FlexiLayout Studio 8.0

Tutorial
Sample 3

Contents

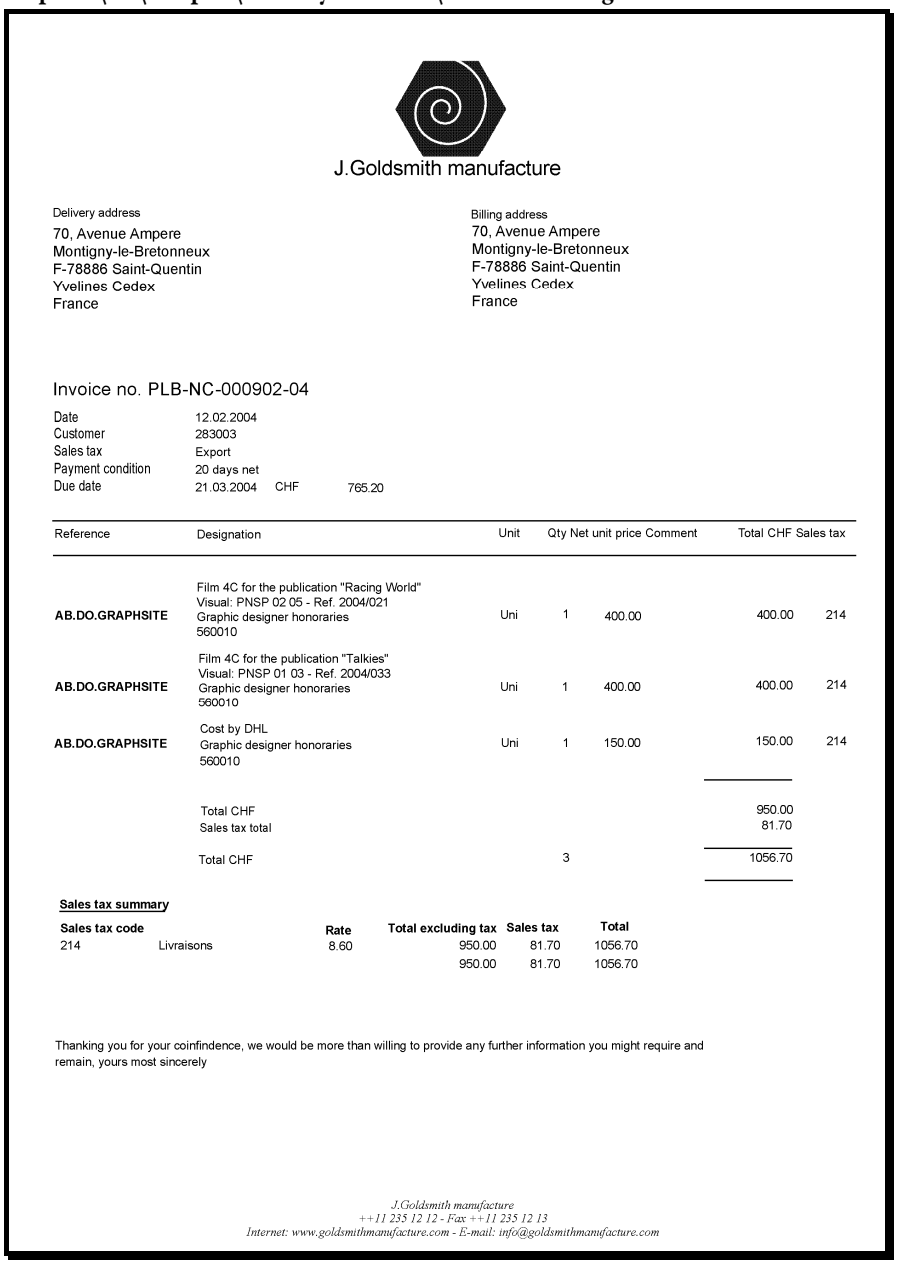
Brief Description of the Form Invoices

Important! For the sake of simplicity, this sample uses a one-page document. This sample describes the creation of a FlexiLayout for a simple invoice. Compared to the previous samples, this one includes a table.

This sample will guide you through all the steps of FlexiLayout creation and at the same time teach you some tricks you can use to describe the position of objects. Most importantly, it will teach you how to detect image objects which correspond to Table elements.

This sample contains references to the following sections of the help file: Help/Tutorial/Sample1 and Help/Tutorial/Sample2, which describe the creation of FlexiLayouts with blocks and elements of various types (except Tables). Some of the basic operations are described there in greater detail.

The images of this document are stored in **<disk name>:\\Documents and Settings\All Users\Application Data\ABBYY\FlexiCapture\8.0\Samples\FlexiLayoutStudio\Invoice One-Page.**



J.Goldsmith manufacture

Delivery address
70, Avenue Ampere
Montigny-le-Bretonneux
F-78886 Saint-Quentin
Yvelines Cedex
France

Billing address
70, Avenue Ampere
Montigny-le-Bretonneux
F-78886 Saint-Quentin
Yvelines Cedex
France

Invoice no. PLB-NC-000902-04

Date	12.02.2004		
Customer	283003		
Sales tax	Export		
Payment condition	20 days net		
Due date	21.03.2004	CHF	765.20

Reference	Designation	Unit	Qty	Net unit price	Comment	Total CHF	Sales tax
AB.DO.GRAPHSITE	Film 4C for the publication "Racing World" Visual: PNSP 02 05 - Ref. 2004/021 Graphic designer honoraries 560010	Uni	1	400.00		400.00	214
AB.DO.GRAPHSITE	Film 4C for the publication "Talkies" Visual: PNSP 01 03 - Ref. 2004/033 Graphic designer honoraries 560010	Uni	1	400.00		400.00	214
AB.DO.GRAPHSITE	Cost by DHL Graphic designer honoraries 560010	Uni	1	150.00		150.00	214
	Total CHF					950.00	
	Sales tax total					81.70	
	Total CHF		3			1056.70	

Sales tax summary

Sales tax code		Rate	Total excluding tax	Sales tax	Total
214	Livraisons	8.60	950.00	81.70	1056.70
			950.00	81.70	1056.70

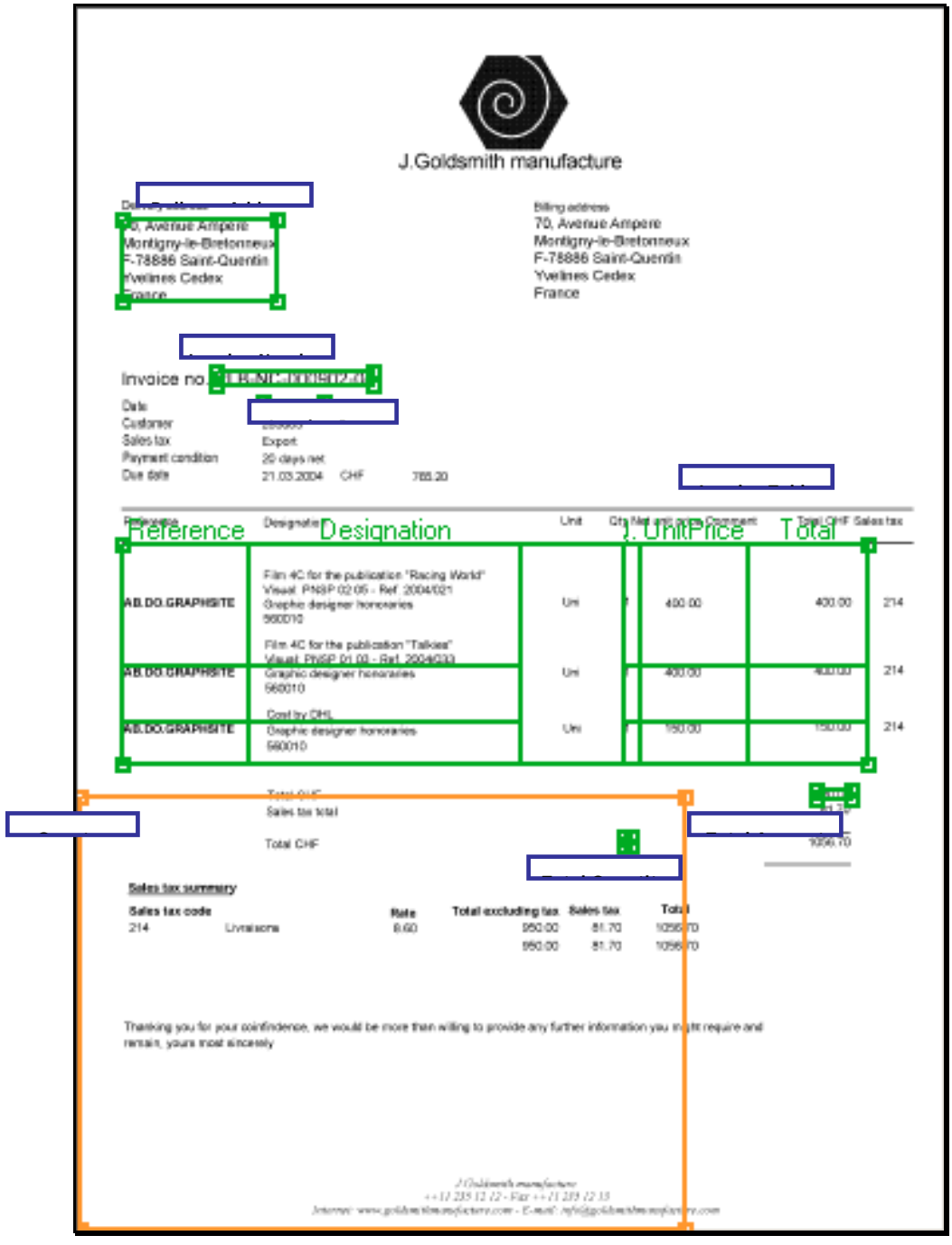
Thanking you for your coinfindence, we would be more than willing to provide any further information you might require and remain, yours most sincerely

J.Goldsmith manufacture
++11 235 12 12 - Fax ++11 235 12 13
Internet: www.goldsmithmanufacture.com - E-mail: info@goldsmithmanufacture.com

The purpose of a FlexiLayout for this document is to ensure reliable detection of the following fields on all the test images (provided they occur on a document):

- **Invoice Number**

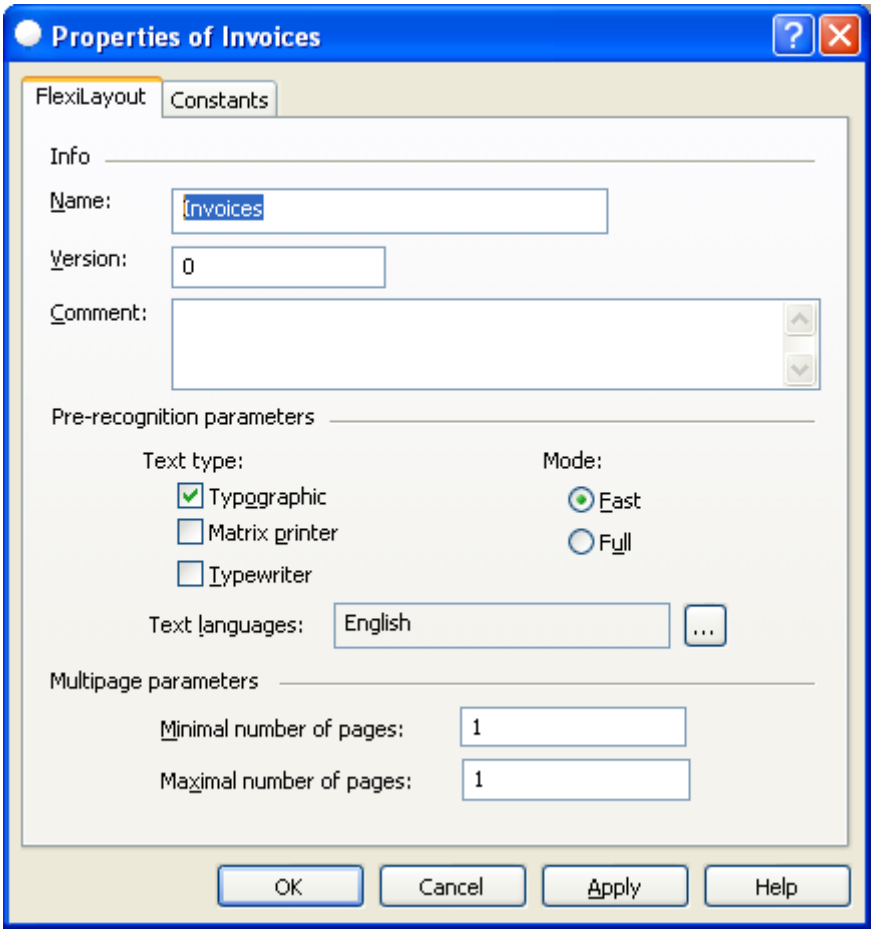
- **Invoice Date**
- **Delivery Address**
- **Total Quantity**
- **Total Amount**
- **Country**
- **Invoice Table** (only the following columns: **Reference**, **Designation**, **Quantity**, **Unit Price**, **Total**)



Step 1. Preparatory settings

1. Create a new folder and name it **Invoices One-Page**.
2. Start ABBYY FlexiLayout Studio 8.0.

3. Create a new project and name it **Sample3** (for more details see Tutorial 1, Step 1).
4. Add images to the batch (for more details see Tutorial 1, Step 2).
   Note: Test images of Tutorial 3 are stored in **<disk name>:\\Documents and Settings\All Users\Application Data\ABBYY\FlexiCapture\8.0\Samples\ FlexiLayoutStudio\Invoice One-Page.**
5. Specify FlexiLayout properties (for more details see Tutorial 1, Step 3). Under **Text type** select **Typographic**, and in **Text languages** select **English**. Note that the program allows you to specify more than one language. Name the FlexiLayout **Invoice**. Do not change the default **Fast** pre–recognition mode, and set **Minimal number of pages** and **Maximal number of pages** to 1.



Note: See Tutorial 1, Step 3 for more about selecting the right pre–recognition parameters.

For one-page documents, you do not need to use predefined compound **Header** and **Footer** elements to indicate the beginning and end of the documents. Therefore, you can remove them from the list of FlexiLayout elements.

Note: You can add **Header** and **Footer** elements to your FlexiLayout to indicate the beginning and end of the documents. If neither the **Header** nor the **Footer** element is found when matching the FlexiLayout, the program will use the maximum number of pages specified in the FlexiLayout as the number of pages in the document. For this particular document we specified 1 as the maximum number of pages.

Step 2. Visual analysis of images and pre–recognition results

Prior to creating elements, we must decide which objects on the form can serve to detect the data fields. These objects can be selected only from such graphical and/or text fragments of the image that are reliably detected during the pre–recognition process.
Start pre–recognition by clicking **Prerecognize** (see Tutorial 1, Step 4 for details).
Reviewing the pre–recognition results reveals that all the text objects corresponding to the titles of the data fields (if a field has a title), names of columns, and horizontal separators have been reliably detected. So they may be used to search for other objects on the images.
Note: For more information on reviewing images and pre–recognition results, see Tutorial 1, Step 5 and Step 6.
Now we can start to create blocks and elements. To do this, activate the **FlexiLayout** window in the main window of ABBYY FlexiLayout Studio.

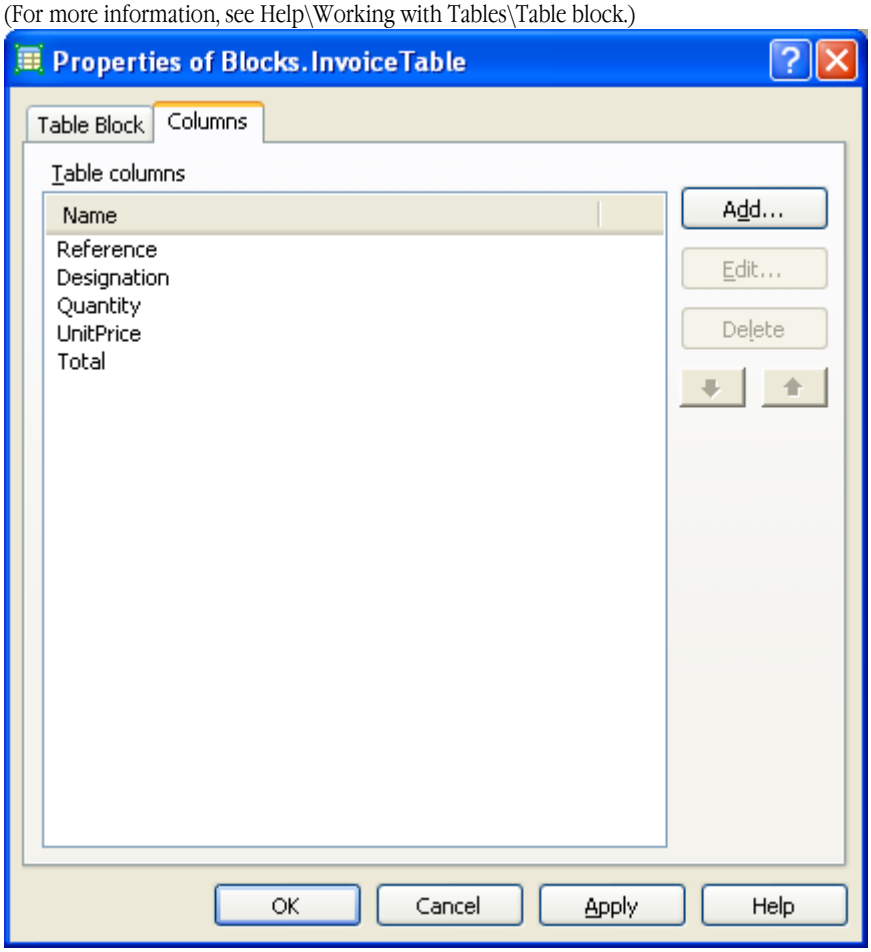
Step 3. Blocks

The main goal of the FlexiLayout creation is a reliable detection of the following blocks (provided they occur on an image):

- Invoice Number
- Invoice Date
- Delivery Address
- Total Quantity
- Total Amount
- Country
- Invoice Table (only the following columns: Reference, Designation, Quantity, Unit Price, Total)

Let us start to create these blocks in order to have their list during FlexiLayout creation.

1. Create a block **InvoiceNumber** of the Text type, which will correspond to the field **Invoice Number**.
2. Create a block **InvoiceDate** of the Text_type, which will correspond to the field **Invoice Date**.
3. Create a block **DeliveryAddress** of the Text_type, which will correspond to the field **Delivery Address**.
4. Create a block **TotalQuantity** of the Text_type, which will correspond to the field **Total Quantity**.
5. Create a block **TotalAmount** of the Text_type, which will correspond to the field **Total Amount**.
6. Create a block **Country** of the Text_type, which will correspond to the field **Country**.
7. Create a block **InvoiceTable** of the Text_type, which will correspond to the field **Invoice Table**. In the **Properties** dialog box of the block, click the **Columns** tab, enter the names of the columns and specify their order, which will later be used in a FlexiCapture template. Add the column names in the following order: **Reference**, **Designation**, **Quantity**, **UnitPrice**, and **Total**. (For more information, see Help\Working with Tables\Table block.)



Note: At this stage we only specify the type of a block and its name (and the order of columns for the Table block). We will specify the location of each block later, when all auxiliary elements have been described and debugged. It is very important that the block of the Table_type be created before the element of the Table type which is used to look for the block, as the Table element must keep a reference to the Table block. This requirement makes Table_blocks different from blocks of the Text, Barcode, Checkmark, and Picture types, which can be created after modifying a FlexiLayout, when all the elements for different blocks have been created (for instance see Tutorial 1, Tutorial 2).

Now that the preparatory settings have been made, we can start to create elements.

Step 4. Analyzing the images to decide in which order elements must be detected

At this step, we are to decide:

- Is there any pattern in the arrangement of the fields on the images?
- Which elements can be used as reference elements for fields detection?
- What sequence of search for elements is the best choice? (At each new step we can only refer to the above described elements.)

Let us analyze the available images.
You will have noticed that the central part of the document contains a table which occurs on all the images.
Note: Please note that in FlexiLayout Studio 8.0, a table is an image object made up of fragments that consist of rows and columns visually separated by separator lines or by white gaps.
Above the table there is a group of fields (which may be named the InvoiceHeader): **Invoice Number**, **Invoice Date**, and **Delivery Address**. Note that the field **Invoice Number** occurs on all the images and can be used as a identifier field, while the fields **Invoice Date** and **Delivery Address** are optional. Below the table there is a group of fields (which may be named the Footer): **TotalQuantity, TotalAmount** and **Country.** These fields also do not occur on all the images.
We will start to create our search for elements in the upper part of the document. To detect the upper fields, we will create a logical group uniting all the elements which are used to look for the fields **Invoice Number**, **Invoice Date**, and **Delivery Address**.

1. Create an element of the Group type and name it **InvoiceHeader**.
   The fields **Invoice Number**, **Invoice Date**, and **Delivery Address** are always located in the upper left corner. Moreover, their order is always the same: **Delivery Address** followed by **Invoice Number** followed by **Invoice Date** (provided they occur on the image). We will look for them in the same order.
   To describe search of the keywords for the titles of the fields **Invoice Number**, **Invoice Date**, and **Delivery Address**, we will use elements of the Static Text_type. The **InvoiceHeader** element must contain the following elements:
2. **kwDeliveryAddress** element of the Static Text type, which will correspond to the title of the field **Delivery Address** (for detailed instructions, see Step 5);
3. **kwInvoiceNumber** element of the Static Text type, which will correspond to the title of the field **Invoice Number** (for detailed instructions, see Step 6);
4. **kwInvoiceDate** element of the Static Text type, which will correspond to the title of the field **Invoice Date** (for detailed instructions, see Step 7).
   As for the fields **Number** and **Invoice Date**, we are going to search for them in the same row as their corresponding titles, to the right of the titles. In the **InvoiceHeader** element, create the following elements:
5. **InvoiceNumber** element of the Character String_type, which will correspond to the field **Invoice Number** (for detailed instructions, see Step 8);
6. a Group element **grDate** of the Static Text type to look for the field **Invoice Date** (for detailed instructions, see Step 9).
   - In the **InvoiceHeader.grDate** element, create:
7. **InvoiceDate** element of the Date type, which will correspond to the **Invoice Date** field in the case of good quality images (for detailed instructions, see Step 9);
8. **InvoiceDateAsString** element of the Character String_type, which will correspond to the **Invoice Date** field if the program fails to find the **InvoiceDate** element (for detailed instructions, see Step 9).
   Note: For more information on finding dates on poor quality images, for detailed instructions, see the **Tips and Tricks** section of the Help file.
   The field **Delivery Address** has multiple lines, so we need an element of the Text Fragment_type to detect it. However, prior to creating this element, we should limit the search area as much as possible. For the right border of the search area, we will use element of the White Gap_type. Afterwards, we will group all the elements which describe the location of the field **Delivery Address** into a Group element:
9. In the element **InvoiceHeader**, create an element of the Group type and name it **grAddress** (for detailed instructions, see Step 10).
   In the **grAddress** element, create the following elements:
10. an auxiliary element **wgAddressRight** of the White Gap type, bordering on the right the field **Delivery Address** (for detailed instructions, see Step 11);
11. an element **DeliveryAddress** of the Text Fragment type, which will correspond to the field **Delivery Address** (for detailed instructions, see Step 12).

Step 5. Detecting the title of the Delivery Address field: the kwDeliveryAddress element

Analysis of the test images reveals that the string “DeliveryAddress” may be used as the keywords for the **Delivery Address** field title. Set the maximum percentage of errors to 20 per cent, i.e. in case of 15 letters string mean no more than 3 errors in the entire string.

Note: The case of the letters does not affect the search and is only used to make the text more readable. The maximum number of errors is determined by method of trial and error. For more information about the parameters of a Static Text element, see **Help/Elements/Element Properties/Static Text**.

To create the **kwDeliveryAddress** element:

1. In the **InvoiceHeader** element, create an element of the Static Text type and name it **kwDeliveryAddress**.
2. Click the **Static Text** tab.
3. In the **Search Text** field, enter the text to find: "**DeliveryAddress**".
4. Set the Max error percentage to 20.
5. Click the **Advanced** tab.
6. Since the **Delivery Address** field is in the upper third and in the left third of the image, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object in the left third of the image. Look for the object in the upper third of the image*. In the FlexiLayout Language:
   Above: PageRect.Top + PageRect.Height/3;
   LeftOf: PageRect.Left + PageRect.Width/3;
7. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.

Note: To match the FlexiLayout with all the test images, you can use the hotkeys:

- Activate the **Batch** window
- Press **Ctrl** + **A** for **Select All** pages
- Press **Ctrl** + **Shift** + **E** for **Analyze Selected**

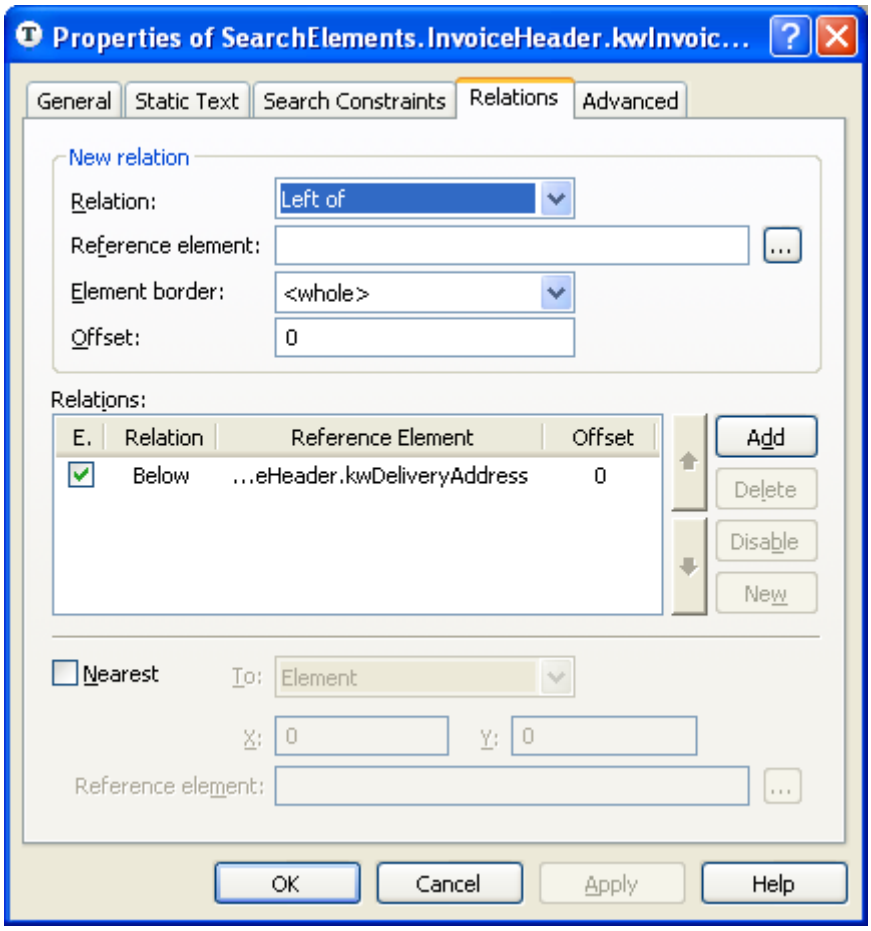
Step 6. Detecting the title of the Invoice Number field: the kwInvoiceNumber element

Analysis of the images reveals that the title of the field **Invoice Number** occurs on all the images and distinguishes them from other document types. When processing semi-structured forms in ABBYY FlexiCapture, you will want to avoid accidental recognition of forms which do not belong to the selected type. One way to identify a form is to create at least one required element. So we will mark this element as required, i.e. use it as an identifier for this type of form.

Note that on some of the images this title is written as **Invoice No.,** while on others it is written as **Credit note no**. Our search string must include both these variants, and we will specify each of the strings without spaces. In this case the maximum number of errors will apply to the entire phrase, not to separate words.

To create the **kwInvoiceNumber** element:

1. In the **InvoiceHeader** element, create an element of the Static Text type and name it **kwInvoiceNumber**.
2. Clear the **Optional** check box on the **General** tab (we have decided to make this element a required element).
3. Click the **Static Text** tab.
4. In the **Search Text** field, enter the text to find: "**InvoiceNo.|CreditNoteNo**".
5. Set Max error percentage to 20.
6. Click the **Relations** tab.
7. On the **Relations** tab, specify that the keywords of the title of the **Invoice Number** field can only occur below the title of the field **Delivery Address**:
   Below the element **InvoiceHeader.kwDeliveryAddress**, **Offset** = 0.



8. Click the **Advanced** tab.
9. Since the **Invoice Number** field is in the upper third and in the left third of the image, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object in the left third of the image; Look for the object in the upper third of the image*. In the FlexiLayout Language:
   Above: PageRect.Top + PageRect.Height/3;
   LeftOf: PageRect.Left + PageRect.Width/3;
10. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images.

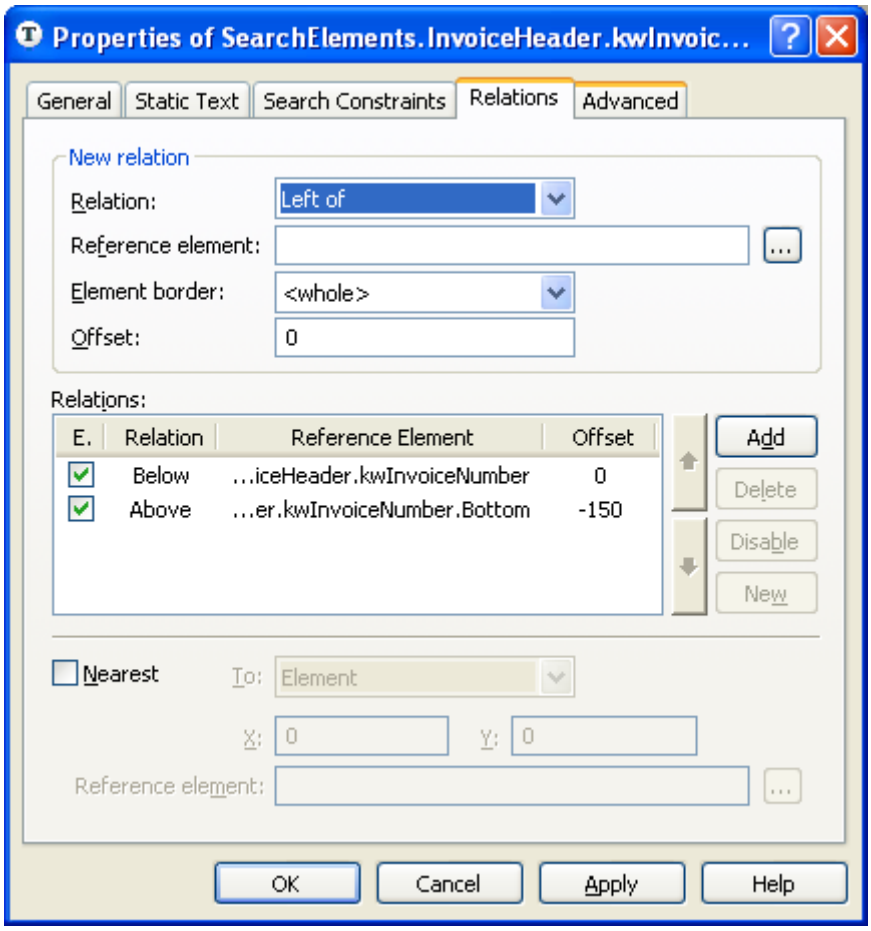
Note: In practice, if the wording of a field title differs from image to image, you must be sure to specify all the possible variants of the title, or at least the majority of them. You may want to ask for more test images or contact your customer to learn about the possible variants of the title on the forms of this type.

Step 7. Detecting the title of the Invoice Date field: the kwInvoiceDate element

Analysis of the test images reveals that the title of the field **Invoice Date** is a short and very common word "Date." This means that we have to restrict the search area as much as possible. You will have noticed that the title of the **Date** field (whenever it occurs on an image) is located very close to the title of the field **Invoice Number**.

To create the **kwInvoiceDate** element:

1. In the **InvoiceHeader** element, create an element of the Static Text type and name it **kwInvoiceDate**.
2. Click the **Static Text** tab.
3. In the **Search Text** field, enter the text to find: "**Date**".
4. Leave 30 as the value for the **Max error percentage** (for the sought word of 4 letters it means no more than 1 error).
5. Click the **Relations** tab.
6. On the **Relations** tab, specify that the keywords of the title can only occur below the title of the field **Invoice Number,** but no more than 150 dots below its bottom border.
   Below the element **kwInvoiceNumber**, **Offset** = 0, and
   Above the **bottom** of the element **kwInvoiceNumber**, **Offset** = –150, **Element border** = **Bottom**.



Note: To select the best values for offsets, you may want to analyze the geometrical properties (i.e. size and borders) of the objects detected during pre–recognition.

7. Click the **Advanced** tab.
8. Since the **Invoice Date** field is in the left half of the image, set the following additional search constraint in the **Advanced pre–search relations** field: *Look for the object in the left half of the image*. In the FlexiLayout Language:
   LeftOf: PageRect.Left + PageRect.Width/2;
9. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.

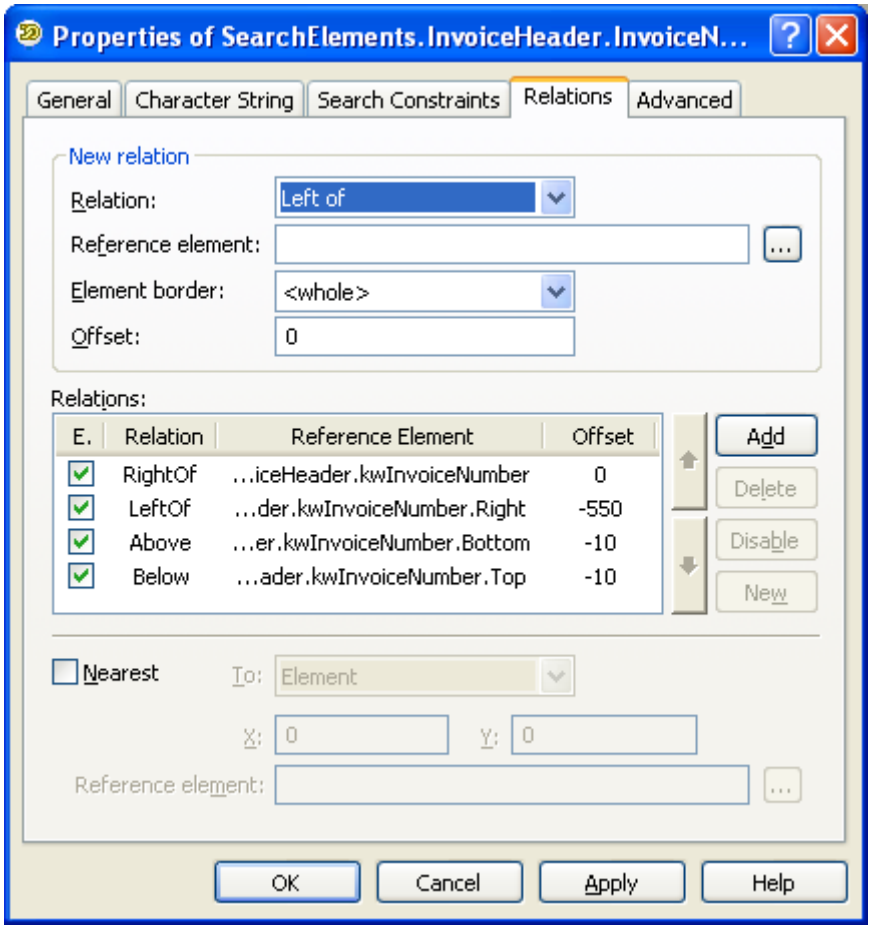
Step 8. Describing the Invoice Number field: the InvoiceNumber element

The field **Invoice Number** is to the right of the title and is located on the same horizontal level. Analysis of the test images reveals that on some of the images the number has been poorly recognized. So we will not restrict the alphabet, but we will restrict the search area and restrict the length of the string to 3–20 characters.

To create the **InvoiceNumber** element:

1. In the **InvoiceHeader** element, create an element of the Character String type and name it **InvoiceNumber**.
   Note: The element and the block which corresponds to the field **InvoiceNumber** need not have identical names, but this is handy when working with the FlexiLayout.
2. Click the **Character String** tab.
3. Set the **Percentage of non–alphabet characters** to 100%, which means that there is no restriction for characters.
4. In the **String Length** field, specify this fuzzy interval: [–1, 3, 20, INF], which restricts the number of characters in the string. We assume that the possible values range from 3 to 20. Any hypotheses outside this range will be penalized.
5. Click the **Relations** tab.
6. Since **InvoiceNumber** can only occur to the right of the keywords of the title **kwInvoiceNumber**, but no farther than 550 dots from the right border of the title **kwInvoiceNumber**, set the following search constraints:
   – **RightOf** the element **kwInvoiceNumber**, **Offset** = 0
   – **LeftOf** the element **kwInvoiceNumber**, **Offset** = –550, **Element border** = Right
   Note: Offset values are selected by method of trial and error.

7. Since **InvoiceNumber** is always located on the same level with the keyword of the title **kwInvoiceNumber**, set the following search constraints:
   – **Above** the element **kwInvoiceNumber**, **Offset** = –10, **Element border** = Bottom
   – **Below** the element **kwInvoiceNumber**, **Offset** = –10, **Element border** = Top



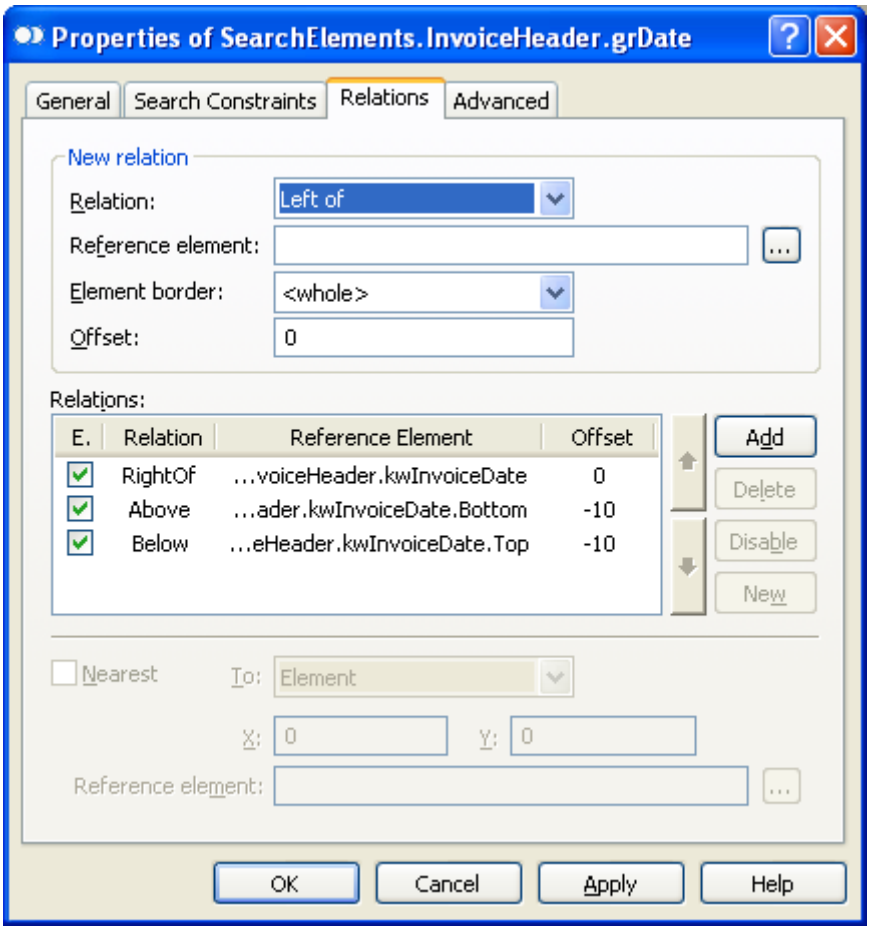
8. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.
9. To describe the location of the **InvoiceNumber** block, click its **Properties** and select the **Source element** option, then click "**...**" and select the **InvoiceNumber** element as the source element.

Step 9. Describing the Invoice Date field: the grDate, InvoiceDate, and InvoiceDateAsString elements

Analysis of the test images reveals that the **Invoice Date** field is located to the right of the title and on the same horizontal level as the title. We will use a Date element to look for the date. In case there are poor quality images where dates may not be recognized properly so that the search for the **Date** gives no results, create an additional element of the **Character String** type. To specify the properties common to all these elements, we need to create a compound **grDate** element.
To create the **grDate** element:

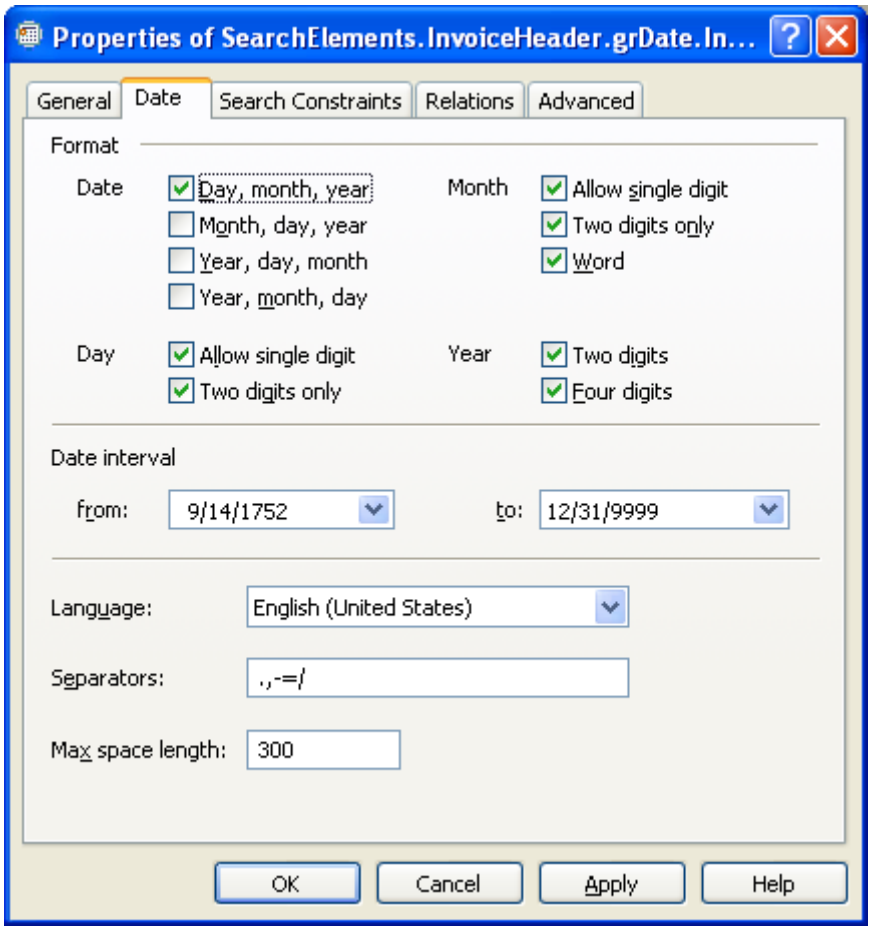
1. In the **InvoiceHeader** element, create an element of the Group type and name it **grDate**.
2. Click the **Relations** tab.
3. Since all the elements used to detect the **Invoice Date** field may only occur to the right of the keywords of the **kwInvoiceDate** title and on the same level as the title, set the following search constraints:

– **RightOf** the element **kwInvoiceDate**, **Offset** = 0
– **Above** the element **kwInvoiceDate**, **Offset** = –10, **Element border** = **Bottom**
– **Below** the element **kwInvoiceDate**, **Offset** = –10, **Element border** = **Top**



To create the **InvoiceDate** element:

1. In the **InvoiceHeader.grDate** element, create an element of the **Date** type and name it **InvoiceDate**.
2. Click the **Date** tab.
3. On the **Date** tab, specify all the possible date formats which may occur on the images. We assume that the date format is **Day, Month, Year**, the language is **English**, and there are no other constraints on the date.



4. Click the **Advanced** tab.
5. Since the **Invoice Date** field is optional on this form but whenever it occurs, it occurs with its title and is always located next to the title, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object on the image only if the **kwInvoiceDate** element has been detected; Look for the image object closest to the **kwInvoiceDate** element.* In the FlexiLayout Language:
   If InvoiceHeader.kwInvoiceDate.IsNull Then DontFind;
   Nearest: InvoiceHeader.kwInvoiceDate;

   Note: The **Nearest** constraint can also be specified via the program's graphical user interface. Click the **Relations** tab, select **Nearest**, and in the **To:** drop-down list select **Element** and specify **kwInvoiceDate** as the **Reference element**.

6. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where there is an **Invoice Date** field.

To create the **InvoiceDateAsString** element:

1. In the **InvoiceHeader.grDate** element, create an element of the Character String type and name it **InvoiceDateAsString**.
2. Click the **Character String** tab.
3. In the **String Length** field, specify this fuzzy interval: [–1, 8, 14, INF], which is an estimate of the string length. We assume that the string may be 8 to 14 characters long. Any hypothesis outside this range will be penalized.
4. Set the **Max space length** to 20, which means that the space may be 20 dots long.
5. For the other properties, leave the default values.
6. Click the **Advanced** tab.
7. Since we want the program to look for this element only if the **InvocieDate** element has not been detected, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object only if the **InvocieDate** element has not been detected.* In the FlexiLayout Language:
   If Not InvoiceDate.IsNull Then DontFind;
8. Since the **Invoice Date** field is optional on this form but whenever it occurs, it is always present with its title and located next to the title, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object on the image only if the element **kwInvoiceDate** has been detected; Look for the image object closest to the **kwInvoiceDate** element.* In the FlexiLayout Language:
   If InvoiceHeader.kwInvoiceDate.IsNull Then DontFind;
   Nearest: InvoiceHeader.kwInvoiceDate;

Note: The **Nearest** constraint can also be specified via the program's graphical user interface. Click the **Relations** tab, select **Nearest**, and in the **To:** drop-down list select **Element** and specify **kwInvoiceDate** as the **Reference element**.

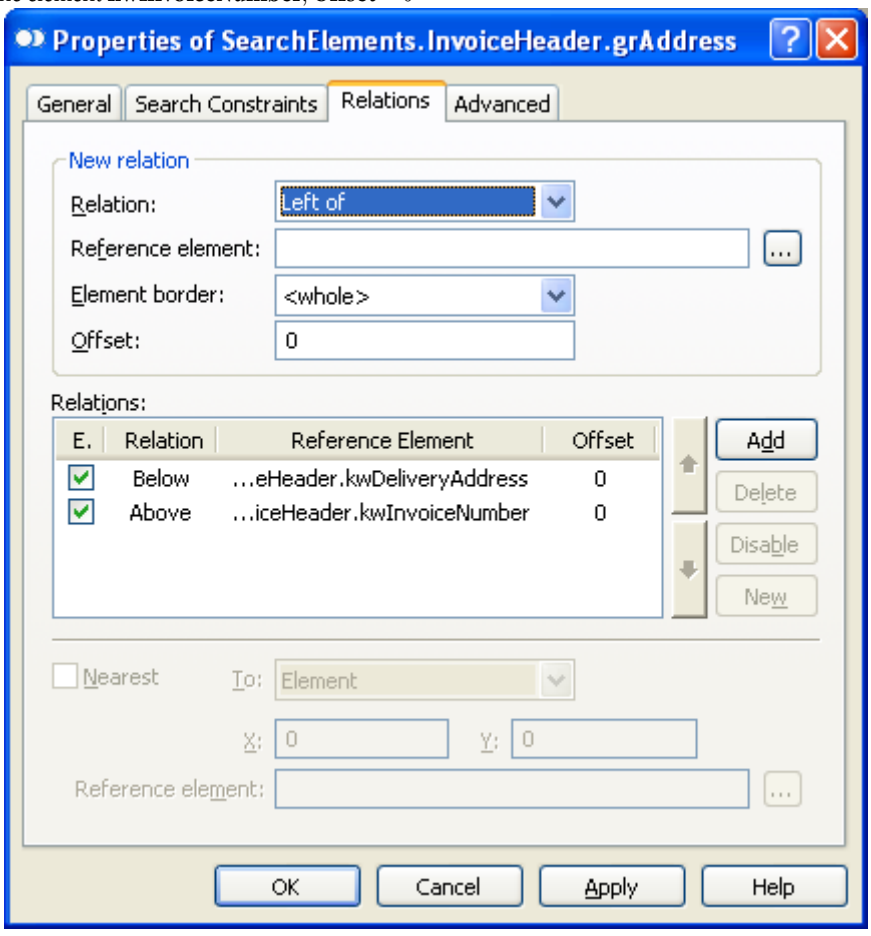
9. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where there is an **Invoice Date** field and generates a null hypothesis for the **InvoiceDate** element.
10. To describe the location of the **InvoiceDate** block, select the **Source element** option, then click **"..."** and select the **InvoiceHeader.grDate** element as the source element.

Note: The region of a Group element is calculated by uniting the regions of all the detected sub–elements.

Step 10. Describing the grAddress element of the Group type

We have decided to look for the **Delivery Address** field by using the following two elements: a White Gap element on the right, and an element of the Text Fragment type. To specify the properties common to all these elements, we will create a Group element.
To create the **grAddress** element:

1. In the **InvoiceHeader** element, create an element of the Group type and name it **grAddress**.
2. Click the **Relations** tab.
3. Since all the elements used to search for the **Delivery Address** field can only occur below the keywords of the title **kwDeliveryAddress** and above the keywords of the title **kwInvoiceNumber**, set the following search constraints:
   – **Below** the element **kwDeliveryAddress**, **Offset** = 0
   – **Above** the element **kwInvoiceNumber**, **Offset** = 0



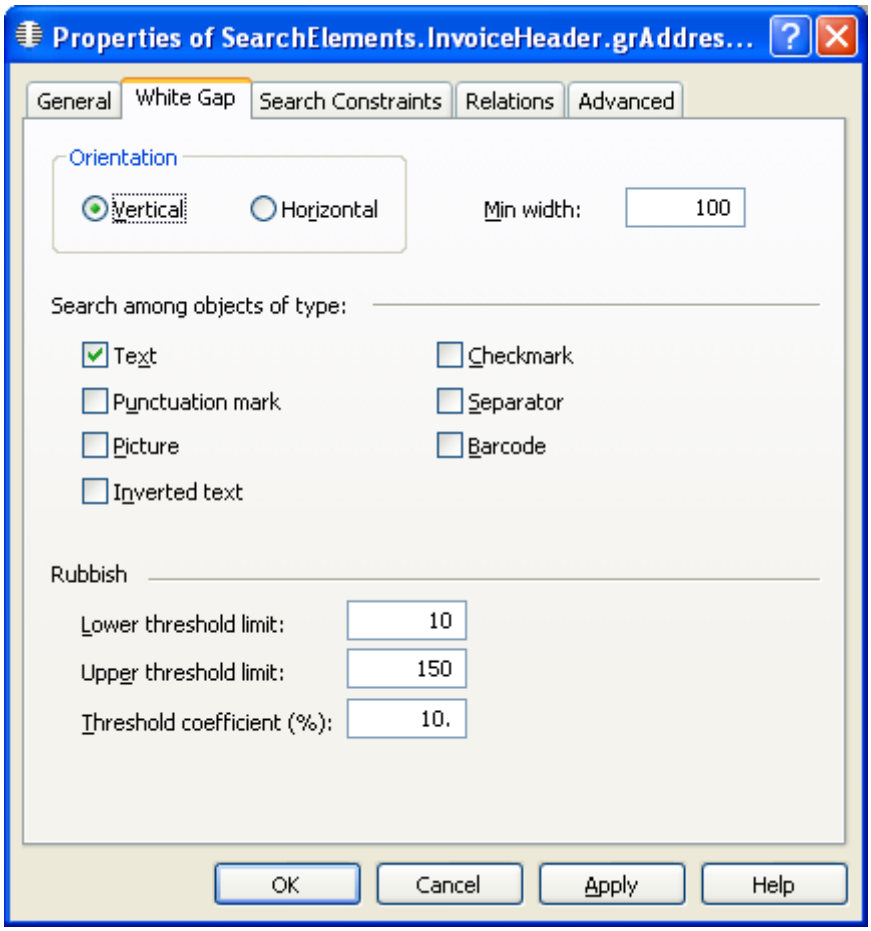
4. Click the **Advanced** tab.
5. Since the **Delivery Address** field and the white gap bordering it are located in the left two fifths of the page, set the following additional search constraint in the **Advanced pre–search relations** field: *Look for the objects in the left two fifths of the page.* In the FlexiLayout Language:
   LeftOf: PageRect.Left + PageRect.Width*2/5;

Step 11. Detecting the right border of the Delivery Address field: the wgAddressRight element

To specify the right border of the search area of the **DeliveryAddress** element, we need to describe the corresponding vertical white gap.

To create the **wgAddressRight** element:

1. In the **InvoiceHeader.grAddress** element, create an element of the White Gap type and name it **wgAddressRight**.
2. Click the **White Gap** tab.
3. In the **Orientation** group, choose **Vertical**. Set **Min Width** to 100.
   Note: Prior to setting values for height or width of white gaps, you may want to analyze the geometrical properties (i.e. size and borders) of the neighboring objects detected during pre-recognition.
4. Set **Lower threshold limit:** and **Threshold coefficient (%):** to 10.
   Note: The values of **Lower threshold limit**, **Upper threshold limit**, and **Threshold coefficient (%)**are selected by method of trial and error.
   For more information, see **Help/Elements/Element Properties/ White Gap**.
5. Leave the default values for the other parameters of the element.



6. Click the **Relations** tab.
7. Since the sought white gap can only be located to the right of the keywords of the title **kwDeliveryAddress**, set the following search constraint:
   – **RightOf** the element **kwDeliveryAddress**, **Offset** = 0
8. Click the **Advanced** tab.
9. Since the **Delivery Address** field is optional on the images but whenever it occurs it has a title, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object only if the **kwDeliveryAddress** element has been detected.* In the FlexiLayout Language:
   If InvoiceHeader.kwDeliveryAddress.IsNull Then DontFind;
10. Since the sought white gap is adjacent to the **Invoice Number** field, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object on the image which is nearest to the **kwInvoiceNumber** element.* In the FlexiLayout Language:
    Nearest: InvoiceHeader.kwInvoiceNumber;

Note: The **Nearest** constraint can also be specified via the program's graphical user interface. Click the **Relations** tab, select **Nearest**, and in the **To:** drop-down list select **Element** and specify **kwInvoiceNumber** as the **Reference** element.

11. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where the **Delivery Address** field occurs.

Step 12. Describing the Delivery Address field: the DeliveryAddress element

To create the **DeliveryAddress** element:

1. In the **InvoiceHeader**.**grAddress** element, create an element of the Text Fragment type and name it **DeliveryAddress**.
2. Click the **Relations** tab.
3. Since the **DeliveryAddress** field can only occur above the title **kwInvoiceNumber**, only to the left of the **wgAddressRight** white gap, and only to the right of the beginning of the title **kwDeliveryAddress** (with small offset), set the following search constraints:
   – **Above** the element **wgAddressBottom**, **Offset** = 0
   – **LeftOf** the element **wgAddressRight**, **Offset** = 0
   – **RightOf** the element **kwDeliveryAddress**, **Offset** = –50, **Element border** = Left
   Note: Offset values are selected by method of trial and error.
4. Since in the Group element **InvoiceHeader**.**grAddress** we have already specified that the **DeliveryAddress** field can only be located below the keywords of the title **kwDeliveryAddress**, we need not duplicate this constraint here.
5. Click the **Advanced** tab.
6. Since the **Delivery Address** field is optional on the images but whenever it occurs it has a title, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object only if the* ***kwDeliveryAddress*** *element has been detected.* In the FlexiLayout Language:
   If InvoiceHeader.kwDeliveryAddress.IsNull Then DontFind;
7. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.
8. To describe the location of the **DeliveryAddress** block, select the **Source element** option, then click **"..."** and select the **DeliveryAddress** element as the source element.

Step 13. Further analysis of the images

So far we have described the **Invoice Number**, **Invoice Date**, and **DeliveryAddress** fields.
Let us now decide in which order the other elements must be detected.
Analysis of the arrangement of the fields on the test images reveals that to detect the **Country** field, we can use the title of this field and an element of the Character String type. But in order to detect the **Total Quantity** and **Total Amount** fields we will need to use not only the title, which is shared by the two fields, but also additional elements. The names of the corresponding columns of the table may be used for the purpose.
We also need to prepare the ground for the search of the **Invoice Table** field, which is to be detected by means of an element of the Table type.
First, we describe the Header of the table. There are two ways to detect column names. The first one is to specify one or several keywords in the **Column Properties** dialog box (the **Properties** dialog box of the Table element > **Columns** tab). This method is quick but not very flexible. The second way is to use a previously detected element as a reference element to find the location of the column title. This method allows you to take full advantage of the additional settings available for elements. Analysis of the pre-recognition results reveals that the recognition quality is good enough, so we can use the first method (the only snag would be the very short title of the **Quantity** column title — 20% of errors in a three-letter keyword "Qty" effectively means that no errors allowed). We will use the first method to detect the first two columns, **Reference** and **Designation**, and we can use the previously described elements to detect the titles of the rest of the columns: **Quantity**, **Unit Price**, and **Total** (as far as we need to create these elements anyway to detect the **TotalQuantity** and **TotalAmount** fields).
Secondly, we describe the Footer of the table. It can also be described either by using keywords or by using an auxiliary element. Analysis of the keywords in the footer reveals that on some of the images the keywords also occur in the first row of the table, so we have to use an auxiliary element because of means to restrict the search area.
Thirdly, we describe the table search area. After we have described the header and the footer, we need to describe the right border of the table (we do not need to describe the left border, as there are no other data in this part of the image). Since we cannot use the context to separate the figures in the **Total** column from the figures in the **Sales** column, we will need a different approach. We can use the name of the last column, **Sales**, to restrict the table search area on the right.
Note: Elements and the order in which they must be detected are selected by trial and error and can be changed during FlexiLayout adjustment.
Prior to describing the remaining fields, we will describe the auxiliary elements.
Analysis of interdependencies among the elements reveals that first we have to detect the names of the **Quantity**, **Unit Price**, **Total**, and **Sales** columns.

The header of the table begins with a horizontal separator between the column names and the **Invoice Number**, **Invoice Date**, and **Delivery Address** fields. This separator will help us to restrict the search areas of the column names.

1. For this purpose, create an element of the **Separator** type and name it **hsTableHeaderTop** (for detailed instructions, see Step 14).
   We will look for the column names in their natural order.
2. Create a Group element and name it **TableHeader** (for detailed instructions, see Step 15).
   This element must include:
3. element **kwQuantity** of the Static Text type, which will correspond to the name of the **Quantity** column of the table **InvoiceTable** (for detailed instructions, see Step 16);
4. element **kwUnitPrice** of the Static Text type, which will correspond to the name of the **Unit Price** column of the table **InvoiceTable** (for detailed instructions, see Step 17);
5. element **kwTotal** of the Static Text type, which will correspond to the name of the **Total** column of the table **InvoiceTable** (for detailed instructions, see Step 18);
6. element **kwSales** of the Static Text type, which will correspond to the name of the **Sales** column of the table **InvoiceTable** (for detailed instructions, see Step 19).
   Let's continue to describe the lower part of the form. The fields we are interested in are the title of the **Total Quantity** and the **Total Amount** fields (which serves as the bottom border of the table) and any other captions that can help locate the bottom border of the table. We will also describe the title element and the source element for the **Country** field in this logic group.
7. Create an element of the Group type and name it **Footer**.
   This element must include:
8. element **kwFooter** of the Static Text type, which will correspond to the footer of the **Invoice Table** (for detailed instructions, see Step 21);
9. element **kwTotal** of the Static Text type, which will correspond to the title of the **Total Quantity** and the **Total Amount** fields (for detailed instructions, see Step 22);
10. element **kwOrigin** of the Static Text type, which will correspond to the title of the **Country** field (for detailed instructions, see Step 23);
11. element **Country** of the Character String type, which will correspond to the **Country** field (for detailed instructions, see Step 24).
    Now that all the preparations have been made and the order of creating additional elements has bee described, we can start looking for the remaining fields. Relying on the previously detected titles of columns and Footer, we will describe the remaining **Total Quantity**, **Total Amount**, and **Invoice Table** fields.
12. For this purpose, create two elements of the Character String type and name them **TotalQuantity** and **TotalAmount**. These elements will correspond to the **Total Quantity** and the **Total Amount** fields respectively (for detailed instructions, see Step 25).
13. Create an element of the Table type and name it **InvoiceTable**. This element describes the **Invoice Table** field (for detailed instructions, see Step 26).

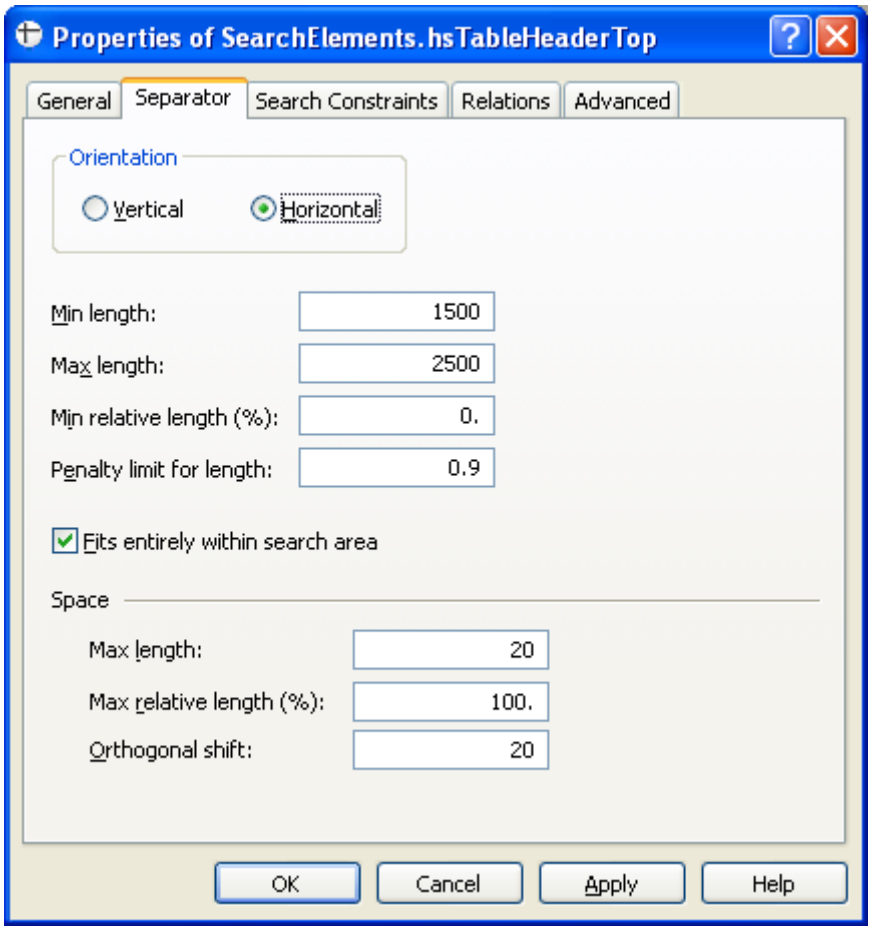
Step 14. Detecting the auxiliary horizontal separator: the hsTableHeaderTop element

To create the **hsTableHeaderTop** element:

1. Create an element of the Separator type and name it **hsTableHeaderTop**.
2. Click the **Separator** tab.
3. In the **Orientation** group, click the **Horizontal** option.
4. Set the **Min/Max length** to 1500 / 2500.
5. Click the Fit entirely within search area option.
6. Set the **Space**, **Max length** to 20 and the **Orthogonal shift** to 20.

Note: Prior to setting the values for the length of the separator, you may want to analyze the geometrical properties (i.e. size and borders) of the neighboring objects detected during pre–recognition. Separator error thresholds are selected by method of trial and error.

Leave the default values for the other parameters of the element.



7. Click the **Relations** tab.
8. Since the separator can only occur below the keywords of the titles **kwInvoiceDate** and **kwInvoiceNumber**, set the following search constraint:
   Below the element InvoiceHeader.kwInvoiceDate, Offset = 0;
   Below the element InvoiceHeader.kwInvoiceNumber, Offset = 0;
9. Click the **Advanced** tab.
10. Since the sought separator is located at the top of the described search area, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for an object on the image nearest to its top border.* In the FlexiLayout Language:
    NearestY: PageRect.Top;
    Note: The **Nearest** constraint can also be specified via the program's graphical user interface. Click the **Relations** tab, select **Nearest**, and in the **To:** drop-down list select **Page top edge**.
11. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images.

Step 15. Analyzing the search constraints for column names. The TableHeader element of the Group type

Analysis of the test images reveals that the column names are located directly below the previously detected horizontal separator. To specify the search constraints common to all the column names, we need to create a Group element — **TableHeader**.

To create the **TableHeader** element:

1. Create an element of the Group type and name it **TableHeader**.
2. Click the **Advanced** tab.
3. You will have noticed that if the **hsTableHeaderTop** separator has been detected on an image, the column names are located below the separator, but no lower than 100 dots from it. Otherwise, they are located in the top three fifths of the page. So set the following additional search constraint in the **Advanced pre-search relations** field: *If the **hsTableHeaderTop** element has been detected, look for the objects of the image below the separator but no lower than 100 dots from it. Otherwise, look for the objects of the image in the top three fifths of the page.* In the FlexiLayout Language:

If Not (hsTableHeaderTop.IsNull) Then

```
{
Below: hsTableHeaderTop, 0 * dot;
Above: hsTableHeaderTop.Bottom, –100 * dot;
} Else
{
Above: PageRect.Top + PageRect.Height*3/5;
}
```

Step 16. Detecting the name of the Quantity column: the kwQuantity element

Analysis of the pre–recognition results reveals that even though the title is a very short keyword "Qty," we can detect the title by using an element of the Static Text_type. Set the maximum error percentage to 35; this allows one wrong character in a three–letter word.

To create the **kwQuantity** element:

1. In the **TableHeader** element, create an element of the Static Text type and name it **kwQuantity**.
2. Click the **Static Text** tab.
3. In the **Search Text** field, enter the text to find: "**Qty**".
4. Set the Max error percentage to 35.
5. Click the **Advanced** tab.
6. Since the name of the **Quantity** column is located to the right of the leftmost third of the image, set the following additional search constraint in the **Advanced pre–search relations** field: *Look for the object to the right of the leftmost third of the page.* In the FlexiLayout Language:
   RightOf: PageRect.Left + PageRect.Width/3;
7. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images.

Step 17. Detecting the name of the Unit Price column: the kwUnitPrice element

To describe the element which will be used to detect the keywords in the **Unit Price** column name, we will take advantage of the fact that the **Unit Price** column is always located to the right of the **Quantity** column.

To create the **kwUnitPrice** element:

1. In the **TableHeader** element, create an element of the Static Text type and name it **kwUnitPrice**.
2. Click the **Static Text** tab.
3. In the **Search Text** field, enter the text to find: "**UnitPrice**".
4. Click the **Advanced** tab.
5. Since the name of the **Unit Price** column is located to the right of the name of the **Quantity** column, set the following additional search constraint in the **Advanced pre–search relations** field: *If the **kwQuantity** element has been detected, look for the **kwUnitPrice** element to the right of it.* In the FlexiLayout Language:
   If Not (TableHeader.kwQuantity.IsNull) Then
   RightOf: TableHeader.kwQuantity, 0 * dot;
6. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images.

Step 18. Detecting the name of the Total column: the kwTotal element

To describe the element which will be used to look for the keywords in the name of the **Total** column, we will take advantage of the fact that the **Total** column is always located to the right of the **Unit Price** column.

To create the **kwTotal** element:

1. In the **TableHeader** element, create an element of the Static Text type and name it **kwTotal**.
2. Click the **Static Text** tab.
3. In the **Search Text** field, enter the text to find: "**TotalChf**".
4. Click the **Advanced** tab.

5. Since the name of the **Total** column is located to the right of the name of the **Unit Price** column (if the latter occurs on the form), set the following additional search constraint in the **Advanced pre-search relations** field: *If the **kwUnitPrice** element has been detected, look for the **kwTotal** element to the right of it.* In the FlexiLayout Language:
   If Not (TableHeader.kwUnitPrice.IsNull) Then
   RightOf: TableHeader.kwUnitPrice, 0 * dot;
6. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images.

Step 19. Detecting the name of the Sales column: the kwSales element

Though we do not need the data from the **Sales** column, we have to detect its name in order to restrict the search area of the right border of the table.

To create the **kwSales** element:
1. In the **TableHeader** element, create an element of the Static Text type and name it **kwSales**.
2. Click the **Static Text** tab.
3. In the **Search Text** field, specify the value for the sought text: "**Sales**".
4. Click the **Advanced** tab.
5. Since the name of the **Sales** column is located to the right of the name of the **Total** column (if the latter occurs on the form), set the following additional search constraint in the **Advanced pre-search relations** field: *If the **kwTotal** element has been detected, look for the **kwSales** element to the right of it.* In the FlexiLayout Language:
   If Not (TableHeader.kwTotal.IsNull) Then
   RightOf: TableHeader.kwTotal, 0 * dot;
6. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images.

Step 20. Describing the Footer group: the Footer element of the Group type

Analysis of the test images reveals that the footer of the table, the **Country**, **Total Quantity**, and **Total Amount** fields and their titles are located in the lower part of the image, under the table header. To keep the first row of the table, where the keywords for the footer may also occur, out of the search area, we will lower the top border of the search area. To specify some common search constraints, we need to create a Group element.

To create the **Footer** element:
1. Create an element of the Group type and name it **Footer**.
2. Click the **Relations** tab.
3. Since the footer of the **Invoice Table** and the **Country**, **Total Quantity**, and **Total Amount** fields can only be located below the separator **hsTableHeaderTop** with some offset, set the following search constraint:
   – **Below** the element **hsTableHeaderTop**, **Offset** = 300

Step 21. Describing the footer of the table: the kwFooter element

Analysis of the test images reveals that the footer of the table can begin with the keywords "Carried Over," "Total Chf," "Goods total excl. tax," "Discount," and "SPECIAL DISCOUNT." Let us list all these variants in a search text for the element of the Static Text type**.**

To create the **kwFooter** element:
1. In the **Footer** element, create an element of the Static Text type and name it **kwFooter**.
2. Click the **Static Text** tab.
3. In the Search Text field, enter the text to find: "Discount|TotalCHF|SPECIALDISCOUNT|Carriedover|Goodstotalexcl.tax".
4. Click the **Advanced** tab.
5. The sought keywords are located in the left half of the image. If the program detects several of the keywords, we will only be interested in the topmost keyword. So set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object in the left half of the page; Look for the object nearest to the top border of the page.* In the FlexiLayout Language:
   LeftOf: PageRect.Left + PageRect.Width/2;
   NearestY: PageRect.Top;

6. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.

Note: If the wording of a field title may differ from image to image, be sure to list all the possible variants of the title, or at least the majority of them. You may want to ask for more test images or contact your customer to learn about the possible variants of field titles on the forms of this type.

Step 22. Describing the title of the Total fields: the kwTotal element

You will have noticed that the title of the **Total Quantity** and **Total Amount** fields may be represented by the keywords "TotalChf" or "Carried Over" and is located in the lower left part of the page.

To create the **kwTotal** element:

1. In the **Footer** element, create an element of the Static Text type and name it **kwTotal**.
2. Click the **Static Text** tab.
3. In the **Search Text** field, specify the values for the sought text: "**TotalChf|CarriedOver**".
4. Set Max error percentage to 20.
5. Click the **Advanced** tab.
6. Since the title is located in the left half of the page and the sought string is nearest to the top border of the page, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object in the left half of the page; Look for the object nearest to the top border of the page.* In the FlexiLayout Language:
   LeftOf: PageRect.Left + PageRect.Width/2;
   NearestY: PageRect.Top;
7. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.

Step 23. Detecting the title of the Country field: the kwOrigin element

We will use an element of the Static Text type and the keyword "OriginOfGoods:" to detect the title of the **Country** field.

To create the **kwOrigin** element:

1. In the **Footer** element, create an element of the Static Text type and name it **kwOrigin**.
2. Click the **Static Text** tab.
3. In the **Search Text** field, enter the text to find: "**OriginOfGoods:".**
4. Click the **Relations** tab.
5. Since the **Country** field can only be located below the footer of the table, set the following search constraint:
   – **Below** the element **Footer.kwFooter**, **Offset** = 0
6. Click the **Advanced** tab.
7. Since the **Country** field is located in the left half of the image, set the following additional search constraint in the **Advanced pre-search relations** field: *Look for the object in the left half of the page.* In the FlexiLayout Language:
   LeftOf: PageRect.Left + PageRect.Width/2;
8. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.

Step 24. Describing the Country field: the Country element

We will use an element of the Character String type to detect the **Country** field. The element must be located on the same horizontal level with the title of this field and will be used only if the title has been detected.

To create the **Country** element:

1. In the **Footer** element, create an element of the Character String type and name it **Country**.
2. Click the **Character String** tab.
3. Select all Latin characters as the alphabet and set the **Percentage of non-alphabet characters** to 20.
4. In the **String Length** field, specify this fuzzy interval: { -1, 2, 25, INF }. This is an estimation of the length of the character string. We assume that the possible values may range from 2 to 25. Any hypotheses outside this range will be penalized.
5. Leave the default values for the other parameters of the element.
6. Click the **Relations** tab.
7. Since the **Country** field can only be located to the right of the keywords of the **kwOrigin** title but no farther than 500 dots from its right border, set the following search constraints:

– **RightOf** the element **kwOrigin**, **Offset** = 10
– **LeftOf** the element **kwOrigin**, **Offset** = –500, **Element border** = Right

8. Since the **Country** element is always located on the same level with the keywords of the **kwOrigin** title, set the following search constraints:
– **Above** the element **kwOrigin**, **Offset** = –10, **Element border** = Bottom
– **Below** the element **kwOrigin**, **Offset** = –10, **Element border** = Top
Note: Prior to setting the values for the offsets, you may want to analyze the geometrical properties (i.e. size and borders) of the neighboring objects detected during pre–recognition. Offset values are selected by trial and error.
9. Click the **Advanced** tab.
10. The **Country** field is optional on this form but whenever it occurs, it is always located next to its title. So set the following additional search constraint in the **Advanced pre–search relations** field: *Look for the object on the image only if the **kwOrigin** element has been detected.* In the FlexiLayout Language:
If Footer.kwOrigin.IsNull Then DontFind;
11. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.
12. To describe the location of the **Country** block, select the **Source element** option, then click **"..."** and select the **Country** element as the source element.

Step 25. Detecting the TotalQuantity and TotalAmount fields: the TotalQuantity and TotalAmount elements

Analysis of the test images reveals that the **Total Quantity** and **Total Amount** fields only occur together with the title **Footer.kwTotal** and are located on the same level. We will use elements of the Character String type to detect the **Total Quantity** and the **Total Amount** fields. To specify the search constraints common to these elements, we need to create a Group element — **grTotal**.

To create the Group element **grTotal**:

1. In the **Footer** element, create an element of the Group type and name it **grTotal**.
2. Select the **Optional** checkbox because we decided to stop the search of the sub–elements in case of title absence.
Note: For some details see Help article Tips and tricks\The "Optional" property of a Group element.
3. Click the **Advanced** tab.
4. Since the **Total Quantity** and the **Total Amount** fields only occur on the image together with the title **Footer.kwTotal**, set the following additional search constraint in the **Advanced pre–search relations** field: *If the **Footer.kwTotal** element has not been detected, do not look for the object on the image.* In the FlexiLayout Language:

If Footer.kwTotal.IsNull Then DontFind;

The **Total Quantity** field is located at the intersection of the **Total** row and the **Quantity** column. The first constraint was already specified at the Group element level. To specify the second constraint, we will use the previously detected column name as a reference element.

To create the **TotalQuantity** element:

1. In the **grTotal** element, create an element of the Character String type and name it **TotalQuantity**.
2. Click the **Character String** tab.
3. In the **String Length** field, specify this fuzzy interval: [–1, 1, 10, INF]. This is an estimation of the length of the character string. We assume that the possible values may range from 1 to 10. Any hypotheses outside this range will be penalized.
4. Leave the default values for the other parameters of the element.
5. Click the **Relations** tab.
6. Since the **Total Quantity** field is always located on the same level with the keywords of the title **Footer.kwTotal** or slightly lower, set the following search constraints:
– **Above** the element **Footer.kwTotal**, **Offset** = –200, **Element border** = Bottom
– **Below** the element **Footer.kwTotal**, **Offset** = –20, **Element border** = Top
Note: Prior to setting the values for the offsets, you may want to analyze the geometrical properties (i.e. size and borders) of the neighboring objects detected during pre–recognition. Offset values are selected by trial and error.
7. Click the **Advanced** tab.
8. Since the **Total Quantity** field is optional on the images but whenever it occurs it occurs together with the **Quantity** column and below its name, set the following additional search constraint in the **Advanced pre–search relations** field: *Look for the object on the image only if the **TableHeader.kwQuantity** element has been detected; Look no farther than 50 dots to the left or to the right of the respective borders of the column name.* In the FlexiLayout Language:

```
If Not (TableHeader.kwQuantity.IsNull) Then
{
LeftOf: TableHeader.kwQuantity.Right, –50 * dot;
RightOf: TableHeader.kwQuantity.Left, –50 * dot;
```

```
}
Else DontFind;
```

9. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.
10. To describe the location of the **TotalQuantity** block, select the **Source element** option, then click **"..."** and select the **TotalQuantity** element as the source element.

The **Total Amount** field is located at the intersection of the **Total** row (the footer of the table) and the **Total** column. The first constraint was already specified at the Group element level. To specify the second constraint, we will use the previously detected column names as reference elements.

To create the **TotalAmount** element:

1. In the **grTotal** element, crate an element of the Character String type and name it **TotalAmount**.
2. Click the **Character String** tab.
3. In the **String Length** field, specify this fuzzy interval: [–1, 1, 20, INF]. This is an estimation of the length of the character string. We assume that the possible values may range from 1 to 20. Any hypotheses outside this range will be penalized.
4. Leave the default values for the other parameters of the element.
5. Click the **Relations** tab.
6. Since the **Total Amount** field is always located on the same level with the keywords of the title **Footer.kwTotal**, set the following search constraints:
   – **Above** the element **Footer.kwTotal**, **Offset** = –20, **Element border** = Bottom
   – **Below** the element **Footer.kwTotal**, **Offset** = –20, **Element border** = Top
7. Click the **Advanced** tab.
8. Since the **Total Amount** field can only be located immediately below (may be with a small shift) the name of the **Total** column and only between the **Unit Price** and **Sales** columns, set the following additional search constraint in the **Advanced pre-search relations** field: *If the **TableHeader.kwTotal** element has been detected, look for the object no farther than 70 dots to the right of the right border of the **TableHeader.kwTotal** element and no farther than 50 dots to the left from the left border of the **TableHeader.kwTotal** element; or, if the **TableHeader.kwUnitPrice** and **TableHeader.kwSales** elements are detected, look for the object no farther than 40 dots to the right of the left border of the **TableHeader.kwSales** element and to the right of the right border of the **TableHeader.kwUnitPrice** element; otherwise do not look for the object.* In the FlexiLayout Language:

   ```
   If Not (TableHeader.kwTotal.IsNull) Then
   {
   LeftOf: TableHeader.kwTotal.Right, –70 * dot;
   RightOf: TableHeader.kwTotal.Left, –50 * dot;
   }
   Else
   If Not (TableHeader.kwUnitPrice.IsNull) and Not (TableHeader.kwSales.IsNull) Then
   {
   LeftOf: TableHeader.kwSales.Left, –40 * dot;
   RightOf: TableHeader.kwUnitPrice, 0 * dot;
   }
   Else DontFind;
   ```

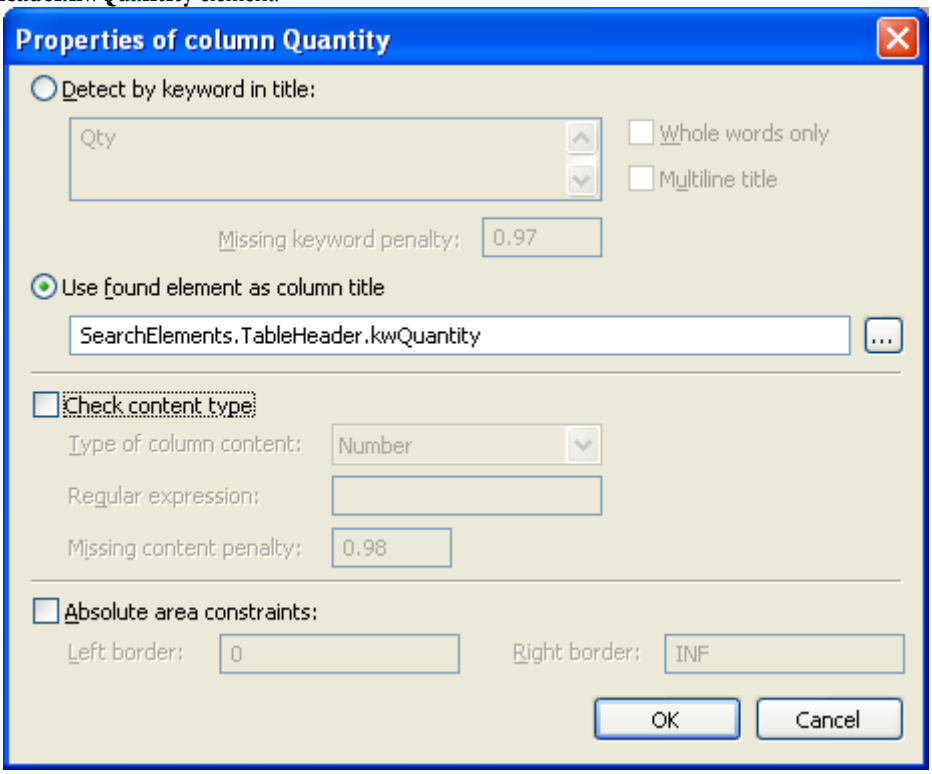
   Note: Prior to setting the values for the offsets, you may want to analyze the geometrical properties (i.e. size and borders) of the neighboring objects detected during pre–recognition. Offset values are selected by trial and error.
9. Try matching the FlexiLayout with the test images and make sure that the program successfully detects the element on all the images where applicable.
10. To describe the location of the **TotalAmount** block, select the **Source element** option, then click **"..."** and select the **TotalAmount** element as the source element.

Step 26. Detecting the Table element: the InvoiceTable element

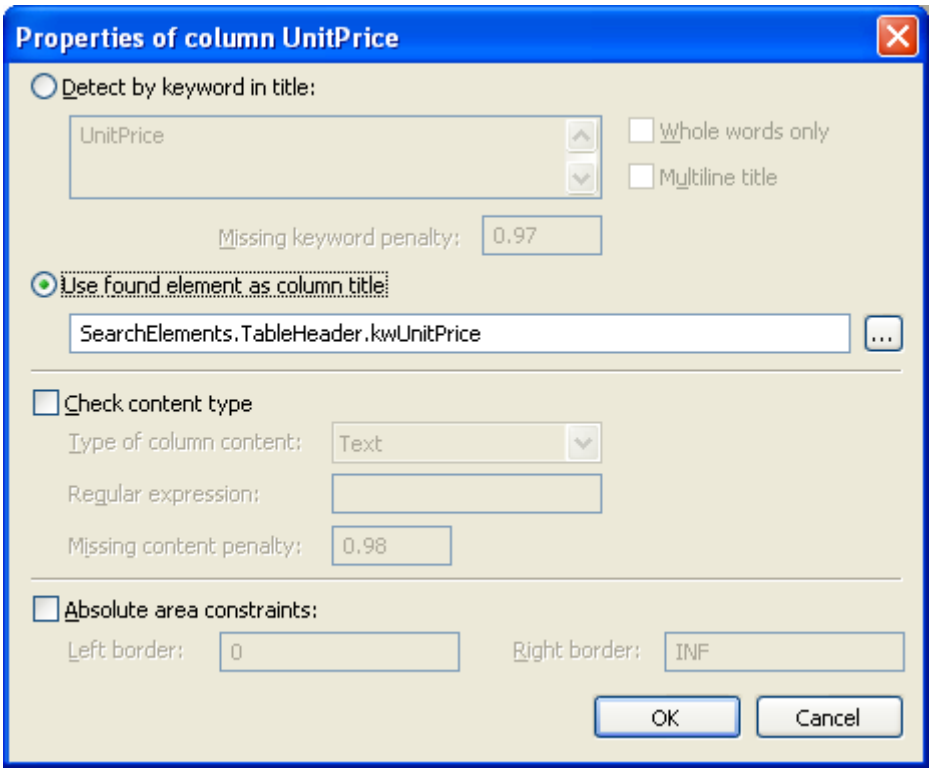
Now that all the preparations have been completed, we can search for the **Invoice Table** field. Let us describe an element of the Table type.
First, we describe the Header of the table. We decided (for detailed instructions, see Step 13) to describe the names of the **Reference** and the **Designation** columns by keywords (**Table** element > **Properties** dialog box > **Columns** tab > dialog box of column **Properties**), while the names of the **Quantity**, **Unit Price**, and **Total** columns will be detected with the help of the previously described auxiliary elements.
Secondly, we describe the Footer of the table. To describe the footer, we decided to use the previously described element which uses keywords and phrases to detect the beginning of the footer (for detailed instructions, see Step 13).
Thirdly, we describe the search area of the table. We will use the **Sales** column name to restrict the search area on the right.

To create the **InvoiceTable** element:

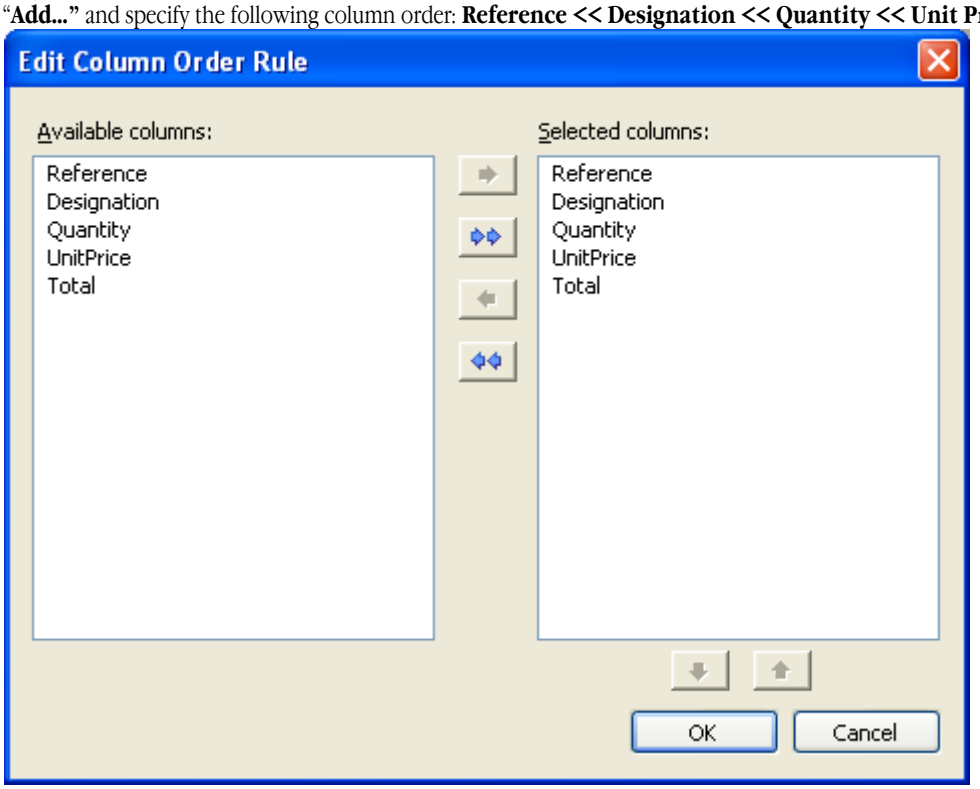
1. Create an element of the Table type and name it **InvoiceTable**.
2. Click the **Columns** tab.
3. Click **"..."** and specify the previously described **InvoiceTable** block as the block to be detected by the **InvoiceTable** element (for detailed instructions, see Step 2).
4. Select the **Reference** column and click **Properties**. We will use the keyword **Reference** to look for this column name, so leave the default values.
5. Select the **Designation** column and click **Properties**. We will use the keyword **Designation** to look for this column name, so leave the default values.
6. Select the **Quantity** column and click **Properties**. We will use the previously detected element **TableHeader.kwQuantity** to find this column name. Select **Use found element as column title** option and click **"..."** to select the **TableHeader.kwQuantity** element.

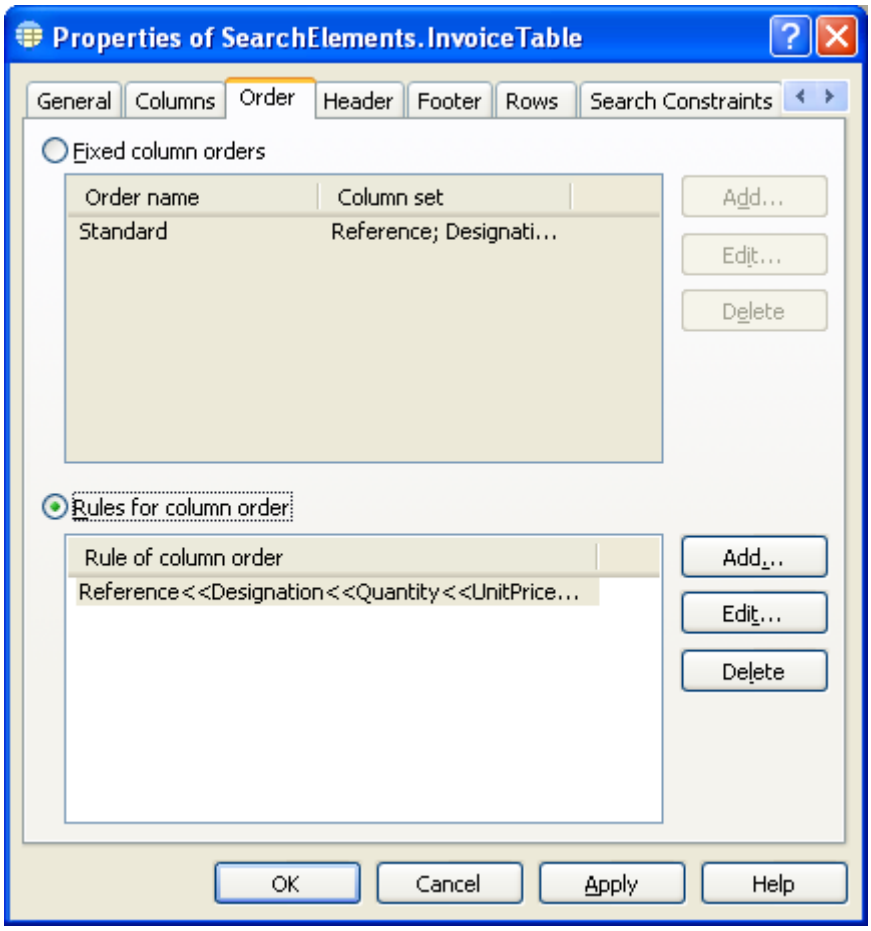


7. Select the **UnitPrice** column and click **Properties**. We will use the previously detected element **TableHeader.kwUnitPrice** to find this column name. Select **Use found element as column title** option and click **"..."** to select the **TableHeader.kwUnitPrice** element.

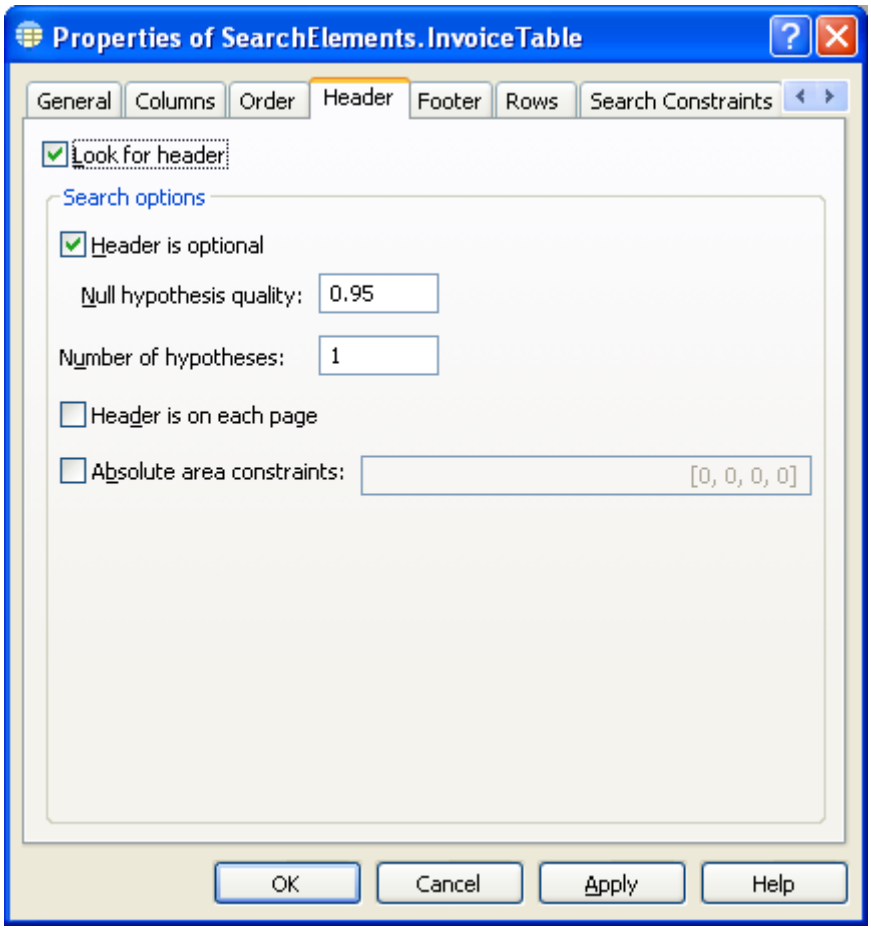


8. Select the **Total** column and click **Properties**. We will use the previously detected element **TableHeader.kwTotal** to find this column name. Select **Use found element as column title** option and click **"..."** to select the **TableHeader.kwTotal** element.
9. Click the **Order** tab.
10. There can be other columns between the sought ones (for instance, **Unit**), but the sought columns are always located in the following order: **Reference**, **Designation**, **Quantity**, **Unit Price**, **Total**. Select the **Rules for column order** option, then click "**Add...**" and specify the following column order: **Reference << Designation << Quantity << Unit Price << Total**.

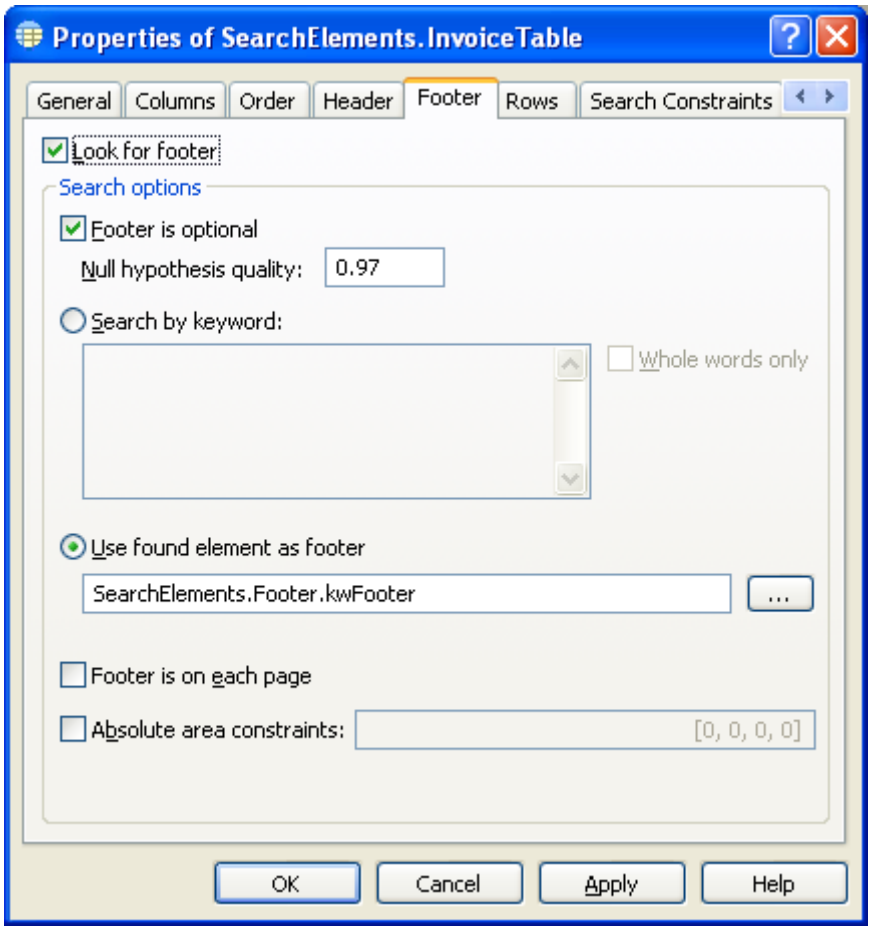




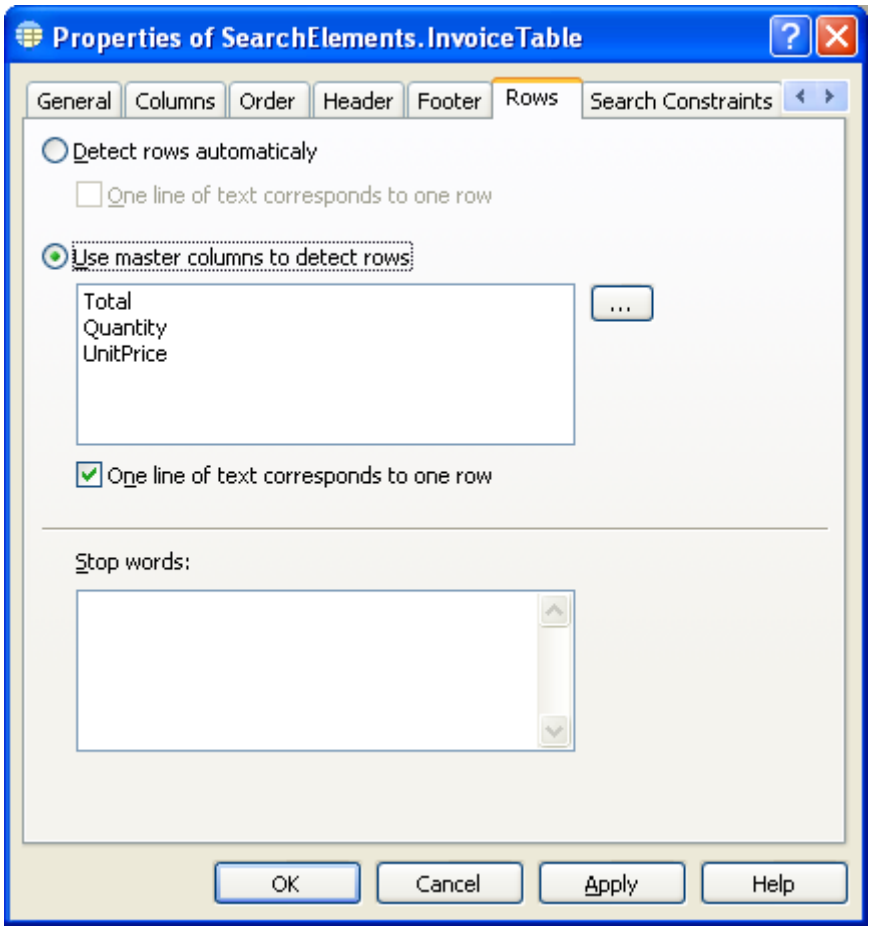
11. Click the **Header** tab. We need the header but we are not going to add any search constraints, so leave the default values.



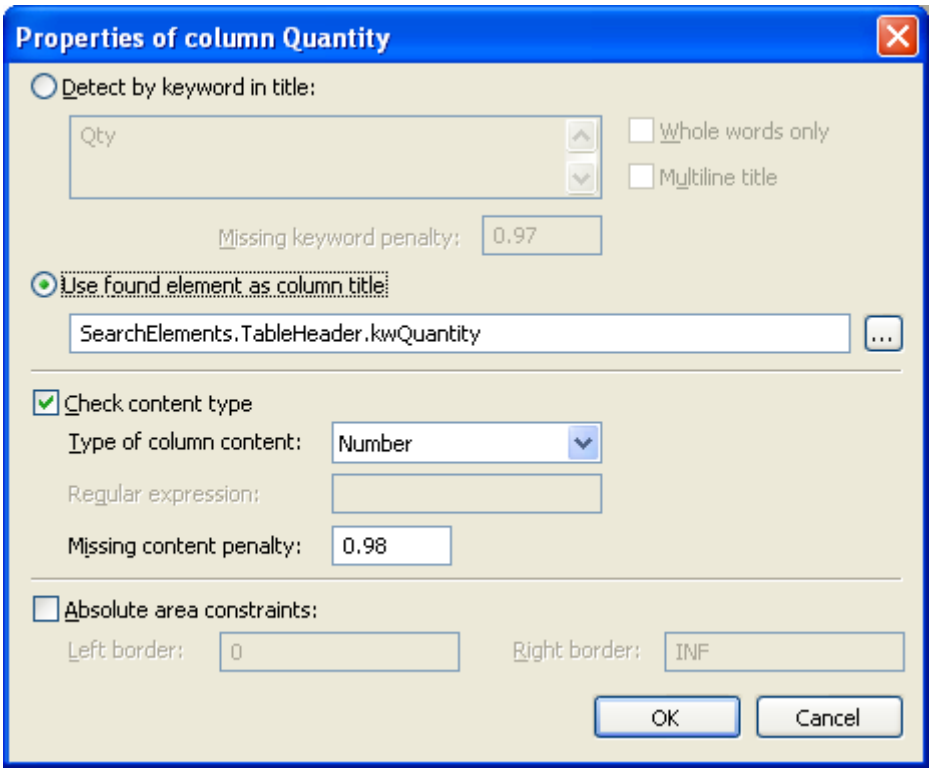
12. Click the **Footer** tab. We will use the previously detected element **Footer.kwFooter** to detect the footer. Select **Use found element as footer** option and click **"..."** to select the **Footer.kwFooter** element.



13. Click the **Rows** tab.
14. Since some columns in the table may have entries consisting of multiple lines, we have to specify master column to detect the rows. Select the **Use master columns to detect rows** option. Specify the **Total** column as the primary master column, as this column always contains data and a single row of the table corresponds to each entry. In case this column is not detected on a form, specify the **Quantity** and **Unit Price** columns as secondary and tertiary master columns respectively. Click "**...**" and specify this order of master columns.



15. Click the **Relations** tab. Since the sought columns are located to the left of the beginning of the name of the **Sales** column (with a small gap), set the following search constraint:
– **LeftOf** the element **TableHeader.kwSales**, **Offset** = –40, **Element border** = Left
Note: When selecting values for offsets, you may want to analyze the geometrical properties (i.e. size and borders) of the reference and the sought objects detected during pre–recognition. Otherwise these values are selected by method of trial and error.
16. Match the FlexiLayout with the test images. You see that the program has successfully detected the table element on all the images where it occurs. But on some of the images, the **Quantity** column also includes unwanted artifacts from the previous column. To separate the useful data from noise, let's specify the type of content which may occur in the column.
17. Open the **Properties** dialog box of the **InvoiceTable** element. Click the **Columns** tab, open the **Properties** dialog box for the **Quantity** column and select the **Check content type** option**.** Select **Number** in the **Type of column content** drop–down list.



18. Try matching the FlexiLayout with the test images and make sure that the **Quantity** column does not contain any unwanted data.
19. To describe the location of the **InvoiceTable** block, select the **Source element** option, then click **"..."** and specify the **InvoiceTable** element as the source element.

Step 27. Exporting the FlexiLayout into ABBYY FlexiCapture

The resulting FlexiLayout can be saved as an *.afl file and exported into ABBYY FlexiCapture.
Note: For more about exporting FlexiLayouts, see Sample 1, Step 21 and Step 22.