

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

1 Goals

Estimated time: **30 Minutes**

Goal of this HOL is to teach the fundamentals of re-bind, the presentation layer of re-motion. It also introduces re-call, the transactional navigation system.

All exercises in this HOL can be solved step by step. Each exercise is designed to be solved 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-bind 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 presentation layer
-

Important:

For this HOL Visual Studio 2010 must be installed on your working PC. We recommend installing the following two third party tools too:

- ▶ JetBrains Resharper (<http://www.jetbrains.com/resharper/>)
- ▶ RedGate Reflector (<http://reflector.red-gate.com/Download.aspx>)

Both tools will help you to understand the samples in this HOL better.

Get the re-motion version from <https://www.re-motion.org/builds/>. This HOL was created with version 1.13.94.

This HOL continues where the re-store HOL has ended. If you have not done the re-store 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.

Every data type has its "default control". Strings can be retrieved from text boxes, date from date time pickers, enums from radio buttons, etc. At some points you have to decide, such as whether you want to display the content of a list in a drop down lists or a list box.

In a good DDD there are well separated entities as domain objects. Every domain object can be represented in a form for CRUD operations. It is obvious to provide at least forms to edit objects and to return all available domain objects in a list.

With the preconditions, it is possible to generate a web application from a template based on the domain objects. 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.

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?
- ▶ Can you create a mapping data type to controls?
- ▶ Discuss the following statements about UIs. Some of them are definitely to be discussed. Which of them do you consider as good which not.
 - ◆ UIs must be consistent. It is hell to solve a problem in one way and in another situation in another way.
 - ◆ UIs must support working without a mouse.
 - ◆ 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.
 - ◆ Modern UI controls with 3D effects are essential. The more new technologies, the better. They improve our performance and give us a new feeling of working.
 - ◆ You can neglect old browsers to make your GUI outstanding
 - ◆ It is hell to have to re enter data, because the data has been lost due to a session time out or something similar.
 - ◆ UIs shall be designed for the power user. Complex UIs designed that the intelligent user can work fast is good. 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.
 - ◆ Users must be prevented to make wrong inputs that can lead to mistakes.
 - ◆ 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 in help files. Even we have to repeat ourselves, the more info the better.
- ▶ Add your own statement!
- ▶ 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?

- ▶ What is, for example, required to design a gui barrier-free? Why do you think barrier-free sides are 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?

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 lab)
2. Prepare a XML template to generate the GUI
3. Generate your UI 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)
- ▶ Xml based Project template

UIGEN is part of the re-motion contribution project. Normally, this tool is shipped with a stable version of the re-motion. If you are working with an unstable version, you might have to check out the re-motion contribution repository and build the Remotion.ObjectBinding.Web.CodeGenerator project.

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. Use the following content for the PhoneBook.xml

```
<?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=1.13.90.2, 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" />
    <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" />
  </projectReplacements>
</applicationGenerator>
```

```
<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`

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
- ▶ As you can find out easily, there is a template for a solution. Therefore you can apply a corporate design. Does this make sense?
- ▶ There is a template tabbed editor. So that the forms are grouped by tabs. Of what can you alternative structures can you think of?
- ▶ What can you steer 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

re-call enables a developer to call re-bind forms similar to subroutines. It is possible to pass (typed) parameters and to return values to the caller.

re-call manages the implementation details to store the state of pages. re-call uses re-store in-memory transactions, removing any need to explicitly access the database.

That is what we do, when we start the application. If we look up the startup link, we get a link such as <http://localhost:22809/EditLocation.wxe?WxeReturnToSelf=true>. The port may vary.

To understand the lab, it is recommended that you understand how handlers work in ASP.NET. There are many tutorials on web handlers in the web. If you are new to this topic, get an overview on web handlers before you read on.

3.1.1 Questions and Discussions

- ▶ If a junior developer now asks you what a web handler is, how would you explain this topic?
- ▶ Try do call some forms in the browser. Adapt the URL from the exercise description
- ▶ **Expert Questions:** Discuss the topic session state management!

3.2 Exercise 1 Get Rid of Exceptions

On saving, you encounter this

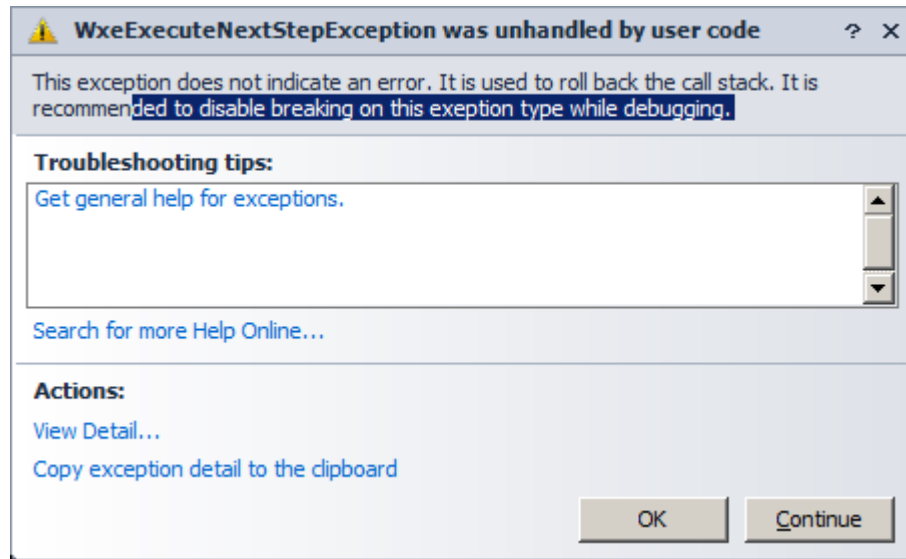


Illustration 1 WxeExecuteNextStepException

3.2.1 Task 1 Get Rid of Exceptions

You can get rid of this by

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

I

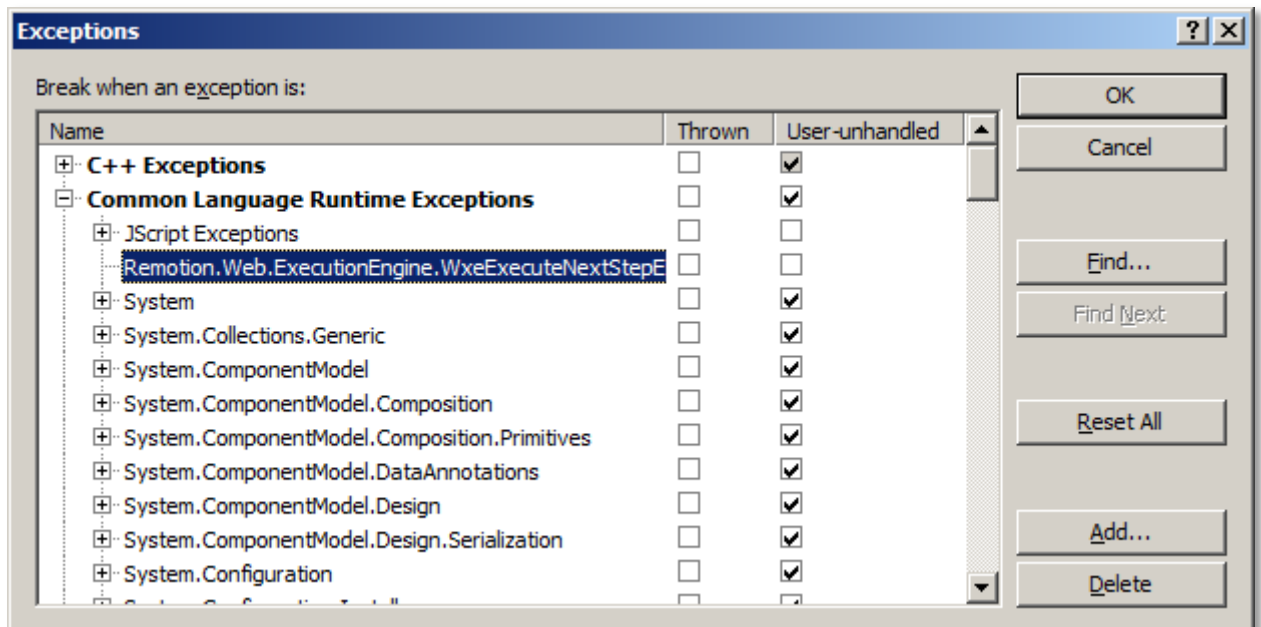


Illustration 2 Exception

3.2.2 Question

- Is this pattern an “exception to the use of exceptions”?

4 Lab 3: First steps with re-bind

4.1 Exercise 1 Globalization

4.1.1 Fixing tab names

We discover that the tab names contain some strange identifier with a ~ symbol. This can be fixed by the following procedure

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 a specific languages.

1. Copy file global.resx to global.es.resx. This would add a Spanish version. Feel free to take your own globalization
2. Translate in your globalization file
3. Start and change language.

4.1.2 Adding Globalization on Properties

1. Add class attributes to the class declaration, 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
```

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. Run application
6. Add properties to Location.resx (Name property:ZipCode,Value Zip)
7. Run application again and check what has changed

4.2 Exercise 3 Customizing Content

4.2.1 Task 1 Display names

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

Illustration 3 start screen

1. Replace

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

With

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

4.2.2 Task 2 Customizing a BOC List

Looking at the list, we see

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

Illustration 4: Boclist

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

1. Replace

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

With

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

We also might want to combine street with number.

1. Replace

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

with

```
<remotion:BocCompoundColumnDefinition ColumnTitle="Name" ItemID="CompoundEdit"
FormatString="{0}, {1}">
```

```
<propertypathbindings>
  <remotion:PropertyPathBinding PropertyPathIdentifier="Surname" />
  <remotion:PropertyPathBinding PropertyPathIdentifier="FirstName" />
</propertypathbindings>
<persistedcommand>
  <remotion:BocListItemCommand Type="Event"/>
</persistedcommand>
</remotion:BocCompoundColumnDefinition>
```

5 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 presentation layer
-

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

- ▶ Designing forms using the re-motion designer integration for visual studio
- ▶ 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 about re-form, a new approach to design forms
 - ▶ Read about re-vision and re-flow on the community page
 - ▶ 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
-