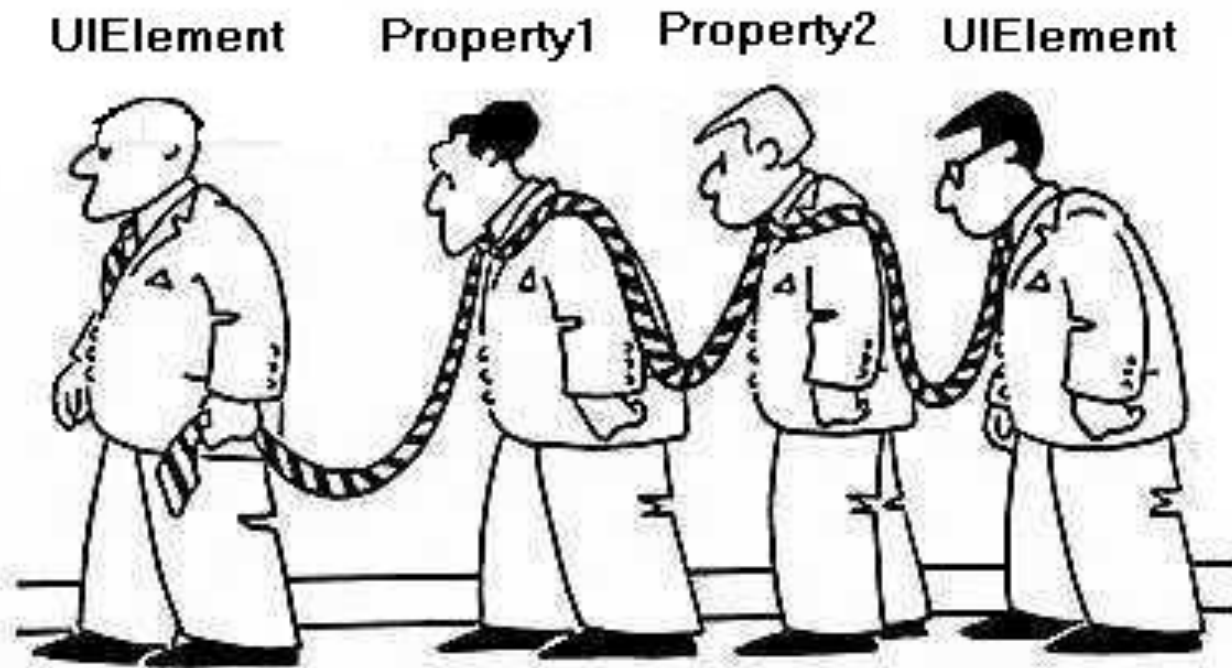


Привязка данных (Data Binding)



Привязка данных — это акт подключения свойств элемента управления к значениям данных, которые могут изменяться на протяжении жизненного цикла приложения.

Привязка данных WPF предоставляет приложениям простой и последовательный способ представления и взаимодействия с данными. Элементы можно связать с данными из разнообразных источников, данных в форме объектов среды CLR и XML .

Класс **Binding** обеспечивает высокоуровневый доступ к определению привязки, соединяющей **свойства объектов**, выступающих в качестве цели привязки (как правило, элементов WPF), с **любым источником данных** (например, с базой данных, файлом XML или любым другим объектом, содержащим данные).

Привязка данных

Источник привязки



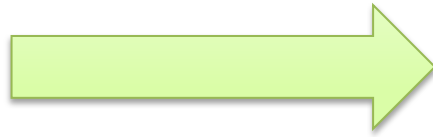
```
<Slider Name="silderFontSize"  
        Minimum="12"  
        Maximum="40"  
        Margin="10"></Slider>
```

```
<TextBlock Margin="10"  
           FontSize="{Binding ElementName=silderFontSize, Path=Value}"  
           Hello world  
</TextBlock>
```

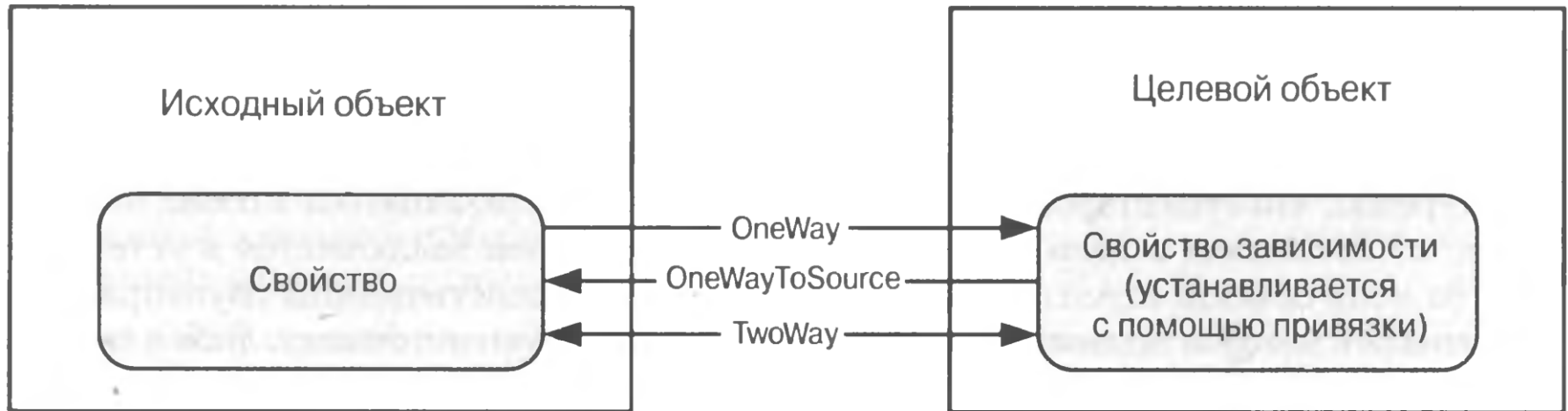


Целевой объект привязки

Источник



Цель



Различные направления привязки свойств

OneWay	Целевое свойство обновляется при изменениях исходного свойства
TwoWay	Целевое свойство обновляется при изменениях исходного свойства, а исходное свойство обновляется при изменении целевого свойства
OneTime	Целевое свойство устанавливается изначально на основе значения исходного свойства. Обычно этот режим используется для сокращения накладных расходов, если известно, что целевое свойство не изменится
OneWayToSource	Подобно OneWay, но действует в обратном направлении. Исходное свойство обновляется, когда изменяется целевое свойство , но целевое свойство никогда не обновляется
Default	Этот тип привязки зависит от целевого свойства. Это либо TwoWay (для устанавливаемых пользователем свойств, таких как TextBox.Text), либо OneWay (для всего остального).

```
// создание объекта привязки данных
Binding binding = new Binding();
// указываем источник привязки (объект с которым связываемся)
binding.Source = sliderFontSize;
// указываем свойство, к которому привязываемся
binding.Path = new PropertyPath("Value");
// указываем тип привязки
binding.Mode = BindingMode.TwoWay;
// устанавливаем целевой объект и целевое свойство
tbText.SetBinding(TextBlock.FontSizeProperty, binding);
```

Привязка в коде применяется при:

- **Создании динамических привязок.**
- **Удалении привязки и установка новой.**

Чтобы удалить привязку недостаточно просто присвоить новое значение свойству. Необходимо получить возможность установки свойства обычным образом, понадобится помощь метода `ClearBinding()` или `ClearAllBindings()`..

- **Создании специальных элементов управления.**

Значения перечисления UpdateSourceTrigger

- **PropertyChanged** – обновление происходят сразу после изменения значения свойства.
- **LostFocus** – обновление происходит после изменения значения и потери фокуса .
- **Explicit** – обновления происходят после вызова метода `BindingExpression.UpdateSource()`;
- **Default** – Для большинства свойств значение `PropertyChanged` для `TextBox.Text` - `LostFocus`

управляют тем, как обновляется источник в привязках `TwoWay` и `OneWayToSource`.

Привязка к объектам, не являющимся элементами

- **Source** – ссылка на объект источник.
 - **RelativeSource** – позволяет создать ссылку на текущий элемент или на другой элемент вверх по дереву.
 - **DataContext** – указание источника для группы элементов управления.
-
- *DataContext* используется в случаях привязки несколько свойств одного объекта к разным элементам другого.
 - *DataContext* устанавливается для контейнера, вместо его установки непосредственно на целевой элемент.