WPF Binding CheatSheet version 1.1

You can always find the latest version of this cheat sheet at http://go.nbdtech.com?94E138EA

Part I – Common Examples

Basic Binding	
{Binding}	Bind to current DataContext.
{Binding Name}	Bind to the “Name” proeprty of the current DataContext.
{Bindind Name.Length}	Bind to the Length property of the object in the Name property of the current DataContext.
{Binding ElementName=SomeTextBox, Path=Text}	Bind to the “Text” property of the element XAML element with name=”SomeTextBox” or x:Name=”SomeTextBox”.

XML Binding	
{Binding Source={StaticResource BooksData} XPath=/books/book}	Bind the result of XPath query “/books/book” from the XML in the XmlDataProvider in a parent’s “Resources” elememt with x:Key=”BooksData”.
{Binding XPath=@name}	Bind to the result of an XPath query run on the XML node in the DataContext (for example in an ItemControl’s DataTemplate when the ItemsControl.ItemsSource is bound to an XML data source).

Relative Source Binding	
{Binding RelativeSource={RelativeSource Self}}	Bind to the target element.
{Binding RelativeSource={RelativeSource Self}, Path=Name}	Bind to the “Name” property of the target element.
{Binding RelativeSource={RelativeSource FindAncestor, AncestorType={x:Type Window}}, Path=Title}	Bind to the title of the parent window.
{Binding RelativeSource={RelativeSource FindAncestor, AncestorType={x:Type ItemsControl}, AncestorLevel=2}, Path=Name}	Bind the the name of the 2nd parent of type ItemsControl.
{Binding RelativeSource={RelativeSource TemplatedParent}, Path=Name}	Inside a control template, bind to the name property of the element the template is applied to.
{TemplateBinding Name}	Shortcut for the previous example.

Collection Current Item Binding	
{Binding /}	Bind to the current item in the DataContext (when DataContext is a collection)
{Binding AllItems/}	Bind to the current item in the “AllItems“ property of the DataContext
{Binding AllItems/Name}	Bind to the “Name” property of the current item in the “AllItems“ property of the DataContext

* Someone has to manage the collection current item, you can do it in code using the CollectionView class or set IsSynchronizedWithCurrentItem="True" on a control that supports it (like ListBox) and is bound to the same collection.

Part II – Alphabetical list of all Binding's properties

Property	Description
BindingGroupName (3.5sp1)	The name of the BindingGroup to which this binding belongs. A BindingGroup is used to validate multiple bindings together (for example when multiple changes should be submitted all at once).
BindsDirectlyToSource	When using a DataSourceProvider derived class (for example a ObjectDataProvider) setting this property to true will bind to the data source provider object itself, leaving it false will bind to the data contained in the data source.
Converter	The converter to use, usually you create the converter in a parent element's Resources element and reference it using a {StaticResource name) or create the converter as a static field and reference it with {x:Static ns:class.field}
ConverterCulture	The culture passed to the converter.
ConverterParameter	The parameter passed to the converter
ElementName	Element name, when binding to an element in the same XAML scope. Can't be used if RelativeSource or Source is set.
FallbackValue	Value to use when the Binding encounters an error
IsAsync	Use when the property's get accessor takes a long time, to avoid blocking the UI thread, While waiting for the value to arrive, the binding reports the FallbackValue.
Mode	Direction of binding, possible options: • `TwoWay` - updates the target property or the source property whenever the other one changes. • `OneWay` - updates the target property only when the source property changes. • `OneTime` - updates the target property only when the application starts or when the DataContext undergoes a change. • `OneWayToSource` - updates the source property when the target property changes, useful the target property is not a dependency property – put the binding on what would normally be the source and point it to the target. • `Default` - causes the default Mode value of target property to be used.
NotifyOnSourceUpdated	Raise the SourceUpdated event when a value is transferred from the binding target to the binding source.
NotifyOnTargetUpdated	Raise the TargetUpdated event when a value is transferred from the binding source to the binding target.
NotifyOnValidationError	Raise the Error attached event on the bound object.
Path	Source property.
RelativeSource	Binding source relative to the target, possible options: • `{x:Static RelativeSource.Self}` or `{RelativeSource Self}` bind to target element. • `{RelativeSource FindAncestor, AncestorType={x:Type TypeName}}` Bind to the first parent of type TypeName • `{RelativeSource FindAncestor, AncestorType={x:Type TypeName}, AnsestorLevel=n}` Bind to the n th parent of type TypeName • `{RelativeSource TemplatedParent}` bind to the element this template is applied to (useful in control templates, consider

	using TemplateBinding instead. Can’t be used if ElementName or Source is set.
Source	Object to use as the binding source. Can’t be used if ElementName or RelativeSource is set
StringFormat (3.5sp1)	Format string to use when converting the bound value to a string. Works only if the target property is of type string.
TargetNullValue (3.5sp1)	Value to use when the bound value is null.
UpdateSourceExceptionFilter	Custom logic for handling exceptions that the binding engine encounters. Only if you add an ExceptionValidationRule to ValidationRules or set ValidatesOnExceptions.
UpdateSourceTrigger	timing of binding source updates, possible options: • `Default` - The default UpdateSourceTrigger value of the binding target property. The default is usually PropertyChanged, while the Text property is LostFocus. • `PropertyChanged` - Updates the binding source immediately whenever the binding target property changes. • `LostFocus` - Updates the binding source whenever the binding target element loses focus. • `Explicit` - Updates the binding source only when you call the UpdateSource method.
ValidatesOnDataErrors (3.5sp1)	Use IDataErrorInfo when validating.
ValidatesOnExceptions (3.5sp1)	Treat exceptions as validation failures.
ValidationRules	Collection of rules that check the validity of the user input.
XPath	XPath query that returns the value on the XML binding source to use. Top

The Internationalization Fix

By default, when you use data binding and the target property is a string, WPF will format your value using the US English culture, to use the correct setting the user seletceted in the control panel add the following code before loading any GUI (the Application.Startup event is a good place)

```
FrameworkElement.LanguageProperty.OverrideMetadata(
                  typeof(FrameworkElement),
                  new FrameworkPropertyMetadata(
                     XmlLanguage.GetLanguage(
                     CultureInfo.CurrentCulture.IetfLanguageTag)));
```