <a href="#">MouseLeftButtonDown</a> 事件提供处理。 ( 重写 <a href="#">Control.OnMouseLeftButtonDown(MouseButtonEventArgs)</a> 。 )
	OnMouseLeftButtonUp	在 <a href="#">MouseLeftButtonUp</a> 事件发生之前调用。 ( 继承自 <a href="#">Control</a> 。 )
	OnMouseMove	在 <a href="#">MouseMove</a> 事件发生之前调用。 ( 继承自 <a href="#">Control</a> 。 )
	OnMouseRightButtonDown	在 <a href="#">MouseRightButtonDown</a> 事件发生之前调用。 ( 继承自 <a href="#">Control</a> 。 )
	OnMouseRightButtonUp	在 <a href="#">MouseRightButtonUp</a> 事件发生之前调用。 ( 继承自 <a href="#">Control</a> 。 )
	OnMouseWheel	在 <a href="#">MouseWheel</a> 事件发生之前调用，用于在不附加委托的情况下为派生类中的事件提供处理。 ( 继承自 <a href="#">Control</a> 。 )
	OnTap	在 <a href="#">Tap</a> 事件发生之前调用。 ( 继承自 <a href="#">Control</a> 。 )
	OnTextInput	在 <a href="#">TextInput</a> 事件发生之前调用。 ( 继承自 <a href="#">Control</a> 。 )
	OnTextInputStart	在 <a href="#">TextInputStart</a> 事件发生之前调用。 ( 继承自 <a href="#">Control</a> 。 )
	OnTextInputUpdate	在 <a href="#">TextInputUpdate</a> 事件发生之前调用。 ( 继承自 <a href="#">Control</a> 。 )
	ReadLocalValue	返回依赖项属性的本地值 ( 如果设置了本地值 )。 ( 继承自 <a href="#">DependencyObject</a> 。 )
	ReleaseMouseCapture	从 <a href="#">UIElement</a> 中移除鼠标捕获。 在此调用后，一般没有对象具有鼠标捕获。 ( 继承自 <a href="#">UIElement</a> 。 )
	RemoveHandler	从此 <a href="#">UIElement</a> 中移除指定的路由事件处理程序。 ( 继承自 <a href="#">UIElement</a> 。 )
	SetBinding	使用提供的绑定对象，将绑定附加到 <a href="#">FrameworkElement</a> 上，并返回 <a href="#">BindingExpressionBase</a> 以供以后使用。 ( 继承自 <a href="#">FrameworkElement</a> 。 )
	SetValue	对 <a href="#">DependencyObject</a> 设置依赖项属性的本地值。 ( 继承自 <a href="#">DependencyObject</a> 。 )







	<a href="#">ToString</a>	返回表示当前对象的字符串。（继承自 <a href="#">Object</a> 。）
	<a href="#">TransformTo Visual</a>	返回一个变换对象，该变换对象可用于将 <a href="#">UIElement</a> 中的坐标变换为指定的对象。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">UpdateLayout</a>	请确保 <a href="#">UIElement</a> 的所有子对象位置都正确进行了布局更新。（继承自 <a href="#">UIElement</a> 。）





[页首](#)

## 事件

显示: ☒ 继承 ☒ 保护

	名称	说明
	<a href="#">BindingValidationError</a>	在绑定源报告数据验证错误时发生。（继承自 <a href="#">FrameworkElement</a> 。）
	<a href="#">DataContext Changed</a>	当此元素的数据上下文更改时发生。（继承自 <a href="#">FrameworkElement</a> 。）
	<a href="#">DoubleTap</a>	在此 <a href="#">UIElement</a> 上方提交 <b>DoubleTap</b> 手势时发生。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">DragEnter</a>	在输入系统报告将此元素作为目标的基础拖动事件时发生。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">DragLeave</a>	在输入系统报告将此元素作为来源的基础拖动事件时发生。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">DragOver</a>	在输入系统报告将此元素作为潜在放置目标的基础拖动事件时发生。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">Drop</a>	在输入系统报告将此元素作为放置目标的基础放置事件时发生。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">GotFocus</a>	当 <a href="#">UIElement</a> 收到焦点时发生。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">Hold</a>	在此 <a href="#">UIElement</a> 上方提交 <b>Hold</b> 手势时发生。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">IsEnabledChanged</a>	当 <a href="#">IsEnabled</a> 属性更改时发生。（继承自 <a href="#">Control</a> 。）
	<a href="#">KeyDown</a>	在 <a href="#">UIElement</a> 具有焦点的情况下按下键盘上的某个键时发生。（继承自 <a href="#">UIElement</a> 。）
	<a href="#">KeyUp</a>	在 <a href="#">UIElement</a> 具有焦点的情况下释放键盘上的某个键时发生。（继承自 <a href="#">UIElement</a> 。）

	LayoutUpdated	当 Silverlight 可视化树的布局更改时发生。（继承自 FrameworkElement。）
	Loaded	当已构造 FrameworkElement 并将其添加到对象树中时发生。（继承自 FrameworkElement。）
	LostFocus	当 UIElement 失去焦点时发生。（继承自 UIElement。）
	LostMouseCapture	在 UIElement 失去鼠标捕获时发生。（继承自 UIElement。）
	ManipulationCompleted	对于 UIElement 的操作和延时完毕时发生。（继承自 UIElement。）
	ManipulationDelta	当输入设备在操作期间更改位置时发生。（继承自 UIElement。）
	ManipulationStarted	当输入设备对 UIElement 开始操作时发生。（继承自 UIElement。）
	MediaCommand	在按下遥控器上的某个按钮时发生。（继承自 UIElement。）
	MouseEnter	当鼠标（或触笔）进入 UIElement 的边界区域时发生。（继承自 UIElement。）
	MouseLeave	当鼠标（或触笔）离开 UIElement 的边界区域时发生。（继承自 UIElement。）
	MouseLeftButtonDown	当按下鼠标左键（或触笔的笔尖接触 Tablet）并且鼠标指针悬停在 UIElement 上时发生。（继承自 UIElement。）
	MouseLeftButtonUp	当鼠标（或触笔）悬停在 UIElement 上（或 UIElement 具有鼠标捕获）并且用户松开鼠标左键（或从 Tablet 上移开触笔的笔尖）时发生。（继承自 UIElement。）
	MouseMove	当鼠标（或触笔）的坐标位置更改并且悬停在 UIElement 上（或 UIElement 具有鼠标捕获）时发生。（继承自 UIElement。）
	MouseRightButtonDown	当鼠标指针位于 UIElement 上并按下鼠标右键时发生。（继承自 UIElement。）
	MouseRightButtonUp	当鼠标指针位于 UIElement 上并释放鼠标右键时发生。但是，只有在调用方将前面的 MouseRightButtonDown 事件标记为“已处理”时才引发此事件；请参见“备注”。（继承自 UIElement。）
	MouseWheel	在以下情况下发生：当鼠标指针悬停在 UIElement 上时或 UIElement 具有焦点时，用户滚动鼠标滚轮。（继承自 UIElement。）
	SizeChanged	当 FrameworkElement 上的 ActualHeight 或 ActualWidth 属性的值发生更改时发生。（继承自 FrameworkElement。）
	Tap	在此 UIElement 上方提交 Tap 手势时发生。（继承自 UIElement。）

	TextInput	在 UI 元素以与设备无关的方式获取文本时发生。（继承自 <a href="#">UIElement</a> 。）
	TextInputStart	在 UI 元素以与设备无关的方式最初获取文本时发生。（继承自 <a href="#">UIElement</a> 。）
	TextInputUpdate	当通过输入方法编辑器 (IME) 继续编写文本时发生。（继承自 <a href="#">UIElement</a> 。）
	Unloaded	当此对象不再与主对象树连接时发生。（继承自 <a href="#">FrameworkElement</a> 。）

[页首](#)

## 字段

显示: ☒ 继承 ☒ 保护

	名称	说明
	IsSelectedProperty	标识 <a href="#">IsSelected</a> 依赖项属性。

[页首](#)

## 备注

[ListBox](#) 包含 [ListBoxItem](#) 对象的集合。若要选择 [ListBox](#) 中的 [ListBoxItem](#)，请将 [IsSelected](#) 属性设置为 **true**。

通过将 [ListBox](#) 的 [ItemsSource](#) 属性绑定到数据源，可将 [ListBoxItem](#) 绑定到数据。


通过将 [ListBox](#) 的 [ItemTemplate](#) 属性设置为 [DataTemplate](#)，可以自定义 [ListBoxItem](#) 的外观。

默认情况下，[ListBoxItem](#) 的 [HorizontalAlignment](#) 设置为 [Stretch](#)。[StackPanel](#) 的默认水平位置为 [Center](#)。如果通过 [StackPanel](#) 设置 [ListBoxItem](#) 的 [Width](#) 属性，将应用面板的默认值，并且该项会居中。[ListBoxItem](#) 为 [ContentControl](#)。其内容属性为 [Content](#)。

### 自定义 ListBoxItem 控件

若要对多个 [ListBoxItem](#) 控件应用相同的属性设置，请使用 [Style](#) 属性。或者，还可以设置父级 [ListBox](#) 的 [ItemContainerStyle](#) 属性以更改 [ListBox](#) 中包含的所有项的样式。若要更改 [ListBoxItem](#) 的可视结构和可视行为，请复制并修改它的默认样式和模板。有关更多信息，请参见 [控件自定义](#)。

可通过控件的默认样式设置此控件的依赖项属性。如果 [ListBoxItem](#) 的依赖项属性由它的默认样式设置，则当 [ListBoxItem](#) 显示在应用程序中时，该属性的值可能会由默认值更改为其他值。有关更多信息，请参见 [依赖项属性值优先级](#)。您可以从 [ListBox 样式和模板](#) 获取 [ListBoxItem](#) 的默认样式和模板。

 说明：

仅当可视属性显示在 [ListBoxItem](#) 控件的默认模板中，并且是使用 [TemplateBinding](#) 设置的时，设置该可视属

性才有效。在[通过使用 ControlTemplate 自定义现有控件的外观](#)的"更改控件的可视结构"部分中，可以找到可视属性的列表。

## 示例

下面的示例演示如何使用 XAML 将 ListBoxItem 对象直接添加到 [ListBox](#) 控件。

### XAML

```
<ListBox Width="150" Margin="0,5,0,5" SelectionChanged="PrintText"
HorizontalAlignment="Left">
    <ListBoxItem Content="Item 1" />
    <ListBoxItem Content="Item 2" />
    <ListBoxItem Content="Item 3" />
    <ListBoxItem Content="Item 4" />
    <ListBoxItem Content="Item 5" />
</ListBox>
<TextBlock Name="textBlock1" />
```

## 版本信息

Silverlight

受以下版本支持：5、4、3

Silverlight for Windows Phone

受以下版本支持：Windows Phone OS 7.1、Windows Phone OS 7.0

## 平台

有关 Silverlight 支持的操作系统和浏览器的列表，请参见 [支持的操作系统和浏览器](#)。

## 线程安全

此类型的任何公共 **static**（在 Visual Basic 中为 **Shared**）成员都是线程安全的。但不保证所有实例成员都是线程安全的。

## 另请参见

参考

[System.Windows.Controls 命名空间](#)

其他资源

[如何：在 ListBox 中显示数据](#)

[ListBox 样式和模板](#)

---

# 社区附加资源

---

© 2017 Microsoft