### Chapter 4 Domain Model -The Business Object Collection

Business Objects are frequently loaded and used in collections. The common use is to load a set of business objects based on some search criteria or to load all business objects. The business object collection is then be used for calculations, to show in a grid, list or report.

#### Loading a business object collection

We will start with the simple loading a collection of Business Objects.

Business Object Collections are loaded in a very similar manner to Individual Business Object i.e. you can use either a string criteria or a criteria object.

The tests for loading Business Object Collections are contained in the Test\_LoadCustomerUsingStringCriteria Class. We are not going to repeat all the code in the book but essentially the code for loading a business object collection is:

const string loadCriteria = "CustomerCode Like Code%";

IBusinessObjectCollection loadedCustomers = GetBusinessObjectLoader().GetBusinessObjectCollection<Customer>(loadCriteria);

The Criteria Objects and Criteria Strings are much more powerful than shown so far and include all the normal operators required i.e. AND, OR, opening ‘(‘, closing brackets ‘)’, <>, =, <>, >=, <=, >, <, IS NOT NULL, IS NULL, NOT LIKE, LIKE. All these operators work for loading collections of objects as well as for loading individual objects.

*Note however that if you try loading an individual object with a Criteria and more than one object is found in the database an error will be raised.*

This test provides an example of how complex loadCriteria can be built up using strings. You can also build complex Criteria using Criteria objects.

public void Test\_LoadCustomerUsingCriteriaString\_Complex()

{

///This test shows that if a persisted object is loaded from the

/// dataStore using the BusinessObjectLoader.GetBusinessObject.

/// Then an object with the exact same status and data as

/// the persisted object is loaded.

//---------------Set up test pack-------------------

Customer customer = CreateSavedCustomer();

Customer customer2 = CreateSavedCustomer();

Customer customerNoMatch = CreateSavedCustomer("CustName", "NotMatch");

Customer customerNoMatch2 = CreateSavedCustomer("CustName", "Code");

Customer customerMatchOnCustomerName = CreateSavedCustomer("AlternateName", "Code");

//---------------Assert Precondition----------------

Assert.IsFalse(customer.Status.IsNew);

//---------------Execute Test ----------------------

const string loadCriteria = "(CustomerCode Like Code% AND CustomerName Like 'Valid Name%') OR CustomerName Like 'AlternateName%' ";

IBusinessObjectCollection loadedCustomers = GetBusinessObjectLoader().GetBusinessObjectCollection<Customer>(loadCriteria);

//---------------Test Result -----------------------

Assert.AreEqual(3, loadedCustomers.Count);

Assert.Contains(customer, loadedCustomers);

Assert.Contains(customer2, loadedCustomers);

Assert.Contains(customerMatchOnCustomerName, loadedCustomers);

Assert.IsFalse(loadedCustomers.Contains(customerNoMatch));

Assert.IsFalse(loadedCustomers.Contains(customerNoMatch2));

}

The business Object collection also provides helper methods that wrap the Business Object loader and provide the same functionality for loading a collection of business objects. These methods are customers.LoadAll(), various overloads of customers.Load() and various overloads of customers.LoadWithLimits() See Test\_LoadCustomer\_UsingCollectionLoad.

Loading also provides a number of other capabilities including loading only a selected number of records (customers.LoadWithLimits()) and loading in a specific order (i.e. OrderBy Critieria). Once again the orderBy is always property names of the business objects and never database field names. See Test\_LoadCustomer\_UsingCollectionLoad\_LoadWithLimit.

For a full set of tests showing all the functionality for Loading Business Object Collections see TestBusinessObjectLoader\_GetBusinessObjectCollection in the Habanero.Test.BO project.

Future must be able to build custom criteria for any object that just need to put values e.g Criteria object called search by Surname, firstname and date of birth. All the UI developer needs to do then is use these criteria objects with the load. This will significantly improve the UIDevelopers experience especially with complex Criteria. see marks stuff

Load record 2-10 instead of only loadwithlimit

In summary you can load a collection of Business Objects in a number of ways always referring only to the business object properties thus totally isolating the application developer from the database implementation.

#### Making edits to a Business Object Collection

In addition to loading a collection the developer can manipulate the collection in a number of ways. The principles of how Habanero manages editing a Business Object Collection are similar to editing a Business Object. To achieve this functionality the Business Object collection maintains 4 collections namely

* The CreatedBusinessObjects, a list of business objects created by the collection.
* The RemovedBusinessObjects, a list of business objects removed from the collection.
* The PersistedBusinessObjects, a list of business objects representing the list of business objects as per the last time the collection loaded from the database or persisted to the database.
* The MarkedForDeleteBusinessObjects, a list of all business objects marked as deleted by the collection.
* The primary collection, a collection showing the PersistedBusinessObjects plus the CreatedBusinessObjects Less the RemovedBusinessObjects and MarkedForDeleteBusinessObjects.

##### Creating business object

The business object collection can create a business object (**customers.CreateBusinessObject()**). The business object of the appropriate type will be created and added to the business objects internal collection of created business objects (**customers.CreatedBusinessObjects**). The Business Object will also be added to the current collection and will thus be available for viewing/editing in grids, lists etc.

If the collection is saved then these created business objects will be saved (customers.SaveAll() and **customers.SaveAllInTransaction(transaction))** to the database.

(create tests and reference)

If the created Business Object is saved independently it will be removed from the created list and added to the Persisted list.

If the collection is restored (all edits cancelled. **Customers.RestoreAll**) then the collection will be reverted to its origional state i.e. the created Business Objects will be cleared and the current collection will be restored from the PersistedBusinessObjects.

(create tests and reference)

##### Add a business object

The developer can add an existing business object to a business object collection.

If the business object is new (customer.Status.IsNew) then the business object will be added to the CreatedBusinessObjects list as well as the primary list. (see test??).

If the business object is persisted (customer.Status.IsNew) then the business object will be added to the Current collection only. see test??).

##### Remove a Business Object

The developer can also remove a business object from a business object collection (Remove() and RemoveAt()).

If the business object is new then it is removed from the current collection and from the CreatedBusinessObjects list. See test

If the business object is not new then it is removed from the current list and added to the RemovedBusinessObjects list See test

##### MarkForDelete a Business Object

The developer can also mark a Business Objects for deletion from a business object collection.

If the business object is new then it is removed from the current collection and from the CreatedBusinessObjects list and MarkedForDeletion. See test

If the business object is not new then it is removed from the current list, marked for deletion and added to the DeletedBusinessObjects list See test

#### Persisting business object collections

A business object collection can be persisted via the .SaveAll method. All the added, removed, created and deleted business objects will be persisted and their collections cleared. See test

#### Refreshing a business object collection

A business object collection can be refreshed from the database. When refreshing a collection the collection will be reloaded.

Any dirty business objects will not be refreshed. Any Business Objects in the Removed or deleted list will not be shown in the Current list. Any items in the Created List and Added will be shown in the current list. See test

#### Restore (Cancel Edits) a business object collection

As with a business object a business object collection can be restored. The business object collection will be restored to its persisted stated. The created, deleted and removed lists will be cleared. (i.e. the items in the persisted list and the current collection will be identical). See test

#### Clear a business object collection

This clears the current collection, persisted, removed, deleted and created list. See test

#### Other business object collection functions

The business object has other methods Sorting, Intersection and Union, Insert.

(TODO: Insert must work the same as add in terms of inserting a new object)