

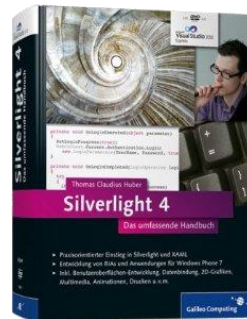
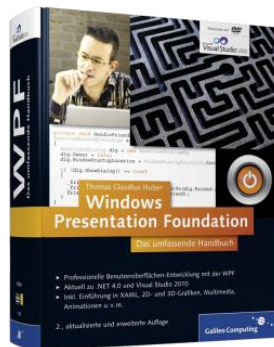


Thomas Claudius Huber | Trivadis AG

Fortgeschrittenes DataBinding bei der WPF

Über Thomas Claudius Huber

- NET Senior Consultant
- Trivadis AG (Basel)
- Spezialisiert auf WPF/Silverlight
- Autor der umfassenden Handbücher zu WPF und Silverlight



- www.thomasclaudiushuber.com

Agenda

- Grundlagen des Data Binding
- Verschiedene Datenquellen
- Validieren von Daten

Das Data-Binding

- In XAML via Binding-Markup Extension

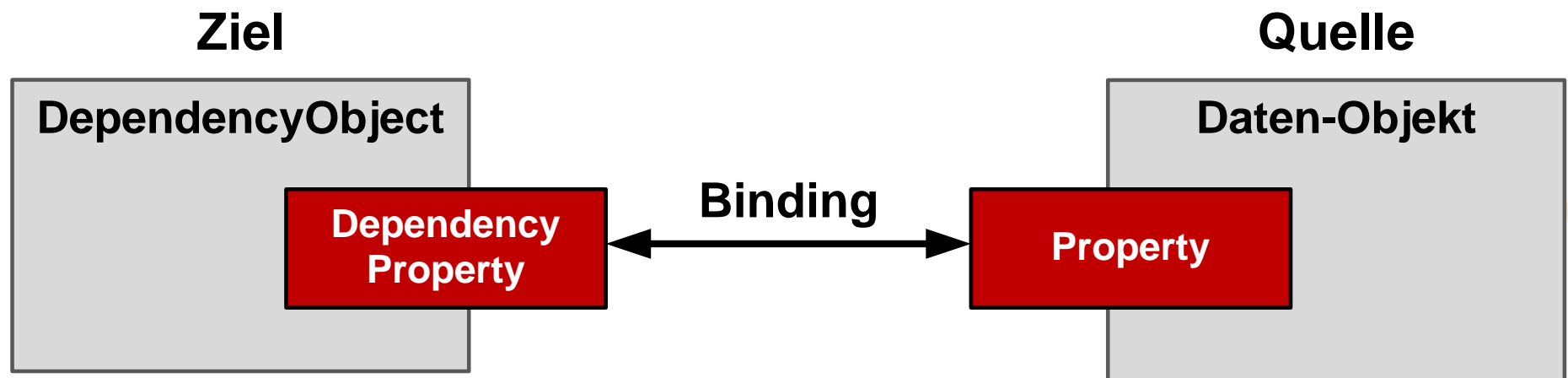
```
<TextBox Text="{Binding ElementName=sliSource,  
                        Path=Value}"/>  
<Slider x:Name="sliSource" Value="0"/>
```

- In C# via SetBinding-Methode

```
Binding b = new Binding();  
b.ElementName = sliSource.Name;  
b.Path = new PropertyPath("Value");  
txtTarget.SetBinding(TextBox.TextProperty, b);
```

Ziel und Quelle

- Das Ziel eines Bindings muss eine Dependency Property sein
- Die Quelle kann eine normale Property sein



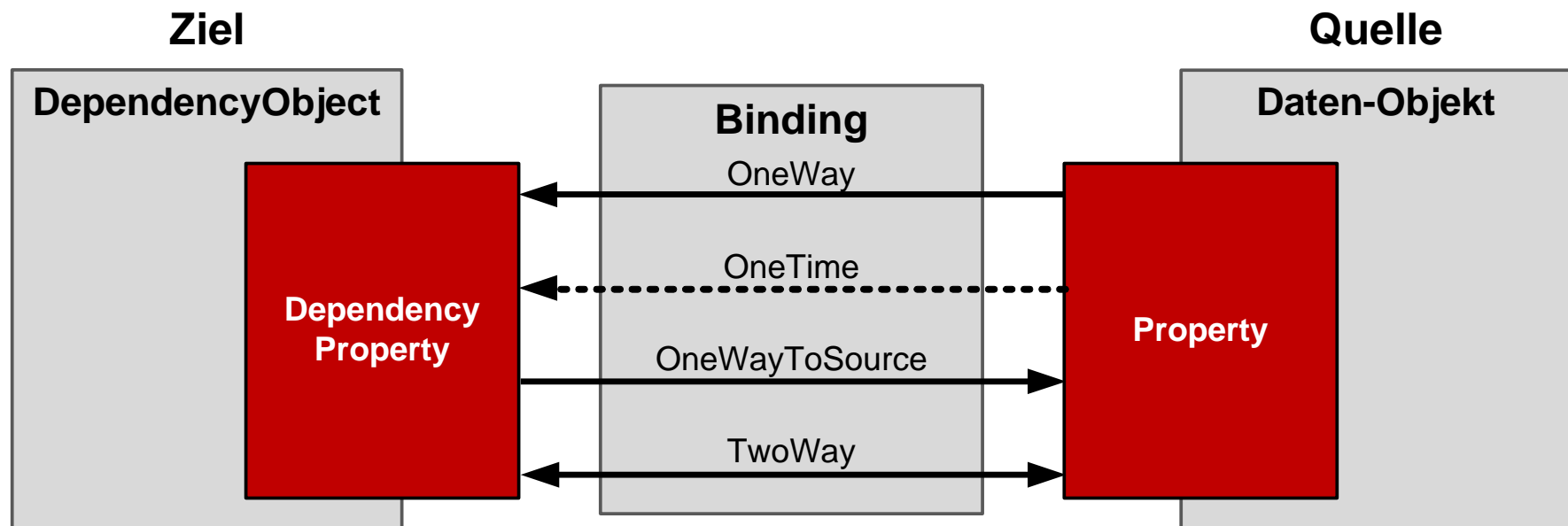
Die Quelle des Data Binding

- Die Quelle wird festgelegt über die
 - Source-Property
 - ElementName-Property
 - RelativeSource-Property
- Ist keine Property gesetzt, wird der DataContext verwendet
- Nach dem DataContext wird aufwärts im Elementbaum gesucht



Der «Mode» des Data Binding

- Mit der Mode-Property wird die Richtung des Bindings angegeben



- Für TwoWay/OneWayToSource wird UpdateSourceTrigger-Property genutzt
 - Zum Bestimmen, wann Quelle aktualisiert wird

Die BindingExpression

- Binding-Instanz definiert nur die Metadaten
 - Die „Arbeit“ wird durch eine BindingExpression-Instanz erledigt
- BindingExpression-Instance gibt es als Rückgabewert von
 - der SetBinding-Methode
 - BindingOperations.GetBindingExpression
- BindingExpression hat Methoden wie UpdateSource oder UpdateTarget

Agenda

- Grundlagen des Data Binding
- Verschiedene Datenquellen
- Validieren von Daten

Datenquellen

- Mit einer Dependency Property ist alles gut 😊
- Normale .NET Property benötigt Änderungsmechanismus
 - Klasse mit der Property implementiert dazu `INotifyPropertyChanged`
- Properties unterschiedlichen Typs mit Convertern versehen
 - In MVVM ist Converter nicht zwingend notwendig

Demo

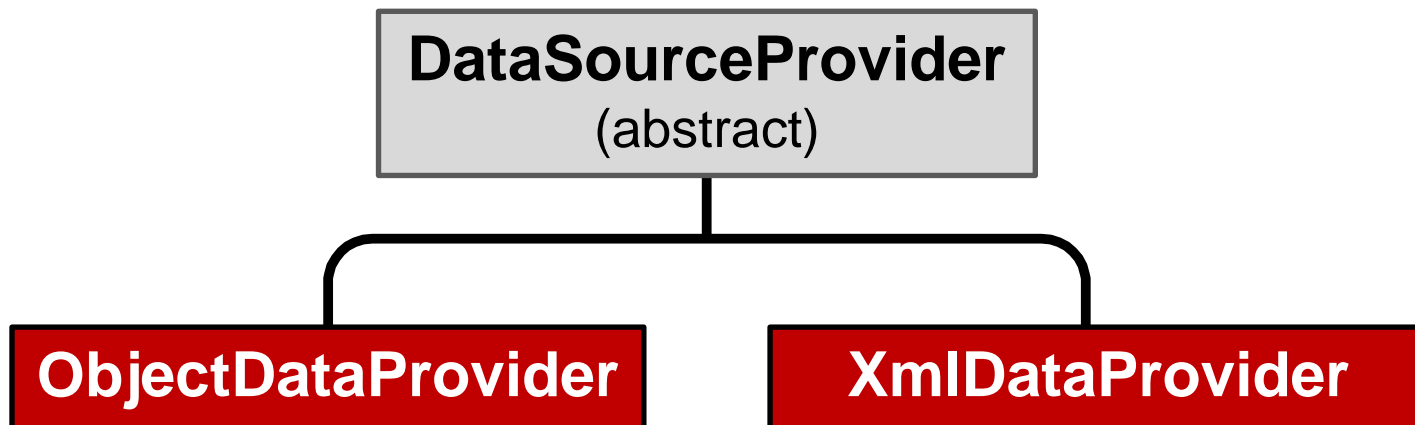
Multibinding & PriorityBinding

- Binding ist für genau eine Ziel- und genau eine Quell-Property
- Multibinding verbindet mehrere Quell-Properties mit einer Ziel-Property
 - Dazwischen ist ein IMultiValueConverter
- PriorityBinding priorisiert Quell-Properties
 - Wert von nur einer Quell-Property wird verwendet

Demo

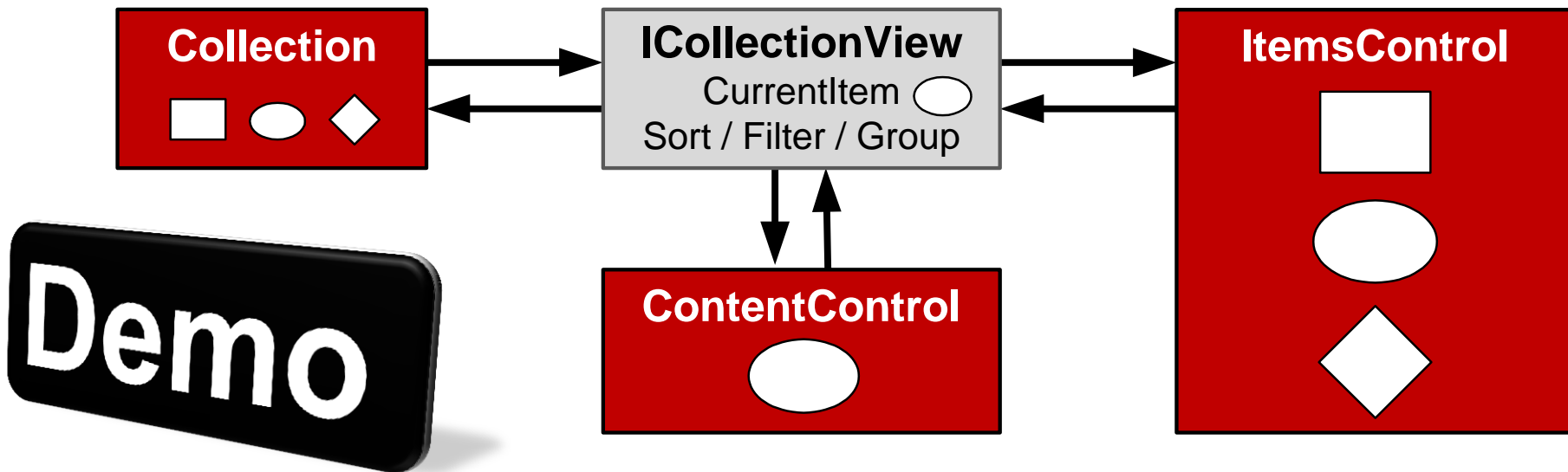
DataSourceProvider

- Es gibt zwei DataSourceProvider, ...
- ... einen zum Erstellen von Objekten, die keinen Default-Konstruktor besitzen
- ... einen zum Anbinden von XML-Daten



Collections als Quelle

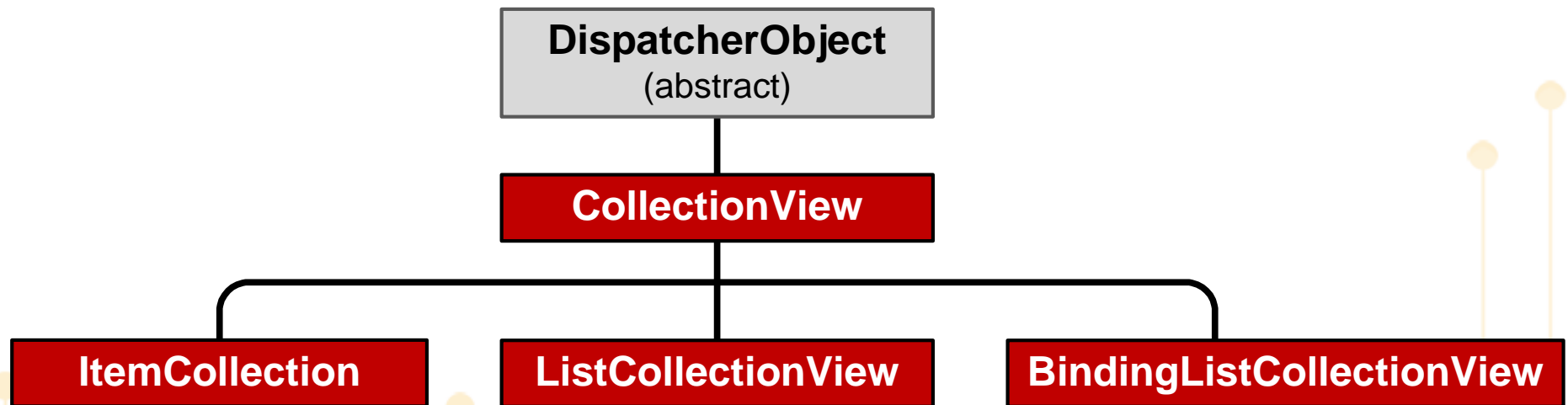
- WPF verwendet ICollectionViews für das Currency-Management



- WPF verwendet eine Default-CollectionView, falls nicht explizit eine erstellt wurde
 - `CollectionViewSource.GetDefaultView(...)`

Die CollectionView-Klassen

- Vom ICollectionView-Interface gibt es verschiedene Implementierungen
 - CollectionView für IEnumerable
 - ListCollectionView für IList
 - BindingListCollectionView für IBindingList



Data Binding an Collections

- Collection muss benachrichtigen, wenn Elemente hinzugefügt/entfernt werden
- Dazu gibt es das Interface **INotifyCollectionChanged**
- `ObservableCollection<T>` implementiert das Interface bereits

Demo

Agenda

- Grundlagen des Data Binding
- Verschiedene Datenquellen
- Validieren von Daten

Validieren von Daten

- Binding-Klasse hat ValidationRules-Property
- Nimmt ValidationRule-Objekte entgegen. Es gibt drei Möglichkeiten:
 - ExceptionValidationRule
 - DataErrorValidationRule
 - Eigene Subklasse von ValidationRule
- Für Exceptions und DataErrors gibt es seit .NET 3.5 auch die Properties ValidatesOnExceptions und ValidatesOnDataError

Demo

Validierungsfehler

- Fehler werden in der `Attached-Property Validation.Errors` gespeichert
- `TextBox` zeigt bei Fehlern automatisch roten Rahmen an
- Der rote Rahmen lässt sich mit einem der `Attached-Property Validation.ErrorTemplate` zugewiesenen `ControlTemplate` ersetzen
 - ein `AdornedElementPlaceholder` legt im `ControlTemplate` die Stelle fest, an welcher die `TextBox` gezeichnet wird.

Demo

Agenda

- Grundlagen des Data Binding
- Verschiedene Datenquellen
- Validieren von Daten

Kernaussagen

- Ziel eines Data-Binding muss eine Dependency Property sein
- CurrencyManagement wird durch ICollectionViews erledigt
- Validierung mit IDataErrorInfo sehr komfortabel

Mail: thomas@thomasclaudiushuber.com
Twitter: www.twitter.com/thomasclaudiush
Blog: www.thomasclaudiushuber.com/blog

Danke!

Fragen?