bea stollnitz
on Silverlight and WPF

contact me

blog about me

Search Blog

Blog

- RSS
- Archive index

Books

Books I tech reviewed and recommend

- Programming WPF
- WPF Unleashed
- WPF in Action
- Data-Driven Services with SL 2

Sorry I can't reply to every email and comment. If you need a fast response, try the WPF Forum or Silverlight Forum.
All original code referenced in this blog is licensed under the Open Software License version 3.0.
March 5, 2006

How do I bind to ADO.NET?

Binding to ADO.NET data is for the most part similar to binding to objects or XML, but there are enough small differences to fill up a blog post. Designing Avalon support for ADO.NET was a little bit of a challenge: on one hand, it was important for us to maintain consistency across the different data sources; on the other, we felt it made sense to delegate as much work as we could to the ADO.NET classes (which at times broke that consistency).

In this post I will:

- Show how to make a 3 level master-detail scenario with ADO.NET.
- Explain how (and why) change notifications work.
- Explain how to filter ADO.NET data.
- How to make a 3 level master-detail scenario with ADO.NET

In the VS project linked below, you'll find a very simple Access "mdb" database file with a hierarchical data source I used in previous posts. There are three tables in this database: Mountains, Lifts, and Runs. There is a one-to-many relationship between Mountains and Lifts and another one-to-many relationship between Lifts and Runs. In the GetData() method in the Window code behind you can find the ADO.NET code I use to read the Access data into a DataSet. I won't go into much detail about that, since there are plenty of resources out there to learn ADO.NET. The only part worth showing here is the creation of the relations between the tables because we will need their names later. The MountainsLifts relation links the Moutains and Lifts tables through the Mountain_ID column. Similarly, the LiftsRuns relation links the Lifts and Runs tables through the Lift_ID column.

```
dataSet.Relations.Add("MountainsLifts",
    tables["Mountains"].Columns["Mountain_ID"],
    tables["Lifts"].Columns["Mountain_ID"]);

dataSet.Relations.Add("LiftsRuns",
    tables["Lifts"].Columns["Lift_ID"],
    tables["Runs"].Columns["Lift_ID"]);
```

Once you have your data in place, making a three-level master-detail application is very simple, as shown in the following markup. Notice that the DataContext is set somewhere up in the tree to the "Mountains" DataTable.

```
<ListBox ItemsSource="{Binding}" DisplayMemberPath="Mountain_Name" IsSynchronizedWithCurrentItem="true" Name="lbMountains"/>
<ListBox ItemsSource="{Binding Path=MountainsLifts}" DisplayMemberPath="Lift_Name" IsSynchronizedWithCurrentItem="true" />
```

```
    <ListBox ItemsSource="{Binding Path=MountainsLifts/LiftsRuns}" DisplayMemberPath="Run_Name" IsSynchronizedWithCurrentItem="true" />
```

You might be wondering how the second binding could possibly work, when there is no property MountainsLifts in DataTable.

When the data binding engine sees that the source of a binding implements IListSource (as DataTable does), it automatically calls the GetList() method, which in this case gives us the DataView. Rows in a DataView are of type DataRowView, which implements ICustomTypeDescriptor. Data binding honors this interface and will call the GetProperties() method on DataView, which will give us a few more properties. If you are curious about what those properties are, you can insert the following code somewhere in your code behind and add a breakpoint to inspect the collection:

```
    IList dataView = ((IListSource)ds.Tables["Mountains"]).GetList();
    PropertyDescriptorCollection coll = ((ICustomTypeDescriptor)dataView[0]).GetProperties();
```

You will see that coll provides three new properties to the DataRowView: the data value for each column (named Mountain_ID and Mountain_Name) and the relation we added earlier (named MountainsLifts). In the markup above, we are binding to the property MountainsLifts of the first table, which returns a DataView containing just those data items in the second table that are related to the currently selected row.

How (and why) change notifications work

If you are using an object data source and you want the UI to be notified of changes in the source, your source object should implement INotifyCollectionChanged and/or INotifyPropertyChanged. INotifyCollectionChanged is used to tell the data binding engine that items have been added, removed, or moved within a collection. INotifyPropertyChanged is used to indicate that a property value has changed. Note that INotifyCollectionChanged does not cover changes to properties of items within a collection — the items should implement INotifyPropertyChanged themselves. I won’t show an example of this here, since we have samples in the Avalon SDK for this. ADO.NET works differently.

The sample application includes a Button that adds a new Mountain to the Mountains table:

```
    private void AddItem(object sender, RoutedEventArgs args)
    {
        DataRow row = ds.Tables["Mountains"].NewRow();
        row["Mountain_ID"] = 4;
        row["Mountain_Name"] = "Big White";
        ds.Tables["Mountains"].Rows.Add(row);
    }
```

It comes as a surprise to some people that with this code, the item is also added to the UI, even though none of the ADO.NET objects implement our INotifyCollectionChanged interface. It turns out that ADO.NET’s DataView already implements its own form of notification using the IBindingList interface, and Avalon’s data binding engine understands that interface in addition to the usual ones. In particular, Avalon listens for the ListChanged event, whose ListChangedEventArgs contain a ListChangedType enum with values such as “ItemAdded”, “ItemDeleted”, and “ItemChanged”.

How to filter ADO.NET data

You can filter the items in XML and CLR collections by writing some custom code in the form of a delegate that gets called for each item. This delegate returns true or false depending on whether we want that particular item to be filtered out. You apply the filter by getting the CollectionView associated with your collection, and setting its Filter property to your delegate.

Filtering in ADO.NET is a little different, though. It is done by assigning an expression (written as a string) to the DataView’s Filter property. When using a DataView as your source, Avalon allows you to use ADO.NET filtering directly (which is very efficient):

```
    private void FilterItems(object sender, RoutedEventArgs args)
    {
        BindingListCollectionView view = (BindingListCollectionView)CollectionViewSource.GetDefaultView(ds.Tables["Mountains"]);
        //bool canFilter = view.CanFilter; // false
        //bool canCustomFilter = view.CanCustomFilter; // true

        view.CustomFilter = "Mountain_Name <> ‘Crystal Mountain’";
    }
```

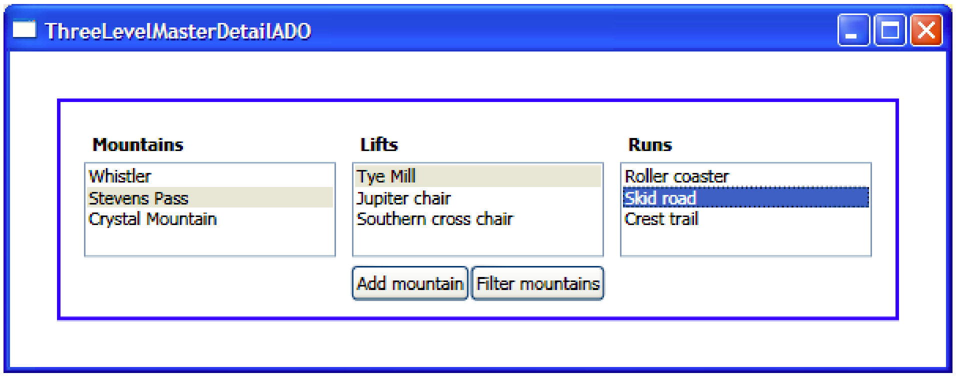
In this sample, I am binding directly to the DataTable, and not to a CollectionViewSource that wraps the DataTable (as I did in a few previous blog posts). In this case, the data binding engine generates a default view of type BindingListCollectionView and binds to it. The static GetDefaultView method on CollectionViewSource returns a handle to the default view we generated internally. Notice that in this case the view type is BindingListCollectionView (and not ListCollectionView as in my previous posts with other data types) because DataView implements IBindingList. BindingListCollectionView has a new property CustomFilter of type string that allows us to set the filter expression to be passed to the DataView’s Filter property. Its CanFilter property will return false because you can’t filter it with the delegate you would use for other source types, but its CanCustomFilter property will return true.

Delegating all our filtering (and sorting) to the underlying DataView has one consequence that you should be aware of. If you have two different

CollectionViewSources that refer to the same object or XML collection, and you filter one of them, the other one remains unfiltered. However, if the source is a DataTable or DataView, filtering the first CollectionViewSource causes the second one to be filtered as well. Because filtering is done at the source level and not at the view level, it affects both views.

The data binding team has been discussing whether or not we should change this behavior for ADO.NET. Maintaining consistency is very important for us because it helps people use their intuition to learn our features. Some of us claim that the platform would be more consistent if filtering affected ADO.NET sources in the same way as other sources, meaning that a filter applied to one CollectionViewSource should have no effect on another. Others argue that our implementation is consistent in the sense that we delegate all operations to the DataView.

If you want an opportunity to help shape Avalon, I would love to hear your opinions on this issue (note that if the current behavior changes, it will be in the Avalon V2 timeframe only).



Here you can find the VS project with this sample code. This works with February CTP WPF bits.

Posted by Bea under WPF | Comments (36)

36 Responses to “How do I bind to ADO.NET?”

1. *Anonymous* says:
March 6, 2006 at 7:36 am

If you need a vote, I think that delegating of operations to DataView is as how it should be.

Speaking of the “use of intuition to learn WPF” – one must be a mind reader plus have Reflector (or WPF source code) open at all times to figure out how to make even the smallest step.

Reference to Avalon SDK samples is also very promising provided that some of them are still May or Sep CTP samples.

Reply

2. *Marc* says:
March 7, 2006 at 12:26 am

Hi Beatriz,

1.- In my opinion, filtering and sorting (F&S) in the presentation (P) and business (B) layers imply different semantics. It is not because the DataView tends to amalgamate both of them that we should carry the same design principles into the WPF architecture.

2.- As a reminder, the semantics of ObservableCollection(T) (WPF’s view on the business layer) and DataView are different. IList(T) is an ordered collection that contractually supports indexing together with Find* and Sort operations within the business layer (.NET documentation: Ilist(T) represents a strongly typed list of objects that can be accessed by index. Provides methods to search, sort, and manipulate lists; ObservableCollection(T) implements IList(T)). On the other hand, the DataView is a reflection of the relational world (unordered collections) that can be subject of ordering and filtering.

3.- IList(T) operations meet business layer requirements, e.g. when I invoice orderLines stored in that layer. In this layer, I might wish to manipulate only non-invoiced orderLines (a B filter excludes the invoiced orderLines). I also might wish to organize the list in such a manner that the items are indexed in decreasing money-value order (a B indexing). On the other hand, I might wish to have one view on the orderLines sorted by date (P sort) in addition to that one sorted by money value. I might also wish to have different P filters in order to be able to delegate the invoicing decision to different departments. In this case, a B filter would probably be less appropriate because I want to keep the order itself consistent (hence do not want to split it up within the B layer) (see point 4.- hereafter).

4.- Let us also not forget the potential consequences of future object/relational mappings (LINQ, etc.) with respect to the present question. That would indeed imply a tighter integration of the business layer with data storage. In this case, the capability to push the F&S operations into the P layer without impact on the B layer seems imperative (think about the ADO.Net alternative where multiple filtered views would be needed within the 'business layer') (see last example in 3.- above).

5.- Summarizing, I feel that F&S operations have their place in both layers be it with different design intentions. Thanks to WPF, a clearer separation (than was the case with ADO.NET) between P and B layers tends to exist and we should take this opportunity to clarify the design options in this respect. In my experience, most people are comfortable with the intuition that F&S can occur within different application layers with different consequences on the meaning of the data being shown.

Kind regards,

Marc

Reply

3. *massimo* says:
March 11, 2006 at 8:58 am

Some comments about the sample:

- I noticed that filtering can also be done using the 'DefaultView' Property of the 'DataTable' class:

DataView view = ds.Tables["Mountains"].DefaultView;
view.RowFilter= "Mountain_Name <> 'Crystal Mountain'";

- I rebuilt the sample using a typed DataSet (a DataSet subclass generated with Visual Studio 2005) named 'SkiMountainsDataSet', in this way it is possible to add an instance of SkiMountainsDataSet to the Window Resources and then assign it to the DataContext of the StackPanel, all in XAML (the Binding Paths of the ListBoxes must be changed to 'Mountains', 'Mountains/MountainsLifts' and 'Mountains/MountainsLifts/LiftsRuns'). The only thing that still needs to be done in code is loading data into the SkiMountainsDataSet DataTables (very easy with the Table Adapters generated by Visual Studio 2005)

- In any case, the sample stops working if you click 'Add mountain', select 'Crystal Mountain' in the first ListBox and then click 'Filter mountains'

Reply

○ *Bea* says:
March 12, 2006 at 3:40 pm

Massimo,

- When you use BindingListCollectionView's CustomFilter, all we do is delegate the work to the DataView's RowFilter. You can think of BLCV as an Avalon wrapper around the ADO.NET objects that delegates as much work as it can to ADO.NET's DataView. So yes, filtering the data using DataView's RowFilter is essentially the same thing.

- Thanks for the details on typed DataSets. I like to hear when people do cool integration experiments with data binbing.

- That is a bug. Thanks for finding it and letting me know about it. I will follow up on that.

Reply

4. *Anonymous* says:
March 15, 2006 at 6:58 pm

Hi Beatriz,

Is it possible to benefits from the master-details relations within a single ListView ?

Suppose i have a BowlerScores table and a Bowlers table, both linked with a bowler Id key.

When i display the scores list for a specific team, i would like to display the full name of the player in the listview and his 3 game scores.

I've set my relations about like you but i can't seem to get that to work for a single ListView.

As show in your Other samples, i've set the listview's view with a series of GridViewColumns. The source for the ListView is the BowlerScores table of the dataset. I just need 1 column to hold the Bowler's name that is in the Bowlers table in the dataset.

I tried figuring playing with Path. Then i tried setting a CellTemplate. I just can't get to figure it out. Yet, i am almost certain that this is possible without having to create a different dataset.

Or are things working the Other way around with WPF ????
I mean, do i have to select the Player so that all the player's scores will become 'filtered' ?

Your help will be very appreciated.

Reply

○ *Bea* says:
March 16, 2006 at 6:01 pm

Replying to your question about using master-detail in a single ListView:

I believe you are confusing two concepts:
- "ADO.NET relationship", which builds up a relation between two DataTables.
- "Master-detail scenario", which is a way to keep the data displayed in two controls in sync through the notion of currency. Typically, one of your controls displays a list of items and the other one displays more details about the selected (and current) item. But you can also do master-detail to keep selection of two ListBoxes in sync (although this is not as common).

If you want to display data in one control only (ListView) that comes from two tables linked by a relationship, I wouldn't call that master-detail. Naming aside, how would you do that scenario? I would suggest combining the data of the two tables you are interested in in an ADO.NET DataView and use that as the source of the ListView.

Let me know if this helped.

Bea

Reply

5. *Tobias* says:
March 16, 2006 at 8:53 am

Hi Beatriz,

I was just trying to throw two of your posts together to get an ADO.NET data bound treeview (most simple hierarchical data in one table with columns like ID / ParentID / Name and a relation to itself). I tried to use a combination of HierarchicalDataTemplate and ADO.NET binding but did not succeed. Any thoughts how this would work best?

Kind Regards,
Tobias

Reply

○ *Tobias* says:
March 17, 2006 at 12:01 pm

Hi Beatriz,

I finally figured out how to setup the treeview to bind to relational typed dataset (table with columns CategoryId / ParentCategoryId / Name).

What I finally did:

1. enhanced the CategoryRow class by wrapping the GetCategoryRows() function in a ChildCategories property

2. enhanced the CategoryDataTable by adding a property RootCategories returning all categories with ParentId NULL

3. instantiated the dataset and assigned the Categories table to the underlying DockPanel

4. assigned the new RootCategories property to the TreeView ItemsSource and assigned the new ChildCategories property to the HierarchicalDataTemplate ItemSource used by the treeview

This is working perfectly fine. Nevertheless I would bet you know a way to accomlish the same (steps 1 and 2) in XAML without code behind :-)

Have a nice weekend!
Tobias

Reply

- *Bea* says:
  March 18, 2006 at 4:29 pm

  Hi Tobias,

  Thank you for the vote of confidence. :) Indeed, I was able to get your scenario to work all in XAML. Let me know if this is what you wanted.

  Bea

  Reply

  - *Tobias* says:
    March 20, 2006 at 1:13 pm

    Hi Beatriz,

    thanks a lot for the sample code. Unfortunately it does not show how to fill the treeview with hierarchical data stored in one table: trying to adapt this example to my scenario, where all tree nodes are defined in one table with a relation (Parent-Child) defined between two fields of the same table (virtually unlimited depth of the tree branches), the data binding seems to take the "wrong direction": Child to Parent instead of Parent to Child. I can't figure out how to set this up correctly without defining an additional property in code.

    Regards,
    Tobias

    Reply

  - *Bea* says:
    March 23, 2006 at 9:15 am

    Hi Tobias,

    Ok, second try here.

    Let me know if this works out for you.

    Bea

    Reply

  - *Bea* says:
    March 25, 2006 at 2:31 am

    Hi Bea,

    this is exactly what I was looking for. The setup of the BindingListCollectionView finally did the trick. Thanks a lot!

    Tobias

    Reply

6. *Anonymous* says:
March 19, 2006 at 12:32 pm

Hi,

I try to understand WPF databinding.

I have a custom component for WinForms applications that abstracts
DataBinding issues in WinForms.

To get notified when the UI changes a DataView (I bind to) (ComboBox
chnanges position) I use the BindingContext of my form in my component. This
works, of course.

Now, in WPF I see that there is no BindingContext anymore.

My question:
Is it possible to get notifed when the UI changes position of an Item outside my Window,
is there something like BindingContext?

I see there is a ItemChanged event in BindingListCollectionView but I want
to bind against System.Data.DataSet.

If I bind to DataSet.Tables["name"] this event will fire, but if i bind to DataSet this event won´t fire anymore ?!

Thanks!

Reply

- *Bea* says:
  March 24, 2006 at 7:34 pm

  Hi,

  About your BindingContext question: what are you trying to do? Were you using the CurrencyManager from WinForms, and want to track currency the same way in Avalon? This has been replaced with BindingListCollectionView as you mentioned. This view tracks currency (or position as you called it) for the ADO.NET scenario.

  You said the ItemChanged event doesn't fire when you bind something to a DataSet. What is your target Control? An ItemsControl can only be bound to a DataTable or DataView. How did you get your source to be a whole DataSet? If your source is a DataSet, you would have to specify the Path of the binding to be, for example, Path=Tables[0] to get the first table.

  If you explain your scenario better (and share some code that will help me re-implement it) I will give it a shot.

  Thanks,
  Bea

  Reply

7. *Anonymous* says:
March 26, 2006 at 10:01 am

Hi Bea, I have one question. How do i filter the datatable if im using a objectdataprovider to utilize asynchrounous loading?

Reply

- *Bea* says:
  March 27, 2006 at 11:11 am

  Hi,

  Regarding your question on filtering an aynchronous DataTable:

  There is a DataChanged event on ObjectDataProvider that fires when the data is ready. So I would add the code that does the custom filtering to the DataChanged event handler. I made a sample application with this, which you can find here.

  Let me know if this works for you.

  Bea

  Reply

8. *Setiono* says:
April 15, 2006 at 4:54 pm

Hi Beatriz,

I'm trying to add to your sample the ability to do a new Query and re-populate the DataSet with a new set of Data. So I added this method in the code behind which is called from a Button in the XAML file:

```
private void Query(object sender, RoutedEventArgs args)
{
```

```
ds.Clear();
ds.ReadXml("c:\\example.xml");
}
```

For testing purpose, the example.xml" is a dump of the original DataSet. So what I expect to happen after this Query is called, is for the ListView to show the original set of data in the UI.

After Query is called, the data seems to show up correctly, but when I start to click around in the ListView, the selection and parent-child relationship is no longer maintained. As a matter of fact, after a few clicks, I get the following exception:

An item with the same key has already been added.

It seems like the DataSet.Clear() does not send a proper notification to the bounded ListView that the Collection underneath has changed.

Is this a bug in the DataBinding engine? I'm still using May CTP.

Thanks,
Setiono

Reply

Bea says:
April 16, 2006 at 11:32 am

Hi,

Yes, I believe this is a bug that we've fixed recently. Can you please let me know when you have a chance to verify this with July CTP? I'm pretty sure you're describing the bug we fixed, but I would like to be 100% positive.

Thanks,
Bea

Reply

Matti says:
May 5, 2006 at 7:24 am

My problem is similar to Setiono's. I'm using RC1.

I have a TextBox bound to a dataset's Table/Field and it shows fine. If I clear the dataset and re-populate it, the UI will be cleared but the new data is not shown.

I tried CollectionViewSource.GetDefaultView(this.DataContext).MoveCurrentToFirst(); after clearing and re-populating but it didn't help. (this fixed a similar problem with ObservableCollection).

Regards, Matti

Reply

Bea says:
May 6, 2006 at 9:35 am

Hi Matti,

First let me explain why moving currency to first worked for you with ObservableCollection. You were probably binding the TextBox to the current item of the collection. When the collection is cleared, currency moves to before first. Then, when you repopulate the collection currency doesn't change, it remains in before first, so your TextBox does not display anything. Once you move currency to first, your TextBox displays the first item as itended.

I tried to repro your bug with a DataTable but I was not able to. Moving currency to the first item after clearing the table displays the item correctly. You can find my repro here.

Let me know what you're doing different. It may still be a bug that comes up when this scenario is coded in a different way.

Thanks,
Bea

Reply

Matti says:
May 7, 2006 at 11:46 pm

Hi Beatriz and thanks for the answer.

The problem arises when DataContext is set to a DataSet.

I modified your sample this way:
XAML: {Binding Path=Places/Name}

Then added a class:

```
public class MyDataSet : DataSet {
public MyDataSet() {
this.Tables.Add(new PlacesDataTable());
}
}
```

In the Window1 ctor set the DataContex to new MyDataSet();

and in ClearRepopulate method
changed the type of the src obj to MyDataSet.

Seems that I'm not handling the currency correctly with DataSet. I would like to set the current item to be the DataSet.Tables[0].Rows[0] as it is by default when the DataContext is set for the first time.

Regards, Matti

Reply

Bea says:
May 8, 2006 at 5:52 pm

Hi Matti,

I was able to repro what you describe easily. We looked into it and unfortunately you're hitting an ADO.NET bug. Hopefully it will be fixed soon…

Sorry about any inconvenience this may cause :(and thanks for letting us know!! We really appreciate feedback!

Bea

Reply

9. *Wayne* says:
June 11, 2006 at 11:44 am

Hi Bea,

Your one-to-many relationship description helped with my understanding of the way binding works with ADO.NET. Thank you.

However, it only gets me half way there. I am working with a many-to-many relationship through a join table. I can get from the one parent to the child using the method outlined in the blog, however, I can't figure out how to get from the child table to the second parent. Do you have any tips?

Thanks,
Wayne.

Reply

Bea says:
May 12, 2006 at 6:40 pm

Hi Wayne,

We don't have any support for Many to Many relationships and we have no plans to add any. I made an attempt to implement master-detail for a Many to Many relationship, but I am unclear of the behavior you expect. You can find my attempt here.

If you explain the exact behavior you would expect us to support, I can add it to the list of scenarios to consider for the next version.

Thanks,
Bea

Reply

10. *Anonymous* says:
June 14, 2006 at 6:13 pm

tell me how you know the internals of binding? How did you peak in? What tools you used? I wasted many hours trying to bing DataRow.

Loved your blog, thanks please keep it coming.

Reply

- *Bea* says:
September 17, 2007 at 7:34 pm

Hi,

Well, I know the internals of binding because I've worked for the WPF Data Binding team for several years. I helped create this API. If you're interested in learning about the internals of binding (or any managed technology) I recommend Dot Net Reflector from Lutz Roeder. In his own words, here's what this tool does:

"Reflector is the class browser, explorer, analyzer and documentation viewer for .NET. Reflector allows to easily view, navigate, search, decompile and analyze .NET assemblies in C#, Visual Basic and IL."

I find myself using this tool every time I need to write code.

And of course, reading blogs and asking questions is a great way to learn any technology.

I hope you're making progress with WPF!

Bea

Reply

11. *sam* says:
August 20, 2006 at 3:59 am

Bea, you wrote:

"Rows in a DataView are of type DataRowView, which implements ICustomTypeDescriptor. Data binding honors this interface and will call the GetProperties() method on DataView, which will give us a few more properties."

Yes, it is called, but it is not enough for a class to be an ICustomTypeDescriptor. There must be something more to a class so databinding will see the content.

How do I know? I wrote a simple wrapper around DataTable/DataView. My wrapper around the DataView contains this source:

```
public object this[int index]
{
get
{
ICustomTypeDescriptor rc = (ICustomTypeDescriptor)m_wrappedDataView[index];
return new MyDataRowViewWrapper(rc);
}
}
```

MyDataRowViewWrapper implements ICustomTypeDescriptor (and IDataErrorInfo for good measure), so it should work as an replacement for the DataRowView. But it does not work.
If I replace "return new MyDataRowViewWrapper(rc);" with just "return rc;" everything works fine, but when I use the wrapper everything falls apart.

So the Binding does use something more than just ICustomTypeDescriptor here, but I can't find what it is?

Why do I need this? I'm trying to bind to a runtime datastructure like dataset, only mine does load the content on demand. And I found no other way to bind to a runtime data structure.

Reply

- *sam* says:
  August 20, 2006 at 6:25 am

  Uhm, stupid me, I didn't see the binding error message in the debugger output window:

  System.Windows.Data Error: 12 : Cannot get 'Nr' value (type 'String') from " (type 'DataRowViewWrapper'). BindingExpression:Path=Nr; DataItem='DataRowViewWrapper' (HashCode=65429669); target element is 'TextBlock' (Name="); target property is 'Text' (type 'String') InvalidCastException:'System.InvalidCastException: Das Objekt des Typs Bsoft.Berta.DataRowViewWrapper kann nicht in Typ System.Data.DataRowView umgewandelt werden.

  I made the jump, scrapped my wrapper and wrote an IListSource, IList, ICustomTypeDescriptor and PropertyDescriptor from scratch, and it works.

  Source is ugly as hell right now, need to clean that stuff up until I understand why it works ;)

  Reply

  - *Bea* says:
    August 26, 2006 at 11:32 pm

    Sam,
    Glad you got it to work :)
    Bea

    Reply

12. *Anonymous* says:
August 25, 2006 at 2:06 pm

Hi,

Thank you for providing very good posts on data binding. It has helped me a lot in getting started.
To understand this topic further I'm using Northwind Database and trying to populate a ListView with Employee data. Now I have a Dataset with table "EmployeeData" which has EmployeeID, FirstName and LastName. I'm binding the ListView with Dataset's table EmployeeData (I'm also using data templates for displaying the data) and the page works fine. My question is If I add a record to the Northwind.Employee table, how can that get updated in the UI. I tried using the INotifyPropertyChanged interface but an not sure on what property should I implement onPropertyChanged event. I tried on the Dataset and that did not work. Every sample that I see on the internet does databinding on simple datatypes. Is it possible to do two way binding on objects.

Reply

- *Bea* says:
  August 26, 2006 at 10:53 am

  Hi,

  Regarding your question on change notifications and ADO. You should not need to do anything special to notify WPF of changes in the DataTable. It's strange that you're seeing something different in your application.

  I made a simple sample that shows how adding an item to a DataTable gets reflected automatically in the UI. You can find it here.

  Hopefully this will help you find the problem in your app.

  Thanks!
  Bea

  Reply

13. *Mike* says:

September 3, 2006 at 2:10 pm

Hi Bea,

Thanks for your tips. I find data binding extremely hard to understand, so seeing your examples if very nice. I feel kind of silly asking this question because it seems like it should be very simple, but I can't seem to get it right.

If I want to build a tree hierarchy from the Lifts table where Mountain_ID designates the parent node and Lift_Name is the child, then how would I do that?

The output would look like this:

1
Big Red Express
Garbanzo Express
Orange chair
2
Tye Mill
Jupiter chair
Southern cross chair
3
Rainier Express
Green Valley

Reply

- *Mike* says:
  September 3, 2006 at 3:41 pm

  Sorry, that output didn't format as I expected. Basically, the numbers are the tree item parent headers (Mountain_ID) and the lift names are the child items.

  Thanks for the help,
  Mike

  Reply

  - *Bea* says:
    September 4, 2006 at 11:09 am

    Hi Mike,

    I'm assuming you would like to display that hierarchy in a TreeView, right? I'm also assuming the data on the first level of the TreeView is stored in a table in a database, and the second level is stored on some other table.

    If that's the case, I would build a middle data layer that reads the information from the database and creates a hierarchical data structure in the format you want. Then you can bind your TreeView directly to that data structure.

    The ideal data structure for this scenario would be a collection of MyMountain objects. MyMountain would contain an int MountainID property and a ObservableCollection Lifts property. MyLift would simply contain a string property with the lift name. Once you have this structure, you can use HierarchicalDataTemplates to help the TreeView display the data.

    Let me know if this helps.
    Bea

    Reply

14. *Reiner* says:
May 23, 2007 at 6:06 am

Hi:

I am new in XAML and WPF. I am trying to use a SQL-SERVER connection in a XBAP I can do it inside a WPF Windows application, but when I try to do the same like Load_Page() ASP.NET I get an error

void mainPage_Loaded(object sender, RoutedEventArgs e)

{

// Connection Fail

}

I have download many good books like:

Applications = Code + Markup: A Guide to the Microsoft® Windows® Presentation Foundation

By Charles Petzold

Programming WCF Services

By Juval Lowy

But no body speak about access data from SQL-SERVER database

may be you can help me.

Excuse my English I speak Spanish .

Thank you

Reiner

reinerra@hotmail.com

Reply

- *Bea* says:
  September 19, 2007 at 4:53 pm

  Hi Reiner,

  My guess is that you should not be able to connect directly to a SQL server from a partial trust application, for security reasons. I am not an expert in partial trust apps or WCF, so it’s hard for me to guide you. Maybe someone reading this post will have better insight.

  Bea

  Reply

Leave a Reply

Name (required)

Mail (will not be published) (required)

Website

Submit Comment