re-store: Hands on Lab

Level 100 – PhoneBook Sample

rubicon
informationstechnologie gmbh

Table of Contents

1 Goal

Goal of this HOL is to teach the fundamentals of re-store, the data storage layer of re-motion. All exercises in this HOL can be done step by step. Each exercise ends with an executable result.

In the first lab you will create a domain layer for a phone book application. You will start with a class without relations to other classes. You will learn how to automatically create your database script. You will not have to write a single sql script by yourself in the whole HOL. In a second step you learn how to add relation between classes.

In the second and third lab you will get a short introduction to re-linq and client transactions.

During this HOL you will encounter sections with questions. Some answers might be easy. Other questions could be tricky. In case you are doing this exercise with other developers, we encourage you to talk about your ideas and your proposed solutions.

Please have also a look on the further readings section in the end of this HOL. It provides suggestions for further readings.

After this hands on lab you will be able to understand the advantages of re-store. You will be able to compare it with other O/R mapping tools and to implement re-store in a project.

Topics:

- Understand the concept of re-store
- Generate a simple domain layer with re-store
- Generate relations between domain classes
- Generate database scripts with dbschema.exe
- Introduction to client transactions
- Introduction to re-linq

Requirements:

Visual Studio 2010 must be installed on your working PC. Additionally, access to a SQL Server 2008 is required. You need at least db_owner rights for the “PhoneBook database”.

Get the re-motion version from https://www.re-motion.org/builds/. This HOL was verified to be working with version 1.13.103.

In this HOL C:\PhoneBook is the base directory for the solution. If you want to use a different path, keep in mind to adapt the directory names.

2 Lab 1: First steps with re-store

Estimated time: **30 Minutes**

2.1 Exercise 1: Preparation

Read in the Internet about O/R Mapping and try to answer the questions below! If you have experience with O/R Mapping tools, you can skip this preparation exercise!

2.1.1 Questions and Discussions

- What is the purpose of an O/R Mapping tool?
- Which products are available?
- Which are the most common O/R Mapping tools for .NET?

2.2 Exercise 2: Ten Minutes SQL Script Generation

2.2.1 Task 1: Create a Domain Object

1. **Start Microsoft Visual Studio 2010.**
2. **Create a new C# Class Library project. Select platform .NET Framework 3.5. Call it 'PhoneBook.Domain'. The solution name is PhoneBook. Select Location C:\ and enable Create directory for solution.**

   QuickCheck: Solution file should be under c:\PhoneBook, the project file under C:\PhoneBook\PhoneBook.Domain.
3. **Press CTRL + SHIFT + A to add a new class named Person to the domain project.**
4. **Replace the generated code for the class with the following code!**

```
using Remotion.Data.DomainObjects;
using Remotion.Data.DomainObjects.ObjectBinding;

namespace PhoneBook.Domain
{
  [DBTable]
  public class Person : BindableDomainObject
  {
    [StringProperty(MaximumLength = 60)]
    public virtual string FirstName { get; set; }

    [StringProperty(IsNullable = false, MaximumLength = 60)]
    public virtual string LastName { get; set; }

    public static Person NewObject()
    {
      return DomainObject.NewObject<Person>();
    }

    public static Person GetObject(ObjectID objid)
    {
      return DomainObject.GetObject<Person>(objid);
    }

    public override string DisplayName
    {
      get { return LastName; }
    }
  }
}
```

You will realize that some attributes and classes are marked red. Some references are missing.

5. **Download the a re-motion build** https://www.re-motion.org/builds

We recommend using the build that the HOL was tested with (See Requirements). You can of course always try the latest build.

6. **Extract the zip file to c:\PhoneBook\remotion**
7. **Add the the following references from c:\PhoneBook\remotion\net-3.5\bin\debug\**

- Remotion
- Remotion.Interfaces
- Remotion.Data.Domain.Objects,
- Remotion.Data.Interfaces.
- Remotion.ObjectBinding,
- Remotion.ObjectBinding.Interfaces

8. **Build**

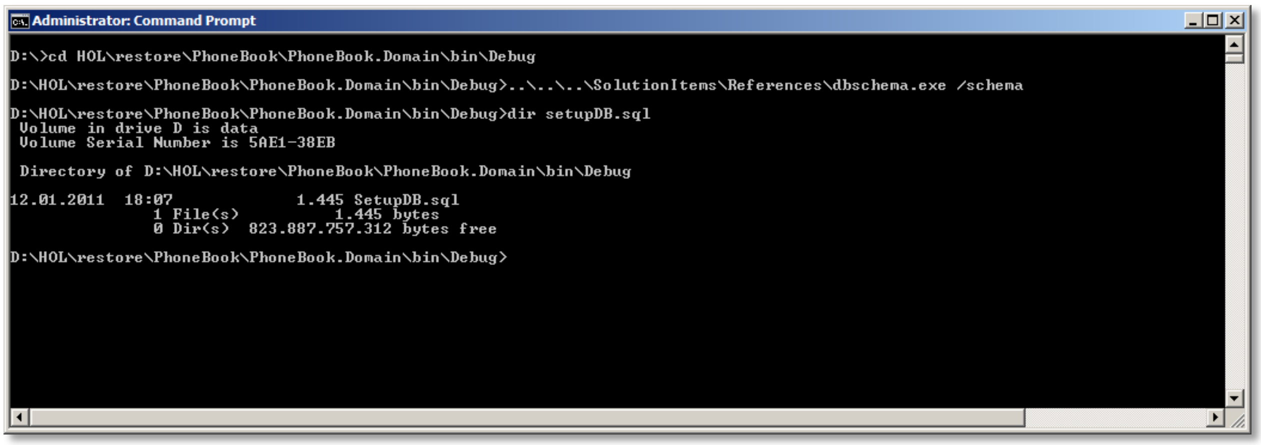
You have created you first domain. In the next steps you will create your database script based on your domain class.

2.2.2 Questions and Discussions

- Open C:\PhoneBook\remotion\net-3.5\doc\remotion.chm. Look up the term *BindableDomainObject*. What is the purpose of this class?

2.2.3 Task 2: Using dbschema.exe

1. **Open the command prompt.**
2. **Enter "`cd C:\PhoneBook\PhoneBook.Domain\bin\Debug`"**
3. **Call "`..\..\..\Remotion\net-3.5\bin\debug\dbschema.exe /schema`" (if you do not use the proposed directory schema, please adapt the path)**
4. **Look for the file `SetupDB.sql`**



```
D:\>cd HOL\restore\PhoneBook\PhoneBook.Domain\bin\Debug

D:\HOL\restore\PhoneBook\PhoneBook.Domain\bin\Debug>..\..\..\SolutionItems\References\dbschema.exe /schema

D:\HOL\restore\PhoneBook\PhoneBook.Domain\bin\Debug>dir setupDB.sql
 Volume in drive D is data
 Volume Serial Number is 5AE1-38EB

 Directory of D:\HOL\restore\PhoneBook\PhoneBook.Domain\bin\Debug

12.01.2011  18:07             1.445 SetupDB.sql
               1 File(s)          1.445 bytes
               0 Dir(s)  823.887.757.312 bytes free

D:\HOL\restore\PhoneBook\PhoneBook.Domain\bin\Debug>
```

Illustration 1: Using dbschema.exe

2.2.4 Questions and Discussions

- Remove one or two assemblies. Which error messages does the compiler return? Can you explain them? Add the assemblies again!
- DomainObject.NewObject<Person>() is a factory method to create new object instances. Can you explain the advantages of using factories to create objects via constructors within a few sentences? Does it make sense? If so, how would you convince a colleague to use factories instead of construction?
- Use Intellisense or the documentation to investigate the StringProperty attribute. Find out the meaning of MaximumLength and IsNullable?
- What happens if you delete the StringProperties attribute, rebuild and rerun dbschema.exe? Does it build? What has changed in the SetupDB.sql?

- Try to add an attribute Age or Birthday to the Person class.
- Explore the other parameters dbschema.exe provides? What does the verbose parameter do?
- The easiest way is to add the database and its tables with SQL Management studio. But is it also the fastest way? Do you want to open SQL Management Studio for every database change? How could you run this script on the console?
- Why do we call our domain project PhoneBook.Domain and not just Domain? Does this make sense?
- Automate dbschema: Changing to the command line and entering commands can get tedious. How can this be improved?
- **Expert question:** dbschema collects data from compiled assemblies not from source files. In theory, you could also read out c# files. Does this make sense? Discuss!
- **Expert question:** What would happen if you would use dbschema on assemblies compiled in release mode?

2.3 Exercise 3: Adding Relations

We will add relations between classes in this exercise. Relations in re-store are defined by virtual properties to reference types.

Attention: The order of properties in the classes matter! It is possible to generate a Visual Studio Web Application project based on the domain later on with UIGen.exe (more on this in Hans on Lab for re-form). The controls will be placed in the same order as the properties in the class in the generated ascx files.

2.3.1 Task 1: Adding classes without relation

First, we add Country, Location and PhoneNumber as new domain classes without relations.

1. **Add the following classes**

```
namespace PhoneBook.Domain
{
  public enum Country
  {
    Austria = 0,
    Australia,
    Germany,
    Switzerland,
    USA
  }
}
```

```
using Remotion.Data.DomainObjects;
using Remotion.Data.DomainObjects.ObjectBinding;

namespace PhoneBook.Domain
{
  [DBTable]
  public class Location : BindableDomainObject
  {
    [StringProperty(IsNullable = false, MaximumLength = 60)]
    public virtual string Street { get; set; }

    [StringProperty(IsNullable = true, MaximumLength = 12)]
    public virtual string Number { get; set; }

    [StringProperty(MaximumLength = 60)]
    public virtual string City { get; set; }

    public virtual int ZipCode { get; set; }

    public static Location NewObject()
    {
      return DomainObject.NewObject<Location>();
    }

    public static Location GetObject(ObjectID objid)
    {
      return DomainObject.GetObject<Location>(objid);
    }

    public override string DisplayName
    {
      get { return Street; }
    }
  }
}
```

```
using Remotion.Data.DomainObjects;
using Remotion.Data.DomainObjects.ObjectBinding;

namespace PhoneBook.Domain
{
  [DBTable]
  public class PhoneNumber : BindableDomainObject
  {
    [StringProperty(MaximumLength = 8)]
    public virtual string CountryCode { get; set; }

    [StringProperty(MaximumLength = 8)]
    public virtual string AreaCode { get; set; }

    [StringProperty(MaximumLength = 12, IsNullable = false)]
    public virtual string Number { get; set; }

    [StringProperty(MaximumLength = 8)]
    public virtual string Extension { get; set; }

    public static PhoneNumber NewObject()
    {
      return DomainObject.NewObject<PhoneNumber>();
    }

    public static PhoneNumber GetObject(ObjectID objid)
    {
      return DomainObject.GetObject<PhoneNumber>(objid);
    }

    public override string DisplayName
    {
      get { return Number; }
    }
  }
}
```

2. **Rebuild**
3. **Call dbschema.exe**

2.3.2 Questions and Discussions

- Identify the changes in the SetupDB.sql

2.3.3 Task 2: Adding unidirectional relations

There are two types of relations. We investigate unidirectional relations firsts. Unlike bidirectional relations, for unidirectional relations, there are no attributes for the properties required.

1. **Add property for Location to Person**

```
    public virtual Location Location { get; set; }
```

   Please place Location as the last property in the class.

2. **Rebuild**
3. **Call dbschema**

   What has changed in your generated script file?

4. **Add property Country to Location**

```
    public virtual Country? Country { get; set; }
```

   Please place Country below property City in the class.

5. **Rebuild**
6. **Call dbschema**

What has changed in your generated script file?

This is a typical foreign key relation in the database. Table Person has a foreign key to table Location.

At run-time a person object knows about his location object, but the location object has no reference to the person object. There is only one direction.

2.3.4 Task 3: Adding bidirectional relations

1. **Add the following code to Person**

```
[DBBidirectionalRelation("Person", SortExpression = "CountryCode,AreaCode,Number,Extension")]
public virtual ObjectList<PhoneNumber> PhoneNumbers { get; set; }
```

   Please place this property as the last property

2. **Build it - Does it build?**
3. **Apply dbschema.exe - Does dbschema.exe return an error?**
4. **Add the following code to PhoneNumbers.**

```
[DBBidirectionalRelation("PhoneNumbers")]
public virtual Person Person { get; set; }
```

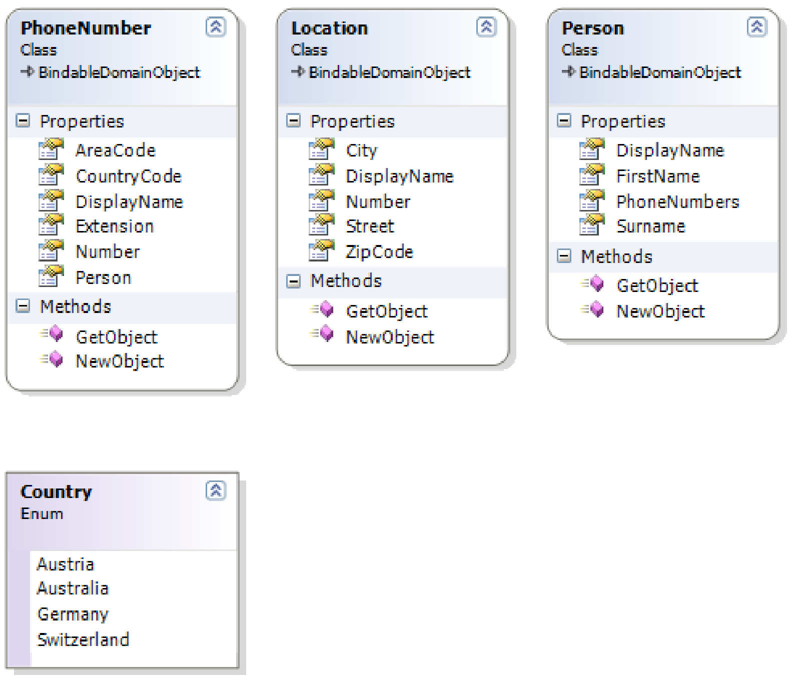
   Please place this property as the last property

5. **Built it. Does it build?**
6. **Does dbschema.exe return an error?**

Every person has zero or more phone numbers. But also each phone number knows to which person it belongs. In the database the table PhoneNumber has a foreign key relation to the table Person as a person can have more phone numbers. There can't be a circular reference between tables and we have a foreign key relation.

The main difference is the run-time. The attribute DBBidirectionalRelation defines that during runtime the objects of both classes must know each other. In other words: If a phone number gets deleted, also the property PhoneNumbers of Person must get updated. re-motion keeps both sides synchronized.

We have now added classes and relations. We have created a domain model that contains the following classes:



2.3.5 Questions and Discussions

- Does it make sense to structure the source code into regions (putting properties, relations or factory methods into #regions… #endregion)?
- Consider two attributes: Mandatory and StorageClassNone. Mandatory specifies that the property is mandatory and StorageClassNone that this property shall not be stored in the database. Apply them in the code, generate scripts and see what happens.
- SetupSQL: Why is LocationID nullable?
- In the last example we saw that there might be problems which can't be detected by the compiler, however they can be detected by generating classes with dbschema.exe. Think about strategies to prevent problems! Does it make sense to add a dbschema generation to your buildscript?
- In the sample we used a 1:N relation for a bidirectional relation. Is it possible to use a 1:1 relation bidirectional? It is possible to use a 1:N relation unidirectional?
- **Expert Question:** Which of the following is a "value object"? Person / Country? How would the other entity be called?
- **Expert Question:** If you have thought about "value type", please look up the term "value object" with "DDD" or "Domain Driven Design". Can you explain the difference between "value type" and "value object"?
- **Expert Question:** re-store identity columns rely on GUIDs. Does this make sense? Is an INT a better alternative?
- **Expert Question:** There are inheritance modes for databases: Single Table Inheritance / Class Table Inheritance / Concrete Table Inheritance. Find out what they do? Which model re-store is using?
- **Expert Question:** The DbTableAttribute and Interitance: Can we add DbTable more than once in an inheritance tree? What if we apply this attribute to leaves, what if we apply this attribute to base classes?
- **Expert Question:** Discuss *eager fetching* and *lazy loading*. Upcoming re-store versions will support both.
- **Expert Question:** What is your opinion about bidirectional relations? Do you think they can be tricky to be implemented in a framework? In which problems a framework developer might run into if he wants to add support for bidirectional relations?

2.4 Exercise 4: Adding parameterized constructors

In this sample we will add some constructors to make construction easier.

2.4.1 Task 1: Adding constructors

1. **Add the following snippet to class Location**

```
public static Location NewObject (string street, string number, string city,
  Country country, int zip)
{
  return DomainObject.NewObject<Location> (ParamList.Create(street, number, city,
   country, zip));
}

protected Location () { }

protected Location (string street, string number, string city, Country country, int zip)
{
  Street = street;
  Number = number;
  City = city;
  Country = country;
  ZipCode = zip;
}
```

2. **Add the following snippet to class Person**

```
protected Person () { }

protected Person (string firstName, string lastName, Location location)
{
  FirstName = firstName;
  LastName = lastName;
  Location = location;
}

public static Person NewObject (string firstName, string lastName, Location location)
{
  return DomainObject.NewObject<Person> (ParamList.Create(firstName, lastName, location));
}
```

3. **Add the following snippet to class Phonenumber**

```
    protected PhoneNumber () { }

    protected PhoneNumber(string cc, string ac, string nu, string ext, Person p)
    {
      CountryCode = cc;
      AreaCode = ac;
      Number = nu;
      Extension = ext;
      Person = p;
    }

    public static PhoneNumber NewObject (string cc, string ac, string nu, string ext, Person p)
    {
      return DomainObject.NewObject<PhoneNumber> (ParamList.Create(cc,ac,nu,ext,p));
    }
```

2.4.2 Questions and Discussions

- Why is a default constructor required?
- Why is a ParamList.Create required?

2.5 Exercise 5: Sample Console App

We will now add a console application to our solution to add data to our database.

1. **Add a console application to the solution**
2. **Add new item "Application Configuration File"**
3. **If not done yet create your SQL Database for this sample.**

   We recommend naming the database "Phonebook". Add tables and views with the generated SetupDB.sql script. Be aware that you have to adapt the use statement in the first line of the SetupDB.sql script to match the name of your database.
4. **Add assembly references to re-store in the console application**
5. **Add a reference to the project of your domain object.**
6. **Add the following code for the program**

```
using PhoneBook.Domain;
using Remotion.Data.DomainObjects;

namespace PhoneBook.Console
{
  class Program
  {
    static void AddSampleData()
    {
      using (ClientTransaction.CreateRootTransaction().EnterDiscardingScope())
      {
        var loc = Location.NewObject();
        loc.Street = "Microsoft Way";
        loc.Number = "4";
        loc.City = "Redmond";
        loc.ZipCode = 1110;

        var p1 = Person.NewObject();
        p1.FirstName = "Steve";
        p1.LastName = "Ballmer";
        p1.Location = loc;

        var pn1 = PhoneNumber.NewObject();
        pn1.CountryCode = "001";
        pn1.AreaCode = "425";
        pn1.Number = "705-1900";
        pn1.Person = p1;
        var p2 = Person.NewObject();
        p2.FirstName = "Scott";
        p2.LastName = "Guthrie";
        p2.Location = loc;

        var pn2 = PhoneNumber.NewObject();
        pn2.CountryCode = "001";
        pn2.AreaCode = "425";
        pn2.Number = "705-1901";
        pn2.Person = p2;

        var locApple = Location.NewObject("Apple Way", "42", "Cupertino", Country.USA, 000);
        var personApple = Person.NewObject("Steve", "Jobs", locApple);
        var phoneApple = PhoneNumber.NewObject("001", "93", "123-4567", "666",personApple);

        ClientTransaction.Current.Commit();
      }
    }

    static void Main(string[] args)
    {
      AddSampleData();
      System.Console.WriteLine (@"Hit any key to continue");
      System.Console.ReadKey();
    }
  }
}
```

7. **Set the following configuration in the application configuration file**

   Change the database name if necessary

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="remotion.data.domainObjects"
type="Remotion.Data.DomainObjects.Configuration.DomainObjectsConfiguration,
Remotion.Data.DomainObjects">
      <section name="storage"
type="Remotion.Data.DomainObjects.Persistence.Configuration.StorageConfiguration,
Remotion.Data.DomainObjects" />
    </sectionGroup>
  </configSections>

  <remotion.data.domainObjects xmlns="http://www.re-
motion.com/Data/DomainObjects/Configuration/2.0">
    <storage defaultProviderDefinition="PhoneBookDB">
      <providerDefinitions>
        <add name="PhoneBookDB"
               type="Remotion.Data.DomainObjects::Persistence.Rdbms.RdbmsProviderDefinition"
               providerType="Remotion.Data.DomainObjects::Persistence.Rdbms.SqlProvider"
               connectionString="PhoneBook"
factoryType="Remotion.Data.DomainObjects::Persistence.Rdbms.SqlServer.Sql2005.SqlStorageObjectFac
tory"
             />
      </providerDefinitions>
    </storage>

  </remotion.data.domainObjects>

  <connectionStrings>
    <add name="PhoneBook" connectionString="Integrated Security=SSPI; Initial Catalog=PhoneBook;
Data Source=localhost" />
  </connectionStrings>
</configuration>
```

8. **Run the sample and verify that the expected content is really stored in the database**

2.5.1 Questions and Discussions

- What is required to add Intellisense to the app.config file?
- **Expert question:** Try to remove the virtual keyword from one of the properties of Person and try to restart the app. There is an exception. Can you explain the reason for this?

3 Lab 2 – Client Transactions

Estimated time: **30 Minutes**

In this Lab we will dig deeper into *ClientTransactions*. We will learn how ClientTransactions are created and how objects can be enlisted into different transactions.

We will also have a look on *SubTransactions* as part of transaction handling.

3.1 Exercise 1: Preparation

Database transactions are an essential mechanism to ensure data consistency.

If you are new to this topic and have no experience with transactions at all, please try to get an overview before reading on. There are a large number of articles on transactions in the Internet.

Nevertheless, a ClientTransaction is more than just a database transaction. Look up the term “Unit of Work” Pattern. A good starting point might be: http://msdn.microsoft.com/en-us/magazine/dd882510.aspx

3.2 Exercise 2: Understanding ClientTransactions

3.2.1 Task 1: Factories

1. **Adapt the sample console application. Try to run the following code.**

```
static void Main(string[] args)
{
  var loc = Location.NewObject();
}
```

What do you expect will happen? Is an object of Location really instantiated?

3.2.2 Questions and Discussions

- To get rid of the exception at the factory method, you have to create a ClientTransaction. Can you explain the difference between the following operations?

```
ClientTransaction.CreateRootTransaction().EnterDiscardingScope();
Location.NewObject();
```

and

```
using (ClientTransaction.CreateRootTransaction().EnterDiscardingScope())
{
  Location.NewObject();
}
```

- We have learned that a domain object cannot live outside a ClientTransaction. Do you think this approach is a good idea? Try to find pros and cons and discuss them.

3.2.3 Task 2: Adding members

1. **Try to to implement**

```
ClientTransaction.CreateRootTransaction().EnterDiscardingScope();
Location.NewObject();
int count1 = ClientTransaction.Current.EnlistedDomainObjectCount;
ClientTransaction.CreateRootTransaction().EnterDiscardingScope();
int count2 = ClientTransaction.Current.EnlistedDomainObjectCount;
```

2. **Find out, if count1 has the same value as count2! Can you explain why they have different values?**

3.2.4 Questions and Discussions

- What needs to be done so that count1 matches count2? **Hint**: Check out static methods of `ClientTransaction.Current`. Is there a method that will help you to solve the problem?
- The return value of `Location.NewObject()` is not assigned to a member or a variable in this sample. Is there a way to access the generated object apart from assigning the return value of `Location.NewObject()` to a variable?

3.2.5 Task 3: Manipulation just on persisted

1. **Add the following code to your console application.**

```
    private static void Demo()
    {
      ClientTransaction.CreateRootTransaction().EnterDiscardingScope();
      var l = Location.NewObject();
      // ClientTransaction.Current.Commit();

      int i1 = ClientTransaction.Current.EnlistedDomainObjectCount;
      using (ClientTransaction.CreateRootTransaction().EnterDiscardingScope())
      {
        ClientTransaction.Current.EnlistDomainObject(l);
        int i2 = ClientTransaction.Current.EnlistedDomainObjectCount;
        l.Street = "test";
      }
    }
```

2. **Try to explain the purpose of this code! The solution for the problem is obviously “marked”. Can you explain why uncommenting the fifth line solves the problem?**

3.2.6 Task 4: Discarding Scope

We have now a look the ClientTransactionScope. Try out the following example:

```
      ClientTransaction.CreateRootTransaction().EnterDiscardingScope();
      var l = Location.NewObject();
      ClientTransaction.Current.Commit();

      using (ClientTransaction.Current.EnterNonDiscardingScope())
      {
        l.ZipCode = 11;
      }
      using (ClientTransaction.Current.EnterDiscardingScope())
      {
        l.ZipCode = 12;
      }

      var obj = ClientTransaction.Current.GetEnlistedDomainObjects();
      string res = l.ZipCode;
```

3.2.7 Questions and Discussions

- Try to explain: What do you think happens in the sample?
- There is also a third option to get into a scope. Discuss on the parameter list of ClientTransaction.Current.EnterScope().
- Why do we need a class ClientTransactionScope anyway? Would it not be better just to work within scopes by usings?
- We might add something like

```
      using (ClientTransaction.Current.EnterDiscardingScope())
      {
         l.ZipCode = 12;
         DoCommit()
      }


      public void DoCommit()
      {
         ClientTransaction.Current.Commit();
      }
```

Do you think it is a good strategy to call commits in public methods? What could happen, if there is a commit in a public method?

3.3 Exercise 2: SubTransactions

1. **The following three code pieces are different in small details. Which results would you expect? Try them out. What would you conclude?**

```
      using (ClientTransaction.CreateRootTransaction().EnterDiscardingScope())
      {
        var loc = Location.NewObject();
        loc.City = "Vienna";

        using (ClientTransaction.Current.CreateSubTransaction().EnterDiscardingScope())
        {
          ClientTransaction.Current.EnlistDomainObject(loc);
          loc.City = "Berlin";
        }
        ClientTransaction.Current.Commit();
      }
    }

      using (ClientTransaction.CreateRootTransaction().EnterDiscardingScope())
      {
        var loc = Location.NewObject();
        loc.City = " Vienna ";

        using (ClientTransaction.Current.CreateSubTransaction().EnterDiscardingScope())
        {
          ClientTransaction.Current.EnlistDomainObject(loc);
          loc.City = "Berlin";
          ClientTransaction.Current.Commit();
        }
        ClientTransaction.Current.Commit();
      }

      using (ClientTransaction.CreateRootTransaction().EnterDiscardingScope())
      {
        var loc = Location.NewObject();
        loc.City = "Vienna";

        using (ClientTransaction.Current.CreateSubTransaction().EnterDiscardingScope())
        {
          ClientTransaction.Current.EnlistDomainObject(loc);
          loc.City = "Berlin";
          ClientTransaction.Current.Commit();
        }
        ClientTransaction.Current.Commit();
      }
```

4 Lab 2: re-linq

4.1 Exercise 1: First re-linq queries

We want to add a report to the sample console app.

4.1.1 Task 1: Report

1. **Be sure to add Remotion.Data.Linq to your project references and add** `using System.Linq;`
2. **So we add the following code to the program file.**

```
static void Report()
{
  using (ClientTransaction.CreateRootTransaction().EnterDiscardingScope())
  {
    foreach (var l in Location.GetLocations())
    {
      System.Console.WriteLine("Location: {0}", l.DisplayName);
      foreach (var p in l.FindPersons())
      {
        System.Console.WriteLine(" ---> Person: {0} ", p.DisplayName);
        foreach (var pn in p.PhoneNumbers)
        {
          System.Console.WriteLine(" -----------------> Tel: {0}", pn.DisplayName);
        }
      }
    }
  }
  System.Console.ReadKey();
```

3. **And we add the following lines to Location.cs**:

```
public static Location[] GetLocations()
{
  var expr = from l in QueryFactory.CreateLinqQuery<Location>()
             select l;
  return expr.ToArray();
}

public Person[] FindPersons()
{
  var expr = from p in QueryFactory.CreateLinqQuery<Person>()
             where p.Location == this
             select p;
  return expr.ToArray();
}
```

4. **Test the query**

We have created our first re-linq query.

4.2 Exercise 2: Further studies

There is a whitepaper on re-linq on https://www.re-motion.org/download/re-linq.pdf. It explains what re-linq does and how it can help you to improve your queries.

5 Final Questions

- What are the advantages or disadvantages of working with a GUI designer, such as a GUI designer from the ADO.NET Entity Framework? Are you faster with a designer or with a code?
- Which alternative O/R Mapper do you know? Can you name some differences to re-store?
- Look up naked objects architecture in Wikipedia? Is re-motion based on a naked objects architecture?
- Re-store uses in many cases lazy loading. Can you explain the difference between lazy loading and eager loading?
- Do you think re-store would be compatible with Mono?
- Bidirectional relations:

  You might want to try the following

```
phoneNumber.Person = person;
person.PhoneNumbers.Add(pn);

person.PhoneNumbers.Add(pn);
phoneNumber.Person = person;
```

The first statement throws an exception: "Cannot add object 'PhoneNumber|a9b8188a-d290-45b0-926a-8bd821832bdf|System.Guid' already part of this collection."

The second does not throw an exception. Why?

- **Expert questions:** On http://madgeek.com/Articles/ORMapping/EN/mapping.htm you will find a selection guide for O/R Mapper. Which of these features are available in re-store?
- **Expert questions:** By default there is a MSSQL provider for re-store. What could be required to implement a provider for a different database?

6 HOL Summary

In this HOL you were able to

- Understand basic concepts of re-store
- Generate a domain layer
- Generate database scripts
- Understand basic concepts of client transactions
- Understand basic concepts of re-linq

Now you might want to

- Go to re-bind hol to generate a web application based on your domain
- Dig a little bit deeper into the topic .NET Enterprise Application. You probably

want to read books like

- http://www.amazon.com/dp/073562609X/ref=rdr_ext_sb_pi_sims_1#_
- http://www.amazon.com/Professional-Enterprise-NET-Wrox-Programmer/dp/0470447613/ref=sr_1_4?s=books&ie=UTF8&qid=1295011417&sr=1-4

- Read Whitepapers to re-linq and have a look on the blogs.