

# PagedCollectionView Class

Represents a view for grouping, sorting, filtering, and navigating a paged data collection.

## Definition

```
public class PagedCollectionView : INotifyPropertyChanged,  
ICollectionView, IPagedCollectionView, IEditableCollectionView
```

## Summary

The following table summarizes the members exposed in this class.

## Public Constructors

<a href="#">PagedCollectionView Constructor(IEnumerable)</a>	Initializes a new instance of the System.Windows.Data.PagedCollectionView class.
<a href="#">PagedCollectionView Constructor(IEnumerable, bool, bool)</a>	Initializes a new instance of the System.Windows.Data.PagedCollectionView class and specifies whether the data is sorted and in group order.

## Public Properties

<a href="#">CanAddNew</a>	Gets a value that indicates whether a new item can be added to the collection.
<a href="#">CanCancelEdit</a>	Gets a value that indicates whether the collection view can discard pending changes and restore the original values of an edited object.
<a href="#">CanChangePage</a>	Gets a value that indicates whether the System.Windows.Data.PagedCollectionView.PageIndex value can change.
<a href="#">CanFilter</a>	Gets a value that indicates whether this view supports filtering by way of the System.Windows.Data.PagedCollectionView.Filter property.
<a href="#">CanGroup</a>	Gets a value that indicates whether this view supports grouping.
<a href="#">CanRemove</a>	Gets a value that indicates whether an item can be removed from the collection.
<a href="#">CanSort</a>	Gets a value that indicates whether this view supports sorting.
<a href="#">Count</a>	Gets the number of records in the view after filtering, sorting, and paging.
<a href="#">Culture</a>	Gets or sets the cultural information for any operations of the view that might differ by culture, such as sorting.
<a href="#">CurrentAddItem</a>	Gets the item that is being added when an System.Windows.Data.PagedCollectionView.AddNew transaction is in progress.
<a href="#">CurrentEditItem</a>	Gets the item in the collection that is being edited when an System.Windows.Data.PagedCollectionView.EditItem(System.Object) transaction is in progress.
<a href="#">CurrentItem</a>	Gets the current item in the view.
<a href="#">CurrentPosition</a>	Gets the ordinal position of the System.Windows.Data.PagedCollectionView.CurrentItem in the view, which might be sorted and filtered.
<a href="#">Filter</a>	Gets or sets a callback that is used to determine whether an item is suited for inclusion in the view.
<a href="#">FilterDescriptions</a>	
<a href="#">GroupDescriptions</a>	Gets a collection of System.ComponentModel.GroupDescription objects that describe how the items in the collection are grouped in the view.
<a href="#">Groups</a>	Gets the top-level groups, constructed according to the descriptions specified in the System.Windows.Data.PagedCollectionView.GroupDescriptions property.
<a href="#">IsAddingNew</a>	Gets a value that indicates whether an System.Windows.Data.PagedCollectionView.AddNew transaction is in progress.
<a href="#">IsCurrentAfterLast</a>	Gets a value that indicates whether the System.Windows.Data.PagedCollectionView.CurrentItem of the view is beyond the end of the collection.
<a href="#">IsCurrentBeforeFirst</a>	Gets a value that indicates whether the System.Windows.Data.PagedCollectionView.CurrentItem of the view is before the start of the collection.

IsEditingItem	Gets a value that indicates whether an System.Windows.Data.PagedCollectionView.EditItem(System.Object) transaction is in progress.
IsEmpty	Gets a value that indicates whether the view is empty.
IsPageChanging	Gets a value that indicates whether the page index is changing.
this[int]	Gets the item at the specified index.
ItemCount	Gets the minimum number of items known to be in the source collection that satisfy the current filter.
NeedsRefresh	Gets a value that indicates whether this view needs to be refreshed.
NewItemPlaceholderPosition	Gets or sets a value that indicates whether to include a new item placeholder in the collection view, and where to include it.
PageIndex	Gets the zero-based index of the current page.
PageSize	Gets or sets the number of items to display on a page.
SortDescriptions	Gets a collection of System.ComponentModel.SortDescription objects that describe how the items in the collection are sorted in the view.
SourceCollection	Gets the System.Collections.IEnumerable collection underlying this view.
TotalItemCount	Gets the total number of items in the view before paging is applied.

## Public Methods

AddNew()	Adds a new item to the underlying collection.
CancelEdit()	Ends the edit transaction and, if it is possible, restores the original value of the item.
CancelNew()	Ends the add transaction and discards the pending new item.
CommitEdit()	Ends the edit transaction and saves the pending changes.
CommitNew()	Ends the add transaction and saves the pending new item.
Contains(object)	Returns a value that indicates whether the specified item belongs to this collection view.
CopyDefinitionsFrom(PagedCollectionView)	Copy the paged collection view definition from the given source.
CopyDefinitionsFrom(PagedCollectionView, bool, bool, bool, bool)	Copy the paged collection view definition from the given source.
DeferRefresh()	Enters a defer cycle that you can use to merge changes to the view and delay automatic refresh.
EditItem(object)	Begins an edit transaction on the specified item.
GetEnumerator()	Returns an System.Collections.IEnumerator object that you can use to enumerate the items in the view.
GetItemAt(int)	Gets the item at the specified zero-based index in this System.Windows.Data.PagedCollectionView, after the source collection is filtered, sorted, and paged.
IndexOf(object)	Returns the zero-based index at which the specified item is located.
MoveCurrentTo(object)	Sets the specified item to be the System.Windows.Data.PagedCollectionView.CurrentItem in the view.
MoveCurrentToFirst()	Sets the first item in the view as the System.Windows.Data.PagedCollectionView.CurrentItem.
MoveCurrentToLast()	Sets the last item in the view as the System.Windows.Data.PagedCollectionView.CurrentItem.
MoveCurrentToNext()	Set the item after the System.Windows.Data.PagedCollectionView.CurrentItem in the view as the System.Windows.Data.PagedCollectionView.CurrentItem.
MoveCurrentToPosition(int)	Sets the item at the specified index to be the System.Windows.Data.PagedCollectionView.CurrentItem in the view.
MoveCurrentToPrevious()	Sets the item before the System.Windows.Data.PagedCollectionView.CurrentItem in the view as the System.Windows.Data.PagedCollectionView.CurrentItem.
MoveToFirstPage()	Sets the first page as the current page.
MoveToLastPage()	Sets the last page as the current page.
MoveToNextPage()	Moves to the page after the current page.
MoveToPage(int)	Moves to the page at the specified index.
MoveToPreviousPage()	Moves to the page before the current page.
PassesFilter(object)	Returns a value that indicates whether the specified item in the underlying collection belongs to the view after filters are applied.

Refresh()	Re-creates the view.
Remove(object)	Removes the specified item from the collection.
RemoveAt(int)	Removes the item at the specified position from the collection.

## Events

CollectionChanged	Occurs when the view has changed.
CurrentChanged	Occurs after the System.Windows.Data.PagedCollectionView.CurrentItem has changed.
CurrentChanging	Occurs when the System.Windows.Data.PagedCollectionView.CurrentItem is changing.
PageChanged	Occurs after the System.Windows.Data.PagedCollectionView.PageIndex has changed.
PageChanging	Occurs when the System.Windows.Data.PagedCollectionView.PageIndex is changing.
PropertyChanged	Occurs after a property value has changed.

## Public Constructors

```
public PagedCollectionView(IEnumerable source)
```

Initializes a new instance of the System.Windows.Data.PagedCollectionView class.

Parameters

source	The source for the collection.
--------	--------------------------------

```
public PagedCollectionView(IEnumerable source, bool isDataSorted, bool isDataInGroupOrder)
```

Initializes a new instance of the System.Windows.Data.PagedCollectionView class and specifies whether the data is sorted and in group order.

Parameters

source	The source for the collection.
isDataSorted	Specifies whether the source is already sorted.
isDataInGroupOrder	Specifies whether the source is already in the correct order for grouping.

Exceptions

ArgumentNullException	<paramref name="source" /> is null.
-----------------------	-------------------------------------

## Public Properties

```
public bool CanAddNew { get; }
```

Gets a value that indicates whether a new item can be added to the collection.

```
public bool CanCancelEdit { get; }
```

Gets a value that indicates whether the collection view can discard pending changes and restore the original values of an edited object.

```
public bool CanChangePage { get; }
```

Gets a value that indicates whether the System.Windows.Data.PagedCollectionView.PageIndex value can change.

```
public bool CanFilter { get; }
```

Gets a value that indicates whether this view supports filtering by way of the System.Windows.Data.PagedCollectionView.Filter property.

```
public bool CanGroup { get; }
```

Gets a value that indicates whether this view supports grouping.

```
public bool CanRemove { get; }
```

Gets a value that indicates whether an item can be removed from the collection.

```
public bool CanSort { get; }
```

Gets a value that indicates whether this view supports sorting.

```
public int Count { get; }
```

Gets the number of records in the view after filtering, sorting, and paging.

```
public CultureInfo Culture { get; set; }
```

Gets or sets the cultural information for any operations of the view that might differ by culture, such as sorting.

Exceptions

ArgumentNullException	The culture information is null.
-----------------------	----------------------------------

```
public object CurrentAddItem { get; }
```

Gets the item that is being added when an `System.Windows.Data.PagedCollectionView.AddNew` transaction is in progress.

```
public object CurrentEditItem { get; }
```

Gets the item in the collection that is being edited when an `System.Windows.Data.PagedCollectionView.EditItem(System.Object)` transaction is in progress.

```
public object CurrentItem { get; }
```

Gets the current item in the view.

```
public int CurrentPosition { get; }
```

Gets the ordinal position of the `System.Windows.Data.PagedCollectionView.CurrentItem` in the view, which might be sorted and filtered.

```
public Predicate<object> Filter { get; set; }
```

Gets or sets a callback that is used to determine whether an item is suited for inclusion in the view.

Exceptions

NotSupportedException	The implementation does not support filtering. Simpler implementations do not support filtering and will throw a <code>System.NotSupportedException</code> . Check the <code>System.Windows.Data.PagedCollectionView.CanFilter</code> property to test whether filtering is supported before assigning a non-null value.
InvalidOperationException	<code>System.Windows.Data.PagedCollectionView.IsAddingNew</code> is true.-or- <code>System.Windows.Data.PagedCollectionView.IsEditingItem</code> is true.

```
public FilterDescriptionCollection FilterDescriptions { get; }
```

```
public ObservableCollection<GroupDescription> GroupDescriptions { get; }
```

Gets a collection of `System.ComponentModel.GroupDescription` objects that describe how the items in the collection are grouped in the view.

```
public ReadOnlyObservableCollection<object> Groups { get; }
```

Gets the top-level groups, constructed according to the descriptions specified in the `System.Windows.Data.PagedCollectionView.GroupDescriptions` property.

```
public bool IsAddingNew { get; }
```

Gets a value that indicates whether an `System.Windows.Data.PagedCollectionView.AddNew` transaction is in progress.

```
public bool IsCurrentAfterLast { get; }
```

Gets a value that indicates whether the `System.Windows.Data.PagedCollectionView.CurrentItem` of the view is beyond the end of the collection.

```
public bool IsCurrentBeforeFirst { get; }
```

Gets a value that indicates whether the `System.Windows.Data.PagedCollectionView.CurrentItem` of the view is before the start of the collection.

```
public bool IsEditingItem { get; }
```

Gets a value that indicates whether an `System.Windows.Data.PagedCollectionView.EditItem(System.Object)` transaction is in progress.

```
public bool IsEmpty { get; }
```

Gets a value that indicates whether the view is empty.

```
public bool IsPageChanging { get; }
```

Gets a value that indicates whether the page index is changing.

```
public object this[int index] { get; }
```

Gets the item at the specified index.

Parameters

index	The index of the item to be retrieved.
-------	--

```
public int ItemCount { get; }
```

Gets the minimum number of items known to be in the source collection that satisfy the current filter.

```
public bool NeedsRefresh { get; }
```

Gets a value that indicates whether this view needs to be refreshed.

```
publicNewItemPlaceholderPosition NewItemPlaceholderPosition { get; set; }
```

Gets or sets a value that indicates whether to include a new item placeholder in the collection view, and where to include it.

Exceptions

ArgumentException	The value is not System.ComponentModel.NewItemPlaceholderPosition.None.
-------------------	---

```
public int PageIndex { get; }
```

Gets the zero-based index of the current page.

```
public int PageSize { get; set; }
```

Gets or sets the number of items to display on a page.

Exceptions

ArgumentOutOfRangeException	The page size is less than 0.
InvalidOperationException	The page size could not be changed because there is a new or edited item that could not be committed.

```
public SortDescriptionCollection SortDescriptions { get; }
```

Gets a collection of System.ComponentModel.SortDescription objects that describe how the items in the collection are sorted in the view.

```
public IEnumerable SourceCollection { get; }
```

Gets the System.Collections.IEnumerable collection underlying this view.

```
public int TotalItemCount { get; }
```

Gets the total number of items in the view before paging is applied.

## Public Methods

```
public object AddNew()
```

Adds a new item to the underlying collection.

Return Types

The new item that is being added.
-----------------------------------

Exceptions

InvalidOperationException	System.Windows.Data.PagedCollectionView.CanAddNew is false.
---------------------------	---

```
public void CancelEdit()
```

Ends the edit transaction and, if it is possible, restores the original value of the item.

Exceptions

InvalidOperationException	System.Windows.Data.PagedCollectionView.CanCancelEdit is false.-or-System.Windows.Data.PagedCollectionView.IsAddingNew is true.
---------------------------	---

```
public void CancelNew()
```

Ends the add transaction and discards the pending new item.

Exceptions

InvalidOperationException	System.Windows.Data.PagedCollectionView.IsEditingItem is true.
---------------------------	--

public void CommitEdit()

Ends the edit transaction and saves the pending changes.

Exceptions

InvalidOperationException	System.Windows.Data.PagedCollectionView.IsAddingNew is true.
---------------------------	--

public void CommitNew()

Ends the add transaction and saves the pending new item.

Exceptions

InvalidOperationException	System.Windows.Data.PagedCollectionView.IsEditingItem is true.
---------------------------	--

public bool Contains(object item)

Returns a value that indicates whether the specified item belongs to this collection view.

Parameters

item	The object to check.
------	----------------------

Return Types

true if the specified item belongs to this collection view; otherwise, false.
---

public void CopyDefinitionsFrom(PagedCollectionView source)

Copy the paged collection view definition from the given source.

Parameters

source	
--------	--

public void CopyDefinitionsFrom(PagedCollectionView source, bool includeGroupDescriptions, bool includeSortDescriptions, bool includeFilterDescriptions, bool includePageDescription)

Copy the paged collection view definition from the given source.

Parameters

source	
includeGroupDescriptions	Copy the group description.
includeSortDescriptions	Copy the sort description.
includeFilterDescriptions	Copy the filter description.
includePageDescription	Copy the paging description.

public IDisposable DeferRefresh()

Enters a defer cycle that you can use to merge changes to the view and delay automatic refresh.

Return Types

An System.IDisposable object that you can use to dispose of the calling object.
---

Exceptions

InvalidOperationException	System.Windows.Data.PagedCollectionView.IsAddingNew is true.-or-System.Windows.Data.PagedCollectionView.IsEditingItem is true.
---------------------------	--

public void EditItem(object item)

Begins an edit transaction on the specified item.

Parameters

item	The item to edit.
------	-------------------

public IEnumerator GetEnumerator()

Returns an System.Collections.IEnumerator object that you can use to enumerate the items in the view.

## Return Types

An `System.Collections.IEnumerator` object that you can use to enumerate the items in the view.

```
public object GetItemAt(int index)
```

Gets the item at the specified zero-based index in this `System.Windows.Data.PagedCollectionView`, after the source collection is filtered, sorted, and paged.

### Parameters

index	The zero-based index of the item to retrieve.
-------	---

## Return Types

The item at the specified zero-based index.

## Exceptions

<code>ArgumentOutOfRangeException</code>	<code>&lt;paramref name="index" /&gt;</code> is less than 0, or greater than or equal to the <code>System.Windows.Data.PagedCollectionView.Count</code> of items in the view.
--	---

```
public int IndexOf(object item)
```

Returns the zero-based index at which the specified item is located.

### Parameters

item	The item to locate.
------	---------------------

## Return Types

The index at which the specified item is located, or `-1` if the item is unknown.

```
public bool MoveCurrentTo(object item)
```

Sets the specified item to be the `System.Windows.Data.PagedCollectionView.CurrentItem` in the view.

### Parameters

item	The item to set as the <code>System.Windows.Data.PagedCollectionView.CurrentItem</code> .
------	---

## Return Types

true if the resulting `System.Windows.Data.PagedCollectionView.CurrentItem` is an item in the view; otherwise, false.

```
public bool MoveCurrentToFirst()
```

Sets the first item in the view as the `System.Windows.Data.PagedCollectionView.CurrentItem`.

### Return Types

true if the resulting `System.Windows.Data.PagedCollectionView.CurrentItem` is an item in the view; otherwise, false.

```
public bool MoveCurrentToLast()
```

Sets the last item in the view as the `System.Windows.Data.PagedCollectionView.CurrentItem`.

### Return Types

true if the resulting `System.Windows.Data.PagedCollectionView.CurrentItem` is an item in the view; otherwise, false.

```
public bool MoveCurrentToNext()
```

Set the item after the `System.Windows.Data.PagedCollectionView.CurrentItem` in the view as the `System.Windows.Data.PagedCollectionView.CurrentItem`.

### Return Types

true if the resulting `System.Windows.Data.PagedCollectionView.CurrentItem` is an item in the view; otherwise, false.

```
public bool MoveCurrentToPosition(int position)
```

Sets the item at the specified index to be the `System.Windows.Data.PagedCollectionView.CurrentItem` in the view.

### Parameters

position	The index to set the System.Windows.Data.PagedCollectionView.CurrentItem to.
----------	--

#### Return Types

true if the resulting System.Windows.Data.PagedCollectionView.CurrentItem is an item in the view; otherwise, false.
---

#### Exceptions

ArgumentOutOfRangeException	<paramref name="position" /> is less than -1 or greater than the System.Windows.Data.PagedCollectionView.Count of items in the view.
-----------------------------	--

public bool MoveCurrentToPrevious()

Sets the item before the System.Windows.Data.PagedCollectionView.CurrentItem in the view as the System.Windows.Data.PagedCollectionView.CurrentItem.

#### Return Types

true if the resulting System.Windows.Data.PagedCollectionView.CurrentItem is an item in the view; otherwise, false.
---

public bool MoveToFirstPage()

Sets the first page as the current page.

#### Return Types

true if the operation was successful; otherwise, false.
---

public bool MoveToLastPage()

Sets the last page as the current page.

#### Return Types

true if the operation was successful; otherwise, false.
---

public bool MoveToNextPage()

Moves to the page after the current page.

#### Return Types

true if the operation was successful; otherwise, false.
---

public bool MoveToPage(int pageIndex)

Moves to the page at the specified index.

#### Parameters

pageIndex	The index of the page to move to.
-----------	-----------------------------------

#### Return Types

true if the operation was successful; otherwise, false.
---

public bool MoveToPreviousPage()

Moves to the page before the current page.

#### Return Types

true if the operation was successful; otherwise, false.
---

public bool PassesFilter(object item)

Returns a value that indicates whether the specified item in the underlying collection belongs to the view after filters are applied.

#### Parameters

item	The item to check.
------	--------------------

#### Return Types

true if the specified item belongs to the filtered view or if no filter is set on the collection view; otherwise, false.
--



public void Refresh()

Re-creates the view.

Exceptions

InvalidOperationException	System.Windows.Data.PagedCollectionView.IsAddingNew is true.-or- System.Windows.Data.PagedCollectionView.IsEditingItem is true.
---------------------------	---

public void Remove(object item)

Removes the specified item from the collection.

Parameters

item	The item to remove.
------	---------------------

public void RemoveAt(int index)

Removes the item at the specified position from the collection.

Parameters

index	The zero-based index of the item to remove.
-------	---

Exceptions

ArgumentOutOfRangeException	<paramref name="index" /> is less than 0, or greater than or equal to the System.Windows.Data.PagedCollectionView.Count of items in the view.
InvalidOperationException	System.Windows.Data.PagedCollectionView.IsAddingNew is true.-or- System.Windows.Data.PagedCollectionView.IsEditingItem is true.-or-System.Windows.Data.PagedCollectionView.CanRemove is false.

## Events

public event NotifyCollectionChangedEventHandler CollectionChanged

Occurs when the view has changed.

public event EventHandler CurrentChanged

Occurs after the System.Windows.Data.PagedCollectionView.CurrentItem has changed.

public event CurrentChangingEventHandler CurrentChanging

Occurs when the System.Windows.Data.PagedCollectionView.CurrentItem is changing.

public event EventHandler<EventArgs> PageChanged

Occurs after the System.Windows.Data.PagedCollectionView.PageIndex has changed.

public event EventHandler<PageChangingEventArgs> PageChanging

Occurs when the System.Windows.Data.PagedCollectionView.PageIndex is changing.

public event PropertyChangedEventHandler PropertyChanged

Occurs after a property value has changed.

---

Namespace: Intersoft.Client.Data.ComponentModel

Assembly Information: Intersoft.Client.Data.ComponentModel, Version 3.0.5000.1

Target Frameworks: Silverlight 5, WPF 4