

Re-form: PhoneBook



Level 100 - Hands on Labs

Table of Contents

1	Goals	1
2	Lab 1: Using UIGEN.....	1
2.1	Exercise: Understanding UI Generators.....	1
2.1.1	Questions and Discussions	3
2.2	Exercise: Using UI Gen	4
2.2.1	Task 1: Preparing UIGEN	4
2.2.2	Task 2: Using UIGEN	5
2.2.3	Task 3: Testing UIGEN.....	5
2.2.4	Questions and Discussions	5
3	Lab 2: First steps with re-call	5
3.1	Exercise: Understanding re-call.....	5
3.1.1	Questions and Discussions	6
3.2	Exercise 1: Get Rid of Exceptions	6
3.2.1	Task 1: Get Rid of WxeExecuteNextStepException.....	7
3.2.2	Questions and Discussions	7
4	Lab 3: First steps with re-bind	8
4.1	Exercise 1: Globalization	8
4.1.1	Task 1: Fixing tab names	8
4.1.2	Task 2: Adding Globalization on Properties	8
4.1.3	Questions and Discussions	8
4.2	Exercise 3: Customizing Content	9
4.2.1	Task 1: Display names	9
4.2.2	Task 2: Customizing a BOC List.....	9
4.2.3	Questions and Discussions	10
5	Final Questions	11
6	Lab Summary.....	11

1 Goals

Estimated time: **30 Minutes**

Goal of this HOL is to teach the fundamentals of re-form, the presentation layer of re-motion. re-form contains re-bind, the data binding component, and re-call, the transactional navigation system.

All exercises in this HOL can be solved step by step. Each exercise is designed to be finishable with an executable result.

During this HOL you will run into sections for questions and discussions. Answers to some questions might be straight forward. 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. Be aware that not for every question, there is a perfect answer.

After this exercise you will be able to understand the advantages of re-form and to compare them with other technologies.

Please have also a look on the further readings section in the end of this HOL. They provide some suggestions on how you might want to proceed to learn more.

In this lab you will continue to work on the phonebook sample, which you started with the HOL for re-store.

Topics:

-
- ▶ Lab1: Using UIGEN to generate a web application project from the domain
 - ▶ Lab2: First steps with re-call, the transactional navigation between forms
 - ▶ Lab3: First steps with re-bind, the binding between control and data
-

Important:

For this HOL Visual Studio 2010 must be installed on your working PC.

Get the re-motion version from <https://www.re-motion.org/builds/>. This HOL was verified to be working with version 1.13.101.

This HOL continues where the re-store HOL has ended. If you have not done the re-store HOL yet, please finish this HOL or copy the files of a finished solution to C:\PhoneBook.

This HOL was tested with IE and Google Chrome.

2 Lab 1: Using UIGEN

2.1 Exercise: Understanding UI Generators

WYSIWYG editors for GUIs are well known to developers. One of the first things you may have learned was to add text box controls and labels to create your first customized “Hello World” application.

Many new applications are created with a “design first” principle in mind, which became popular with RAD development. .NET Developers create their Visual Studio GUI project, drag some controls from the Toolbox to the Designer and arrange them. Once the GUIs are finished, they add the corresponding logic by adding class libraries.

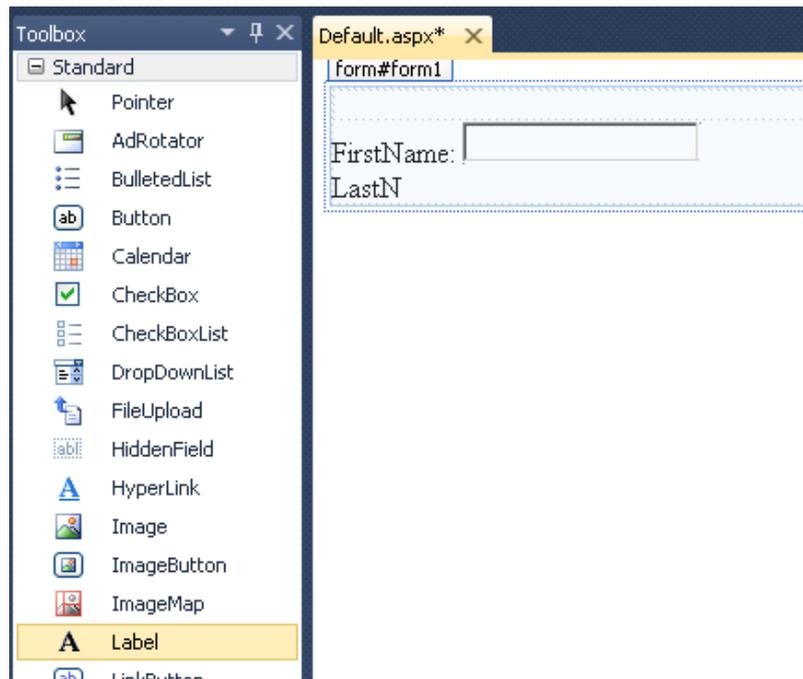


Illustration 1: Designing a Web GUI from the Scratch

Once you have a GUI you are able to define what the application is doing. The next step would be to take care about how the application is processing the user input.

Working with re-motion is different. It can be defined in four steps.

1. **Create your Domain**
2. **Generate your Database Layer**
3. **Generate your Presentation Layer**
4. **Customize and Adapt**

Step 1 and 2 shall already be done in the HOL for re-store. The next question is how a web application layer can be generated from classes.

In simplified words: Every data type has its “default control”. Strings can be entered or changed in text boxes, date in date time pickers, enums in radio buttons, etc. Therefore, a class person with the properties string FirstName, string LastName, DateTime BirthDay would be two TextBoxes and one DateTimePicker with a Save and Cancel Button in a standard data input form.

So why do we need to draw the CRUD pages by designer, if we could generate search and edit forms for each entity. With the preconditions, it is possible to generate a web application based on the domain objects from a template. The generated forms contain an edit and search form for every domain object. Each property is represented by a control corresponding to its data type.

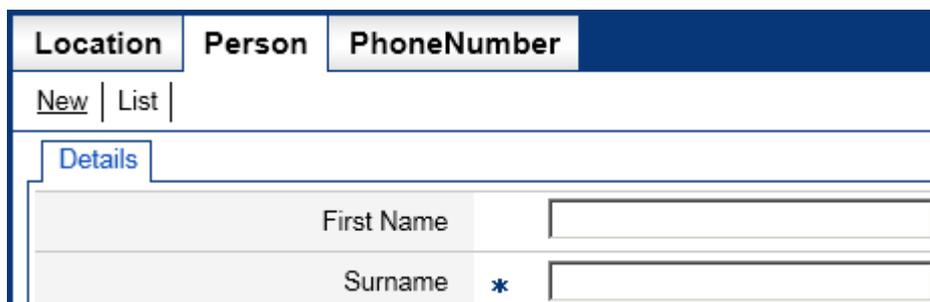


Abbildung 2: Generated GUI for Person

The generated template can then be edited and adapted to individual needs.

2.1.1 Questions and Discussions

- ▶ WYSIWYG editors can be troublesome. Many products have been known for the instability or the lack of performance of their designers. Do you know products or frameworks, where the designer is the weak spot? Why do you think the designer is the weak spot?
- ▶ Are there some additional common page controls that make sense to be generated for a form?
- ▶ Are there standard C# data types that cannot be mapped to standard controls?
- ▶ What are the advantages by generating web projects over designing them?
- ▶ Discuss the following statements about UIs. Some of them are definitely to be discussed or dangerous. Which of them do you consider as good which not?
 - ◆ UIs must be consistent. It is a bad idea to solve a problem in a specific situation in a way and in similar situation in a different way.
 - ◆ Every UI must support keyboard only input
 - ◆ UI must be responsive. Every action must trigger an immediate response.
 - ◆ If the UI is bad, the whole application is worthless, even if the business logic is perfect.
 - ◆ Users love a well-designed GUI that looks good. If your UI does not look good, your app lacks quality (even though there are no functional problems).
 - ◆ Modern UI controls with 3D effects are essential. The more new technologies, the better. They improve the general work performance and give the user a new feeling of working.
 - ◆ You can neglect old browsers to make your GUI outstanding.
 - ◆ It is hell, if you have to reenter data, because the data has been lost due to a session time out or something similar.
 - ◆ UIs shall be designed for the power user. The intelligent user shall be enabled to work fast and he shall also be supported by some advanced complex controls. The lame shall learn or quit using the app.
 - ◆ People are different and their perception and the way they work is different. So there is no good UI, but only UIs that meet the requirement of users.
 - ◆ The best way to prevent problems with the UI is to prevent that a user enters wrong data.
 - ◆ If the application does not work appropriate, it is the users fault! The best way of fixing this is to log every user action, pack the logs, mail them to the boss and instruct the boss how he can blame his employee for obvious wrong usage of the apps.
 - ◆ The more information, the less focus. Some people say that google became dominant as they kept the design simple and focused whereas the competition had stuffed their site with too much information.
 - ◆ Everyone loves to read. So it is essential to pack as much information as possible on a page. Even we have to repeat ourselves, the more information the better.
- ▶ Check out some additional thoughts about usability (see further links as samples). Discuss approaches such as:
 - ◆ <http://www.smashingmagazine.com/2008/01/31/10-principles-of-effective-web-design/>
 - ◆ http://www.sylvantech.com/~talin/projects/ui_design.html
 - ◆ <http://www.ambysoft.com/essays/userInterfaceDesign.html>
- ▶ What do you think makes a UI successful? Add your own statements about a good UI!
- ▶ What is required to design a GUI to meet Web Accessibility features? Why do you think this important?
- ▶ One of the techniques to master in UI development is data binding. If you are new to this topic, research: What is data binding? How can this be achieved? How does a two way data binding differ from a one way data binding? What is an observer pattern?

- ▶ **Expert Question:** In which use cases would you recommend Microsoft Sharepoint, in which re-motion?

2.2 Exercise: Using UI Gen

The logical steps to create the GUI of your DDD application are:

1. **Create your domain (which should already be done in our previous lab for re-store)**
2. **Prepare a XML template to generate the GUI**
3. **Use UIGen to generate your Visual Studio Web Project containing all Forms from the command line**
4. **Adapt your generated UI to your own needs.**

2.2.1 Task 1: Preparing UIGEN

UIGEN.EXE is a command line tool to generate a Visual Studio Web Project based on

- ▶ Compiled domain objects (from your domain project)
- ▶ A Xml based project template

UIGEN is part of the re-motion contribution project. UIGEN is shipped with a stable version of re-motion. If you are working with a development version, you might have to check out the latest version of the re-motion contribution repository and build the Remotion.ObjectBinding.Web.CodeGenerator project (<https://svn.re-motion.org/svn/Remotion-Contrib/>) to get the UIGEN.exe. The latest template is also included at that location.

To generate the UI for the phonebook application with UIGEN, you need to

1. **Verify that your domain project is compiled without any errors.**
2. **Add a PhoneBook.xml file to the directory C:\PhoneBook**
3. **Make sure that you have the latest UIGenTemplate available (if not, the latest Templates are available from the Remotion.Object.WebCodeGenerator project).**
4. **Use the following content for PhoneBook.xml and adapt the directories to match your projects and template**

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- ATTENTION! Only ABSOLUTE PATHS are supported by uigen.exe -->
<!-- DO NOT USE RELATIVE PATHS OR YOUR APPLICATION WILL BREAK -->

<applicationGenerator
  template="\PhoneBook\Remotion\UIGenTemplates\TabbedEditor\TabbedEditor.xml">
  <settings
    templateRoot="\PhoneBook\Remotion\UIGenTemplates\TabbedEditor"
    targetRoot="\PhoneBook"
    projectNamespaceRoot="PhoneBook.Web"
    domainNamespaceRoot="PhoneBook.Domain"
  />
  <projectReplacements>
    <replace from="$ReferencesDir$" to="References" />
    <!-- Domain -->
    <!-- Copy the DomainProjectGuid from your PhoneBook.Domain.csproj file -->
    <replace from="$RemotionResDirectory$" to="\PhoneBook\Remotion\res" />
    <replace from="$DomainProjectGuid$" to="{E715FCED-AD7C-4EB7-9E30-EDF67D904A48}" />
    <replace from="$DomainProjectName$" to="PhoneBook.Domain" />
    <replace from="$DomainProjectAssembly$" to="PhoneBook.Domain" />
    <replace from="$RemotionAssembly$" to="\PhoneBook\Remotion\net-3.5\bin\Debug" />
    <replace from="$WxeEngine$" to="\PhoneBook\Remotion\net-3.5\bin\Debug" />
    <replace from="$WebClientName$" to="PhoneBook.Web" />
    <replace from="$WebClientAssembly$" to="PhoneBook.Web" />
    <replace from="$STRONG_SUPPLEMENT$" to="Version=1.13.101.2, Culture=neutral,
      PublicKeyToken=fee00910d6e5f53b" />
    <!-- USER -->
    <replace from="$USER_APPNAME$" to="PhoneBook Web Sample" />
    <replace from="$USER_DEFAULT_ASPX_TOPIC$" to="This application was generated by UIGen.exe" />
  </projectReplacements>
</applicationGenerator>
```

Re-form

Level 100 - Hands on Labs

```
<replace from="$USER_DEFAULT_ASPX_STARTINFO$" to="To start the application click 'Start'" />
<replace from="$USER_DEFAULT_STARTPAGE$" to="EditLocation.wxe?WxeReturnToSelf=true" />

<replace from="$USER_STATUSBAR$" to="yes" />
<replace from="$USER_CLASSIC_APPSTYLE$" to="false" />

<replace from="$USER_STORAGEPROVIDER$" to="PhoneBookDB" />
<replace from="$USER_CATALOGNAME$" to="PhoneBook" />
</projectReplacements>

</applicationGenerator>
```

2.2.2 Task 2: Using UIGEN

1. Go to the command line
2. Call `c:\PhoneBook\Remotion\net-3.5\bin\debug\UIGen.exe`
`/uigen:C:\PhoneBook\PhoneBook.xml`
`/asmdir:C:\PhoneBook\PhoneBook.Domain\bin\Debug`



Illustration 3: UIGen on Console

2.2.3 Task 3: Testing UIGEN

The UI Gen should have generated PhoneBook.Web Directory in C:\PhoneBook\

1. Add the generated project to your solution in Visual Studio
2. Set the generated project as startup project
3. Build and run

2.2.4 Questions and Discussions

- ▶ Investigate the phonebook.xml file and try to find out what you can adapt!
- ▶ You can create a corporate design via templates to enable you and your fellow developers to create branded applications on a mouse click. On a scale from one to ten, how powerful is this feature?
- ▶ The default template is a tabbed editor: Forms are grouped by tabs. Of what alternative structures can you think of?
- ▶ What can you configure in the TabbedEditor.xml?
- ▶ **Expert Questions:** Could you also create a Silverlight project based on the domain with UIGen? What would be required to create a tool that creates a Silverlight project?

3 Lab 2: First steps with re-call

3.1 Exercise: Understanding re-call

In the first Lab, you have learned how to generate a web project from the domain layer.

This generated web project contains Edit- and SearchForms for all domain objects. But before, we cover the forms in detail; we have a look on the navigation between the pages.

Traditionally, navigation is done the following way in a web application:

- ▶ user clicks control
- ▶ control fires event
- ▶ event-handler causes `Server.Transfer` or `Response.Redirect` to some specific page

Re-form

Level 100 - Hands on Labs

Depending on the implementation details, the information in caller page is either lost or kept in a state.

The idea of re-call is: *Treat page calls like method calls!*

Considering this in detail, it means:

- ▶ user clicks control
- ▶ control fires event
- ▶ event-handler calls other page with or without parameters
- ▶ Closing the called page is can return a value
- ▶ Once the page is closed, the calling page must be restored.



Illustration 4: Start Screen

3.1.1 Questions and Discussions

- ▶ **Expert Question:** If you want to dig deeper it is essential to understand the concept of Web Handlers. Read about them and discuss what they do in groups.
- ▶ **Expert Question:** Talk about the idea: Calling pages like a method. What impact does it have?
- ▶ **Expert Question:** Discuss the topic session state management in web pages!
- ▶ **Expert Question:** Why is calling browsers back, forward or refresh buttons a problem in this concept?

3.2 Exercise 1: Get Rid of Exceptions

If you debug the application and add a new entry, you encounter the following error message when you try to click the save button:

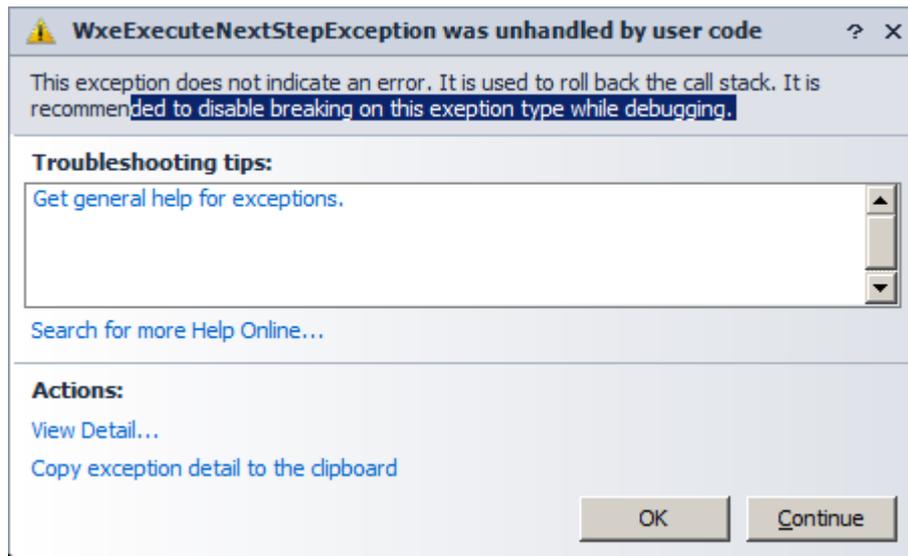


Illustration 5 WxeExecuteNextStepException

The reason for this is that due to the concept of ASP.NET this is the only way to navigate between pages in the code. Of course, we want to remove this message when we are debugging.

3.2.1 Task 1: Get Rid of WxeExecuteNextStepException

You can get rid of this by

1. CTRL + ALT + E (Editing Exception)
2. Add Remotion.Web.ExecutionEngine.WxeExecuteNextStepException to the list
3. Uncheck User-unhandled

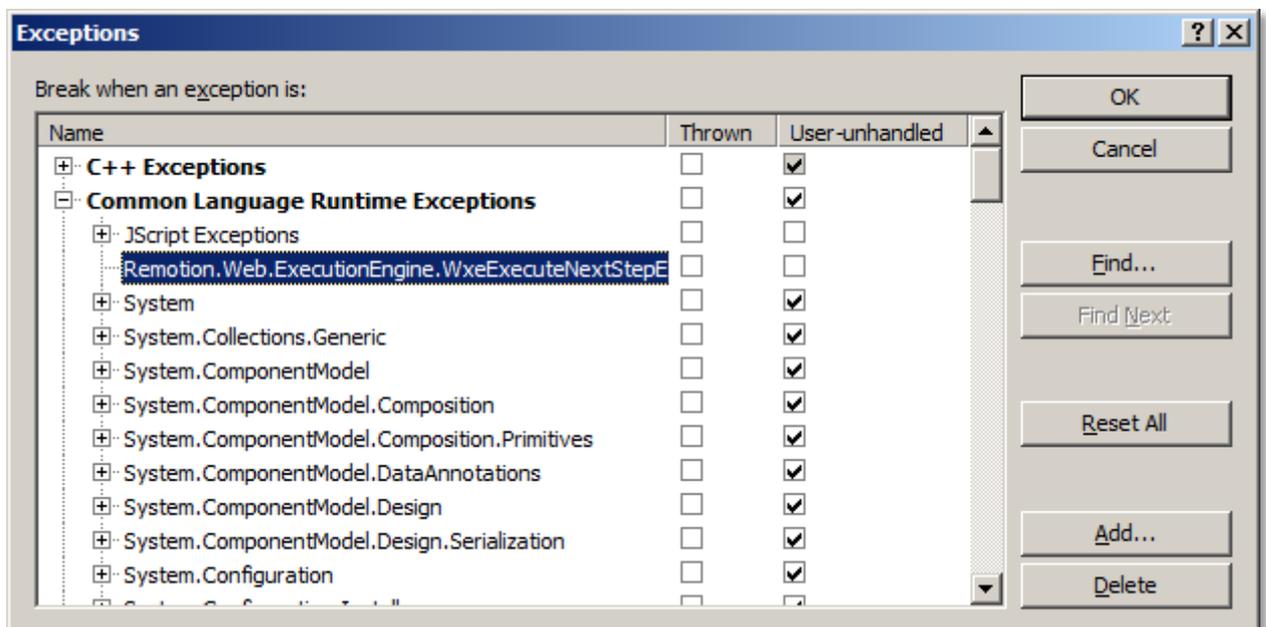


Illustration 6 Exception

3.2.2 Questions and Discussions

- ▶ **Expert Question:** This behavior to roll back a call stack is an “exception to the usage of exceptions” Why is this required? Try to explain it with your own words.

4 Lab 3: First steps with re-bind

re-motion comes with a rich set of controls. These controls are can bind the properties of the domain objects.

4.1 Exercise 1: Globalization

4.1.1 Task 1: Fixing tab names

The tab names contain identifiers with a ~ symbol. We can change this in the following steps

1. **Open global.resx file in PhoneBook.Web project**
2. **Change names**

As you can see, you can add more languages. You might want to try out to add a globalized version specific languages.

1. **Copy file global.resx to global.es.resx. This would add a Spanish version. Feel free to take any language you want**
2. **Translate identifiers in your ressource file**
3. **Start and change language in the browser settings.**

4.1.2 Task 2: Adding Globalization on Properties

After we have changed the tab names, we adapt identifiers in the form like PhoneNumbers or FirstName.

1. **Add class attributes to the class declaration of Person, PhoneNumber and Location**

```
[MultiLingualResources("PhoneBook.Domain.Globalization.PhoneNumber")]  
[MultiLingualResources("PhoneBook.Domain.Globalization.Person")]  
[MultiLingualResources("PhoneBook.Domain.Globalization.Location")]
```

The Location class should look like

```
[DBTable]  
[MultiLingualResources ("PhoneBook.Domain.Globalization.Location")]  
public class Location : BindableDomainObject
```

Attention: If your domain is not called PhoneBook.Domain, you will run into trouble with the namespace. You can adapt either the project name or the string identifier in the class attribute.

2. **Add the class attribute**

```
[EnumDescriptionResource("PhoneBook.Domain.Globalization.Country")]  
to enum Country
```

3. **Add a folder Globalization in the Domain Project**
4. **Add the files Location.resx, Person.resx, PhoneNumber.resx, Country.resx**
5. **Add properties to ressource file**
 - ▶ Location.resx: Name property:ZipCode, Value: Zip
 - ▶ Person.resx: Name property:FirstName, Value: First Name
 - ▶ Person.resx: Name property:SurName, Value: Last Name
6. **Run application again and check what has changed**

4.1.3 Questions and Discussions

- ▶ What are possible problems if you translate your page into another language?

4.2 Exercise 3: Customizing Content

4.2.1 Task 1: Display names

Looking on the form PhoneNumber, we get the last name of a person in the drop down list. This can be changed

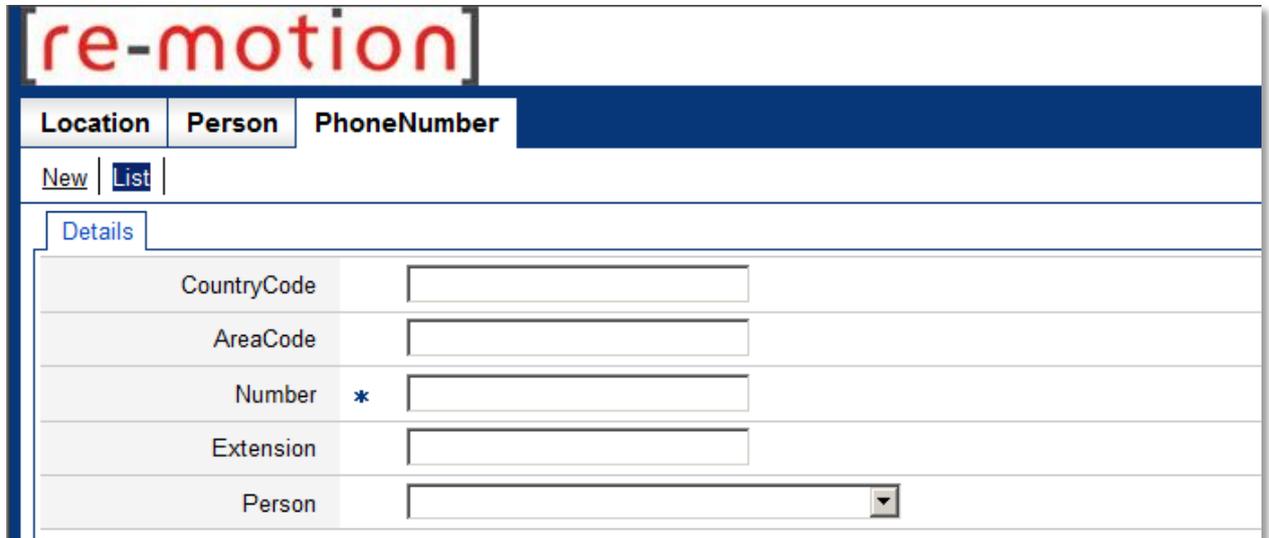


Illustration 7 start screen

1. Replace in class PhoneNumber

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

With

```
public override string DisplayName
{
    get { return LastName + ", " + FirstName; }
}
```

Of course, it is possible to change the DisplayNames of the other classes too.

4.2.2 Task 2: Customizing a BOC List

Looking at the search result of Location, we might get a list such as:

Street	Number	City	Zip code	DisplayName	
Microsoft Way	4	Redmond	1110	Microsoft Way	Edit
Hauptstrasse	10	Wien	1030	Hauptstrasse	Edit

Illustration 8: Boclist

First, we get rid of the DisplayName as this makes no sense in the list

1. Replace in SearchResultLocationControl

```
<remotion:BocAllPropertiesPlaceholderColumnDefinition />
```

with

```
<remotion:BocSimpleColumnDefinition PropertyPathIdentifier="Street" />
<remotion:BocSimpleColumnDefinition PropertyPathIdentifier="Number" />
<remotion:BocSimpleColumnDefinition PropertyPathIdentifier="City" />
<remotion:BocSimpleColumnDefinition PropertyPathIdentifier="ZipCode" />
```

2. We also might want to combine street with number.

```
<remotion:BocSimpleColumnDefinition PropertyPathIdentifier="Street" />
```

```
<remotion:BocSimpleColumnDefinition PropertyPathIdentifier="Number" />
```

with

```
<remotion:BocCompoundColumnDefinition ColumnTitle="Name" ItemID="CompoundEdit"
  FormatString="{0}, {1}">
  <propertypathbindings>
    <remotion:PropertyPathBinding PropertyPathIdentifier="Street" />
    <remotion:PropertyPathBinding PropertyPathIdentifier="Number" />
  </propertypathbindings>
  <persistedcommand>
    <remotion:BocListItemCommand Type="Event"/>
  </persistedcommand>
</remotion:BocCompoundColumnDefinition>
```

4.2.3 Questions and Discussions

- ▶ If you compare this strategy to XAML, what are the advantages and disadvantages?

4.2.4 Task 3: Adding Icons

1. Add a File `WebUIService.cs` to the Web project in Folder `Classes`
2. Paste the following code in `WebUIService.cs`

```
using System;
using System.Web;
using Remotion.Data.DomainObjects;
using Remotion.ObjectBinding;
using Remotion.ObjectBinding.Web;
using Remotion.ObjectBinding.Web.UI.Controls;
using Remotion.Web.UI.Controls;

namespace PhoneBook.Web.Classes
{
    public class WebUIService : IBusinessObjectWebUIService
    {
        public IconInfo GetIcon (IBusinessObject obj)
        {
            if (obj == null)
            {
                return new IconInfo ("~/Images/Icon-Null.gif", "16px", "16px");
            }
            else
            {
                // determine the static type of the
                // passed object, i.e. the originally declared domain object
                // class (Person, PhoneNumber or Location).
                Type staticType = ((DomainObject) obj).GetPublicDomainObjectType ();
                // assemble the path to the icon bitmap based on the class of the
                // object
                return new IconInfo ("~/Images/Icon-" + staticType.Name + ".gif", "16px", "16px");
            }
        }

        public string GetToolTip (IBusinessObject obj)
        {
            return null;
        }

        public HelpInfo GetHelpInfo (IBusinessObjectBoundWebControl businessObjectBoundWebControl,
            IBusinessObjectClass businessObjectClass,
            IBusinessObjectProperty businessObjectProperty,
            IBusinessObject businessObject)
        {
            return null;
        }
    }
}
```

3. Paste the following code in `WebUIService.cs`
4. Add to `global.asax.cs` in `Application_Start`

```
BindableObjectProvider.GetProvider<BindableDomainObjectProviderAttribute>
().AddService (
    typeof (IBusinessObjectWebUIService), new WebUIService ());
```

5. Verify that `icon-Location.gif`, etc. files exists in the `Images` folder

5 Final Questions

- ▶ Discuss: Many people claim RIA technologies are better than web based technologies. What is your opinion about this? Is it a good idea to replace ASP.NET based pages with new Silverlight pages?

6 Lab Summary

You have learned

- ▶ Lab1: Using UIGEN to generate a web application project from the domain
 - ▶ Lab2: First steps with re-call, the transactional navigation between forms
 - ▶ Lab3: First steps with re-bind, the layer for data binding
-

Please note: This is a Level 100 tutorial. Much has not been covered here, such as

- ▶ Re-call parameters
- ▶ Changing default values for Boc controls

No HOL is currently available to cover these advanced topics.

Now you might want to

- ▶ Learn about re-strict, the security component (no HOL available until now)
 - ▶ Learn more about UI principles. It is highly recommended that even as developer you have a basic understanding of UI principles. There are some very good books on this topic such as
 - ◆ http://www.amazon.com/Designing-Mind-Understanding-Interface-ebook/dp/B003H3IOXM/ref=sr_1_2?ie=UTF8&qid=1297333053&sr=8-2
-