

Binding Allgemein

Unter Binding versteht man, eine Eigenschaft von dem Quellobjekt an eine Eigenschaft des Zielobjekts bindet.

Dabei gibt es 2 Arten:

Element an ein anderes Element / Element an Datenquelle z.B. Datenbank



Dabei bindet man eine Abhängigkeits-Eigenschaft (Dependency Property) an die andere. Diese müssen gleichen Typs sein. Sind diese Anforderungen erfüllt, kann man alles binden. Breite eines Steuerelements an die Fensterbreite, selektiertes Element einer Combobox an ein Label und sogar ganze Spalte aus einer Datenbank.

Das Visual Studio bietet hierfür eine Hilfe in den Eigenschaften der Steuerelemente an.



Hier sehen wir nun, 7 Bindungstypen, die uns zur Verfügung stehen. Jedes dieser Bindungen umfasst eine Ansammlung von Eigenschaften aus seinem Gebiet.

1. **Datenkontext:** Jede WPF Anwendung hat zu Beginn 2 Schichten. Eine für UI (User Interface) und die andere

für Daten. Die Datenschicht ist zu Beginn Null, das heißt ohne Inhalt. Indem wir deren Eigenschaft DataContext setzen umfasst dieser Kontext nun alle angegebenen Daten. Solange man nichts weiter angibt, erben nun die Kindobjekte (Steuerelemente wie Label, Textbox usw.) von dem Elternobjekten diese Daten.

2. **Datenquelle:**

3. **ElementName:** Umfasst alle visuelle und nicht visuelle Elemente aus dem XAML Projekt. Hier stehen einem sämtliche Eigenschaften der Steuerelemente wie Textbox, Label, Combobox und auch nicht die Eigenschaften vom Window zur Verfügung.

4. **RelativeSource FindAncestor:** Hier werden sämtliche Eigenschaften von allen hierarchisch überliegenden Objekten zur Verfügung gestellt.

5. **RelativeSource PreviousData:** Gegenteil vom FindAncestor, werden hier alle unterliegenden Objekteigenschaften aufgelistet

6. **RelativeSource Self:** Hier bekommt man Eigenschaften vom selben Objekt zur Verfügung gestellt

7. **RelativeSource TemplatedParent:**

8. **StaticResource:** Bezeichnet eine schon zuvor in die Resources eingeschriebene Datenquelle