

C1Toolbar指令教程（Silverlight版）

Silverlight 4 支持在任意继承ButtonBase类的对象中使用ICommand接口，包括C1ToolbarButton在内的任意按钮可以通过控件的Command属性关联实现ICommand的对象，尽管支持ICommand接口，但Silverlight并不提供任何内置的实现，Silverlight Edition包括C1Command 及C1ToolbarCommand 两个实现，因此您不需要自己来编写。C1Command 类被包含在Cl.Silverlight 配置中，C1ToolbarCommand被包含在Cl.Silverlight.Toolbar配置中，您可以通过添加几个额外的工具栏，并关联相应的标签和图片属性以扩展 C1Command 。以下步骤演示了如何使用C1Toolbar指令。

使用使用C1ToolbarCommand

- 1. 打开或创建一个新的Silverlight 应用程序。
- 2. 添加两个RowDefinitions 到默认的Grid 元素，将第一行设置为自动高度，如：

XAML

```
<Grid.RowDefinitions>
<RowDefinition Height="Auto" />

<RowDefinition />
</Grid.RowDefinitions>
```

- 3. 添加一个C1Toolbar以填充第一行。
- 4. 在第二行中添加一个TextBox。
- 5. 粘贴下述C1ToolbarCommand 到UserControl资源：

XAML

```
<UserControl.Resources>
<c1:C1ToolbarCommand x:Key="cmdClear" LabelTitle="Clear Text"

LargeImageSource="/Resources/delete.png" />
</UserControl.Resources>
```

- 6. 该指令将用于清除TextBox中内容，您可以为C1ToolbarCommand指定一个LargeImageSource 和SmallImageSource，但这不是必须的。
- 7. 粘贴以下XAML 来填充带有标签或者分组按钮的C1Toolbar：

XAML

```
<c1:C1Toolbar Name="c1Toolbar1">
<c1:C1ToolbarTabControl>
<c1:C1ToolbarTabItem Header="Home">
<c1:C1ToolbarGroup Header="Application">
<c1:C1ToolbarButton c1:CommandExtensions.Command="{StaticResource cmdClear}" />

</c1:C1ToolbarGroup>
</c1:C1ToolbarTabItem> </c1:C1ToolbarTabControl>
</c1:C1Toolbar>
```

- 8. 这里我们设置c1:CommandExtensions.Command 属性到我们的指令中，注意C1ToolbarButton 也有一个继承的Command属性，当使用C1Commands时最好使用附加属性CommandExtensions.Command 来设置指令。
- 9. 添加一个TextBox 到页面下方名为“textBox1”的工具栏。
- 10. 接下来，您需要注册该指令后，在页面初始化使用此代码：

C#

```
// 注册指令方法
CommandManager.RegisterClassCommandBinding(GetType(), new
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1"
ac:macro-id="f00821c1-3cf9-4eeb-9927-8942218c8dfe"><ac:plain-text-body><![CDATA[CommandBinding((C1ToolbarCommand)Resources[
"cmdClear"], Clear, CanClear));
]]></ac:plain-text-body></ac:structured-macro>
// 检测已注册指令的执行能力
CommandManager.InvalidateRequerySuggested();
```

- 1. Cl.Silverlight.CommandManager 提供用于注册指令的实用方法。 12. 粘贴以下Clear 及 CanClear 事件处理以执行指令逻辑：

C#

```
private void Clear(object sender, ExecutedRoutedEventArgs e)
{
    textBox1.Text = "";
}
private void CanClear(object sender, CanExecuteRoutedEventArgs e)
{
    if (textBox1.Text.Length > 0)
    {
        e.CanExecute = true;
    }
    else
    {
        e.CanExecute = false;
    }
}
```

2. 在Silverlight版中您需要显式的在代码中检查已注册指令的能力，需要在TextBox 的TextChanged 事件中添加以下代码：

```
C#

private void textBox1_TextChanged(object sender, TextChangedEventArgs e) {
    CommandManager.InvalidateRequerySuggested();
}
```

3. 运行示例，您现在可以通过在C1Toolbar中使用C1Command来清除文本框中的内容。

