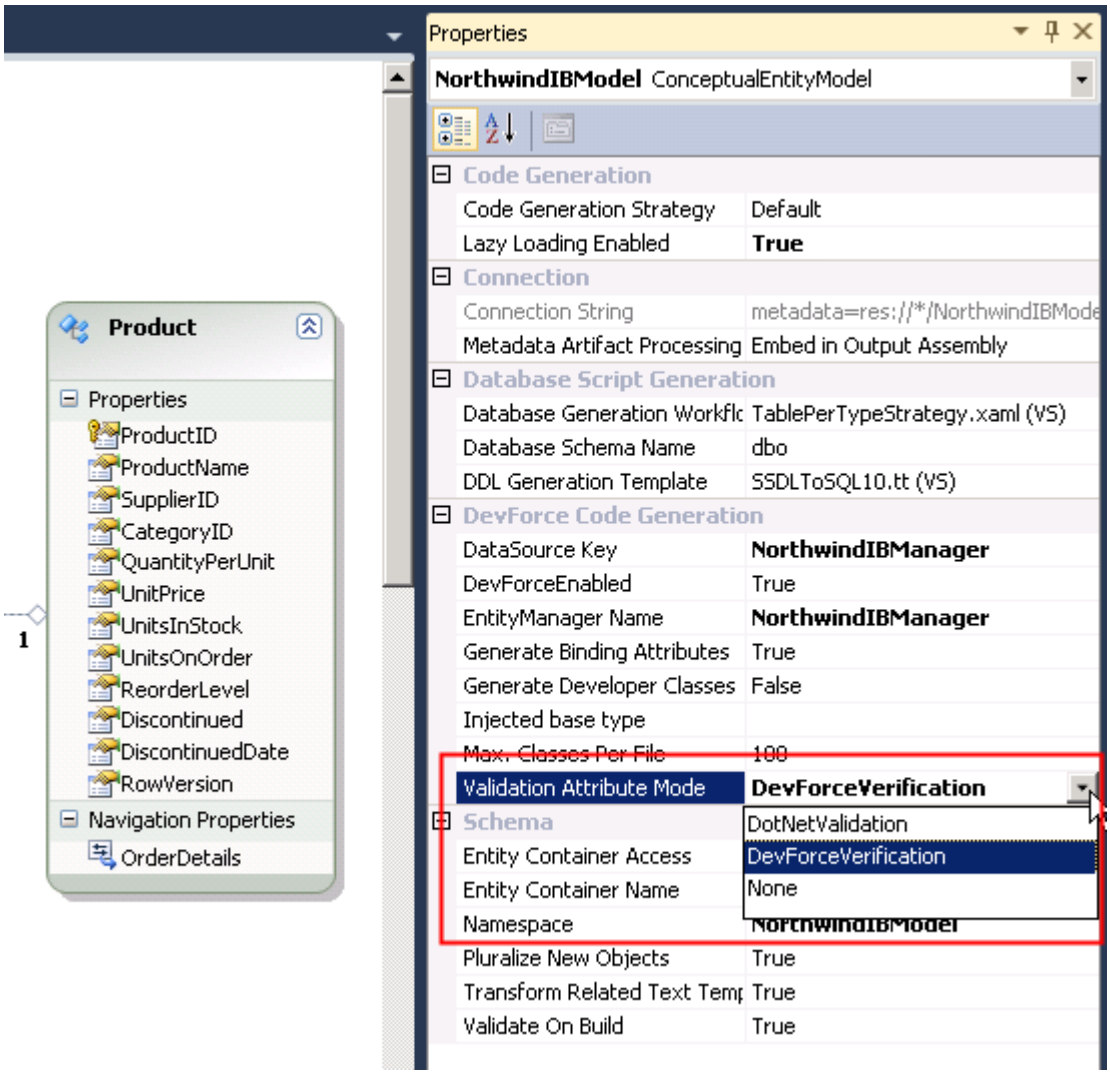
Contents

You can **automatically generate validation attributes** in the code generated from your Entity Data Model.

Validation-related code generation options

1. Open an Entity Data Model in the Visual Studio Entity Data Model Designer.
2. Display the Properties panel and then click in white space in the designer window to display the properties of the *ConceptualEntityModel*. In the DevForce Code Generation section, note the property *Validation Attribute Mode*:



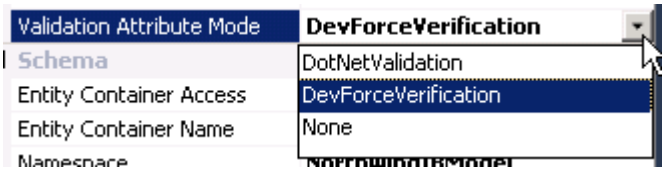
1. You can choose to have DevForce generate DevForce-style validation attributes for entities and their properties, or .NET-style.

We recommend that you use use *DevForceVerification* with Database First models. DevForce verification provides a superset of the capabilities provided by .NET validation.

Generated property code

Here we'll show you the results of the three different values for Validation Attribute Mode:

Generate DevForceVerification Attributes



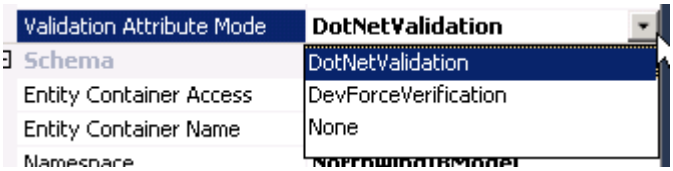
Here is the FirstName property of an Employee object as generated with the settings shown above:

```
#region FirstName property
/// <summary>Gets or sets the FirstName. </summary>
[Bindable(true, BindingDirection.TwoWay)]
[Editable(true)]
[Display(Name="FirstName", AutoGenerateField=true)]
[IbVal.StringLengthVerifier(MaxValue=30, IsRequired=true,
  ErrorMessageResourceName="Employee_FirstName")]
[DataMember]
public string FirstName {
get { return PropertyMetadata.FirstName.GetValue(this); }
set { PropertyMetadata.FirstName.SetValue(this, value); }
}
#endregion FirstName property
```

```
#Region "FirstName property"
 ''' <summary>Gets or sets the FirstName. </summary>
 <Bindable(True, BindingDirection.TwoWay), Editable(True), _
    Display(Name:="FirstName", AutoGenerateField:=True), _
    IbVal.StringLengthVerifier(MaxValue:=30, IsRequired:=True, _
    ErrorMessageResourceName:="Employee_FirstName"), DataMember()> _
 Public Property FirstName() As String
   Get
     Return PropertyMetadata.FirstName.GetValue(Me)
   End Get
   Set(ByVal value As String)
     PropertyMetadata.FirstName.SetValue(Me, value)
   End Set
 End Property
#End Region ' FirstName property
```

IbVal is an alias for the *IdeaBlade.Validation* namespace, defined at the top of the code file. The *IbVal.StringLengthVerifier* sets a maximum length on the (text) value, and its *IsRequired* argument declares the property non-nullable.

Generate .NET validation attributes



Here is the generated code that the above settings in the EDM designer:

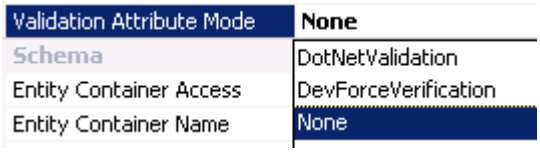
```
#region FirstName property
/// <summary>Gets or sets the FirstName. </summary>
[Bindable(true, BindingDirection.TwoWay)]
[Editable(true)]
[Display(Name = "FirstName", AutoGenerateField = true)]
[Required()]
[StringLength(30)]
[DataMember]
public string FirstName {
get { return PropertyMetadata.FirstName.GetValue(this); }
set { PropertyMetadata.FirstName.SetValue(this, value); }
}
#endregion FirstName property
```

```
#Region "FirstName property"
 ''' <summary>Gets or sets the FirstName. </summary>
 <Bindable(True, BindingDirection.TwoWay), Editable(True), _
    Display(Name:="FirstName", AutoGenerateField:=True), _
    Required(), _
```

```
    StringLength(30), _
    DataMember()> _
  Public Property FirstName() As String
    Get
      Return PropertyMetadata.FirstName.GetValue(Me)
    End Get
    Set(ByVal value As String)
      PropertyMetadata.FirstName.SetValue(Me, value)
    End Set
  End Property
#End Region ' FirstName property
```

This time the non-nullability (i.e., Required) and string length constraints are specified using the .NET validation attributes.

Generate No Verification or Validation Attributes



This setting results in the absence of validation-related attributes of any sort:

```
#region FirstName property
/// <summary>Gets or sets the FirstName. </summary>
[Bindable(true, BindingDirection.TwoWay)]
[Editable(true)]
[Display(Name = "FirstName", AutoGenerateField = true)]
[DataMember]
public string FirstName {
get { return PropertyMetadata.FirstName.GetValue(this); }
set { PropertyMetadata.FirstName.SetValue(this, value); }
}
#endregion FirstName property
```

```
#Region "FirstName property"
  ''' <summary>Gets or sets the FirstName. </summary>
  <Bindable(True, BindingDirection.TwoWay), Editable(True), _
    Display(Name:="FirstName", AutoGenerateField:=True), DataMember()> _
  Public Property FirstName() As String
    Get
      Return PropertyMetadata.FirstName.GetValue(Me)
    End Get
    Set(ByVal value As String)
      PropertyMetadata.FirstName.SetValue(Me, value)
    End Set
  End Property
#End Region ' FirstName property
```