Contents

This sample shows DevForce validation at work in a Silverlight application.

- **Platform:** Silverlight
- **Language:** C#, VB
- **Download:** Validation (Silverlight)

Problem

You'd like to better understand DevForce validation and see it in action.

Solution

The sample shows a simple application displaying employees and their orders. As you modify various fields you can see property-level validation in action. For example, the *First Name* and *Last Name* fields can't contain more than 30 characters, or the *Birth Date* can't come after the *Hire Date*.

These are examples of both simple declarative validation through standard attributes and more complex custom verifiers.

By default when you generated the code for your model DevForce added validation attributes for string length restrictions and required fields:

```
[IbVal.RequiredValueVerifier( ErrorMessageResourceName="Employee_EmployeeID")]
public int EmployeeID {...}
[IbVal.StringLengthVerifier(MaxValue=30, IsRequired=true, ErrorMessageResourceName="Employee_LastName")]
public string LastName {..}
```

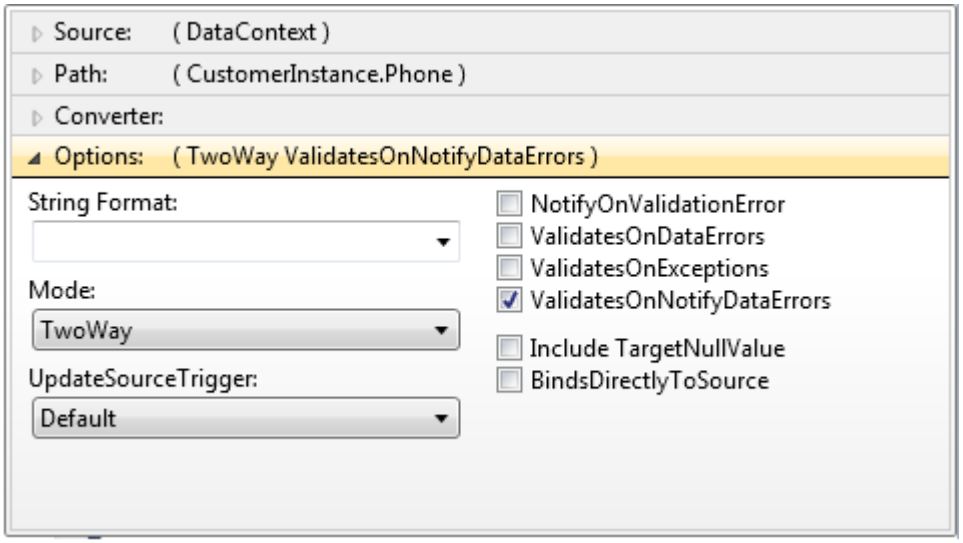
```
<IbVal.RequiredValueVerifier(ErrorMessageResourceName:="Employee_EmployeeID")>
Public ReadOnly Property EmployeeID() As Integer
End Property
<IbVal.StringLengthVerifier(MaxValue:=30, IsRequired:=True, ErrorMessageResourceName:="Employee_LastName")>
Public ReadOnly Property LastName() As String
End Property
```

These validation attributes help ensure that your simple database-defined rules are enforced.

You can also add complex validation rules to enforce your business logic. In the sample you'll see several: rules which extend the DevForce *RegexVerifier* and *DataTimeRangeVerifier*, and a fully custom rule using the *DelegateVerifier* .

If you take a peek at the XAML for the MainPage you'll see ... only that the *DataForm* and *DataGrid* are auto-generating their content. By default, Silverlight bindings *listen* on the INotifyDataErrorInfo and IDataErrorInfo interfaces, both of which DevForce entities implement. Unless you need to customize when validation is performed or the display of validation errors, you don't need to do anything special to get validation in Silverlight.

When you do need to set the validation options on a binding, it's easy using the designer:



For more information on the data binding interfaces implemented by entities, see the Display topic.

Prerequisites

- The Silverlight 4 Toolkit