



FlexiLayout Studio 9.0

Обучение на примерах
Пример 2

Содержание

Краткое описание формы Easiest Recipes	3
Шаг 1. Создание нового проекта	3
Шаг 2. Добавление изображений в пакет	4
Шаг 3. Настройка свойств гибкого описания	6
Шаг 4. Предраспознавание	7
Шаг 5. Просмотр изображений и результатов предраспознавания	7
Шаг 6. Создание идентификационного элемента	8
Шаг 7. Проверка нахождения заголовка формы, соответствующего идентификационному элементу	10
Шаг 8. Определение последовательности поиска полей Recipe Name и Recipe #	11
Шаг 9. Описание поля Recipe #	11
Шаг 10. Описание поля Recipe #. Элемент Recipe	11
Шаг 11. Описание поля Recipe #. Элемент RecipeNumber	12
Шаг 12. Описание поля Recipe #. Блок RecipeNumber	16
Шаг 13. Описание поля Recipe Name	18
Шаг 14. Описание поля Ingredients	19
Шаг 15. Описание полей Cooking Time и Cooking Instructions	21
Шаг 16. Создание элемента CookingTimeHeader	22
Шаг 17. Создание элемента CookingTime	22
Шаг 18. Описание месторасположения блока CookingTime	22
Шаг 19. Создание элемента InvertedHeader	23
Шаг 20. Описание поля Cooking Description	24
Шаг 21. Создание элемента Serves	24
Шаг 22. Создание элемента Servings	25
Шаг 23. Описание поля Cooking Instructions	25
Шаг 24. Создание блока CookingInstructions	26
Шаг 25. Гибкое описание готово	26

Краткое описание формы Easiest Recipes

Внимание! Для простоты в данном примере рассматривается одностраничный документ.

В данном руководстве ABBYY FlexiLayout Studio описывается процесс создания гибкого описания для документа с большим разнообразием полей по сравнению с Halloween Registration Form. Здесь Вы сможете не только по шагам проследить процесс создания гибкого описания, но и разобрать некоторые нестандартные подходы, применяемые при описании местонахождения объектов изображения документа.

Проект FlexiLayout Studio, содержащий тестовые изображения и отлаженное гибкое описание, находится в <имя диска>:\Documents and Settings\All Users\Application Data\ABBYY\FlexiCapture\9.0\Samples\FlexiLayoutStudio\Recipe.

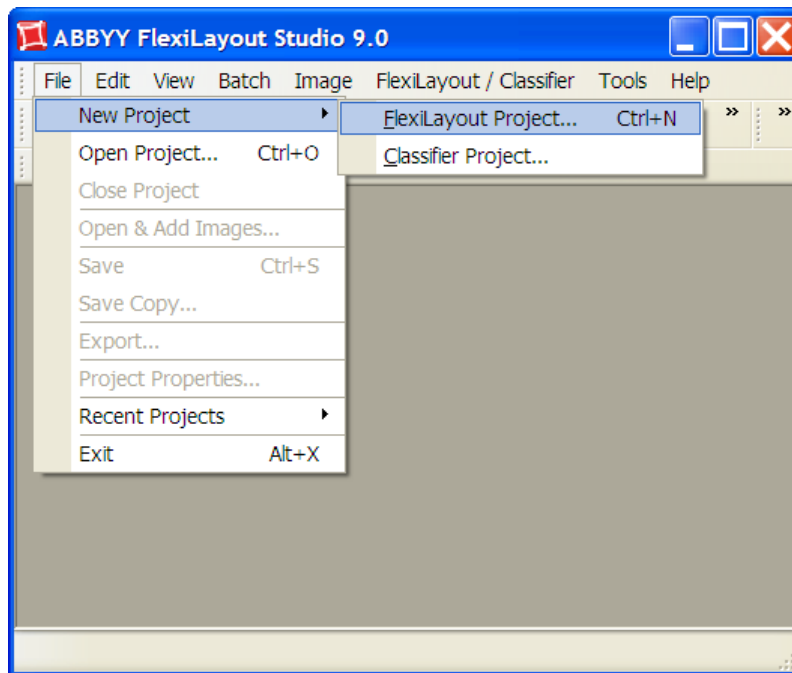
Конечной целью создания гибкого описания данной формы в ABBYY FlexiLayout Studio является устойчивое нахождение на всех тестовых страницах местоположения следующих полей:

Recipe Name
Recipe #
Ingredients
Cooking Time
Cooking Description

Easiest Recipes Barbecued Beefy Beans Baked barbecued beans are a delicious side dish for a cookout, potluck, or family meal. Recipe #: 878VNN233 INGREDIENTS: - 1 tablespoon vegetable oil - 1/2 pound lean ground beef - 1/4 cup chopped green pepper - 1/2 cup chopped onions - 1/2 cup chopped celery 1 can (8oz) tomato sauce - 1/2 cup water - 1 large clove garlic, minced - 2 tablespoons vinegar - 1 tablespoon dry mustard - 1/2 teaspoon thyme - 1 tablespoon brown sugar salt & pepper to taste - 1 large can (approximately 28 ounces) pork and beans Cooking: Heat vegetable oil in a large skillet over medium heat; saute ground beef, onions, green bell pepper, and celery until beef is browned and vegetables are tender. Add tomato sauce, water, garlic, vinegar, mustard, thyme, and brown sugar. Blend well; simmer for 5 minutes. Taste and add salt and pepper as needed. Pour canned pork and beans into a 2-quart casserole; gently stir in meat mixture. Bake at 375° for about 30 minutes. Barbecued beans serves 4 to 6. 1 hour
--

Шаг 1. Создание нового проекта

1. Создайте новую папку **Recipes**.
2. Запустите программу ABBYY FlexiLayout Studio.
3. Создайте новый проект (меню **File>New Project...**).



4. В открывшемся диалоге задайте имя проекта **Sample2**.

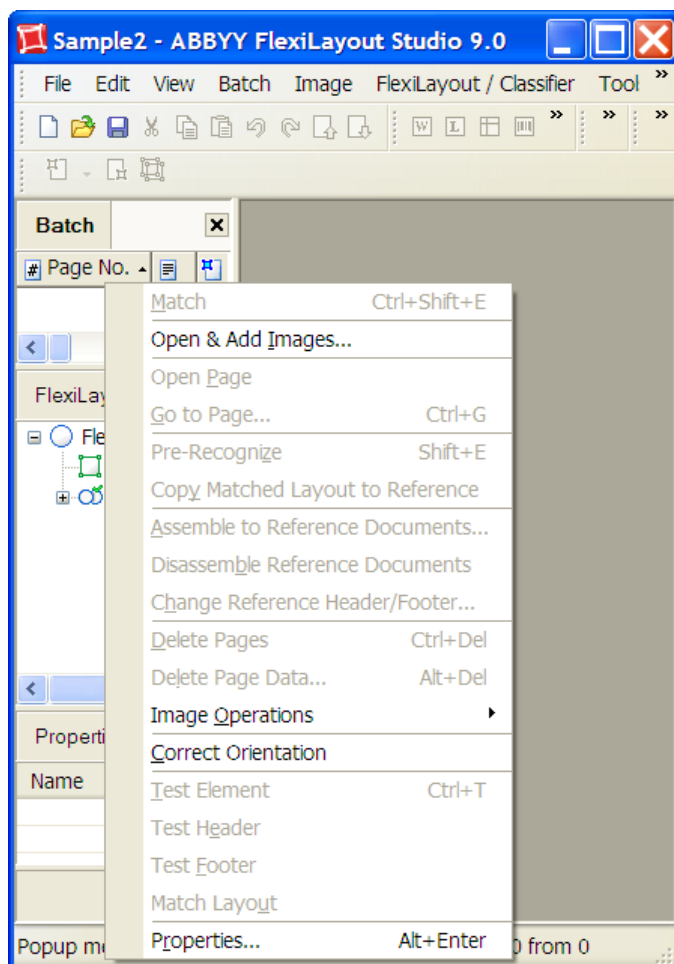
После создания проекта в папке **Recipes** будут созданы:

- **Sample2.fsp** - файл проекта,
- **Sample2Batch** – папка, в которую по умолчанию сохраняются изображения, добавляемые в пакет FlexiLayout,
- **Sample2Templates** - папка, в которую по умолчанию сохраняется гибкое описание.

Шаг 2. Добавление изображений в пакет

После создания нового проекта необходимо добавить в пакет все тестовые изображения, на которых будет осуществляться настройка и отладка гибкого описания. Для этого:

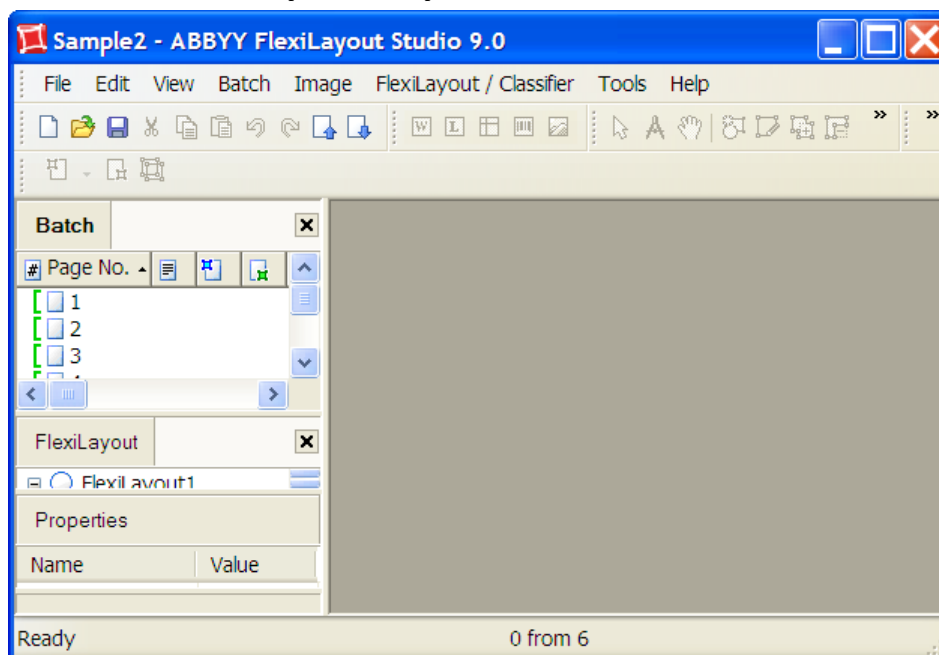
1. Перейдите в окно **Batch**.
2. Выберите команду **Open & Add Images** в контекстном меню, либо в меню **File** основного меню программы.



3. В открывшемся диалоге **Open Images**, отметьте опцию **Document per file** и укажите файлы тестовых изображений

(тестовые изображения этого примера хранятся в
 <имя диска>:\Documents and Settings\All Users\Application Data\ABBYY\FlexiCapture\9.0\Samples\FlexiLayoutStudio\Recipe\RecipeBatch)

Добавленные в пакет тестовые изображения отобразятся в окне **Batch**.



 **Замечание:** При добавлении изображений можно сразу определить принцип эталонной сборки изображений в документы. Выбрав опцию **Document per file**, мы указываем программе, что каждое изображение является отдельным документом. Подробнее об эталонной сборке см. Справка\Отладка гибкого описания\Эталонная сборка страниц в документы.

Шаг 3. Настройка свойств гибкого описания

После создания нового проекта гибкое описание по умолчанию называется **FlexiLayout1** (окно **FlexiLayout**).

Для удобства работы рекомендуется переименовать гибкое описание. Также необходимо указать параметры предраспознавания.

Для того чтобы задать свойства гибкого описания (в том числе, для его переименования):

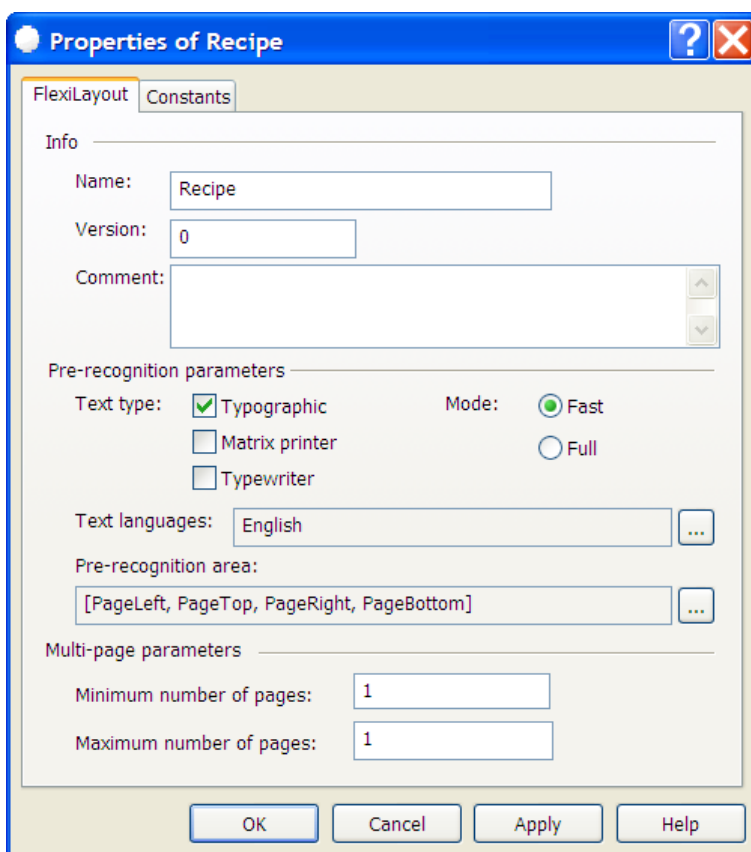
1. Дважды щелкните по имени гибкого описания (в данном случае **FlexiLayout1**) левой кнопкой мыши (либо выберите **Properties** в локальном меню, которое вызывается по нажатию правой кнопки мыши).
2. Задайте имя гибкого описания в поле **Name**, например, **Recipe**.
3. Укажите параметры предраспознавания:

Text Language – наш документ является англоязычным, поэтому в качестве языка предраспознавания в списке выберите **English**.

В поле **Text Type** оставляем **Typographic**.

Выбираем режим предраспознавания **Fast**, предлагаемый по умолчанию.

Multi-page parameters – наш документ является одностраничным. Поэтому задайте значения параметров **Minimal number of pages** и **Maximal number of pages** равными 1.



 **Замечание.** Более подробно о критериях выбора параметров предраспознавания см. Пример 1, Шаг 3.

Для одностраничных документов нет необходимости в описании начала и конца документа с помощью предопределенных составных элементов **Header** и **Footer**. Поэтому их можно удалить из списка элементов гибкого описания.

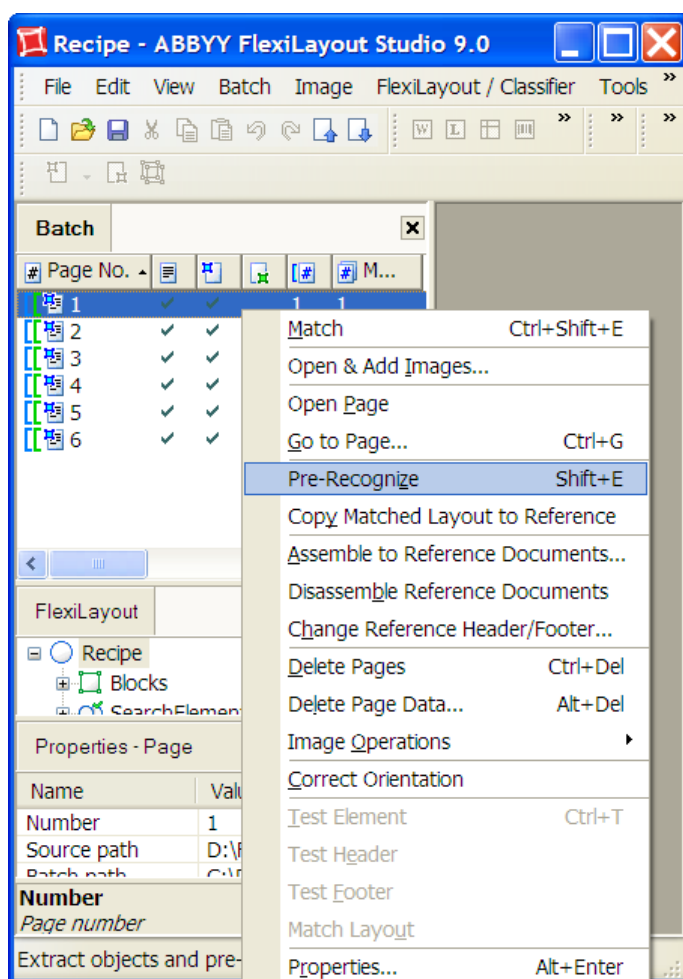
Замечание. Для определения начала и конца документа в гибкое описание могут быть добавлены специальные элементы: **Header** и **Footer**. Если в процессе наложения гибкого описания не найден ни элемент **Header** ни элемент **Footer**, то число страниц в документе считается равным наибольшему числу страниц, указанному в свойствах гибкого описания. Для данного документа в свойствах гибкого описания мы указали 1.

Шаг 4. Предраспознавание

Перед созданием элементов необходимо понять, на какие объекты изображений документа мы можем опереться для поиска ее полей. Такими объектами могут служить только такие графические и/или текстовые фрагменты изображения, которые устойчиво находятся в процессе предраспознавания изображений документа.

Для того чтобы запустить процесс предраспознавания:

1. Выделите все изображения в пакете.
2. Выберите команду **Prerecognize** (меню **Batch** или локальное меню изображения).



Замечание. Если тестовые изображения в пакете меняются и/или добавляются, необходимо перезапустить процедуру предраспознавания. Если предраспознавание страницы не было выполнено предварительно, то оно будет проделано в процессе наложения гибкого описания.

Шаг 5. Просмотр изображений и результатов предраспознавания

Предварительный просмотр результатов предраспознавания всех тестовых страниц показывает, что все текстовые объекты, соответствующие заголовкам искомым полей (для полей с заголовком), инвертированный текст и горизонтальные разделители обнаруживаются устойчиво. Поэтому на них можно будет опираться при поиске других объектов изображения.

Теперь можно приступать непосредственно к созданию элементов. Для этого перейдем окно **FlexiLayout** в главном окне программы ABBYY FlexiLayout Studio.

 **Замечание.** Более подробно о просмотре страниц и результатах предраспознавания см. Пример 1, шаг 5, шаг 6.

 **Замечание.** Для каждого изображения пакета можно создать эталонную разметку, которая будет отражать желаемое расположение блоков на изображении. Эталонная разметка может быть создана вручную, также можно создать эталонную разметку на основе результатов наложения гибкого описания. Эталонная разметка нужна на этапе отладки гибкого описания, она помогает создать по возможности оптимальное описание. Эталонная разметка дает возможность сравнивать результаты наложения гибкого описания и желаемое расположение блоков на каждом изображении пакета. Подробнее см. Эталонная разметка

Шаг 6. Создание идентификационного элемента

При обработке гибких форм в программе ABBYY FlexiCapture хотелось бы по возможности полностью исключить случайное распознавание документов, не относящихся к заданному типу. Одним из способов идентификации документов является задание, по крайней мере, одного обязательного элемента.

В нашем случае в качестве идентификатора будем использовать четкий, хорошо распознающийся текст заголовка формы.

 **Замечание.** Идентификационный элемент или идентификационный набор элементов можно описать в предопределенном составном элементе Header. (В данном примере не рассматривается).

Заголовок формы нам понадобится исключительно для идентификации формы, и он не будет впоследствии распознаваться в программе ABBYY FlexiCapture. Для его нахождения создадим в гибком описании элемент типа **Static text**.

Для того чтобы создать элемент:

1. Перейдите в окно **FlexiLayout** в главном окне программы ABBYY FlexiLayout Studio.
2. В ветви элементов выделите **SearchElements**.
3. Выберите команду **Add Element** (меню **FlexiLayout>Add Element>Static Text** или в локальном меню элемента).
4. Задайте имя элемента, например **FormHeader**, в поле **Name**.
5. Отмените заданное по умолчанию свойство **Optional** на закладке **General** (поскольку данный элемент мы решили сделать обязательным).

Properties of SearchElements.FormHeader

General Static Text Search Constraints Relations Advanced

Name: FormHeader

Full name: SearchElements.FormHeader

Comment:

Search control

☐ Optional element
Null hypothesis quality: 0.97

☒ Required element

☐ Prohibited element

Number of surviving hypotheses: 5

OK Cancel Apply Help

6. Перейдите на закладку **Static text**.

Properties of SearchElements.FormHeader

General Static Text Search Constraints Relations Advanced

☒ Search text:
Easiestrecipes|Easytocookrecipes

☐ Search text from database:
Connection string:
Query text:

☐ Search text from file:
Browse...

☐ Max number of errors: INF
Max error percentage: 20

Max space length: 300

☐ Whole words only
☐ Match case

☐ Take spaces into account
☐ Permit multiple line
Line break penalty: 0
☐ Permit missing word
Missing word: 0

OK Cancel Apply Help

7. В поле **Search Text** укажите значение искомого текста.
Как можно видеть, в пакете присутствуют тестовые изображения, содержащие в заголовке разный текст: **Easiest Recipes** и **Easy to cook Recipes**. В качестве значения зададим оба варианта текста заголовка.
На всех тестовых изображениях заголовок формы однострочный. Поэтому в качестве искомого текста мы

можем задать оба варианта текста заголовка без пробелов. Это позволит ускорить поиск однострочного статического текста. Для обозначения выбора из нескольких вариантов между ними ставится значок «|».

8. Задайте максимально допустимое количество ошибок в искомой фразе в процентах или в явной форме (числом). Зададим в поле **Max error percentage** максимально допустимый процент ошибок 20. Таким образом, для нашего искомого заголовка в 14/17 символов мы допускаем 2/3 ошибки.

 **Замечание.** Значение допустимого числа ошибок подбирается опытным путем.

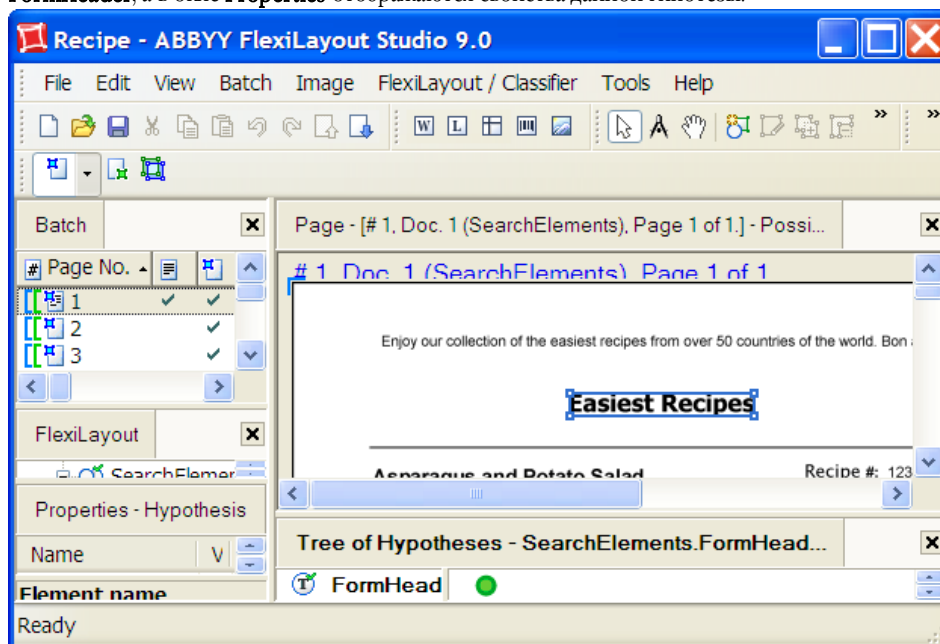
Шаг 7. Проверка нахождения заголовка формы, соответствующего идентификационному элементу

Для проверки нахождения заголовка формы, соответствующего идентификационному элементу **FormHeader**, необходимо наложить гибкое описание на каждое изображение пакета.

Для этого:

1. Откройте каждое изображение пакета.
2. Выберите команду **Match FlexiLayout & Show hypotheses** (меню **Page** или локальное меню изображения).

При успешном наложении гибкого описания в окне **Tree of Hypotheses** сформированная гипотеза отображается значком . При нажатии на него курсором мыши на изображении синим цветом выделяется регион объекта, соответствующего элементу **FormHeader**, а в окне **Properties** отображаются свойства данной гипотезы.



Наложение гибкого описания показывает, что на некоторых изображениях был обнаружен не заголовок документа, а часть фразы, находящейся над заголовком. Как можно заметить фраза "Enjoy our collection of the easiest recipes from over 50 countries of the world." встречается почти на всех изображениях пакета. Чтобы исключить подобную ошибку надо скорректировать свойства идентификационного элемента. Поскольку заголовок формы находится всегда ниже фразы "Enjoy our collection of the easiest recipes from over 50 countries of the world.", то будем искать объект изображения, удовлетворяющий всем свойствам и ближайший к нижнему краю изображения.

Для того чтобы задать дополнительное условие поиска:

1. Откройте диалог **Properties** со свойствами элемента **FormHeader**.
2. Выберите закладку **Advanced**.
3. В поле **Advanced pre-search relations** задайте дополнительное условие поиска: *Искать объект изображения, ближайший к нижнему краю изображения*. На языке описания FlexiLayout данное условие выглядит следующим образом: **NearestY: PageRect.Bottom;**

 **Замечание.** Данное условие также можно задать через интерфейс программы. Для этого перейдите на закладку **Relations**, отметьте опцию **Nearest** и в выпадающем списке **To:** выберите **Page bottom edge**.

 **Замечание.** Функция **Nearest** указывает программе, что из нескольких гипотез элемента необходимо искать ближайшую к некоторому элементу или точке на изображении (сравнивается расстояние между центрами элементов). После ее срабатывания из нескольких рассматриваемых гипотез остается только одна. Данная функция не учитывает качество гипотез, так как выбор происходит уже на этапе создания гипотез.

Повторное наложение гибкого описания показывает, что заголовок формы, соответствующий идентификационному элементу, устойчиво обнаруживается на всех тестовых изображениях.

Шаг 8. Определение последовательности поиска полей **Recipe Name** и **Recipe #**

Обычно месторасположение поля документа ищется относительно заголовка этого поля. Но на нашей форме не у всех полей присутствуют заголовки, например, у поля **Recipe Name** он отсутствует. Поэтому поиск месторасположения поля **Recipe Name** следует организовать иным способом.

Например, можно попытаться найти его, как ближайший (снизу) к заголовку формы объект изображения. Однако, при просмотре тестовых изображений видно, что на изображении 6 ближайшим к заголовку формы является поле **Recipe #**. В данном случае сложно наложить дополнительные условия для поиска поля **Recipe Name**, так как поля **Recipe Name** и **Recipe #** располагаются рядом друг с другом и имеют схожую структуру:

1. Поле **Recipe #** и его заголовок пишутся в одну строку, в которую могут входить те же символы, что и в название рецепта.
2. Нельзя гарантировать, что длина строки, содержащей название рецепта, будет заведомо больше длины строки, содержащей заголовок поля **Recipe #** и само поле.

Обратим внимание, что при предраспознавании заголовок поля **Recipe #** обнаруживается устойчиво на всех тестовых изображениях. Поэтому будем искать сначала поле **Recipe #**. А затем будем искать местоположение поля **Recipe Name** как ближайший (снизу) к заголовку формы объект изображения, исключая из области поиска регионы элементов, соответствующих полю **Recipe #** и его заголовку.

Шаг 9. Описание поля **Recipe #**

На всех тестовых изображениях заголовок поля **Recipe #** обнаруживается устойчиво и является уникальными в рамках данной формы. Поэтому для описания поля **Recipe #** необходимы:

1. элемент **Recipe** (тип **Static Text**), соответствующий заголовку поля **Recipe #**.
2. элемент **RecipeNumber** (тип **Character String**), соответствующий самому полю **Recipe #**.
3. блок **RecipeNumber**, соответствующий полю **Recipe #**.

Шаг 10. Описание поля **Recipe #**. Элемент **Recipe**

Для того чтобы создать элемент **Recipe**:

1. Создайте элемент **Static text** с именем **Recipe**.
Оставим выбранным пункт **Optional element**. В отличие от заголовка формы, который в целях идентификации формы был задан как обязательный, предполагается, что в общем случае все остальные элементы формы могут быть необязательными. Это позволит сформировать гипотезы, даже для тех элементов, для которых не будут найдены соответствующие объекты изображения.

2. Выберите закладку **Static Text**.
3. В поле **Search Text** укажите значение искомого текста. Как можно видеть, на всех тестовых формах заголовок является однострочным. Поэтому мы можем написать заголовок поля без пробелов **Recipe#**.
4. Задайте значение свойства **Max error percentage** равным 10, поскольку заголовок поля состоит всего из одного слова, поэтому в нем не может встретиться значительное число ошибок.

Проверка качества наложения гибкого описания показывает, что заголовок поля, содержащего номер рецепта, успешно находится на всех тестовых изображениях.

Шаг 11. Описание поля Recipe #. Элемент RecipeNumber

Будем полагать, что значение поля **Recipe #**: всегда является однострочным. Поскольку значение этого поля не является фиксированным, в отличие от заголовка поля, для его нахождения следует определить элемент типа **Character String**. Для того чтобы создать элемент **RecipeNumber**:

1. Создайте элемент типа **Character String** с именем **RecipeNumber**.

Properties of SearchElements.RecipeNumber

General Character String Search Constraints Relations Advanced

Name:

Full name:

Comment:

Search control

☒ Optional element
Null hypothesis quality:

☐ Required element

☐ Prohibited element

Number of surviving hypotheses:

OK Cancel Apply Help

2. Выберите закладку **Character String**.

Properties of SearchElements.RecipeNumber

General Character String Search Constraints Relations Advanced

☐ Regular expression:

☒ Alphabets:

Character Set	Portio...
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ...	100.

Add... Edit... Delete

☐ Allow embedded hypotheses

Percentage of non-alphabet characters:

☐ Whole words only ☐ Detect words by interword space

Max space length: Min interword space:

Character count:

Word count:

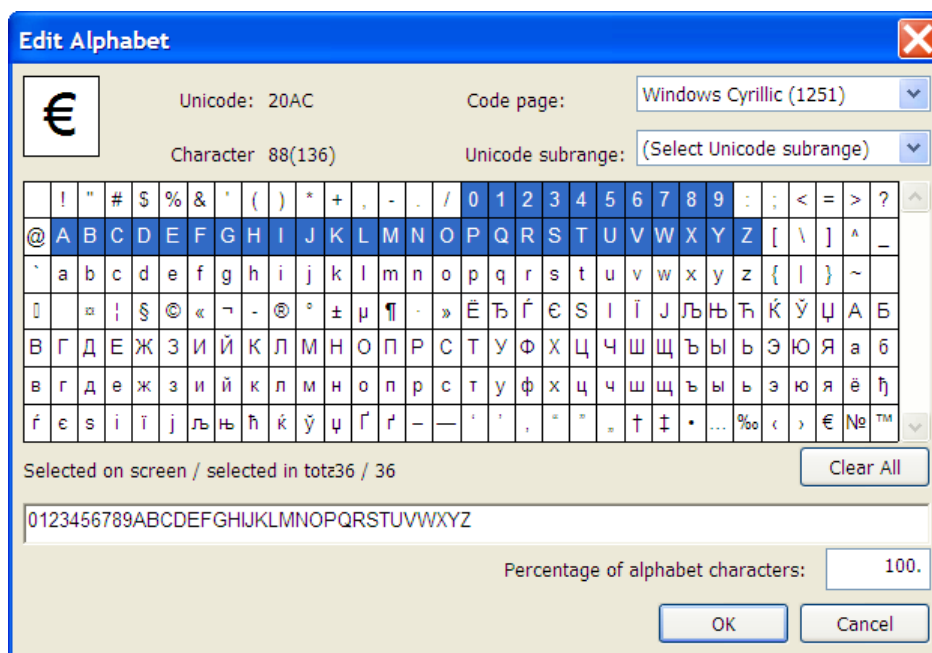
OK Cancel Apply Help

3. Задайте алфавит (все возможные символы, которые могут встретиться в номере рецепта). Исходя из информации, полученной при просмотре тестовых изображений, предполагаем, что это – все заглавные буквы латинского алфавита и цифры. Причем их порядок и количественное соотношение произвольно.

Для задания символов алфавита следует нажать кнопку **Add** и выделить соответствующие символы в открывшемся диалоге **Add New Alphabet**.

 **Замечание.** Выбранные символы отобразятся в столбце **Character set**, поля **Alphabets**, закладки **Character String**.

4. Задайте значение параметра **Percentage of alphabet characters** равным 100. Это значит, что при поиске элемента **RecipeNumber** в расчет будут приниматься только символы данного алфавита.



The screenshot shows the 'Edit Alphabet' dialog box. It has a title bar 'Edit Alphabet' with a close button. Inside, there's a character set grid with columns for characters and their Unicode values. The 'Unicode' field is set to 20AC, 'Code page' is Windows Cyrillic (1251), and 'Character' is 88(136). The 'Percentage of alphabet characters' is set to 100. The 'String Length' is set to 10. The 'Total length of spaces' is set to 10. The 'Relations' tab is selected.

 **Замечание.** В общем случае для элемента может быть создано несколько разных алфавитов. В поле **Percentage of alphabet characters** можно задать максимально допустимую долю участия символов каждого из них в данном элементе.

5. Снимите отметку с пункта **Allow embedded hypotheses**. Это позволит формировать гипотезы максимальной длины, удовлетворяющие всем условиям поиска. Иначе может быть сформировано несколько пересекающихся гипотез меньшей длины, каждая из которых состоит из кусочков, относящихся к одному из заданных алфавитов (с учетом параметра **Percentage of non-alphabet characters**, т.е. с добавлением допустимого числа неалфавитных символов), и удовлетворяет всем условиям поиска.
6. В поле **Percentage of non-alphabet characters** задайте допустимую долю символов, не входящих в алфавит данного элемента, равную 20%.

 **Замечание.** Значение данного свойства подбирается опытным путем и может быть изменено в процессе отладки гибкого описания.

7. В поле **String Length** задайте нечеткий интервал { 0, 8, 10, 16 }, оценивающий длину цепочки символов. Предполагаем, что длина номера составляет всегда 8 символов. Запишем с запасом, что длина номера должна находиться в интервале от 8 до 10 символов. Любая гипотеза, выходящая за рамки данного интервала, будет штрафоваться.

 **Замечание.** Значение данного свойства подбирается опытным путем и может быть изменено в процессе отладки гибкого описания.

8. В поле **Total length of spaces** задайте нечеткий интервал, оценивающий суммарную длину всех пробелов между символами в гипотезе (по умолчанию измеряется в dot, 1 dot = 1/300 дюйма). Оставляем значение по умолчанию { -1, -1, INF, INF }. Таким образом, мы не накладываем никаких ограничений на длину пробелов в гипотезе.
9. Перейдите на закладку **Relations**.

Properties of SearchElements.RecipeNumber

General Character String Search Constraints **Relations** Advanced

New relation

Relation: Left of

Reference

Element boundary: <whole>

Offset: 0

Add Preview

Relations:

E...	Relation	Reference Element	Offset
☒	RightOf	SearchElements.Recipe.Right	0
☒	Above	SearchElements.Recipe	-50
☒	Below	SearchElements.Recipe	-50

☐ Nearest To: Element

X: 0 Y: 0

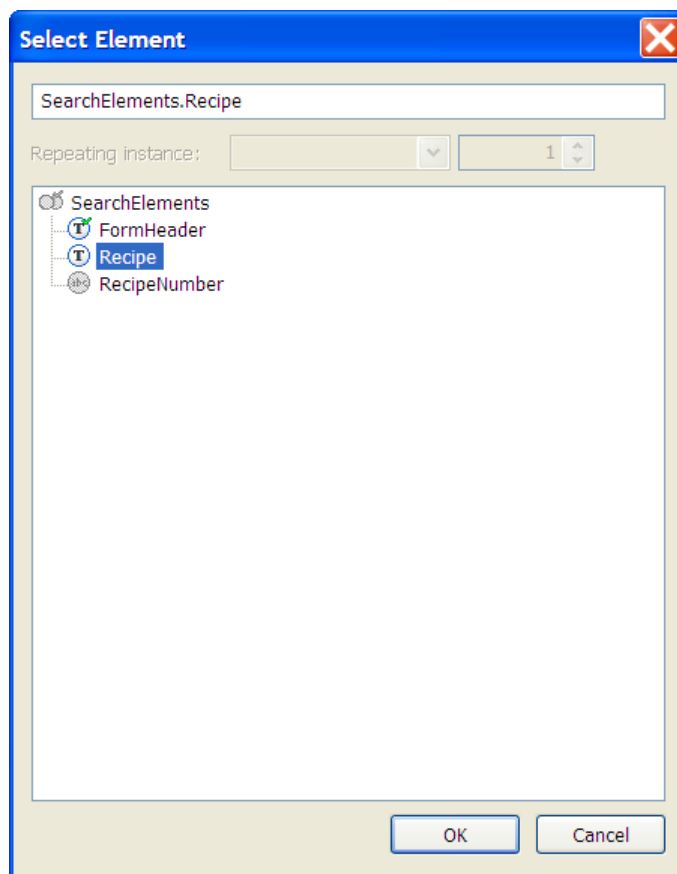
Reference

OK Cancel Apply Help

В качестве опорного элемента (элемента, относительно которого задается положение искомого элемента) для элемента **RecipeNumber** будем использовать элемент **Recipe**. Номер рецепта всегда находится справа от заголовка поля, а также на одном уровне с ним. Поэтому помимо отношения правее, необходимо задать положение номера рецепта относительно верхней и нижней границы элемента заголовка. Иначе в качестве гипотезы может быть найден любой текст, соответствующий заданному алфавиту, и лежащий в правой полуплоскости относительно элемента заголовка, в том числе выше или ниже него.

Итак, будем искать номер рецепта, заданный элементом **RecipeNumber**, справа от заголовка, заданного элементом **Recipe** и находящийся на одном уровне с заголовком **Recipe**.

10. Задайте положение элемента **RecipeNumber** справа от правой границы элемента **Recipe**. Имя элемента **Recipe** следует выделить курсором в окне **Select Element**. В выпадающем списке **Relation** выберите значение **RightOf**, в выпадающем списке **Element border** выберите значение **right** и оставьте значение свойства **Offset** равным 0. Нажмите кнопку **Add**.



11. Задайте положение элемента **RecipeNumber** не ниже элемента **Recipe**.
Имя элемента **Recipe** следует выделить курсором в окне **Select element**. В выпадающем списке **Relation** выберите значение **Above** и задайте значение свойства **Offset** равным **-50** (значение подбирается опытным путем), чтобы описать область поиска с небольшим запасом по высоте заголовка. Отрицательное значение смещения позволяет задавать положение нижней границы элемента ниже нижней границы заголовка. Нажмите кнопку **Add**.
12. Задайте положение элемента **RecipeNumber** не выше элемента **Recipe**.
Имя элемента **Recipe** следует выделить курсором в окне **Select element**. В выпадающем списке **Relation** выберите значение **Below** и задайте значение свойства **Offset** равным **-50** (значение подбирается опытным путем), чтобы описать область поиска с небольшим запасом по высоте заголовка. Отрицательное значение смещения позволяет задавать положение верхней границы элемента выше верхней границы заголовка. Нажмите кнопку **Add**.

 **Замечание.** Неотрицательные смещения при указании в поле **Element border** значения **<whole>** позволяют задавать только положение типа правее правой границы элемента, левее левой границы элемента, выше верхней границы элемента и ниже нижней границы элемента. Задание области поиска между верхней и нижней границами заголовка при указании в свойстве **Element border** значений **<top>** и **<bottom>** с неотрицательным значением свойства **Offset** ограничивает область поиска высотой заголовка, что исключает обнаружение гипотез, слегка выходящих по вертикали за горизонтальные границы заголовка.

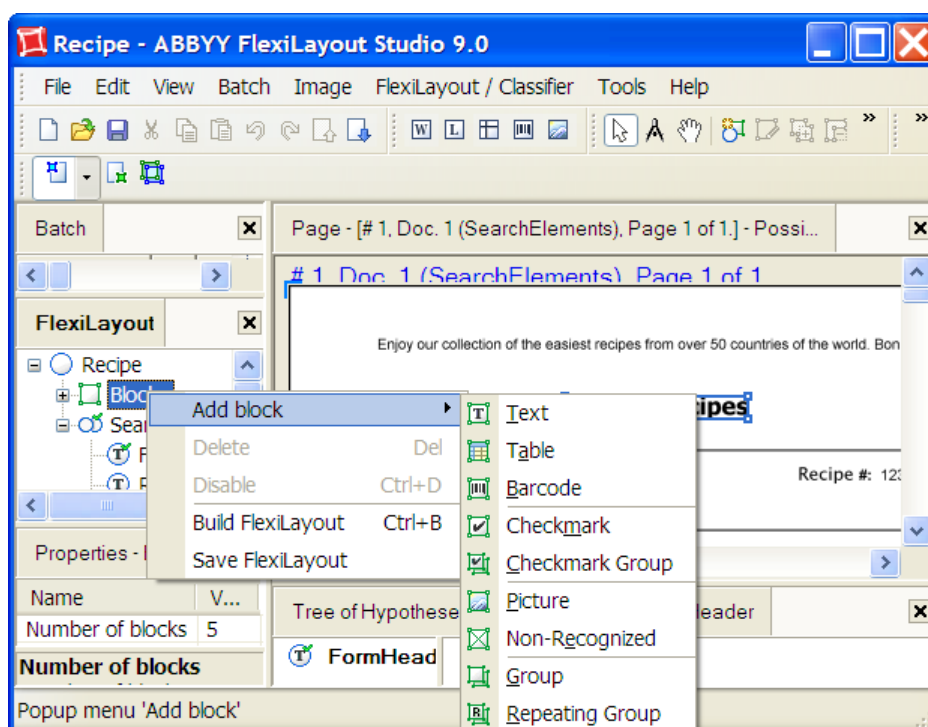
Наложив гибкое описание, увидим, что поле, содержащее номер рецепта, успешно найдено на всех тестовых изображениях.

Шаг 12. Описание поля Recipe #. Блок RecipeNumber

Для последующего распознавания номера рецепта в программе ABBYY FlexiCapture необходимо создать соответствующий блок и задать его месторасположение.

Для того чтобы создать блок, соответствующий полю **Recipe #**:

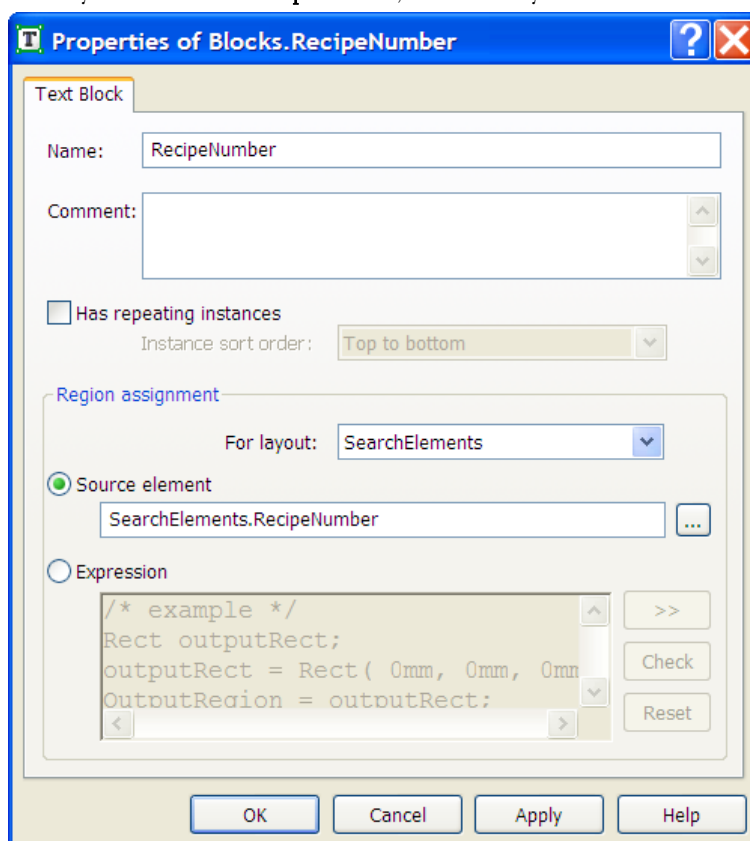
1. Перейдите в окно **FlexiLayout** в главном окне программы ABBYY FlexiLayout Studio.
2. В дереве проекта выделите объект **Blocks**.
3. Выберите команду **Add Block>Text** в меню **FlexiLayout** (или в локальном меню **New>Text**).



4. В открывшемся диалоге **Properties** задайте имя блока, например, **RecipeNumber** в поле **Name**.

Замечание. Совпадение имени блока с именем элемента, соответствующего полю **Recipe #**: не обязательно, но удобно для дальнейшей работы с гибким описанием.

5. Выберите в качестве описания местоположения блока опцию **Source element**. В качестве **Source element** укажите элемент **RecipeNumber**, нажав кнопку "...".



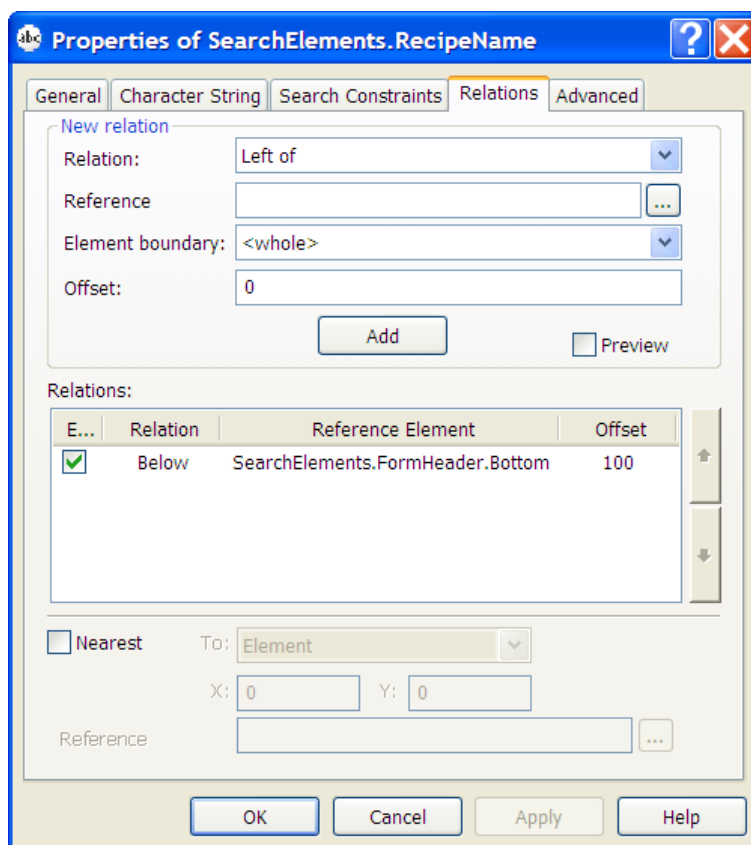
Шаг 13. Описание поля Recipe Name

Будем искать местоположение поля **Recipe Name** как ближайший (снизу) к заголовку формы объект изображения, исключая из области поиска регионы элементов **Recipe** и **RecipeNumber** (см. Шаг 8). Для этого необходимы:

1. элемент **RecipeName** (тип **Character String**), соответствующий самому полю **Recipe Name**.
2. блок **RecipeName**, соответствующий полю **Recipe Name**.

Для того чтобы создать элемент **RecipeName**:

1. Создайте элемент типа **Character String** с именем **RecipeName**.
2. Выберите закладку **Character String**.
3. Задайте алфавит. Исходя из информации, полученной при просмотре тестовых изображений, предполагаем, что это – все буквы латинского алфавита, цифры и символы «#», «-», «&», «.». Причем их порядок и количественное соотношение произвольно.
4. Задайте значение параметра **Persantage of alphabet characters** равным 100.
5. Снимите отметку с пункта **Allow embedded hypotheses**.
6. В поле **Persantage of non-alphabet characters** задайте допустимую долю символов, не входящих в алфавит данного элемента, равную 20%.
7. В поле **String Length** задайте нечеткий интервал, оценивающий суммарную длину цепочки символов. Оставляем значение по умолчанию { -1, -1, INF, INF }. Таким образом, мы не накладываем никаких ограничений на длину цепочки символов.
8. В поле **Total length of spaces** задайте нечеткий интервал, оценивающий суммарную длину всех пробелов между символами в гипотезе (по умолчанию измеряется в dot, 1 dot = 1/300 дюйма). Оставляем значение по умолчанию { -1, -1, INF, INF }. Таким образом, мы не накладываем никаких ограничений на длину пробелов в гипотезе.
9. Перейдите на закладку **Relations**.
На закладке **Relations** зададим, что название рецепта может находиться ниже нижней границы заголовка формы с отступом 100 dot, т.е. между заголовком формы и названием рецепта всегда есть зазор, не меньше 100 dot.



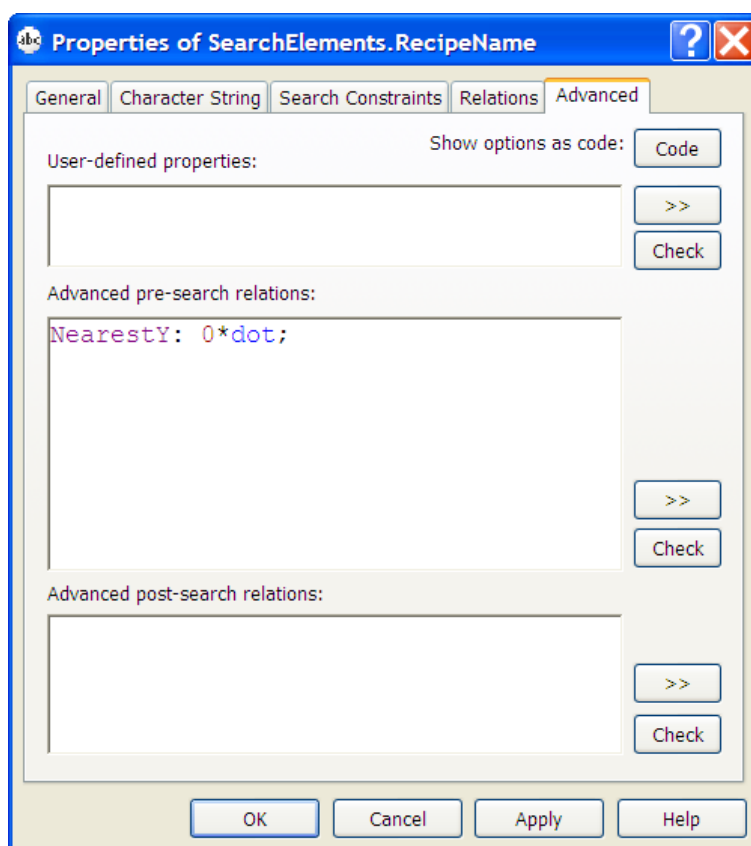
Как можно видеть, на изображениях этим условиям поиска удовлетворяет каждая строка, расположенная ниже заголовка формы. Необходимо сузить область поиска.

10. Перейдите на закладку **Advanced**.

В поле **Advanced pre-search relations** задайте дополнительное условие поиска: *Искать строку, ближайшую к верхнему краю формы*. На языке описания FlexiLayout данное условие выглядит следующим образом: **NearestY: 0*dot;**

В сочетании с условием поиска, заданным на закладке **Relations**, это позволяет сузить поиск до одной строки, ближайшей к заголовку формы.

 **Замечание.** Данное условие также можно задать через интерфейс программы. Для этого перейдите на закладку **Relations**, отметьте опцию **Nearest** и в выпадающем списке **To:** выберите **Page top edge**.



11. Вернитесь на закладку **Search Constraints**.
12. Как отмечалось выше (см. Шаг 8), из области поиска элемента **RecipeName** необходимо исключить регионы элементов **Recipe** и **RecipeNumber**. Для этого добавьте в поле **Exclude elements** элементы **Recipe** и **RecipeNumber**, нажав кнопку **Add...** (рядом с полем **Exclude elements**). В открывшемся диалоге **Select Element to Exclude** выберите соответствующий элемент.

Просмотр результатов наложения гибкого описания показывает, что на всех тестовых изображениях поле **Recipe Name** находится устойчиво.

Процесс создания блока **RecipeName** аналогичен процессу создания блока **RecipeNumber**. Отличие заключается в том, что для блока **RecipeName** в качестве исходного элемента необходимо указать элемент **RecipeName**.

Шаг 14. Описание поля Ingredients

Как можно видеть, на всех тестовых изображениях список ингредиентов представляет собой текст произвольной формы, всегда расположенный под заголовком поля **Ingredients**, но выше разделителя, расположенного над заголовком поля **Cooking**.

Для удобства описания вышеперечисленных объектов и ограничения числа гипотез необходимо объединить их в составной элемент типа **Group**. Для этого:

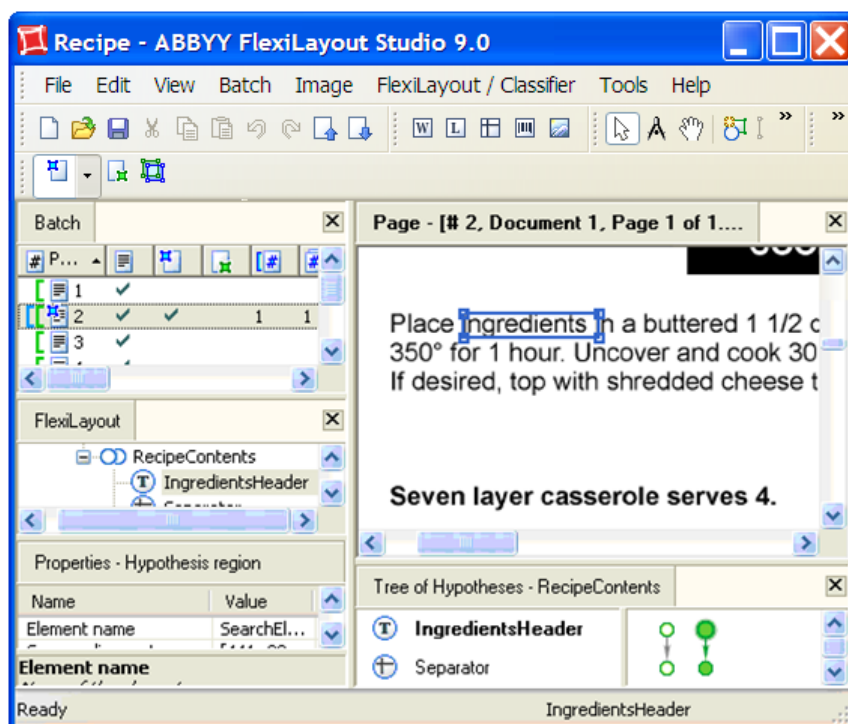
1. Создайте составной элемент **RecipeContents** (тип **Group**).
2. В составе элемента **RecipeContents** создайте элемент **IngredientsHeader** (тип **Static text**), соответствующий заголовку поля **Ingredients**. Свойства элемента задаются по аналогии со свойствами элемента **Recipe** (см. Шаг 10).
3. В составе элемента **RecipeContents** создайте элемент **Separator** (тип **Separator**), соответствующий разделителю, расположенному над заголовком поля **Cooking**. В свойствах элемента (закладка **Separator**) укажите ориентацию разделителя **Horizontal**. Задайте условия поиска (диалог свойств элемента, закладка **Relations**) ниже, чем заголовок **Ingredients: Below** элемента **IngredientsHeader**. При наложении гибкого описания можно увидеть, что иногда за разделитель принимается нижняя граница изображения.

 **Замечание.** При некоторых настройках сканирования, края формы проявляются на изображении в виде темной полосы и могут восприниматься как разделители.

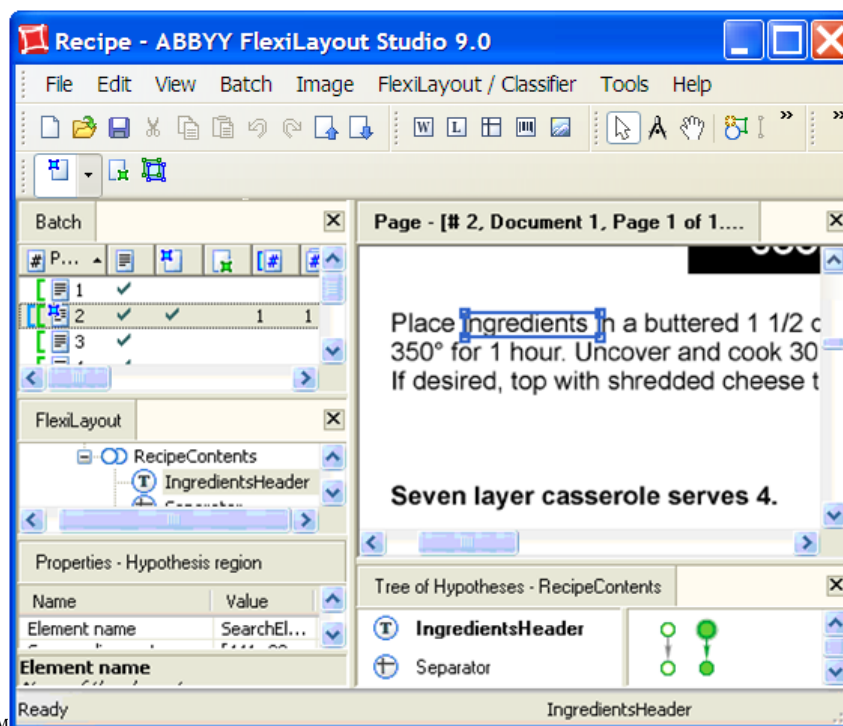
Во избежание подобной ситуации необходимо задать дополнительное условие поиска. Для этого в поле **Advanced pre_search relations** (диалог свойств элемента, закладка **Advanced**) задайте дополнительное условие поиска: Искать разделитель ближайший к заголовку **Ingredients**. На языке описания гибких шаблонов данное условие выглядит следующим образом: **Nearest: SearchElements.RecipeContents.IngredientsHeader;**

 **Замечание.** Данное условие также можно задать через интерфейс программы. Для этого перейдите на закладку **Relations**, отметьте опцию **Nearest** и в выпадающем списке **To:** выберите **Element** и укажите **IngredientsHeader** в качестве **Reference element**.

4. Произведем наложение гибкого описания еще раз.
5. При наложении гибкого описания на второе изображение были сгенерированы 2 гипотезы для элемента **IngredientsHeader** и 2 гипотезы для элемента **Separator**. Обратите внимание, что программа выбрала как лучший тот путь дерева гипотез, в котором было ошибочно распознано слово "**ingredients**" в рецепте как заголовок поля и нижняя граница изображения в качестве разделителя.



При наложении гибкого описания мы видим, что на изображении 2 вместо заголовка было обнаружено слово "**ingredients**" в тексте описания процедуры приготовления. Такая ситуация вполне обоснована, поскольку отличие найденного участка текста от искомого, заданного в свойствах элемента **IngredientsHeader** (закладка **Static text**, поле **Search text**) очень незначительное (один символ «»), и не может быть отсечено допустимым процентом ошибочных символов,



заданным в поле **Max error percentage** (20).

Замечание: Мы не можем задать **Max error percentage** = 0 (или **Max number of errors** = 0), поскольку из-за разного качества изображений ошибки предраспознавания вполне возможны. А в случае не нахождения заголовка **Ingredients**: не будет найден и сам список ингредиентов.

Во избежание подобной ситуации необходимо задать дополнительное условие поиска. Для этого в поле **Advanced pre-search relations** (диалог свойств элемента, закладка **Advanced**) задайте дополнительное условие поиска: Искать заголовок в верхней половине изображения. На языке описания FlexiLayout данное условие выглядит следующим образом:

Above: PageRect.Top + PageRect.Height/2;

6. В составе элемента **RecipeContents** создайте элемент **Ingredients** (тип **Text Fragment**), соответствующий полю **Ingredients**.
 Задайте условия поиска (диалог свойств элемента, закладка **Relations**) ниже, чем заголовок **Ingredients**: (**Below** элемента **IngredientsHeader**), но выше разделителя (**Above** элемента **Separator**).
 Просмотр результатов наложения шаблона показывает, что на всех тестовых изображениях поле **Ingredients** находится устойчиво.

Процесс создания блока **Ingredients** аналогичен процессу создания блока **RecipeNumber**. Отличие заключается в том, что для блока **Ingredients** в качестве исходного элемента необходимо указать элемент **Ingredients**.

Шаг 15. Описание полей Cooking Time и Cooking Instructions

Для описания поиска оставшихся полей **Cooking Time** и **Cooking Instructions** создадим составной элемент. Для этого:

1. Создайте составной элемент **Cooking** (тип **Group**).
2. В составе элемента **Cooking** необходимы
 - 2.1 элемент **CookingTimeHeader** (тип **Static text**), соответствующий единицам времени, указанным в поля **Cooking Time** (подробнее см. Шаг 16).
 - 2.2 элемент **CookingTime** (тип **Character String**), соответствующий полю **Cooking Time** (подробнее см. Шаг 17).
 - 2.3 элемент **InvertedHeader** (тип **Object Collection**), соответствующий заголовку поля **Cooking Instructions** (подробнее см. Шаг 19).
 - 2.4 составной элемент **Instructions** (тип **Group**), соответствующий полю **Cooking Instructions** (подробнее см. Шаг 20).

Шаг 16. Создание элемента CookingTimeHeader

Для того чтобы создать элемент **CookingTimeHeader**:

1. Создайте элемент **CookingTimeHeader** (тип **Static text**).
2. Выберите закладку **Static Text**.
3. В поле **Search Text** укажите значение искомого текста: **minutes|hour|hours**.
4. Задайте значение свойства **Max error percentage** равным 10.
5. Выберите закладку **Advanced**.
6. Поскольку поле **Cooking Time** находится ниже всех остальных объектов изображения, в поле **Advanced pre-search relations** задайте дополнительное условие поиска: *Искать объект изображения, ближайший к нижнему краю изображения*. На языке описания FlexiLayout данное условие выглядит следующим образом: **NearestY: PageRect.Bottom**;

 **Замечание.** Данное условие также можно задать через интерфейс программы. Для этого перейдите на закладку **Relations**, отметьте опцию **Nearest** и в выпадающем списке **To**: выберите **Page bottom edge**.

Шаг 17. Создание элемента CookingTime

Для того чтобы создать элемент **CookingTime**:

1. Создайте элемент **CookingTime** (тип **Character String**).
2. Выберите закладку **Character String**.
3. Задайте алфавит. Исходя из информации, полученной при просмотре тестовых изображений, предполагаем, что это – все цифры.
4. Задайте значение параметра **Persantage of alphabet characters** равным 100.
5. Снимите отметку с пункта **Allow embedded hypotheses**.
6. В поле **Persantage of non-alphabet characters** задайте допустимую долю символов, не входящих в алфавит данного элемента, равную 10%.
7. В поле **String Length** задайте нечеткий интервал [0, 1, 3, 4], оценивающий длину цепочки символов. Предполагаем, что все возможные значения будут находиться в интервале от 1 до 3 символов. Любая гипотеза, выходящая за рамки данного интервала, будет штрафиться.
8. В поле **Total length of spaces** задайте нечеткий интервал, оценивающий суммарную длину всех пробелов между символами в гипотезе (по умолчанию измеряется в dot, 1 dot = 1/300 дюйма). Оставляем значение по умолчанию { -1, -1, INF, INF }. Таким образом, мы не накладываем никаких ограничений на длину пробелов в гипотезе.
9. Перейдите на закладку **Relations**.
На закладке **Relations** зададим, что время приготовления может находиться только слева от единиц времени, но не далее, чем на расстоянии 100 dot от его левой границы (**LeftOf** левой границы региона элемента **CookingTimeHeader**, **Offset** = 0, **Element border** = left и **RightOf** левой границы региона элемента **CookingTimeHeader**, **Offset** = -100, **Element border** = left).
10. Поскольку время всегда находится на одном уровне с единицами времени, поэтому необходимо добавить следующие условия поиска элемента:
 - **Above** элемента **CookingTimeHeader**, **Offset** = -20, **Element border** = bottom
 - **Below** элемента **CookingTimeHeader**, **Offset** = -20, **Element border** = top.

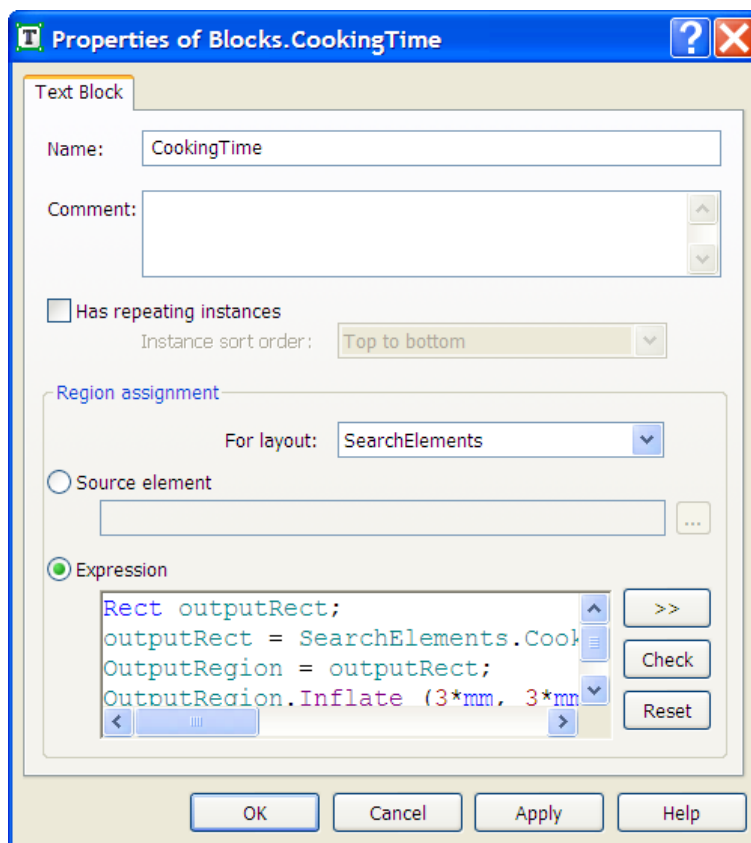
Шаг 18. Описание месторасположения блока CookingTime

Для того чтобы создать блок **CookingTime**:

1. Перейдите в окно **FlexiLayout** в главном окне программы ABBYY FlexiLayout Studio.
2. В дереве FlexiLayout выделите объект **Blocks**.
3. Выберите команду **Add Block>Text** в меню **FlexiLayout** (или в локальном меню **New>Text**).

4. В открывшемся диалоге **Properties** задайте имя блока **CookingTime** в поле **Name**.
5. Выберите в качестве описания местоположения блока опцию **Region expression**.
6. Задайте месторасположение блока в виде описывающего прямоугольника от объединения описывающих прямоугольников регионов двух элементов **CookingTimeHeader** и **CookingTime**, увеличенного по высоте и ширине на 3 мм.
На языке описания FlexiLayout описание месторасположения выглядит следующим образом:

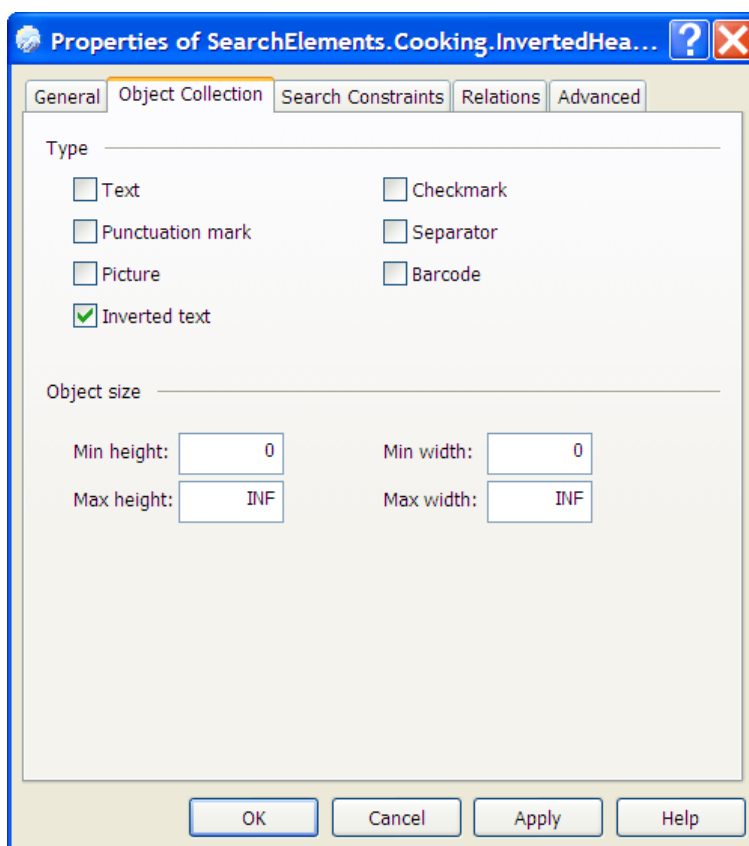
```
Rect outputRect;  
outputRect = SearchElements.Cooking.CookingTime.Rect Or  
SearchElements.Cooking.CookingTimeHeader.Rect;  
OutputRegion = outputRect;  
OutputRegion.Inflate (3*mm, 3*mm);
```



Шаг 19. Создание элемента InvertedHeader

Заголовок **Cooking** представляет собой инвертированный текст (для данного типа документов светлые буквы на темном фоне). Для того чтобы создать элемент **InvertedHeader**:

1. Создайте элемент **InvertedHeader** (тип **Object Collection**).
2. Перейдите на закладку **Object Collection**.
3. В группе **Type** отметьте тип объекта изображения **Inverted text**.



При наложении гибкого описания на все тестовые изображения убедимся, что инвертированный текст обнаруживается устойчиво.

Шаг 20. Описание поля Cooking Description

Нам осталось задать условия поиска поля **Cooking Instructions**, содержащего описание процедуры приготовления блюда и информацию о количестве порций.

Заметим, что на всех изображениях информация о количестве порций находится справа от слова **Serves**. Однако на изображениях 2, 3, 4 слову **Serves** предшествует некоторый текст, не содержащий важной информации. Опишем поле **Cooking Instructions** так, чтобы этот текст не попал в распознаваемый блок, соответствующий полю **Cooking Instructions**.

Для удобства навигации по дереву FlexiLayout, а также для уменьшения общего числа гипотез в элементе **Cooking** создайте составной элемент **Instructions**. Для этого:

1. В составе элемента **Cooking** создайте составной элемент **Instructions** (тип **Group**).
2. В составе элемента **Instructions** создайте элемент **Serves** (тип **Static text**), для нахождения слова **Serves** (подробнее см. Шаг 21).
3. В составе элемента **Instructions** создайте элемент **Servings** (тип **Character String**), для нахождения информации о количестве порций (подробнее см. Шаг 22).
4. В составе элемента **Instructions** создайте элемент **Cooking Instructions** (тип **Text Fragment**), соответствующий полю **Cooking Instructions** (подробнее см. Шаг 23).
5. Создайте блок **CookingInstructions**, соответствующий полю **Cooking Instructions** (подробнее см. Шаг 24).

Шаг 21. Создание элемента Serves

Для того чтобы создать элемент **Serves**:

1. Создайте элемент **Serves** (тип **Static text**).

2. Выберите закладку **Static Text**.
3. В поле **Search Text** укажите значение искомого текста: **Serves**.
4. Задайте значение свойства **Max error percentage** равным 20.
5. Выберите закладку **Advanced**.
6. В поле **Advanced pre-search relations** задайте дополнительное условие поиска: *Искать объект изображения, ближайший к заголовку поля **Cooking time***. На языке описания FlexiLayout данное условие выглядит следующим образом:
Nearest: SearchElements.Cooking.CookingTimeHeader;

 **Замечание.** Данное условие также можно задать через интерфейс программы. Для этого перейдите на закладку **Relations**, отметьте опцию **Nearest** и в выпадающем списке **To:** выберите **Element** и укажите **CookingTimeHeader** в качестве **Reference element**.

Шаг 22. Создание элемента Servings

Для того чтобы создать элемент **Servings**:

1. Создайте элемент **Servings** (тип **Character String**).
2. Выберите закладку **Character String**.
3. Задайте алфавит. Исходя из информации, полученной при просмотре тестовых изображений, предполагаем, что это – все цифры и буквы «Т», «т», «О», «о».
4. Задайте значение параметра **Persantage of alphabet characters** равным 100.
5. Снимите отметку с пункта **Allow embedded hypotheses**.
6. В поле **Persantage of non-alphabet characters** задайте допустимую долю символов, не входящих в алфавит данного элемента, равную 20%.
7. В поле **String Length** задайте нечеткий интервал [0, 1, 10, 15], оценивающий длину цепочки символов. Предполагаем, что все возможные значения будут находиться в интервале от 1 до 10 символов. Любая гипотеза, выходящая за рамки данного интервала, будет штрафовать.
8. В поле **Total length of spaces** задайте нечеткий интервал, оценивающий суммарную длину всех пробелов между символами в гипотезе (по умолчанию измеряется в dot = 1/300 дюйма). Оставляем значение по умолчанию { -1, -1, INF, INF }. Таким образом, мы не накладываем никаких ограничений на длину пробелов в гипотезе.
9. Перейдите на закладку **Relations**.
10. Поскольку количество порций может находиться только справа от слова **Serves**, но не далее, чем на расстоянии 200 dot от его правой границы, задайте следующие условия поиска:
 - **RightOf** элемента **Serves**, **Offset** = 0
 - **LeftOf** правой границы региона элемента **Serves**, **Offset** = -200, **Element border** = right
11. Поскольку количество порций всегда находится на одном уровне со словом **Serves**, добавьте следующие условия поиска:
 - **Above** элемента **Serves**, **Offset** = -50
 - **Below** элемента **Serves**, **Offset** = -50

Шаг 23. Описание поля Cooking Instructions

Для того чтобы создать элемент **CookingInstructions**:

1. Создайте элемент с именем **CookingInstructions** (тип **Text Fragment**).
2. Перейдите на закладку **Relations**.
3. Поскольку описание процесса приготовления находится ниже заголовка, но выше слова **Serves**, задайте условия поиска:
 - **Below** элемента **InvertedHeader**, **Offset** = 0
 - **Above** элемента **Serves**, **Offset** = 0

Шаг 24. Создание блока CookingInstructions

Для того чтобы создать блок **CookingInstructions**:

1. Перейдите в окно **FlexiLayout** в главном окне программы ABBYY FlexiLayout Studio.
2. В дереве FlexiLayout выделите объект **Blocks**.
3. Выберите команду **Add Block>Text** в меню **FlexiLayout** (или в локальном меню **New>Text**).
4. В открывшемся диалоге **Properties** задайте имя блока **CookingInstructions** в поле **Name**.
5. Выберите в качестве описания местоположения блока опцию **Region expression**.
6. Необходимо задать месторасположения блока так чтобы в него попали только данные о приготовлении блюда и данные о количестве порций. При этом в блок не должны попасть лишние данные (весь текст, находящийся слева от слова **Serves**). Для этого задайте месторасположение блока в виде массива прямоугольников регионов элементов: **CookingInstructions**, **Serves** и **Portions**. Такой способ описания региона позволяет задавать области сложной (не прямоугольной) конфигурации, что бывает необходимо, как в нашем случае.
На языке описания FlexiLayout описание месторасположения выглядит следующим образом:

```
RectArray outputRectArray;  
// инициализация переменной с использованием пустого конструктора  
outputRectArray = RectArray();  
// добавление прямоугольников гипотез 3-х элементов  
outputRectArray = RectArray();  
outputRectArray.Add ( SearchElements.Cooking.Instructions.Cooking Instructions.Rect );  
outputRectArray.Add( SearchElements.Cooking. Instructions.Serves.Rect );  
outputRectArray.Add( SearchElements.Cooking. Instructions.Servings.Rect );  
OutputRegion = Region( outputRectArray );
```

Шаг 25. Гибкое описание готово

Полученное гибкое описание можно экспортировать в формат ABBYY FlexiLayout (*.afl) и подключить к шаблону в программе ABBYY FlexiCapture.

Подробнее об экспорте и подключении смотрите Пример 1, Шаг 21 – Шаг 22.