Examples

The following examples create an application that prompts the user to enter the description and price of an item and the date that the offer expires. The application displays the current information for the item below the form. The user can submit or cancel the changes.

The application does the following to achieve this behavior.

* Creates a BindingGroup and adds it the root [StackPanel](https://msdn.microsoft.com/en-us/library/system.windows.controls.stackpanel(v=vs.110).aspx) when it creates the user interface (UI) of the application.
* Calls [BeginEdit](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup.beginedit(v=vs.110).aspx), [CommitEdit](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup.commitedit(v=vs.110).aspx), and [CancelEdit](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup.canceledit(v=vs.110).aspx) in the application's logic to enable rolling back changes.
* Calls [TryGetValue](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup.trygetvalue(v=vs.110).aspx) in a [Validate](https://msdn.microsoft.com/en-us/library/ms590035(v=vs.110).aspx) method to get the user's input and then check that an item over 100 dollars is available for at least seven days.

The following example creates the user interface (UI) of the application. The root [StackPanel](https://msdn.microsoft.com/en-us/library/system.windows.controls.stackpanel(v=vs.110).aspx) has a BindingGroup that contains a [ValidationRule](https://msdn.microsoft.com/en-us/library/system.windows.controls.validationrule(v=vs.110).aspx) that validates the item, as described previously. The binding objects on the Price property and the OfferExpires property become part of the BindingGroup and each binding has a [ValidationRule](https://msdn.microsoft.com/en-us/library/system.windows.controls.validationrule(v=vs.110).aspx) to make sure that price and date, respectively, are valid values. The validation rules for the individual properties run before the [ValidationRule](https://msdn.microsoft.com/en-us/library/system.windows.controls.validationrule(v=vs.110).aspx) on the BindingGroup.

XAML

<StackPanel Name="stackPanel1" Margin="10" Width="250"

Loaded="stackPanel1\_Loaded"

Validation.Error="ItemError">

<StackPanel.Resources>

<Style TargetType="HeaderedContentControl">

<Setter Property="Margin" Value="2"/>

<Setter Property="Focusable" Value="False"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="HeaderedContentControl">

<DockPanel LastChildFill="False">

<ContentPresenter ContentSource="Header" DockPanel.Dock="Left" Focusable="False" VerticalAlignment="Center"/>

<ContentPresenter ContentSource="Content" Margin="5,0,0,0" DockPanel.Dock="Right" VerticalAlignment="Center"/>

</DockPanel>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style TargetType="Button">

<Setter Property="Width" Value="100"/>

<Setter Property="Margin" Value="10,15,15,15"/>

</Style>

</StackPanel.Resources>

<StackPanel.BindingGroup>

<BindingGroup NotifyOnValidationError="True">

<BindingGroup.ValidationRules>

<src:ValidateDateAndPrice ValidationStep="ConvertedProposedValue" />

</BindingGroup.ValidationRules>

</BindingGroup>

</StackPanel.BindingGroup>

<TextBlock FontSize="14" Text="Enter an item for sale"/>

<HeaderedContentControl Header="Description">

<TextBox Width="150" Text="{Binding Path=Description, Mode=TwoWay}"/>

</HeaderedContentControl>

<HeaderedContentControl Header="Price">

<TextBox Name="priceField" Width="150">

<TextBox.Text>

<Binding Path="Price" Mode="TwoWay" >

<Binding.ValidationRules>

<src:PriceIsAPositiveNumber/>

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

</HeaderedContentControl>

<HeaderedContentControl Header="Date Offer Ends">

<TextBox Name="dateField" Width="150" >

<TextBox.Text>

<Binding Path="OfferExpires" StringFormat="d" >

<Binding.ValidationRules>

<src:FutureDateRule/>

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

</HeaderedContentControl>

<StackPanel Orientation="Horizontal">

<Button IsDefault="True" Click="Submit\_Click">\_Submit</Button>

<Button IsCancel="True" Click="Cancel\_Click">\_Cancel</Button>

</StackPanel>

<HeaderedContentControl Header="Description">

<TextBlock Width="150" Text="{Binding Path=Description}"/>

</HeaderedContentControl>

<HeaderedContentControl Header="Price">

<TextBlock Width="150" Text="{Binding Path=Price, StringFormat=c}"/>

</HeaderedContentControl>

<HeaderedContentControl Header="Date Offer Ends">

<TextBlock Width="150" Text="{Binding Path=OfferExpires, StringFormat=d}"/>

</HeaderedContentControl>

</StackPanel>

The following example shows the event handlers for the application. When the user clicks the Submit button, the application calls [CommitEdit](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup.commitedit(v=vs.110).aspx) to run each [ValidationRule](https://msdn.microsoft.com/en-us/library/system.windows.controls.validationrule(v=vs.110).aspx) that is associated with the BindingGroup. If each [ValidationRule](https://msdn.microsoft.com/en-us/library/system.windows.controls.validationrule(v=vs.110).aspx) succeeds, [CommitEdit](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup.commitedit(v=vs.110).aspx) saves the values to the object and ends the edit transaction. If [CommitEdit](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup.commitedit(v=vs.110).aspx) is successful, the application begins another edit transaction. When a [ValidationRule](https://msdn.microsoft.com/en-us/library/system.windows.controls.validationrule(v=vs.110).aspx) fails, the [Validation.Error](https://msdn.microsoft.com/en-us/library/system.windows.controls.validation.error(v=vs.110).aspx) event occurs because the application set [NotifyOnValidationError](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup.notifyonvalidationerror(v=vs.110).aspx) to **true** on the BindingGroup (in the previous example). ItemError handles the [Validation.Error](https://msdn.microsoft.com/en-us/library/system.windows.controls.validation.error(v=vs.110).aspx) event and displays information about the validation error to the user. The example also handles the [Loaded](https://msdn.microsoft.com/en-us/library/system.windows.frameworkelement.loaded(v=vs.110).aspx) event for the [StackPanel](https://msdn.microsoft.com/en-us/library/system.windows.controls.stackpanel(v=vs.110).aspx)and the [Click](https://msdn.microsoft.com/en-us/library/system.windows.controls.primitives.buttonbase.click(v=vs.110).aspx) event for the **Cancel** button.

C#

[**VB**](https://msdn.microsoft.com/en-us/library/system.windows.data.bindinggroup(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

private void Submit\_Click(object sender, RoutedEventArgs e)

{

if (stackPanel1.BindingGroup.CommitEdit())

{

MessageBox.Show("Item submitted");

stackPanel1.BindingGroup.BeginEdit();

}

}

// This event occurs when a ValidationRule in the BindingGroup

// or in a Binding fails.

private void ItemError(object sender, ValidationErrorEventArgs e)

{

if (e.Action == ValidationErrorEventAction.Added)

{

MessageBox.Show(e.Error.ErrorContent.ToString());

}

}

void stackPanel1\_Loaded(object sender, RoutedEventArgs e)

{

// Set the DataContext to a PurchaseItem object.

// The BindingGroup and Binding objects use this as

// the source.

stackPanel1.DataContext = new PurchaseItem();

// Begin an edit transaction that enables

// the object to accept or roll back changes.

stackPanel1.BindingGroup.BeginEdit();

}

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

// Cancel the pending changes and begin a new edit transaction.

stackPanel1.BindingGroup.CancelEdit();

stackPanel1.BindingGroup.BeginEdit();

}

The following example shows the custom [ValidationRule](https://msdn.microsoft.com/en-us/library/system.windows.controls.validationrule(v=vs.110).aspx) ValidateDateAndPrice, which was added to the BindingGroup in the first example. The [ValidationRule](https://msdn.microsoft.com/en-us/library/system.windows.controls.validationrule(v=vs.110).aspx) uses the BindingGroup in its [Validate](https://msdn.microsoft.com/en-us/library/ms590035(v=vs.110).aspx) method to get the values the user entered into the form, and checks that if an item is over 100 dollars, it will be available for at least seven days.

C#

public class ValidateDateAndPrice : ValidationRule

{

// Ensure that an item over $100 is available for at least 7 days.

public override ValidationResult Validate(object value, CultureInfo cultureInfo)

{

BindingGroup bg = value as BindingGroup;

// Get the source object.

PurchaseItem item = bg.Items[0] as PurchaseItem;

object doubleValue;

object dateTimeValue;

// Get the proposed values for Price and OfferExpires.

bool priceResult = bg.TryGetValue(item, "Price", out doubleValue);

bool dateResult = bg.TryGetValue(item, "OfferExpires", out dateTimeValue);

if (!priceResult || !dateResult)

{

return new ValidationResult(false, "Properties not found");

}

double price = (double)doubleValue;

DateTime offerExpires = (DateTime)dateTimeValue;

// Check that an item over $100 is available for at least 7 days.

if (price > 100)

{

if (offerExpires < DateTime.Today + new TimeSpan(7, 0, 0, 0))

{

return new ValidationResult(false, "Items over $100 must be available for at least 7 days.");

}

}

return ValidationResult.ValidResult;

}

}

Version Information

**.NET Framework**   
Available since 3.0