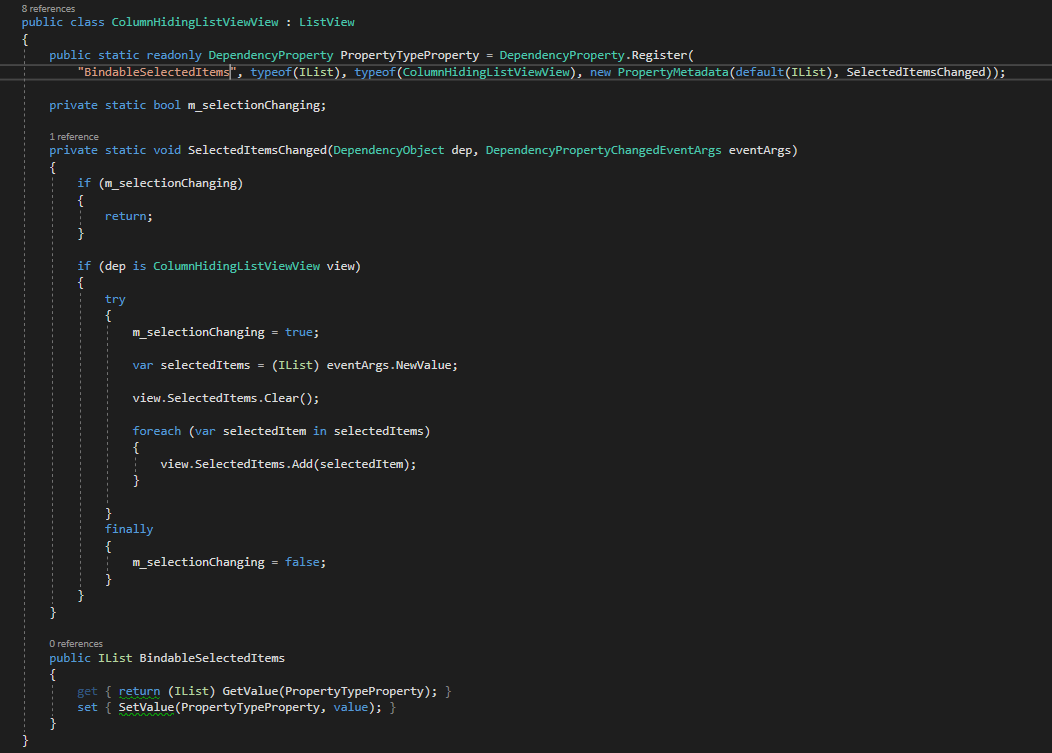
Bindable Selected Items

Postupovano podle: [C:\Users\phlavenka\OneDrive\Nielsen prace\Moje poznamky Nielsen\SelectedItems RadGridView Telerik.docx](file:///C:\Users\phlavenka\OneDrive\Nielsen%20%20prace\Moje%20poznamky%20Nielsen\SelectedItems%20RadGridView%20Telerik.docx)

Z vesmiru je postup popsany zde : [C:\Users\phlavenka\OneDrive\Nielsen prace\Moje poznamky Nielsen\Skoleni - ( po nastupu )\WPF.rtf](file:///C:\Users\phlavenka\OneDrive\Nielsen%20%20prace\Moje%20poznamky%20Nielsen\Skoleni%20-%20(%20po%20nastupu%20)\WPF.rtf)

V Normovadle potrebuju pridat kolekci SelectedItems do ColumnHidingListview.

* Otevru si ColumnHidingListViewView : ListView
* Vytvorim dep property (dep + tab)
* typeOf (IList)
* Nazev property BindableSelectedItems
* Do metadat pridam svoji metodu **SelectedItemsChanged**
* Touto metodou muzu zmenit kolekci z viewModelu



* Ve chvili kdy se zmeni kolekce na ViewModelu, se bindingem (protoze kolekce na VM bude twoWay bindovana na BindableSelectedItems) dostaneme do metody SelectedItemsChanded.

Tady rekneme **prekopiruj moji kolekci z ViewModelu do kolekce SelectedItems na ListView** (ze ktereho dedi ColumnHidingListView)

private static void SelectedItemsChanged(DependencyObject dep, DependencyPropertyChangedEventArgs eventArgs)

{

if (m\_selectionChanging)

{

return;

}

if (dep is ColumnHidingListViewView view)

{

try

{

m\_selectionChanging = true;

**var selectedItems = (IList) eventArgs.NewValue;**

view.SelectedItems.Clear();

**foreach (var selectedItem in selectedItems)**

**{**

**view.SelectedItems.Add(selectedItem);**

**}**

}

finally

{

m\_selectionChanging = false;

}

}

}

* Jeste musime nabindovat novou propertu BindableSelectedItems v xamlu na propertu ve viewModelu.

BindableSelectedItems="{Binding Path=SelectedAlliedMessages}"

**Pozor na to co chces vrazit do kolekce , musi to byt stejne objekty . Tady jsem se divil proc mi nejde do kolekce view.SelectedItems vlozit selectedItem .**

**Pak jsem zkusil setnout prvni tri polozky a zjistil jsem ze musim vkladat itemy z kolekce ktera je source . Jine objekty sem vlozit nejdou I kdyz jsou stejneho typu.**

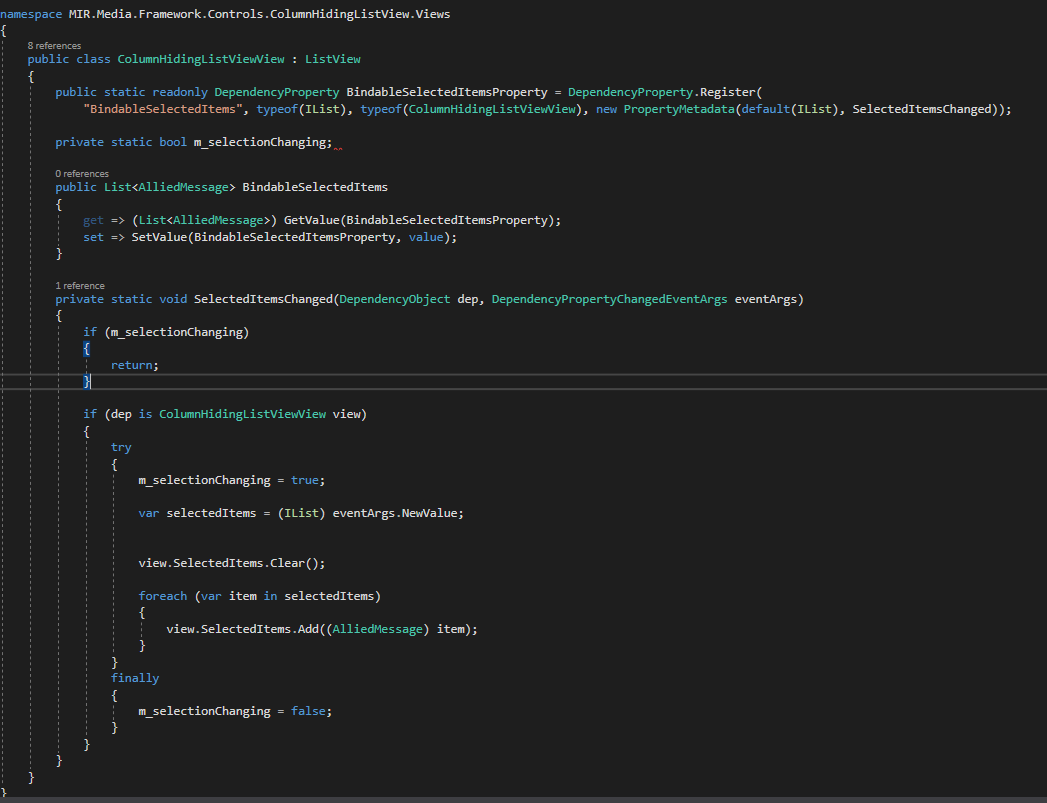
if (SelectedAlliedMessages != null)

{

var list = SelectedAlliedMessages.Select(d => d.Id).ToList();

SelectedAlliedMessages = list.Count > 0 ? new List<AlliedMessage>(**AlliedMessages.Where(d => list.Contains(d.Id)))** : new List<AlliedMessage> {AlliedMessages[0]};

**Konecna verze ColumnHidingListView:**



**Z vesmiru – funguje pro obyc datagrid:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

namespace WpfUniverse.Gui.Controls

{

public class ExtendedDataGrid : DataGrid

{

private List<object> mojeKolekce;

public static readonly DependencyProperty BindableSelectedItemsProperty = DependencyProperty.Register(

"BindableSelectedItems", typeof(IList), typeof(ExtendedDataGrid), new PropertyMetadata(default(IList), SelectedItemsChanged));

private static bool m\_selectionChanging;

public ExtendedDataGrid()

{

}

private static void SelectedItemsChanged(DependencyObject dependencyObject, DependencyPropertyChangedEventArgs d) // Sem proleze z metody OnSelectionChanged

{

if (m\_selectionChanging) // Tady potrebujeme false abychom mohli dal

{

return; // Pri prvnim spusteni tady koncime a jdeme znovu do metody OnSelectionChanged

} // Prolezeme i set v kolekci Bindable a v OnSel prohodime ve finaly boolean.

ExtendedDataGrid dataGrid = dependencyObject as ExtendedDataGrid; // Nacastujeme si nas dependency objekt jako dataGrid

if (dataGrid != null)

{

try

{

m\_selectionChanging = true;

IList selectedItems = (IList) d.NewValue; // Nova hodnota predana v argumentu metody bude IList selectedItems

dataGrid.SelectedItems.Clear(); // Tim ze nas dependency objekt ma predka , ma i kolekci SelectedItems . Vyprazdnime ji.

foreach (var item in selectedItems) // Pro kazdou polozku z Listu ( neboli z d.NewValue )

{

dataGrid.SelectedItems.Add(item); // Pridej polozku do kolekce SelectedItems na datagridu. Ty budou vybrane.

/\*BindableSelectedItems.Add(item)\*/; // STATIC

}

}

finally

{

m\_selectionChanging = false; // Nezapomenout na boolean.

}

}

}

// Metoda vyvolana kliknutim na polozku datagridu event

protected override void OnSelectionChanged(SelectionChangedEventArgs e) // Kliknutim do gridu vlezeme sem

{

base.OnSelectionChanged(e); // Zavolame event rodice

if (m\_selectionChanging) // Pokud je boolean true vratime se

{

return;

}

try // Jinak

{

m\_selectionChanging = true; // Boolean zmenime na true

ArrayList items = new ArrayList();

foreach (var selectedItem in SelectedItems) // Projdeme kolekci na rodicovske tride SelectedItems .

{

items.Add(selectedItem); // Do prazdne kolekce si pridame obsah rodicovske kolekce .

//BindableSelectedItems.Add(selectedItem);

}

BindableSelectedItems = items; //TOHLE UZ JE NULL (reference ??) // Rekneme , ze kolekce BindableSelectedItems (na kterou je navazana kolekce

Console.WriteLine($@"Pocet : {BindableSelectedItems?.Count}"); // v GalaxyViewModelu (na kterou binduje xaml)) bude nase pozbirana kolekce

}

finally

{

m\_selectionChanging = false; // Nakonec prohodime boolean aby se kolekce porad neprepisovala

}

}

// Tato kolekce je stejna jako ta v GalaxyViewModelu (SelectedGalaxies)

public IList BindableSelectedItems //

{

get { return (IList) GetValue(BindableSelectedItemsProperty); } // Vraci hodnotu nasi dependency property v podobe listu

set

{

SetValue(BindableSelectedItemsProperty, value); // Muze dostat list ktery ulozi do dependency propr

}

}

}

}