ClipboardSpy

<Separator/>分隔线

IDataObject \_dataObject= Clipboard.GetDataObject();

判断包含的内容

Clipboard.ContainsAudio();

Clipboard.ContainsFileDropList();

Clipboard.ContainsImage();

Clipboard.ContainsText();

Returns a list of all formats that the data in this data object is stored in,

or can be converted to.

var formats = \_dataObject.GetFormats();

\_dataObject.GetData(format, true).ToString()

ClipboardViewer

Clipboard.Clear();

Checks to see whether the data is available in, or can be converted to, a specified

// format; the data format is specified by a string.

dataObject.GetDataPresent(DataFormats.WaveAudio);

dataObject.GetDataPresent(DataFormats.FileDrop);

dataObject.GetDataPresent(DataFormats.Bitmap);

dataObject.GetDataPresent(DataFormats.Text);

dataObject.GetDataPresent(DataFormats.Rtf);

dataObject.GetDataPresent(DataFormats.Xaml);

ApplicationShutdown

Application.Current.ShutdownMode = ShutdownMode.OnLastWindowClose;

Application.Current.ShutdownMode =

(ShutdownMode) Enum.Parse(typeof (ShutdownMode), shutdownModeListBox.SelectedValue.ToString());

var exitCode = 0;

int.TryParse(appExitCodeTextBox.Text, out exitCode);

Application.Current.Shutdown(exitCode);

CodeOnlyWindowsApplication

Wpf默认会为程序生成main方法。可以把实例化mainwindow传给app.Run()，或设置app.Mainwindow=window

public static void Main()

{

// Start the WPF application

var app = new App();

app.Run();

}

Application的事件可以使用覆盖的方式处理，或用handler订阅事件

Startup中可以处理对程序的传参，或者在这里指定程序的mainwindow

protected override void OnStartup(StartupEventArgs e)

{

base.OnStartup(e);

foreach (var arg in e.Args)

{}

// Show main application window.

// NOTE: this window is automatically set as

// App.Current.MainWindow and App.Current.Windows[0]

var window = new MainWindow();

window.Show();

}

ExceptionHandlingSecondaryUIThread

在子线程中开启另一个子窗口

子窗口的Dispatcher.Thread.ManagedThreadId和子线程的id相等Thread.CurrentThread.ManagedThreadId;

子线程的方法

private void MethodRunningOnSecondaryUIThread()

{

var secondaryUiThreadId = Thread.CurrentThread.ManagedThreadId;

try

{

// On secondary thread, show a new Window before starting a new Dispatcher

// ie turn secondary thread into a UI thread

var window = new SecondaryUIThreadWindow();

window.Show();

Dispatcher.Run();

}

catch (Exception ex)

{

// Dispatch the exception back to the main ui thread and reraise it

Application.Current.Dispatcher.Invoke(

DispatcherPriority.Send,

(DispatcherOperationCallback) delegate

{

// THIS CODE RUNS BACK ON THE MAIN UI THREAD

string msg = $"Exception forwarded from secondary UI thread {secondaryUiThreadId}.";

throw new Exception(msg, ex);

}

, null);

// NOTE - Application execution will only continue from this point

// onwards if the exception was handled on the main UI thread

// by Application.DispatcherUnhandledException

}

}

}

App的xmal中定义DispatcherUnhandledException="App\_DispatcherUnhandledException"

private void App\_DispatcherUnhandledException(object sender, DispatcherUnhandledExceptionEventArgs e)

{

主窗口的Thread.CurrentThread.ManagedThreadId和app中处理异常e.Dispatcher.Thread.ManagedThreadId的id相等

sb.AppendFormat("Exception handled on main UI thread {0}.", e.Dispatcher.Thread.ManagedThreadId);

// Keep application running in the face of this exception

e.Handled = true;

}

SkinnedApplication

private void App\_Startup(object sender, StartupEventArgs e)

{

Properties["Blue"] = (ResourceDictionary) LoadComponent(new Uri("BlueSkin.xaml", UriKind.Relative));

}

Application.Current.Resources = (ResourceDictionary) Application.Current.Properties[selectedValue];

ADODataSet

\_appPath = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);

\_myDataSet = new DataSet();

adapter.Fill(\_myDataSet, "BookTable");

// myListBox is a ListBox control.

// Set the DataContext of the ListBox to myDataSet

myListBox.DataContext = \_myDataSet;

<ListBox Name="myListBox" Height="200"

ItemsSource="{Binding Path=BookTable}"

ItemTemplate ="{StaticResource BookItemTemplate}"/>

DataTemplate中的binding只有path是已ItemsSource为源？

BindConversion

public class MyData : INotifyPropertyChanged{

OnPropertyChanged都要调用PropertyChanged?

public DateTime TheDate

{

get { return \_thedate; }

set

{

\_thedate = value;

OnPropertyChanged("TheDate");

}

}

// Declare event

public event PropertyChangedEventHandler PropertyChanged;

// OnPropertyChanged method to update property value in binding

private void OnPropertyChanged(string info)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(info));

}

}

<TextBlock Name="myconvertedtext"

Foreground="{Binding Path=TheDate,

Converter={StaticResource MyConverterReference}}">

<TextBlock.Text>

<Binding Path="TheDate"

Converter="{StaticResource MyConverterReference}"/>

</TextBlock.Text>

</TextBlock>

// 参数:

// value:

// The value produced by the binding source.

//

// targetType:

// The type of the binding target property.

//

// parameter:

// The converter parameter to use.

//

// culture:

// The culture to use in the converter.

public class MyConverter : IValueConverter

{

public object Convert(object o, Type type,

object parameter, CultureInfo culture)

{

var date = (DateTime) o;

switch (type.Name)

{

case "String":

return date.ToString("F", culture);

case "Brush":

return Brushes.Red;

default:

return o;

}

}

BindingDPToDP

DependencyProperty

<ComboBoxItem>Green</ComboBoxItem>

<Canvas.Background>

<Binding ElementName="myComboBox" Path="SelectedItem.Content"/>

</Canvas.Background>

BindingToMethod

<ObjectDataProvider ObjectType="{x:Type local:TemperatureScale}"

MethodName="ConvertTemp" x:Key="ConvertTemp">

<ObjectDataProvider.MethodParameters>

<system:Double>0</system:Double>

<local:TempType>Celsius</local:TempType>

</ObjectDataProvider.MethodParameters>

</ObjectDataProvider>

<TextBox.Text>

<Binding Source="{StaticResource ConvertTemp}" Path="MethodParameters[0]"

BindsDirectlyToSource="true" UpdateSourceTrigger="PropertyChanged"

Converter="{StaticResource DoubleToString}">

<Binding.ValidationRules>

<local:InvalidCharacterRule/>

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

BindsDirectlyToSource获取或设置一个值，该值指示是相对于数据项还是 [DataSourceProvider](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.data.datasourceprovider?view=netframework-4.8) 对象计算 [Path](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.data.binding.path?view=netframework-4.8#System_Windows_Data_Binding_Path)。

若要相对于数据项自身计算路径，则为 false；否则为 true。 默认值为 false。

BindValidation

Target

<Label Grid.Column="0" Grid.Row="1" FontSize="15" Margin="2"

Target="{Binding ElementName=textBox1}">TextBox with \_custom ErrorTemplate and ToolTip:</Label>

都是使用Validation附加属性

<Style x:Key="TextBoxInError" TargetType="{x:Type TextBox}">

<Style.Triggers>

<Trigger Property="Validation.HasError" Value="true">

<Setter Property="ToolTip"

Value="{Binding RelativeSource={x:Static RelativeSource.Self},

Path=(Validation.Errors)[0].ErrorContent}"/>

</Trigger>

</Style.Triggers>

</Style>

Validation.Error事件

Validation.Errors依赖属性

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

Loaded="StackPanel1\_Loaded"

Validation.Error="ItemError">

自定义错误模板

<TextBox Name="textBox1" Width="50" FontSize="15"

Validation.ErrorTemplate="{StaticResource ValidationTemplate}"

Style="{StaticResource TextBoxInError}"

Grid.Row="1" Grid.Column="1" Margin="2" />

<ControlTemplate x:Key="ValidationTemplate">

<DockPanel>

<TextBlock Foreground="Red" FontSize="20">!</TextBlock>

<AdornedElementPlaceholder/>

</DockPanel>

</ControlTemplate>

<Binding.ValidationRules>

<local:AgeRangeRule Min="21" Max="130"/>

</Binding.ValidationRules>

<Binding.ValidationRules>

<ExceptionValidationRule />

</Binding.ValidationRules>

public override ValidationResult Validate(object value, CultureInfo cultureInfo)

{

var age = 0;

try

{

if (((string) value).Length > 0)

age = int.Parse((string) value);

}

catch (Exception e)

{

return new ValidationResult(false, "Illegal characters or " + e.Message);

}

if ((age < Min) || (age > Max))

{

return new ValidationResult(false,

"Please enter an age in the range: " + Min + " - " + Max + ".");

}

return new ValidationResult(true, null);

}

Model实现IDataErrorInfo接口

// 参数:

// columnName:

// The name of the property whose error message to get.

//

// 返回结果:

// The error message for the property. The default is an empty string ("").

string this[string columnName] { get; }

// 返回结果:

// An error message indicating what is wrong with this object. The default is an

// empty string ("").

string Error { get; }

public string this[string name]

{

get

{

string result = null;

if (name == "Age")

{

if (Age < 0 || Age > 150)

{

result = "Age must not be less than 0 or greater than 150.";

}

}

return result;

}

}

<TextBox.Text>

<!--By setting ValidatesOnExceptions to True, it checks for exceptions

that are thrown during the update of the source property.

An alternative syntax is to add <ExceptionValidationRule/> within

the <Binding.ValidationRules> section.-->

<Binding Path="Age" Source="{StaticResource Data}"

ValidatesOnExceptions="True"

UpdateSourceTrigger="PropertyChanged">

<Binding.ValidationRules>

<!--DataErrorValidationRule checks for validation

errors raised by the IDataErrorInfo object.-->

<!--Alternatively, you can set ValidationOnDataErrors="True" on the Binding.-->

<DataErrorValidationRule/>

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

CodeOnlyBinding

清除目标元素上的binding

BindingOperations.ClearBinding(MyText, TextBlock.TextProperty);

CollectionBinding

IsSynchronizedWithCurrentItem

// Gets or sets a value that indicates whether a System.Windows.Controls.Primitives.Selector

// should keep the System.Windows.Controls.Primitives.Selector.SelectedItem synchronized

// with the current item in the System.Windows.Controls.ItemsControl.Items property.

<ListBox Width="200" IsSynchronizedWithCurrentItem="True"

ItemsSource="{Binding Source={StaticResource MyFriends}}"/>

ContentControl默认绑到集合的当前元素？

<ContentControl Content="{Binding Source={StaticResource MyFriends}}"

ContentTemplate="{StaticResource DetailTemplate}"/>

等价

Content="{Binding Path=/, Source={StaticResource MyFriends}}"

<DataTemplate x:Key="DetailTemplate">

<TextBlock Grid.Row="0" Grid.Column="1" Text="{Binding Path=FirstName}"/>

</DataTemplate>

CollectionViewSource

Xaml中的xml数据

<XmlDataProvider x:Key="MyTasks" XPath="Tasks/Task">

<x:XData>

<Tasks xmlns="">

<Task Name="Groceries" Priority="2" Type="Home">

<Description>Pick up Groceries and Detergent</Description>

</Task>

</Tasks>

</x:XData>

</XmlDataProvider>

<CollectionViewSource x:Key="MySortedTasks"

Source="{StaticResource MyTasks}">

<CollectionViewSource.SortDescriptions>

<componentModel:SortDescription PropertyName="@Priority" />

</CollectionViewSource.SortDescriptions>

<CollectionViewSource.GroupDescriptions>

<PropertyGroupDescription PropertyName="@Priority" />

</CollectionViewSource.GroupDescriptions>

/CollectionViewSource>

？

<ListBox ItemsSource="{Binding Source={StaticResource Cvs}}"

DisplayMemberPath="CityName" Name="lb">

<ListBox.GroupStyle>

<x:Static Member="GroupStyle.Default"/>

</ListBox.GroupStyle>

</ListBox>

Colors

<RadioButton GroupName="SortBy" Checked="OnSortByChanged">Name</RadioButton>

<RadioButton GroupName="SortBy" Checked="OnSortByChanged">Luminance</RadioButton>

获取RadioButton继承的DataContext，并从中得到default view

var cv = (CollectionView) CollectionViewSource.GetDefaultView((IEnumerable) rb.DataContext);

cv.SortDescriptions.Clear();

cv.SortDescriptions.Add(new SortDescription(sortBy, ListSortDirection.Descending));

ItemsSource="{Binding}"从datacontext中获取binding？

<ListBox Name="colorListBox" Height="455" Width="200" DockPanel.Dock="Left"

ItemsSource="{Binding}"

ItemTemplate = "{StaticResource ColorItemTemplate}"

ItemContainerStyle="{StaticResource ListItemStyle}"

IsSynchronizedWithCurrentItem="True"/>

明细和汇总都共用父元素的datacontext

var newItem = new ColorItem((ColorItem) cv.CurrentItem);

colorList.Add(newItem);

cv.MoveCurrentTo(newItem);

CompositeCollections

<ListBox.ItemsSource>

<CompositeCollection>

<CollectionContainer

Collection="{Binding Source={StaticResource GreekGodsData}}" />

<CollectionContainer

Collection="{Binding Source={StaticResource GreekHeroesData}}" />

<ListBoxItem Foreground="Red">Other Listbox Item 1</ListBoxItem>

<ListBoxItem Foreground="Red">Other Listbox Item 2</ListBoxItem>

</CompositeCollection>

</ListBox.ItemsSource>

<DataTemplate DataType="{x:Type local:GreekGod}">

<TextBlock Text="{Binding Path=Name}" Foreground="Gold"/>

</DataTemplate>

<DataTemplate DataType="Hero">

<TextBlock Text="{Binding XPath=@Name}" Foreground="Blue"/>

</DataTemplate>

DataBindingToStringFomat

<GridViewColumn DisplayMemberBinding="{Binding Path=Price, StringFormat=Now {0:c}!}"/>

<TextBlock.Text>

<MultiBinding StringFormat="{}{0} -- Now only {1:C}!">

<Binding Path="Description"/>

<Binding Path="Price"/>

</MultiBinding>

</TextBlock.Text>

DataTemplatingIntro

DataTrigger在datatemplate中

<DataTemplate.Triggers>

<DataTrigger Binding="{Binding Path=TaskType}">

<DataTrigger.Value>

<local:TaskType>Home</local:TaskType>

</DataTrigger.Value>

<Setter TargetName="border" Property="BorderBrush" Value="Yellow"/>

</DataTrigger>

</DataTemplate.Triggers>

<ListBox Width="400" Margin="10"

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

ItemTemplateSelector="{StaticResource MyDataTemplateSelector}"/>

FindResource用法

public override DataTemplate

SelectTemplate(object item, DependencyObject container)

{

if (item != null && item is Task)

{

var taskitem = (Task) item;

var window = Application.Current.MainWindow;

if (taskitem.Priority == 1)

return

window.FindResource("ImportantTaskTemplate") as DataTemplate;

return

window.FindResource("MyTaskTemplate") as DataTemplate;

}

return null;

}

MultiBinding

ImultiValueConverter

object Convert(object[] values, Type targetType, object parameter, CultureInfo culture);

IValueConverter

object Convert(object value, Type targetType, object parameter, CultureInfo culture);

<TextBlock.Text>

<MultiBinding Converter="{StaticResource MyNameConverter}"

ConverterParameter="FormatLastFirst">

<Binding Path="FirstName"/>

<Binding Path="LastName"/>

</MultiBinding>

</TextBlock.Text>

public object Convert(object[] values, Type targetType, object parameter, CultureInfo culture)

{

string name;

switch ((string) parameter)

{

case "FormatLastFirst":

name = values[1] + ", " + values[0];

break;

default:

name = values[0] + " " + values[1];

break;

}

return name;

}

DataTrigger

<Style TargetType="ListBoxItem">

<Style.Triggers>

<DataTrigger Binding="{Binding Path=State}" Value="WA">

<Setter Property="Foreground" Value="Red" />

</DataTrigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding Path=Name}" Value="Portland" />

<Condition Binding="{Binding Path=State}" Value="OR" />

</MultiDataTrigger.Conditions>

<Setter Property="Background" Value="Cyan" />

</MultiDataTrigger>

</Style.Triggers>

</Style>

DirectionalBinding

OneWay Binding, with TargetUpdated event handling

<TextBlock Grid.Row="1" Grid.Column="1" Name="RentText"

Text="{Binding Path=Rent, Mode=OneWay, NotifyOnTargetUpdated=True}"

TargetUpdated="OnTargetUpdated"/>

后台更新binding源的值

var bindingExpression =

BindingOperations.GetBindingExpression(SavingsText, TextBlock.TextProperty);

var sourceData = (NetIncome) bindingExpression.DataItem;

sourceData.Rent = (int) ((1 + i/100)\*sourceData.Rent);

<!-- TwoWay binding example (default for TextBox), with

UpdateSourceTrigger=LostFocus (default for TextBox) -->

<!-- OneWay binding example (which is default for Text) -->

<TextBlock Grid.Row="4" Grid.Column="1" Name="SavingsText" Text="{Binding Path=Savings}"/>

Text="{Binding Path=TotalIncome, Mode=OneTime}"

EditingCollections

有多个属性变化，NotifyPropertyChanged不传参数名？

public void CancelEdit()

{

\_currentData = \_copyData;

NotifyPropertyChanged("");

}

实现接口

public interface IEditableObject

{

void BeginEdit();

void CancelEdit();

void EndEdit();

}

private ItemData \_copyData;

private ItemData \_currentData;

ItemData 是结构体类型，没有引用，都是值复制

private struct ItemData

{

internal string Description;

internal DateTime OfferExpires;

internal double Price;

}

var editableCollectionView = itemsControl.Items as IEditableCollectionView;

if (!editableCollectionView.CanAddNew)

{

MessageBox.Show("You cannot add items to the list.");

return;

}

// Create a window that prompts the user to enter a new

// item to sell.

var win = new ChangeItem {DataContext = editableCollectionView.AddNew()};

//Create a new item to be added to the collection.

// If the user submits the new item, commit the new

// object to the collection. If the user cancels

// adding the new item, discard the new item.

if ((bool) win.ShowDialog())

{

editableCollectionView.CommitNew();

}

else

{

editableCollectionView.CancelNew();

}

删除

if (!editableCollectionView.CanRemove)

{

MessageBox.Show("You cannot remove items from the list.");

return;

}

if (MessageBox.Show("Are you sure you want to remove " + item.Description,

"Remove Item", MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

editableCollectionView.Remove(itemsControl.SelectedItem);

}

editableCollectionView.EditItem(itemsControl.SelectedItem);

if ((bool) win.ShowDialog())

{

editableCollectionView.CommitEdit();

}

else

{

editableCollectionView.CancelEdit();

}

Grouping

\_myView = (CollectionView) CollectionViewSource.GetDefaultView(myItemsControl.ItemsSource);

if (\_myView.CanGroup)

{

var groupDescription

= new PropertyGroupDescription("@Type");

\_myView.GroupDescriptions.Add(groupDescription);

}

<ItemsControl.GroupStyle>

<GroupStyle>

<GroupStyle.HeaderTemplate>

<DataTemplate>

<TextBlock FontWeight="Bold" FontSize="15"

Text="{Binding Path=Name}"/>

</DataTemplate>

</GroupStyle.HeaderTemplate>

</GroupStyle>

</ItemsControl.GroupStyle>

HierarchicalDataTemplate

<TreeView>

<TreeViewItem ItemsSource="{Binding Source={StaticResource MyList}}" Header="My Soccer Leagues" />

</TreeView>

<HierarchicalDataTemplate DataType = "{x:Type local:League}"

ItemsSource = "{Binding Path=Divisions}">

<TextBlock Text="{Binding Path=Name}"/>

</HierarchicalDataTemplate>

<HierarchicalDataTemplate DataType = "{x:Type local:Division}"

ItemsSource = "{Binding Path=Teams}">

<TextBlock Text="{Binding Path=Name}"/>

</HierarchicalDataTemplate>

<DataTemplate DataType="{x:Type local:Team}">

<TextBlock Text="{Binding Path=Name}"/>

</DataTemplate>

ListBox选择项的类型？

int.Parse(((sender as ListBox).SelectedItem as ListBoxItem).Content.ToString());

MasterDetail

<StackPanel>

<Label>My Soccer Leagues</Label>

<ListBox ItemsSource="{Binding}"

IsSynchronizedWithCurrentItem="true"/>

</StackPanel>

<StackPanel>

<Label Content="{Binding Path=Name}"/>

<ListBox ItemsSource="{Binding Path=Divisions}"

IsSynchronizedWithCurrentItem="true"/>

</StackPanel>

<StackPanel>

<Label Content="{Binding Path=Divisions/Name}"/>

<ListBox ItemsSource="{Binding Path=Divisions/Teams}"/>

</StackPanel>

PriorityBinding

字段用attr设置

<local:AsyncDataSource SlowestDp="Slowest Value" SlowerDp="Slower Value"

FastDp="Fast Value" x:Key="AsyncDs" />

public string SlowerDp

{

get

{

// This simulates a lengthy time before the

// data being bound to is actualy available.

Thread.Sleep(3000);

return \_slowerDp;

}

set { \_slowerDp = value; }

}

<TextBlock.Text>

<PriorityBinding FallbackValue="defaultvalue">

<Binding Path="SlowestDp" IsAsync="True"/>

<Binding Path="SlowerDp" IsAsync="True"/>

<Binding Path="FastDp" />

</PriorityBinding>

</TextBlock.Text>

SortFilter

MyCollectionView = (ListCollectionView) CollectionViewSource.GetDefaultView(rootElement.DataContext);

MyCollectionView.SortDescriptions.Add(new SortDescription("OrderItem",

ListSortDirection.Ascending));

// Gets or sets a method that is used to determine whether an item is suitable for

// inclusion in the view.

case "Filter":

MyCollectionView.Filter = Contains;

break;

case "Unfilter":

MyCollectionView.Filter = null;

case "Previous":

if (MyCollectionView.MoveCurrentToPrevious())

{

feedbackText.Text = "";

}

else

{

MyCollectionView.MoveCurrentToFirst();

feedbackText.Text = "At first record";

}

break;

case "Next":

if (MyCollectionView.MoveCurrentToNext())

{

feedbackText.Text = "";

}

else

{

MyCollectionView.MoveCurrentToLast();

feedbackText.Text = "At last record";

}

UpdateSource

// itemNameTextBox is an instance of a TextBox

var be = itemNameTextBox.GetBindingExpression(TextBox.TextProperty);

be.UpdateSource();

ValidateItemSample

When a ValidationRule

is added to a BindingGroup, the rule can get the properties of

the source item in the Validate method.

<StackPanel.BindingGroup>

<BindingGroup NotifyOnValidationError="True">

<BindingGroup.ValidationRules>

<local:ValidateDateAndPrice ValidationStep="ConvertedProposedValue" />

</BindingGroup.ValidationRules>

</BindingGroup>

</StackPanel.BindingGroup>

<LineBreak/>

public override ValidationResult Validate(object value, CultureInfo cultureInfo)

{

var bg = value as BindingGroup;

// Get the source object.

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

}

private 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 Submit\_Click(object sender, RoutedEventArgs e)

{

if (stackPanel1.BindingGroup.CommitEdit())

{

MessageBox.Show("Item submitted");

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();

}

<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>

private void ItemError(object sender, ValidationErrorEventArgs e)

{

if (e.Action == ValidationErrorEventAction.Added)

{

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

}

}

实现了IEditableObject接口的数据源，需要手动提交，texbox默认的twoway和 UpdateSourceTrigger="PropertyChanged"都不管用？

ValidateItemsInItemsControl

The ValidationRule assigned to ItemsControl.ItemBindingGroup checks

multiple properties in the item

<ObjectDataProvider MethodName="GetValues"

ObjectType="{x:Type sys:Enum}"

x:Key="RegionValues">

<ObjectDataProvider.MethodParameters>

<x:Type TypeName="local:Region" />

</ObjectDataProvider.MethodParameters>

</ObjectDataProvider>

Container是ContentPresenter

var container = (FrameworkElement) customerList.ContainerFromElement(btn);

if (container.BindingGroup.ValidateWithoutUpdate())

{

container.BindingGroup.UpdateSources();

\_bindingGroupInError = null;

MessageBox.Show("Item Saved");

}

public override ValidationResult Validate(object value, CultureInfo cultureInfo)

{

var bg = value as BindingGroup;

var cust = bg.Items[0] as Customer;

if (cust == null)

{

return new ValidationResult(false, "Customer is not the source object");

}

var region = (Region) bg.GetValue(cust, "Location");

var rep = bg.GetValue(cust, "ServiceRepresentative") as ServiceRep;

}

将错误显示在label元素上

<ItemsControl.ItemContainerStyle>

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

<Setter Property="Validation.ValidationAdornerSite"

Value="{Binding ElementName=validationErrorReport}"/>

</Style>

</ItemsControl.ItemContainerStyle>

<Label Name="validationErrorReport"

Content="{Binding RelativeSource={RelativeSource Self},

Path=(Validation.ValidationAdornerSiteFor).(Validation.Errors)[0].ErrorContent}"

Margin="5" Foreground="Red" HorizontalAlignment="Center"/>

Xpath用来指定使用数据源中哪些节点数据

<XmlDataProvider x:Key="InventoryData" XPath="Inventory/Books">

Xpath用来过滤数据

<ListBox.ItemsSource>

<Binding Source="{StaticResource InventoryData}"

XPath="\*[@Stock='out'] | \*[@Number>=8 or @Number=3]"/>

</ListBox.ItemsSource>

stackPanel.FindName("dog");

ContextMenuOpening

<Rectangle.ContextMenu>

<ContextMenu>

<MenuItem Header="Item1"/>

<MenuItem Header="Item2"/>

</ContextMenu>

</Rectangle.ContextMenu>

ContextMenuOpening事件，可以添加或覆盖xmal中定义的contextmenu

private void HandlerForCMO(object sender, ContextMenuEventArgs e)

{

var fe = e.Source as FrameworkElement;

fe.ContextMenu = BuildMenu();

}

Alternative approach for custom classes: override OnContextMenuOpening class handler

protected override void OnContextMenuOpening(ContextMenuEventArgs e)

{

base.OnContextMenuOpening(e);

var buttonMenu = new ContextMenu();

var mia = new MenuItem {Header = "Item1"};

buttonMenu.Items.Add(mia);

var fe = e.Source as FrameworkElement;

fe.ContextMenu = buttonMenu;

}

FocusVisualStyle

用Template？FocusVisualStyle只对用键盘产生的focus生效。鼠标选中或代码实现的focus无效

<Style x:Key="MyFocusVisual">

<Setter Property="Control.Template">

<Setter.Value>

<ControlTemplate>

<Rectangle Margin="-5" StrokeThickness="2" Stroke="Red" StrokeDashArray="1 2"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

HeightProperties

ActualHeight

优先级

MinHeight > MaxHeight >Height

btn1.Margin = marginThickness.Left == 10 ? new Thickness(60) : new Thickness(10);

ThicknessConverter

ThicknessConverter不是binding的converter，继承自TypeConverter

Binding 的是IValueConverter

var myThicknessConverter = new System.Windows.ThicknessConverter();

var th1 = (Thickness) myThicknessConverter.ConvertFromString(li.Content.ToString());

var myBrushConverter = new BrushConverter();

border1.BorderBrush = (Brush) myBrushConverter.ConvertFromString((string) li2.Content);

// Create a custom routed event by first registering a RoutedEventID

// This event uses the bubbling routing strategy

public static readonly RoutedEvent TapEvent = EventManager.RegisterRoutedEvent(

"Tap", RoutingStrategy.Bubble, typeof (RoutedEventHandler), typeof (MyButtonSimple));

public event RoutedEventHandler Tap

{

add { AddHandler(TapEvent, value); }

remove { RemoveHandler(TapEvent, value); }

}

// This method raises the Tap event

private void RaiseTapEvent()

{

var newEventArgs = new RoutedEventArgs(TapEvent);

RaiseEvent(newEventArgs);

}

private void HandleClick(object sender, RoutedEventArgs args)

{

var fe = (FrameworkElement) sender;

\_eventstr.Append("Event handled by element named ");

\_eventstr.Append(fe.Name);

\_eventstr.Append("\n");

var fe2 = (FrameworkElement) args.Source;

\_eventstr.Append("Event originated from source element of type ");

\_eventstr.Append(args.Source.GetType());

\_eventstr.Append(" with Name ");

\_eventstr.Append(fe2.Name);

}

<Expander Margin="0,0,0,10" DockPanel.Dock="top" Background="white" FocusVisualStyle="{x:Null}">

<Expander.Header>

<TextBlock Margin="10,0,0,0" FontFamily="Calibri" FontWeight="bold" >Layout</TextBlock>

</Expander.Header>

<ListBox Name="LayoutListBox" DataContext="{Binding Source={StaticResource SamplesList}, XPath=/Samples/Category[1]/Sample}" ItemsSource="{Binding}" SelectionChanged="HandleSelectionChanged" Style="{DynamicResource SamplesListBox}" IsSynchronizedWithCurrentItem="True" />

</Expander>

ControlsAndLayout

PreviewRow.Height = new GridLength(1, GridUnitType.Star);

CodeRow.Height = new GridLength(0);

Page跳转

<Hyperlink NavigateUri="Page1.xaml">Go To The Start Page</Hyperlink>

\_2DTransforms

Frame标签source是page标签？

<TabControl Background="White" BorderBrush="Orange">

<TabItem Header="Transform Example">

<Frame Background="White" Source="TransformExample.xaml" />

</TabItem>

</ TabControl>

绝对定位旋转

<Polyline

Points="25,25 0,50 25,75 50,50 25,25 25,0"

Stroke="Blue" StrokeThickness="10"

Canvas.Left="75" Canvas.Top="50"

RenderTransformOrigin="0.5,0.5">

<Polyline.RenderTransform>

<RotateTransform Angle="45" />

</Polyline.RenderTransform>

</Polyline>

相对定位旋转

<!-- Rotates the Polyline 45 degrees about the point (25,50). -->

<Polyline Points="25,25 0,50 25,75 50,50 25,25 25,0"

Stroke="Blue" StrokeThickness="10"

Canvas.Left="75" Canvas.Top="50">

<Polyline.RenderTransform>

<RotateTransform CenterX="25" CenterY="50" Angle="45" />

</Polyline.RenderTransform>

</Polyline>

<StackPanel.Triggers>

<EventTrigger RoutedEvent="Button.Click" SourceName="startButton">

</ StackPanel.Triggers>

AutoReverse配合RepeatBehavior时，一个来回算一次

<DoubleAnimation

Storyboard.TargetName="AnimatedRotateTransform1"

Storyboard.TargetProperty="(RotateTransform.Angle)"

RepeatBehavior="2x" AutoReverse="True"

From="0" To="180" Duration="0:0:2" />

多个包含在animation包含在ParallelTimeline中

<Storyboard>

<ParallelTimeline RepeatBehavior="Forever">

<DoubleAnimation

Storyboard.TargetName="AnimatedRotateTransform2"

Storyboard.TargetProperty="Angle"

From="0" To="360" Duration="0:0:2.5" RepeatBehavior="4x" />

<DoubleAnimation

Storyboard.TargetName="AnimatedRotateTransform2"

Storyboard.TargetProperty="CenterX"

From="0" To="25" Duration="0:0:5" AutoReverse="True" />

拉伸

<Rectangle.RenderTransform>

<ScaleTransform CenterX="25" CenterY="25" ScaleX="0.5" ScaleY="0.5" />

</Rectangle.RenderTransform>

移动

<Rectangle.RenderTransform>

<TranslateTransform X="50" Y="50" />

</Rectangle.RenderTransform>

元素沿着x轴扭曲(倾斜)45度

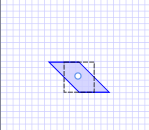
<Rectangle.RenderTransform>

<!-- Applies a horizontal skew of 45 degrees

from a center point of (25,25). -->

<SkewTransform CenterX="25" CenterY="25" AngleX="45" AngleY="0" />

</Rectangle.RenderTransform>



组合transform

<Polyline.RenderTransform>

<TransformGroup>

<TransformGroup.Children>

<TransformCollection>

<RotateTransform CenterX="25" CenterY="50" Angle="45" />

<TranslateTransform X="50" Y="0" />

</TransformCollection>

</TransformGroup.Children>

</TransformGroup>

ClipRegion

<Image

Source="sampleImages\Waterlilies.jpg"

Width="200" Height="150" HorizontalAlignment="Left">

<Image.Clip>

<EllipseGeometry

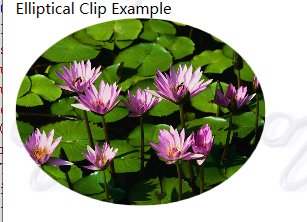
RadiusX="100"

RadiusY="75"

Center="100,75"/>

</Image.Clip>

</Image>



Height不写的话，默认是1\*

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="20" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

Geometry用于图形的路径

GeometryGroup中，两个路径相交部分镂空

<Path.Data>

<GeometryGroup>

<RectangleGeometry Rect="50,5 100,10" />

<RectangleGeometry Rect="5,5 95,180" />

<EllipseGeometry Center="100, 100" RadiusX="20" RadiusY="30"/>

<RectangleGeometry Rect="50,175 100,10" />

<PathGeometry>

<PathGeometry.Figures>

<PathFigureCollection>

<PathFigure IsClosed="true" StartPoint="50,50">

<PathFigure.Segments>

<PathSegmentCollection>

<BezierSegment Point1="75,300" Point2="125,100" Point3="150,50"/>

<BezierSegment Point1="125,300" Point2="75,100" Point3="50,50"/>

</PathSegmentCollection>

</PathFigure.Segments>

</PathFigure>

</PathFigureCollection>

</PathGeometry.Figures>

</PathGeometry>

</GeometryGroup>

</Path.Data>

用于GeometryDrawing

<DrawingBrush.Drawing>

<GeometryDrawing Brush="#CCCCFF">

<GeometryDrawing.Pen>

<Pen Thickness="1" Brush="Black" />

</GeometryDrawing.Pen>

<GeometryDrawing.Geometry>

用于Clip

<Image.Clip>

<EllipseGeometry

RadiusX="100"

RadiusY="75"

Center="100,75"/>

</Image.Clip>

ImageBrush

<Ellipse.Fill>

<ImageBrush ImageSource="sampleImages\blueberries.jpg" />

</Ellipse.Fill>

Because the stroke extends

past the shape's bounding box, the image brush's Viewport property

is made larger so that it completely fills the stroke. -->

<Ellipse Grid.Row="6" Grid.Column="0" Height="150" Width="150" StrokeThickness="20"

HorizontalAlignment="Left">

<Ellipse.Stroke>

<ImageBrush ImageSource="sampleImages\blueberries.jpg" Viewport="-10,-10,160,160" ViewportUnits="Absolute" />

</Ellipse.Stroke>

</Ellipse>

<Button.Background>

<ImageBrush ImageSource="sampleImages\blueberries.jpg" />

</Button.Background>

<TextBlock.Foreground>

<ImageBrush ImageSource="sampleImages\blueberries.jpg" />

</TextBlock.Foreground>

An ImageBrush's content alignment can be specified using

the AlignmentX and AlignmentY properties.

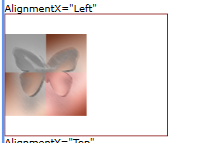
The following are some examples.

需要设置Stretch="None"？

<Rectangle.Fill>

<ImageBrush Stretch="None" AlignmentX="Left" ImageSource="sampleImages\butterfly2.jpg" />

</Rectangle.Fill>



ListBox的ItemContainerStyle是用来设置每一个集合控件的Item的样式的属性

在其中设置<Setter Property="Control.Template">，可以设置datatemplate？

var selLoc = new Uri(selection);

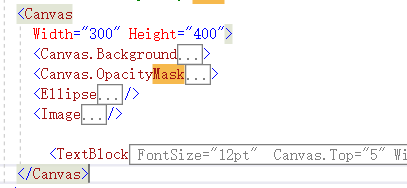
var id = new BitmapImage(selLoc);

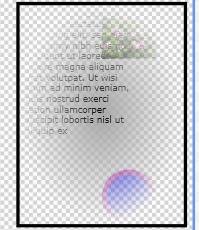
currentImage.Source = id;

OpacityMask是brush类型

public Brush OpacityMask { get; set; }

容器cavas设置OpacityMask后，它包含的元素都会应用上OpacityMask





# FillRule 枚举

Specifies how the intersecting areas of [PathFigure](https://docs.microsoft.com/en-us/dotnet/api/system.windows.media.pathfigure?view=netframework-4.8) objects contained in a [Geometry](https://docs.microsoft.com/en-us/dotnet/api/system.windows.media.geometry?view=netframework-4.8) are combined to form the area of the [Geometry](https://docs.microsoft.com/en-us/dotnet/api/system.windows.media.geometry?view=netframework-4.8).

Stroke也是brush类型

<Line

X1="10" Y1="10"

X2="50" Y2="50"

StrokeThickness="4"

Canvas.Left="100">

<Line.Stroke>

<RadialGradientBrush GradientOrigin="0.5,0.5" Center="0.5,0.5" RadiusX="0.5" RadiusY="0.5">

<RadialGradientBrush.GradientStops>

<GradientStop Color="Red" Offset="0" />

<GradientStop Color="Blue" Offset="0.25" />

</RadialGradientBrush.GradientStops>

</RadialGradientBrush>

</Line.Stroke>

</Line>

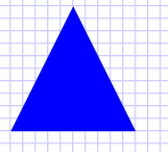
# Polygon Class

Draws a polygon, which is a connected series of lines that form a closed shape.

<!-- Draws a triangle with a blue interior. -->

<Polygon Points="10,110 60,10 110,110"

Fill="Blue" />



也有Stretch属性，和image的stetch属性一样

<Border Height="100" Width="50" Canvas.Top="50" Canvas.Left="70"

BorderBrush="Black" BorderThickness="1" >

<Polygon Height="100" Width="50" Points="0,0 10,10 0,10" Fill="Blue" Stretch="Fill"/>

</Border>

Mouse.Capture(\_elementToCapture);

Mouse.Capture(null);

VisualBrush

<TextBlock Margin="10">

<Bold>Painting with a VisualBrush</Bold><LineBreak/><LineBreak/>

A VisualBrush can be used to paint areas with text, shapes, controls, and more.

</TextBlock>

<VisualBrush Viewport="0,0,95,35" ViewportUnits="Absolute" TileMode="Tile">

<VisualBrush.Visual>

<StackPanel Background="Transparent">

<TextBlock FontSize="10pt" Margin="10">Hello, World!</TextBlock>

</StackPanel>

</VisualBrush.Visual>

<VisualBrush.RelativeTransform>

<RotateTransform Angle="45" CenterX="0.5" CenterY="0.5" />

</VisualBrush.RelativeTransform>

</VisualBrush>

CommandSourceControlUsingSystemTimer

触发slider的增加命令

<Button Command="Slider.IncreaseSmall"

CommandTarget="{Binding ElementName=secondSlider}"

Height="25"

Width="25"

Content="+"/>

var timer = new Timer();

timer.Elapsed += timer\_Elapsed;

timer.Interval = 1000;

timer.Enabled = true;

private void timer\_Elapsed(object sender, ElapsedEventArgs e)

{

// Place delegate on the Dispatcher.

Dispatcher.Invoke(DispatcherPriority.Normal,

new TimerDispatcherDelegate(TimerWorkItem));

}

private void TimerWorkItem()

{

// Update current second display.

lblSeconds.Content = DateTime.Now.Second;

// Forcing the CommandManager to raie the RequerySuggested event.

CommandManager.InvalidateRequerySuggested();

}

private void OnSliderMouseUp(object sender, MouseButtonEventArgs e)

{

var source = e.Source as Slider;

if (source != null)

{

if (e.ChangedButton == MouseButton.XButton1)

{

// Execute the Slider DecreaseSmall RoutedCommand.

// The slider.value propety is passed as the command parameter.

Slider.DecreaseSmall.Execute(

source.Value, source);

}

private void OnSliderMouseWheel(object sender, MouseWheelEventArgs e)

{

var source = e.Source as Slider;

if (source != null)

{

if (e.Delta > 0)

{

// Execute the Slider DecreaseSmall RoutedCommand.

// The slider.value propety is passed as the command parameter.

Slider.DecreaseSmall.Execute(

source.Value, source);

}

CommandSourceControlWithDispatcherTimer

\_dispatcherTimer = new DispatcherTimer();

\_dispatcherTimer.Tick += dispatcherTimer\_Tick;

\_dispatcherTimer.Interval = new TimeSpan(0, 0, 1);

\_dispatcherTimer.Start();

private void dispatcherTimer\_Tick(object sender, EventArgs e)

{

// Updating the Label which displays the current second

lblSeconds.Content = DateTime.Now.Second;

// Forcing the CommandManager to raise the RequerySuggested event

CommandManager.InvalidateRequerySuggested();

}

自定义命令

静态的？

public static RoutedCommand ColorCmd = new RoutedCommand();

<CommandBinding Command="{x:Static local:MainWindow.ColorCmd}"

Executed="ColorCmdExecuted"

CanExecute="ColorCmdCanExecute"/>

private void OpenCmdExecuted(object target, ExecutedRoutedEventArgs e)

{

string command, targetobj;

command = ((RoutedCommand) e.Command).Name;

targetobj = ((FrameworkElement) target).Name;

}

CursorType

Mouse.OverrideCursor = DisplayArea.Cursor;

Mouse.SetCursor(DisplayArea.Cursor);

DisplayArea.Cursor = Cursors.SizeNS;

<Button Command="{x:Static local:MainWindow.ColorCmd}"

CommandParameter="ButtonOne"

CommandTarget="{Binding ElementName=FirstStackPanel}"

Content="CommandTarget = FristStackPanel" />

CanExecute处理方法的e.Source是命令源指定的命令目标FirstStackPanel

private void ColorCmdCanExecute(object sender, CanExecuteRoutedEventArgs e)

{

if (e.Source is Panel)

{

e.CanExecute = true;

}

else

{

e.CanExecute = false;

}

}

EditingCommands

系统内置命令

System.Windows.Documents.EditingCommands.AlignCenter.Execute(null, target);

System.Windows.Documents.EditingCommands.IncreaseFontSize.Execute(null, target);

焦点事件

GotKeyboardFocus="OnGotFocusHandler"

逻辑区域焦点事件

GotFocus="OnGotFocusHandler"

ICommandSourceImplementation

自己实现命令源，需要订阅commad的CanExecuteChanged事件

newCommand.CanExecuteChanged += \_canExecuteChangedHandler;

KeyStrokeCounter

var converter = new KeyConverter();

var target = Key.None;

// Verifying there is only one character in the string.

if (txtTargetKey.Text.Length == 1)

{

// Converting the string to a Key.

target = (Key) converter.ConvertFromString(txtTargetKey.Text);

}

MouseLeftButtonDown

private void HandleButtonDown(object sender, MouseButtonEventArgs e)

{

//Casting the source to a StackPanel

var sourceStackPanel = e.Source as StackPanel;

//If the button is pressed then make dimensions larger.

if (e.ButtonState == MouseButtonState.Pressed)

{

sourceStackPanel.Width = 200;

sourceStackPanel.Height = 200;

}

private void MouseMoveHandler(object sender, MouseEventArgs e)

{

// Get the x and y coordinates of the mouse pointer.

var position = e.GetPosition(this);

var pX = position.X;

var pY = position.Y;

// Creating a FocusNavigationDirection object and setting it to a

// local field that contains the direction selected.

var focusDirection = \_focusMoveValue;

// MoveFocus takes a TraveralReqest as its argument.

var request = new TraversalRequest(focusDirection);

// Gets the element with keyboard focus.

var elementWithFocus = Keyboard.FocusedElement as UIElement;

// Change keyboard focus.

elementWithFocus?.MoveFocus(request);

var elementWithFocus = Keyboard.FocusedElement as UIElement;

// Get the element which would receive focus if focus were changed.

predictionElement = elementWithFocus.PredictFocus(\_focusMoveValue);

The following example calls [CoerceValue](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dependencyobject.coercevalue?view=netframework-4.7.2) within a [PropertyChangedCallback](https://docs.microsoft.com/en-us/dotnet/api/system.windows.propertychangedcallback?view=netframework-4.7.2) implementation that is used as the [PropertyChangedCallback](https://docs.microsoft.com/en-us/dotnet/api/system.windows.propertymetadata.propertychangedcallback?view=netframework-4.7.2) for a different dependency properties on the same class. This is a common pattern for introducing true value dependencies between dependency properties.

In addition to being explicitly invoked through calling [CoerceValue](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dependencyobject.coercevalue?view=netframework-4.7.2), the [CoerceValueCallback](https://docs.microsoft.com/en-us/dotnet/api/system.windows.coercevaluecallback?view=netframework-4.7.2) for a dependency property is also invoked internally whenever the dependency property value is being re-evaluated by the WPF property system.