public sealed class FullObservableCollection<T> : ObservableCollection<T>

where T : INotifyPropertyChanged

{

public FullObservableCollection()

{

CollectionChanged += FullObservableCollectionCollectionChanged;

}

public FullObservableCollection(IEnumerable<T> pItems) : this()

{

foreach (var item in pItems)

{

this.Add(item);

}

}

private void FullObservableCollectionCollectionChanged(object? sender, NotifyCollectionChangedEventArgs e)

{

if (e.NewItems != null)

{

foreach (Object item in e.NewItems)

{

((INotifyPropertyChanged)item).PropertyChanged += ItemPropertyChanged;

}

}

if (e.OldItems != null)

{

foreach (Object item in e.OldItems)

{

((INotifyPropertyChanged)item).PropertyChanged -= ItemPropertyChanged;

}

}

}

private void ItemPropertyChanged(object? sender, PropertyChangedEventArgs e)

{

var args = new NotifyCollectionChangedEventArgs(

NotifyCollectionChangedAction.Replace, sender, sender, IndexOf((T)sender!)

);

OnCollectionChanged(args);

}

}