События

xmlns:i="clr-namespace:System.Windows.Interactivity;assembly=System.Windows.Interactivity"

Title="MainWindow" Height="350" Width="525">

<i:Interaction.Triggers >

<i:EventTrigger EventName="Closed">

<i:InvokeCommandAction Command="{Binding CloseCommand}"/>

</i:EventTrigger>

<i:EventTrigger EventName="Loaded">

<i:InvokeCommandAction Command="{Binding LoadCommand}"

CommandParameter="{Binding RelativeSource={RelativeSource FindAncestor, AncestorType={x:Type Window}}}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

Подключение к главному окну класса

<Window.DataContext>

<local:MainWindowViewModel></local:MainWindowViewModel>

</Window.DataContext>

Подключение ресурсов

<Window.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="Dictionary/Dictionary.xaml"/>

<ResourceDictionary Source="Dictionary/ScrollBar.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Window.Resources>

Подключение ресурсов для DataGrid или ComboBox

<Grid>

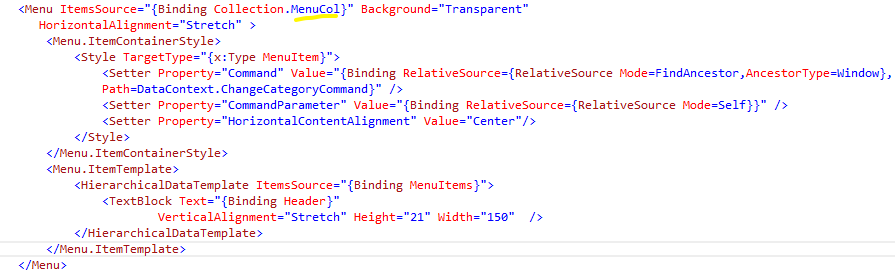
<Grid.Resources>

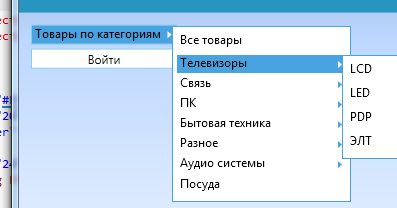
<CollectionViewSource Source="{Binding Collection.RolesCol}" x:Key="RoleCol"/>

<CollectionViewSource Source="{Binding Collection.SubCategoriesCol}" x:Key="SubCol"/>

</Grid.Resources>

Меню





<Menu ItemsSource="{Binding Collection.MenuCol}" Background="Transparent"

HorizontalAlignment="Stretch" >

<Menu.ItemContainerStyle>

<Style TargetType="{x:Type MenuItem}">

<Setter Property="Command" Value="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window},

Path=DataContext.ChangeCategoryCommand}" />

<Setter Property="CommandParameter" Value="{Binding RelativeSource={RelativeSource Mode=Self}}" />

<Setter Property="HorizontalContentAlignment" Value="Center"/>

</Style>

</Menu.ItemContainerStyle>

<Menu.ItemTemplate>

<HierarchicalDataTemplate ItemsSource="{Binding MenuItems}">

<TextBlock Text="{Binding Header}"

VerticalAlignment="Stretch" Height="21" Width="150" />

</HierarchicalDataTemplate>

</Menu.ItemTemplate>

</Menu>

В DataFGrid выпадающий список (ComboBox)

DataGridComboBoxColumn Header="Статус"

ItemsSource="{Binding Source={StaticResource Roles}}"

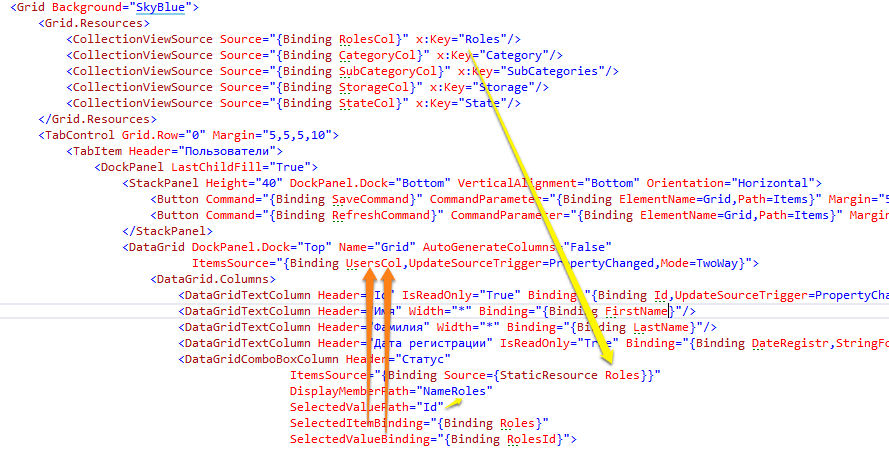
DisplayMemberPath="NameRoles"

SelectedValuePath="Id"

SelectedItemBinding="{Binding Roles}"

SelectedValueBinding="{Binding RolesId}">

</DataGridComboBoxColumn>



Дата в DataGrid

<DataGridTemplateColumn Header="Дата рождения" CanUserSort="True" SortMemberPath="LatestVictory">

<DataGridTemplateColumn.CellTemplate >

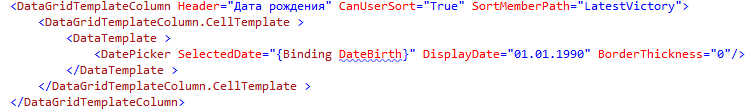
<DataTemplate >

<DatePicker SelectedDate="{Binding DateBirth}" DisplayDate="01.01.1990" BorderThickness="0"/>

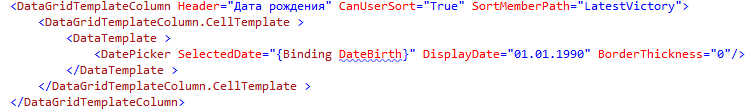
</DataTemplate >

</DataGridTemplateColumn.CellTemplate >

</DataGridTemplateColumn>



Обычный ComboBox



Конвертирование значений

<Label Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Left" Content="{Binding ElementName=OrderGrid,Path=Items.Count,Converter={StaticResource ConvertStat},

UpdateSourceTrigger=PropertyChanged}"></Label>



Передача двух параметров

<Button Command="{Binding FindCommand}" Margin="5"

Template="{StaticResource AnotherButton}" Content="Поиск">

<Button.CommandParameter>

<MultiBinding Converter="{StaticResource Convert}">

<Binding ElementName="RolesGrid" Path="Items"/>

<Binding RelativeSource="{RelativeSource Mode=FindAncestor,AncestorType=StackPanel}" Path="Children[2].Text"/>

</MultiBinding>

</Button.CommandParameter>

</Button>

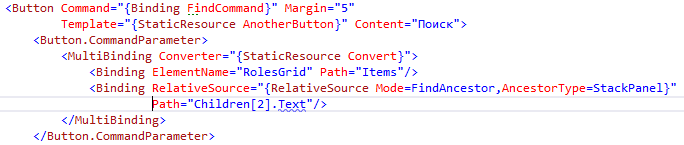
Объявление конвертора

<Window.Resources>

<ResourceDictionary>

<converters:ConverterBindingWrapper x:Key="Convert"/>

<convert:ConverterStatistics x:Key="ConvertStat"/>



Форматирование строки

Не нужны скобки и тогда, когда строка начинается с любого символа, кроме {:

<TextBlock x:Name="numItemsLabel"

Text="{Binding StringFormat=Number of items: {0},

Source={StaticResource photos}, Path=Count}"

DockPanel.Dock="Bottom"/>

Можно также улучшить форматирование, добавив спецификатор N0, который расставляет разделители между группами по три цифры. Например, если Count равно 54, то будет показана строка «54 item(s)», а если Count равно 1001, то строка «1,001 item(s)» (по крайней мере, в культуре en-US):

<TextBlock x:Name="numItemsLabel"

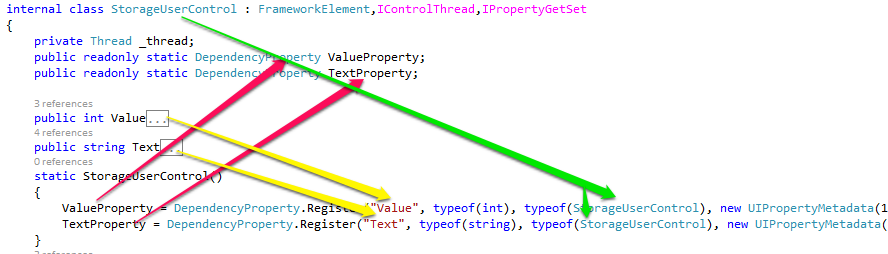
Text="{Binding StringFormat={}{0:N0} item(s),

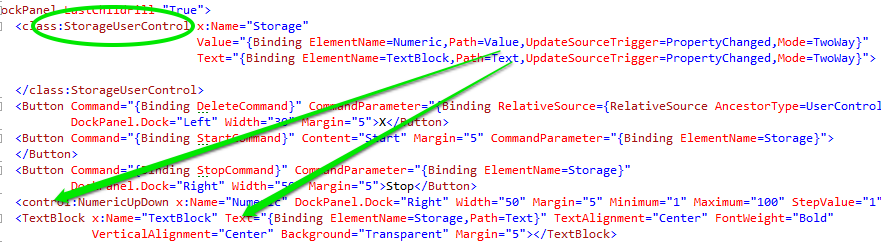
Source={StaticResource photos}, Path=Count}"

DockPanel.Dock="Bottom"/>

<TextBlock Text="{Binding StringFormat=\{0:C\}}" />

Получаем класс который объявлен в контроле и в этом классе свойства привязаны к значениям элемента





Тул тип, ошибки при неправильных данных