Reporting

User Manual

Report Internals

This section describes the internal components and features of Stimulsoft Reports, including Expressions, Appearance, Text Formatting, Barcodes, Watermarks and more.

Expressions

Expressions are a key part of Stimulsoft Reports - without them it would not be possible to produce any reports at all. An expression is a combination of one or more of the following:

- Text;
- Mathematical and Logical operators;
- Constants;
- Functions;
- Field names;
- Controls;
- Properties.

Stimulsoft Reports processes the expressions defined in a report in order to calculate the value to be displayed or printed for each one. This value is saved and can be used in further calculations when generating the report output.

The most common expressions used in the report generator are text expressions. These expressions are used to define any text displayed or printed in the report that is not the subject of any calculation such as a text heading. Text expressions are always converted into strings.

Text Expressions

The simplest expressions are Text expressions. For example:

```
MyText
```

```
12345
```

```
Test
```

All three expressions above consist of one string and there are no calculations - the expression will be printed in the report exactly as it has been defined. Such expressions are typically used to indicate simple string constants, column names, reports, links etc.

Calculating Values in Expressions

An expression can contain many different types of variable as well as functions and field values from databases. These various parts can be combined to calculate a value to be printed or displayed within a report.

Using Code in an Expression
When calculating a value within an expression you may also include code written in the programming language of a report. Curly braces (the “{“ and “}” symbols) are used to separate code item from other text. The opening brace symbol “{“ indicates the beginning of a calculation. The closing brace symbol “}” indicates the end of a calculation. The code between symbols is calculated and the value included in the result of calculation. In text expressions the result of the calculation is automatically converted into a string. For example, if you enter the following expression:

Value = {1 + 2}

then after calculation the result appearing in the report will be:

Value = 3

Multiple Code Insertions
When using calculations an unlimited number of code insertions are allowed in any one expression. For example, if you enter the following expression:

```
ValueA = {1 + 2}, ValueB = {2 + 3}
```

then after calculation the result appearing in the report will be:

```
ValueA = 3, ValueB = 5
```

Nested Code Insertions
When you perform calculations in an expression the nesting of code sections is not allowed. For example, the following expression is not correct and will cause the calculation to fail:

```
Value = {1 + 2 + {2 + 3}}
```

Important: Code nesting is not allowed when making calculations in expressions.

Multi-line Expressions

It is possible for a single expression to output multiple lines of text within a report. To create a multi-line expression simply insert a line feed before any new line. You can do this by simply pressing the Enter key at the appropriate place

in the code editor. There is no limit to the number of lines that can be included in an expression. For example, if you enter the following expression:

```
Value:
{1+2}
```

then after calculation the result appearing in the report will be:

```
Value:
3
```

In other words, the text output will contain two lines.

Note: An expression may contain any number of lines.

Using Code in Multi Line Expressions

Multi line expressions do not have limitations on using code to calculate values other than those for single line expressions.

Using Dictionary Variables

You can create variables in the designer data dictionary which can then be used in expressions. When you specify the name of a variable in the expression the value of the variable will be included in the report. The syntax is simply the name of the variable surrounded by curly braces. For example if you set the value of the variable to 5 and you enter the following expression:

```
Value = {MyVariable}
```

then after calculation the result appearing in the report will be:

```
Value = 5
```

Calculating with Variables

Variables can also be used in calculations. For example if the value of **MyVariable** is 15 and you enter the following expression:

```
Value = {MyVariable + 10}
```

then after calculation the result appearing in the report will be:

```
Value = 25
```

Important: If the report language is **C#** then variable names are case sensitive. If the report language is **VB.Net** then variable names are not case sensitive.

Using Report Code Variables

It is possible to add programming code and additional variables into the underlying code of a report within the report generator. If you choose to add variables in this way they can be used within expressions in the report.

The example below adds two variables to the report in code:

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.Data;
using Stimulsoft.Report;
using Stimulsoft.Report.Components;
using Stimulsoft.Base.Drawing;

namespace StiReports
{

    public class Report : Stimulsoft.Report.StiReport
    {

        private int MyVatiable = 15;
        private string MyName = "Test";

        public Report()
        {
            this.InitializeComponent();
        }

        #region StiReport Designer generated code - do not modify
        public Stimulsoft.Report.Dictionary.StiDataRelation ParentSu
        public Stimulsoft.Report.Dictionary.StiDataRelation ParentCa
```

You can use these variables in exactly the same way that you would use variables from the report data dictionary. However, it is important to remember that only variables declared in the data dictionary can be used to calculate global values such as a running total.

Important! Only variables declared in the data dictionary can be used to calculate global values.

Using Data Fields

Values from data sources can be used in expressions. To reference a field from the data source you must provide a string representation of the field. The syntax of the reference is simple - you give the name of the data source and the field name separated by a decimal point or full stop character, surrounded by curly braces:

```
{DataSource.Column}
```

For example, if you have an entry in the customers table with the company name field set to "The Big Company" and you enter the following expression:

```
Company Name: {Customers.CompanyName}
```

then after calculation the result appearing in the report will be:

```
Company Name: The Big Company
```

Note: In order to avoid having to create this sort of expression manually you can use drag and drop from the data dictionary directly to the page of a report or within the expression editor to insert the necessary information automatically and with the correct syntax.

Parent Relationships

If the data source has a **parent** relationship with other data sources you can directly reference fields from the **parent** data source. The syntax of the reference is similar to the examples already given - you give the name of the data source, then the relation name, and then the field name each separated by a decimal point or full stop character, and the whole thing surrounded by curly braces. For example:

```
{Datasource.Relation.Field}
```

Assuming that you have a set of information like this:

- **Products** is a name of a data source;
- **ParentCategories** is a name of relation, with what two data sources are related. In this case, two data sources are related:
- **Products** is a list of products, and **Categories** is a list of categories of these products.
- **CategoryName** is a column name in the **Categories** data source.

if you enter the following expression:

```
{Products.ParentCategories.CategoryName}
```

then after calculation the result appearing in the report will be the name of a category for a product.

There are no limits on the number of relationships you can use in Stimulsoft Reports. Therefore a column can be called through two or three or even more relationships. For example, Assuming that you have a set of information like this:

- **OrderDetails** is a name of a data source;
- **ParentProducts** is a name of relations between **OrdersDetails** and **Products** data sources;
- **ParentCategories**. is a name of relation between **Products** and **Categories** data sources;
- **CategoryName** is a field in the **Categories** data source.

if you enter the following expression:

```
{OrderDetails.ParentProducts.ParentCategories.CategoryName}
```

then after calculation the result appearing in the report will still be the name of a category for a product butthe value of the **CategoryName** field has been obtained using relationships and bypassing the **OrderDetails** data source to get to the **Categories** data source. No direct call to the **Categories** data source has been used

Important: If the report language is **C#** then names are case sensitive. If the report language is **VB.Net** then names are not case sensitive.

It should be remembered that all the values in data sources are typed. This means that all data items are dynamically converted to the type that is specified in the options column which helps to accelerate the development of reports. However, if you need to get data from a column without conversion you will need to specify the data source directly. For example, in C#:

```
{Products["ProductName"]}
```

This expression will return data from the **Products** data source "as is" without conversion. The example below shows the same expression for VB.Net:

```
{Products.Item("ProductName")}
```

Using Component Properties

When creating an expressions you can use the properties of any component contained within a report.

Syntax

The syntax is the same whether the report language is **C#** or **VB.NET**. You simply enterthe name of the component and the property name separated by a decimal point or full stop character, surrounded by curly braces:

```
{Component.Property}
```

Important: If the report language is **C#** then names are case sensitive. If the report language is **VB.NET** then names are not case sensitive.

For example, to display the name of a component called MyComponent you would enter the expression:

```
{MyComponent.Name}
```

If you wish to access a calculated value from within a component you should use the property that contains the result you require. For example, if the component has ahyperlink value which calculates a hyperlink from the other component properties you would access it by entering the expression:

```
{MyComponent.HyperlinkValue}
```

You can use component properties in calculations should this be necessary. For example, the following would display the area taken up by the component:

```
{MyComponent.Width*MyComponent.Height}
```

Using Functions in Expressions

Built In Functions

Stimulsoft Reports has a large number of built in functions available for you to use. You can access these functions directly from the data dictionary and within the Expression Editor. Examples of built in functions and their usage would be:

```
{Trim(MyString)}
```

or

```
{Trim(MyDataSource, MyDataColumn)}
```

In each case the use of the **Trim** function removes leading and trailing spaces from the result shown in the report.

.NET Framework Methods

In addition to the built in functions you can use any available .Net Framework methods. For string expressions you could use any of the following examples:

```
{MyString.Trim()} // Removes leading and trailing spaces
```

```
{"Test".ToUpper()}// Converts the value to upper case "TEST"
```

```
{MyString.Length}// Returns the length of the string - if the value of MyString is "Test" then the method will return 4
```

For numerical expressions you could use any of the following examples:

```
{Math.Round(MyValue, 2)}// Rounds the value to two decimal places
```

```
{Math.Sqrt(MyValue)}// Returns the square root of MyValue
```

```
{MyValue.ToString() + " times"} // Converts the number to a string and adds the word "times" -
                                // if MyValue is 5 this returns "5 times"
```

There are no limits to the number of Framework methods you can access - if they are available within .Net for the type you are using in a report you can use them without restriction.

Conditional Expressions

Conditional Expressions are not allowed in Stimulsoft Reports by default. However, there are two ways force conditional behaviour should you find it necessary to do so:

The IIF Function

Firstly you can use the built-in **IIF** function which you can insert from the data dictionary. The function uses the following syntax:

```
{IIF(Condition, Value1, Value2}
```

This evaluates **Condition**, and if the **Condition** returns **true**, then the expression will return **Value1**. If it returns **false**, then it will return **Value2**. For example, if you enter the following expression:

```
Number of Stores: {Store.Count > 0 ? Store.Count : "None"}
```

then if the value of Store.Count is 10 after calculation the result appearing in the report will be:

```
Number of Stores: 10
```

If the value of Store.Count is 0 after calculation the result appearing in the report will be:

```
Number of Stores: None
```

The C# Ternary Operator

If you are using **C#** as your report languageit is also possible to use the ternary operator. The syntax for the ternary operator is as follows:

```
{Condition ? Value1 : Value2}
```

In exactly the same way as the IIF function, if **Condition** evaluates to **true**, then the expression will return **Value1**. If**false**, then it will return**Value2**.

Using Aliases in Expressions

To make it easier to understand expressions in a report you can use aliases instead of explicitly specifying the variable or data source and column details. For example, if you have a variable in the data dictionary called "MyVariable" and you have set its alias to "my best variable" you can reference that variable directly by Name or by Alias.

To use the variable by name you would create an expression like this:

```
{MyVariable}
```

To use the variable by alias you would create an expression like this:

```
{[my best variable]}
```

Syntax - Variables

If you use spaces, punctuation, or characters within an alias that are not permitted under C# or VB.Net then you MUST enclose the string representation of the alias in square brackets []. If no such characters are used then the square brackets are optional.

For example, if the alias was**"MyBestVariable"**then the expressioncan be written without brackets:

```
{MyBestVariable}
```

Otherwise you MUST enclose the variable in square brackets. Examples of valid alias usage:

```
{Variable1}

{VariableAndValue}

{[Variable and Value]}

{[Variable and Value]}

{[Variable&Values]}

{[Variable-First]}
```

Just for extra clarification, examples of some **INVALID** alias usage

```
{Variable and Value}// spaces in the name cause this to fail

{Variable&Values}// reserved character causes this to fail
```

Syntax - Data

The same rule is used and when creating the names of data sources and columns. But there is one exception. When referring to the data column, only a part with incorrect characters for identifier should be bracketed.For example:

```
{DataSource.[Data Column]}

{[Data-Source].DataColumn}
```

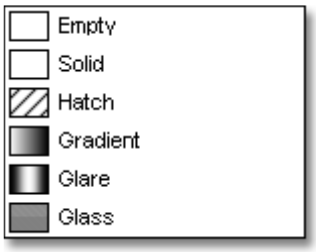
```
{[Data=Source].[Data=Column]}
```

Appearance

Stimulsoft Reports offers many ways to control the appearance of your reports. These include text brushes, brushes to fill background, font types, component borders, and horizontal and vertical alignment of the contents of components. Styles can be used to simplify setting the appearance of your reports and to standardize the look and feel of them.

Background Brushes

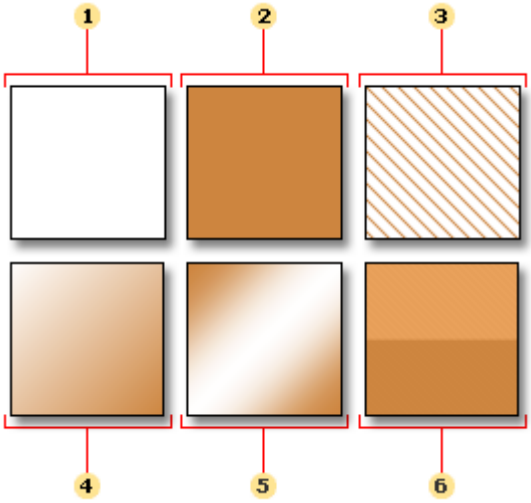
Brushes are used to fill a background, and to draw a text within a report. Brushes have several styles and colors. To change the background color and appearance of a component use the **Brush** property within the Object Inspector.



Six types of Brushes are available within Stimulsoft Reports:

- **Empty**;
- **Solid**;
- **Hatch**;
- **Gradient**;
- **Glare**;
- **Glass**.

Below are representations of the results all six Brush types:



1 **Empty**	The background of a component is transparent.
2 **Solid**	The background of a component is filled with the color you specify.
3 **Hatch**	The background of a component is filled with a texture. The background and foreground colors of the select can be specified individually.
4 **Gradient**	The background of a component is filled with gradient. A Start color, an End color, and a Gradient ang specified.
5 **Glare**	The background of a component is filled using the Glare effect.
6 **Glass**	The background of a component is filled using the Glass effect.

Fonts and Font Brushes

A **font** is a complete set of characters - letters, numbers, and symbols - that share a common weight, width, and style. Stimulsoft Reports has two components which are used to setup and draw text, the **Text** and **Rich Text** components. The font for these components can be set using the **Font** property within the Object Inspector.

Selecting Fonts

Text within a report can be output using different fonts. The font is set using the Font.Name property. Three examples fonts are shown below:

AaBbCcDd

AaBbCcDd

AaBbCcDd

Any font that is installed on your machine can be used in a report. However, when choosing a font try to select one that will also be present on a user machine or a report may not render as you would wish at runtime.

Font Size

You may well wish to change the size of font on some components, for example a heading may require a much larger font size than a copyright notice.

The font size can be changed using the **Font.Size** property. For example:

AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd

Font Styles

Different styles can be applied to the font. A font may include one or more styles such as regular, bold, semibold, italic, underlined, and strikeout. You can control the styles using properties such as **Font.Bold**, **Font.Italic**, **Font.Underline**, and **Font.Strikeout**, and the styles may be combined to produce different effects like bold and underlined or bold and italic. Examples of font styles are shown below:

AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
~~AaBbCcDd~~

Font Brushes

Five types of brushes are used to draw a text: **Solid**, **Hatch**, **Gradient**, **Glare**, and **Glass**. The **TextBrush** property is used to control brushes. An example of using the five different brushes is shown below:

AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd

Borders

Many components in Stimulsoft Reports can have borders. Where they have been set borders may have different thicknesses, colors, and styles, and there be a drop shadow applied. The Border property of a component is used to control the appearance of the border, and this property can be manipulated either from the Object Inspector or using controls within the Ribbon or the Toolbar depending on whether you are using the Standard of Ribbon interface.

There are two types of borders in Stimulsoft Reports: Simple and Advanced. The Borders can be included in component styles so that they can be automatically applied to multiple components.

Articles in this section describe both types of border and the differences between them.

Simple Borders

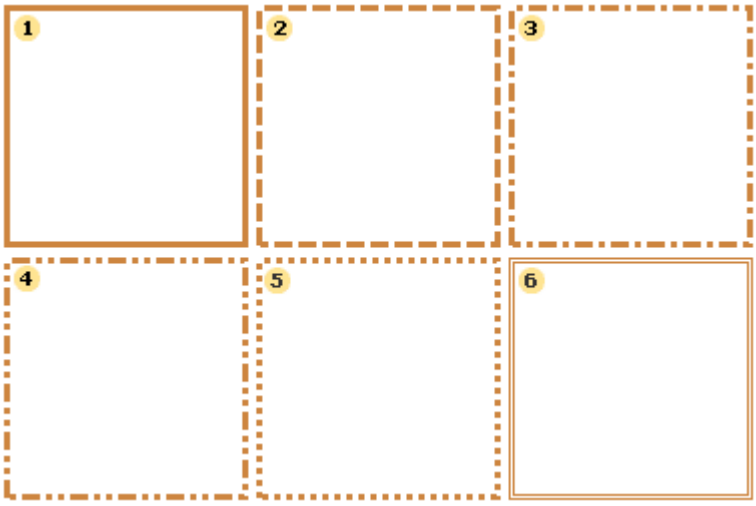
Border Sides

Each border consist of 4 segments: **top side**, **left side**, **bottom side**, **right side**. These segments may be shown together or in different combinations. For example:

Using the **Border.Side** property it is possible to setup on which sides a border will be visible.

Border Style

Seven styles of border are available - **Solid**, **Dash**, **Dash Dot**, **Dash Dot Dot**, **Dot**, **Double,** and **None**. With simple borders a selected style is applied to all sides of the border at the same time. Examples of each type of border are shown below:



1 **Solid**;
2 **Dash**;
3 **Dash Dot**;
4 **Dash Dot Dot**;
5 **Dot**;
6 **Double**.

The style of border can be selected using the **Border.Style** property. You can also set the border color and thickness.

Border Color

The border color can be set using the **Border.Color** property. When using simple borders the selected color is applied to all visible border sides. The image below demonstrates components with different border colors.

Border Thickness

When using simple borders the border thickness is applied to all visible border sides. The border thickness can be set using the **Border.Size** property. The image below demonstrates components with different border thicknesses.

It is important to know that the border thickness is ignored if the **Double** border style is enabled.

Notice. The border size is ignored if the **Double** style is set in the **Border.Style** property.

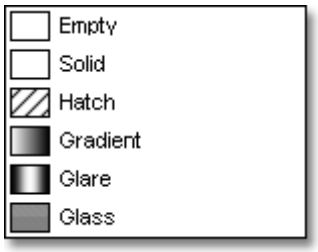
Shadow

A component that has borders may have shadow. A shadow has three parameters:

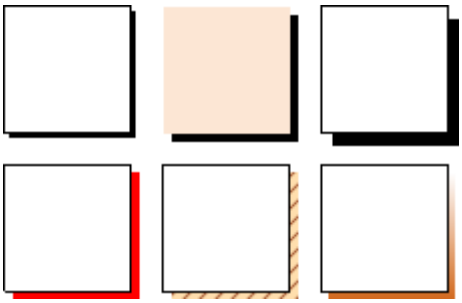
- **Border.DropShadow** - a boolean property. If it is set to true, then a Shadow will be shown
- **Border.ShadowBrush** - the brush to use to draw a shadow;
- **Border.ShadowSize** - the size of a shadow.

Shadow Styles

Five types of brushes are used to draw a border: Solid, Hatch, Gradient, Glare, and Glass.



These styles can be combined with the other shadow properties to apply a wide range of different appearances to report components. A few examples:

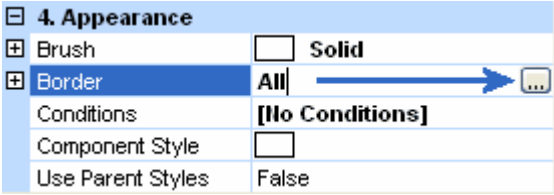


Setting Simple Border Properties

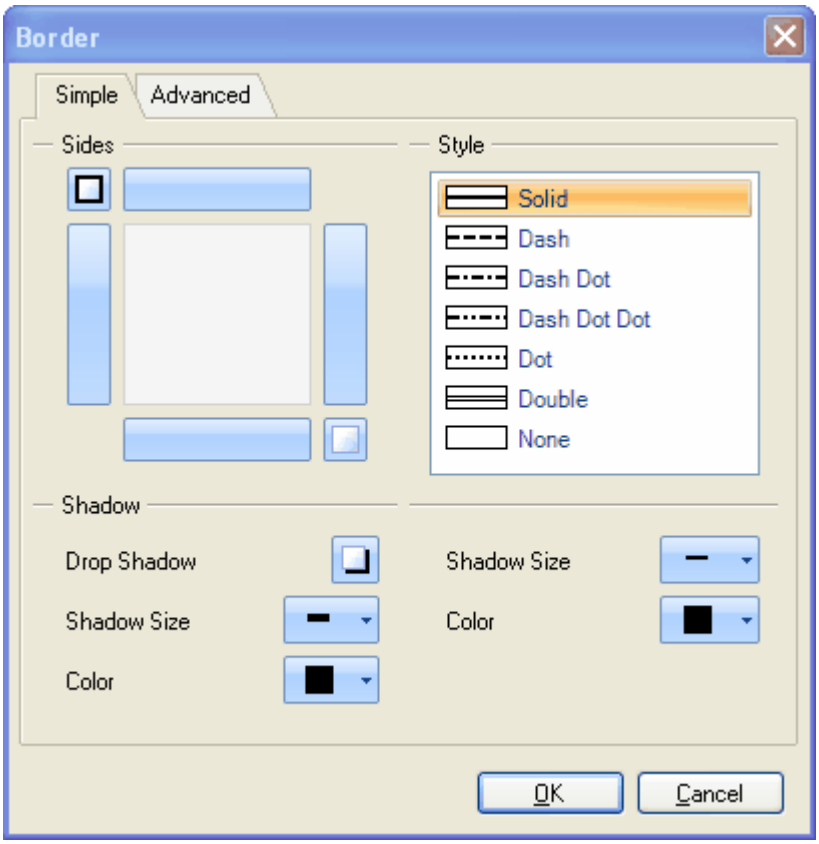
You can set simple Border properties directly from the Object Inspector, or using the Borders Toolbar.

Object Inspector

To set properties from the Object Inspector click the ellipsis button beside the Border property



A new dialog will be displayed that allows you to set the options for the border of the component:



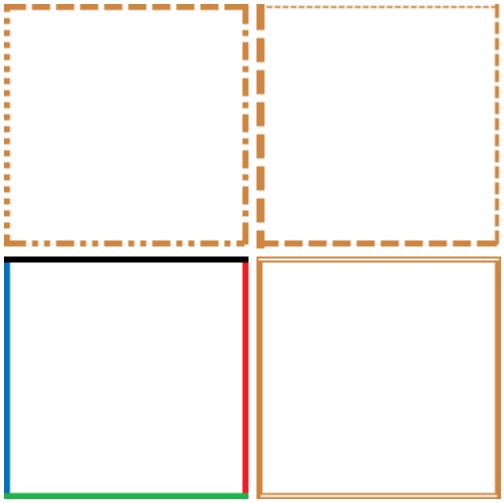
Simply select the settings you would like to apply and click the OK button to close the dialog and update the border.

Advanced Borders

The main difference between simple and advanced border types is that the style, color and thickness of the border can be set separately for each side - **Top Side**, **Left Side, Bottom Side**, **Right Side**. This provides additional opportunities to produce cleverly formatted reports.

Note: The advanced border type allows the style, color and thickness to be set separately for each side.

Some examples of advanced borders with different features applied to different sides:



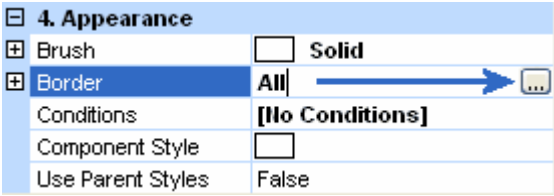
Setting Advanced Border Properties

You can set Advanced Border properties only from the Object Inspector.

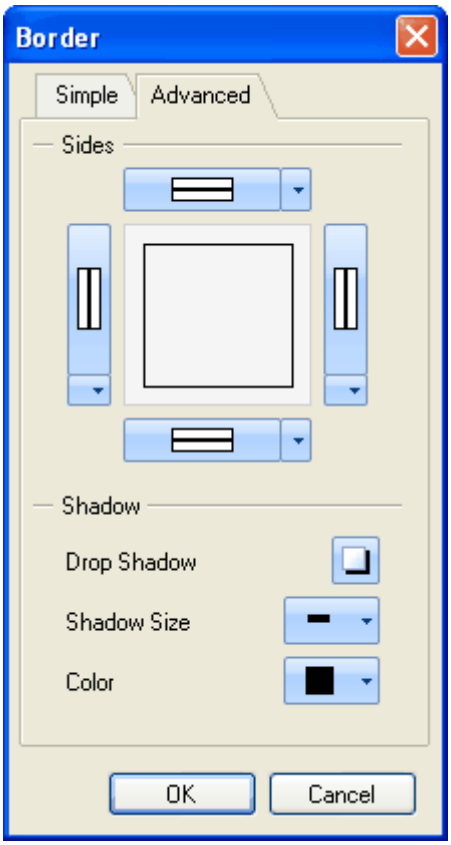
Important: You cannot set advanced border properties from the toolbar.

Object Inspector

To set **Advanced** border properties from the Object Inspector click the ellipsis button beside the Border property



The simple **Border** dialog will be displayed. To access the advanced border features simply click the Advanced tab at the top to bring it to the front.



Simply select the settings you would like to apply and click the OK button to close the dialog and update the border.

Conditional Borders

It is possible to conditionally select a border based on any condition arising within a report. For example, you may choose to display a red border if a total is negative, and a black border or no border at all if it is positive.

You can set a condition for a border using the Object Inspector in the designer. For more information on this topic please see the Conditional Formatting section.

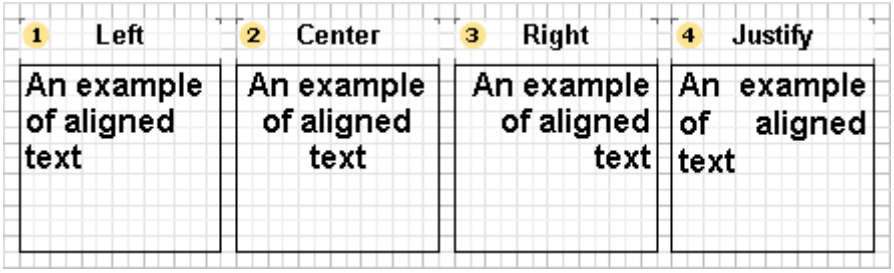
Horizontal Alignment

Some components (such as Text and Image components) allow the horizontal alignment of their content to be specified when creating reports.

To set the horizontal alignment use the Horizontal Alignment property in the Object Inspector or the alignment controls within the Ribbon or the Toolbar depending on whether you are using the Standard of Ribbon interface.

Horizontal Text Alignment

The most common alignment for text is Left aligned, where the left hand edge of each line of text starts at the same position in relation to the left hand edge of the component. However, modern design needs more flexibility so Stimulsoft Reports allows a choice of alignments: **Left**, **Center**, **Right**, and **Justify**. These are assigned using the HorizontalAlignment property of the component.



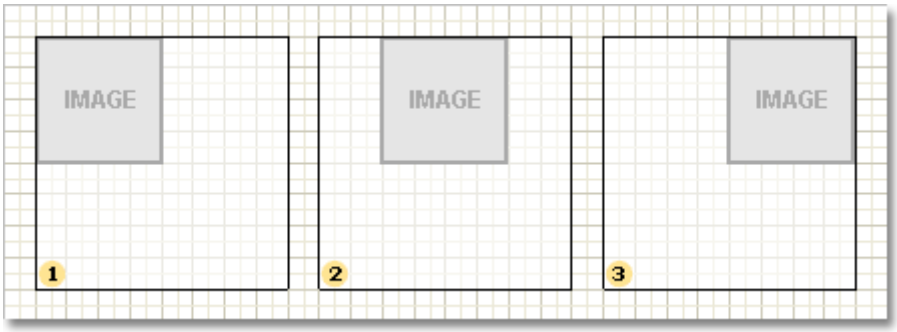
1	**Left**	The text is aligned on the right edge with a ragged right edge.
2	**Center**	The text is aligned centrally within the component with ragged left and right edges.
3	**Right**	The text is aligned on the right edge with a ragged left edge.
3	**Justify**	The text is aligned evenly across the width of the component, providing smooth edges to the text on both sides. This is achieved by automatically adjusting the amount of space between words.

Alternative Text Alignment

In addition the the alignment property it is possible to set text alignment using HTML tags.

Horizontal Image Alignment

Stimulsoft Reports allows a choice of three alignments of an image within an image component: **Left**,**Center**, and **Right**.



1	**Left**	The image is aligned on the right edge.
2	**Center**	The image is aligned on the center on the left and right edges of the component.
3	**Right**	The image is aligned on the right edge.

Images will be aligned only when the Stretch property of the image component is set to **false**. If the Stretch property is true then alignment settings will be ignored.

Important: Image alignment will be ignored if the Stretch property is set to **true**.

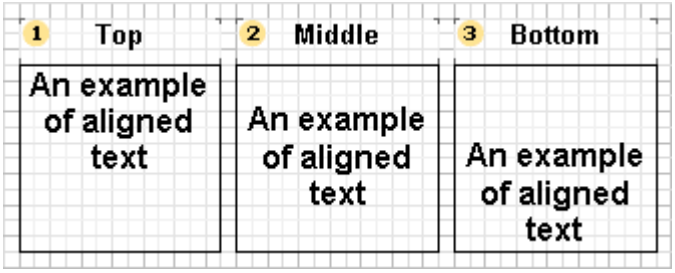
Vertical Alignment

Some components (such as Text components) allow the vertical alignment of their content to be specified when creating reports.

To set the vertical alignment use the Vertical Alignment property in the Object Inspector or the alignment controls within the Ribbon or the Toolbar depending on whether you are using the Standard of Ribbon interface.

Vertical Text Alignment

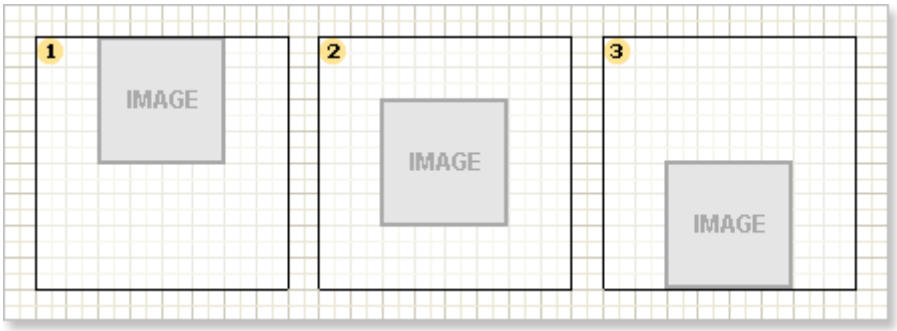
By default a text is aligned with the top edge of a component. But if the need arises, you can install the necessary alignment. In doing so, if there is alignment on the lower side and the text does not fit vertically within the boundaries of the component, it will be truncated on the upper side. If it is aligned to the center, in the case if the text does not fit, he will cut off both the top and bottom side.



1	**Top**	Text is aligned with the top edge of the component.
2	**Center**	Text is aligned centrally between the top and bottom edges of the component.
3	**Bottom**	Text is aligned with by the bottom edge of the component.

Vertical Image Alignment

To control the vertical alignment for the Image component the same property is used as for the Text component. Images are aligned only if the Stretch property is set to false. Otherwise, alignment will be ignored.



1	**Top**	The image is aligned with the top edge of the component.
2	**Center**	The image is aligned centrally between the top and bottom edges of the component.
3	**Bottom**	The image is aligned with the bottom edge of the component.

Images will be aligned only when the Stretch property of the image component is set to **false**. If the **Stretch** property is true then alignment settings will be ignored.

Important: Image alignment will be ignored if the Stretch property is set to **true**.

Styles

A style is a combination of various design attributes which can be applied to report components.

Instead of manually formatting each component, you can create a new style in a report and set its parameters (such as font name, size, and font style) exactly as you want them. The stylecan then be assigned to any component within the report and it will automatically take on the features of that style.

Another advantage of using styles is that should it become necessary or desirable to change the formatting of a report simply changing the settings of the relevant style will automatically propagate those changes across the entire report.In addition, a specific report style can be saved to a file and can then be used in other reports. This allows a common appearance to be applied to all reports where a corporate style or standard output format is required.

Name

Each style has its own name. This name must be unique within a report.

Description

Each style also has adescription which can be used toexplain the intended purpose of the style to others. For example if you create a style called 'Section Heading' you might assign a description 'Bold heading for use at the start of a section'

Style Types

There are three types of style:

1. **Component**;
2. **Cross-Tab**;
3. **Chart**.

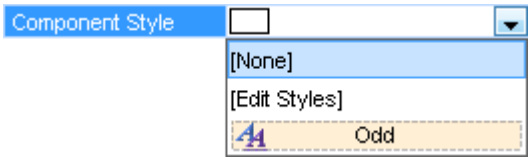
The Component style is designed to be used with all components except the Cross-Tab and Chart components which have their own dedicated style types because they have style features not included in other components.

The component style contains all the basic elements of the appearance of a component including **Font**,**Text Brush**,**Brush**,**Border**,**Horizontal Alignment**, and **Vertical Alignment**. In addition to these parameters, the simple style has parameter flags that determine whether certain style parameters can be modified by the user at design time.

1. Main	
Name	**Odd**
Description	
2. Appearance	
Text Brush	**Solid**
Brush	**Solid**
Border	**None**
Font	**Arial; 8pt**
Horizontal Alignment	Left
Vertical Alignment	Top
3. Parameters	
Allow Use Font	True
Allow Use Border	True
Allow Use Brush	True
Allow Use Text Brush	True
Allow Use Text Options	True
Allow Use Hor Alignment	False
Allow Use Vert Alignment	False

Applying Styles

Each component in the report has a **Component Style** property. In the object inspector you can specify any style that exists within the report by clicking the drop down button at the right of the property and selecting it from the list. You can also create or edit styles by clicking the [Edit Styles] option:



After a style has been assigned to a component the report generator will ensure that the appearance of the component consistently matches that of the specified style. Changes to the style will automatically cascade to all components to which the style has been assigned.

For example, if the developer changes the background color of the style all the components in the report that use that style will take on the new background color.

It is important to remember that even though they may share a style each component has its own design parameters which may not include some of those set in the style. For example, the **Panel** component has no **Font** parameter. If you apply a style to a panel, this parameter will be ignored. In other words the component will use only the design parameters of the style that it actively supports.

Note: The component will use only those parameters of the style that it supports.

Alternate Row Styles

The **Data** component has more than one property to which it is possible to assign a style. In addition to the standard **ComponentStyle** property this component has two additional properties: **OddStyle** and **EvenStyle**. These properties are used to highlight alternate lines of a report.

By default these properties are not set, but if you allocate suitable styles to each property the report generator will apply those styles to the even and odd numbered lines when rendering the report. In the example below a style with a different background color has been applied to alternate rows:

UseParentStyles Property

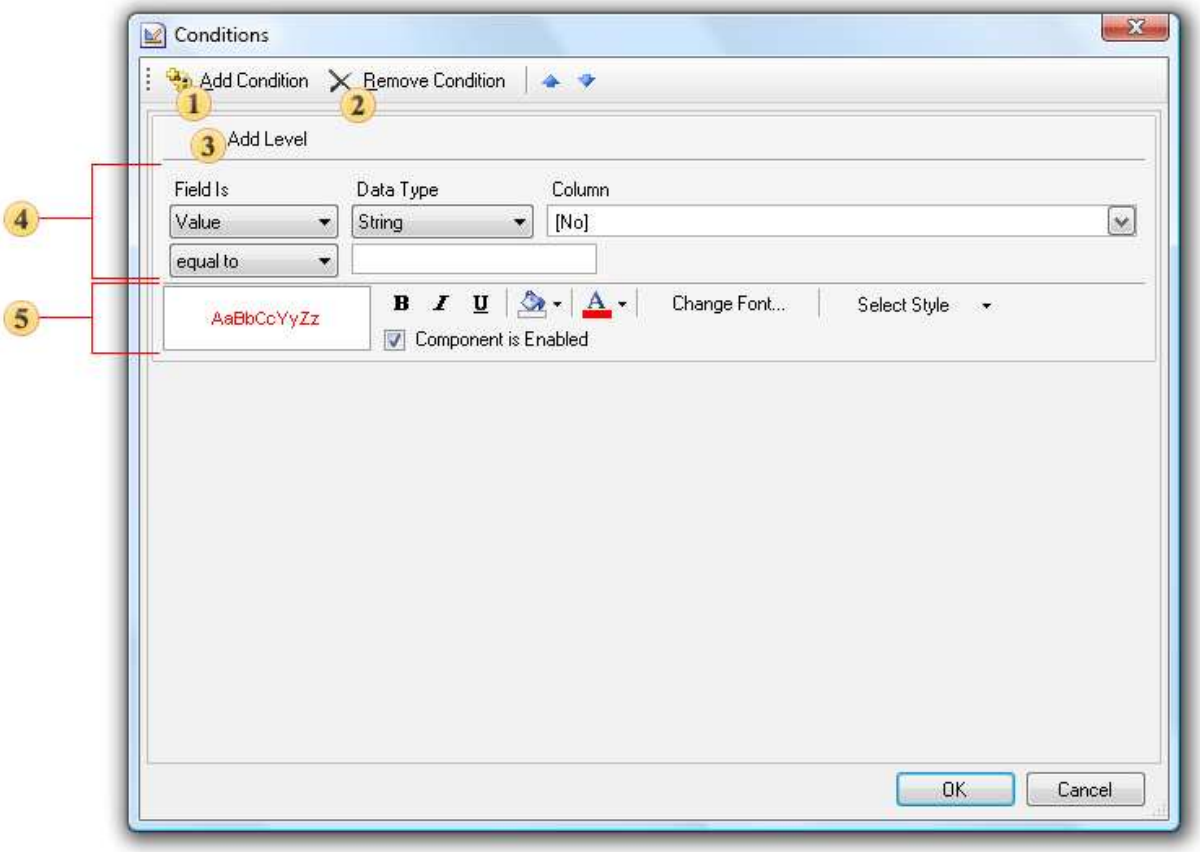
Each component has an additional property management style: the **UseParentStyles** property.

If this property is set to **true**, then the component will use the style of the component on which it is located. For example, if the component is on a page, it will automatically use the style set for that page. If the component is on a

panel, then it will use the panel style. If the **UseParentStyles** property is set to true for the panel, then both components will use the page style.

Conditional Formatting

Conditional formatting allows you to change the design of components, depending on certain conditions. For each component in the record, you can set the conditions that define its formatting, such as font style, text color and background color. You can also hide or disable the component. For a component, you can set several conditions, ie appearance of the component may change in different ways depending on the conditions. Setting up conditional formatting is done using the properties of conditions (Conditions). Using this property is called the editor environment. The figure below presents the main elements of the editor of conditions:



1 **Add condition**
This button adds a new conditional formatting to component conditions.
2 **Remove condition**
This button removes a new conditional formatting from component conditions. It is necessary to select the conditional formatting.
3 **Add level**
This button adds one level of the condition parameter.
4 **Parameters of condition**
Parameters of condition are specified on this panel.
5 **Parameters of formatting**
Parameters of the component appearance are setup on this panel.

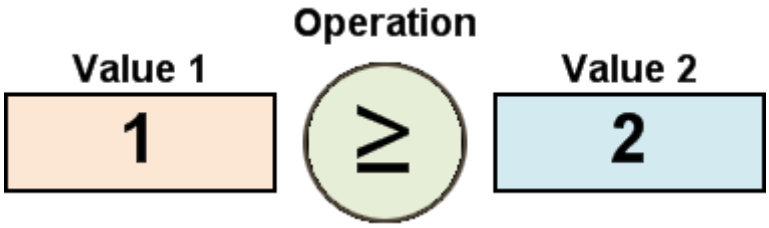
There are two types of conditions - **Value** and **Expression**. How to set a condition is reviewed on next topics.

Value Condition

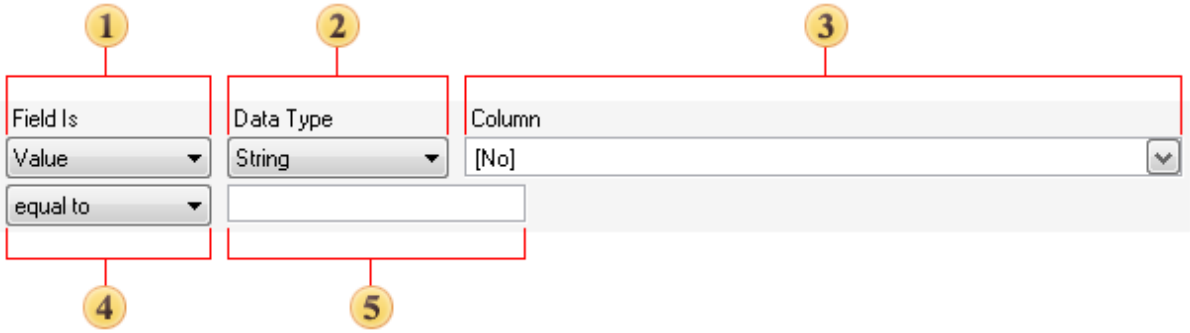
If you use a Value condition you will need to set the condition using a special format which consists of three elements:

1.	**The column in the data source**	The column in the data source from which the first value is taken for comparison with the second value of the condition.
2.	**Operator**	The selected operator lets the report generator know how to process the first and second values to obtain the result. For example, the comparison operator tells to the reporting tool to compare the first and the second values to produce the result.
3.	**The value to calculate a condition**	This is the second value used to calculate the condition (the first is taken from the data source). The value can be either a constant (for all types of data except for the Expression type), or an expression (for the Expression type).

If you were writing a value condition in code, it would look like this:



For several types of operation three values are used in calculating the condition. These are operations in which the value is checked to determine whether or not it is within a specified range, defined by two values. In addition to the elements described, the condition also includes a data type. The data type helps the reporting tool to identify the type of the second condition, and to automatically modify the list of available types of conditional operator. The picture below shows the panel used to set a value condition:



1	**Field Is combo**	This is used to select the type of condition.
2	**Data Type combo**	This field specifies the type of data with which a condition will work. There are five types of data: String, Numeric, DateTime, Boolean, and Expression. The data type affects how the reporting tool processes the condition. For example, if the data type is a string, then the methods that work with strings are used. In addition, depending on the type of data the list of available operators is automatically changed. For example, the Contains operator is available only for the String data type. The Expression data type provides the ability to specify an expression instead of the second value. In this case the reporting tool will not check the compatibility of the first and the second values of the condition. Therefore, the user should ensure that the expression entered is valid in order to prevent runtime errors.

3	**Column combo**	This is used to specify the column of the data source. The value of the column will be used as the first value of the condition.
4	**Operator combo**	This is used to specify the type of operator to be used when calculating the value of the condition.
5	**Value box**	This is used to specify the comparison value to be used when calculating the value of a condition. For some operations you may need to specify three values.

Operators

Operators enable you to define the circumstances in which a condition is deemed to be true. The operators available depend on the data type being operated upon, so only the appropriate operators will be available. For example, a logical condition can only be true or false, so it cannot be greater than anything making the greater than operator inappropriate for that data type.

The table below shows a list of operators and the data with which they can be used:

Operator	Types of data					Description
	String	Numerical	Date	Logic	Expression	
equal to	✓	✓	✓	✓	✓	If the first value is equal to the second, then the condition is true.
not equal to	✓	✓	✓	✓	✓	If the first value is not equal to the second, then the condition is true.
between		✓	✓		✓	If the first value is in the range, then the condition is true.
not between		✓	✓		✓	If the first value is not in the range, then the condition is true.
greater than		✓	✓		✓	If the first value is greater then the second value, then the condition is true.
greater than or equal to		✓	✓		✓	If the first value is greater then the second value of equal to the second value, then the condition is true.
less than		✓	✓		✓	If the first value is less then the second value, then the condition is true.
less then or equal to		✓	✓		✓	If the first value is less then the second value or equal to the second value, then the condition is true.

Operator	Types of data					Description
containing	✓					If the first value contains the second value, then the condition is true. This operator is used only for strings.
not containing	✓					If the first value does not contain the second value, then the condition is true. This operator is used only for strings.
beginning with	✓					If the first value starts with the second value, then the condition is true. This operatior is used only for strings.
ending with	✓					If the first value ends with the second value, then the condition is true. This operator is used only for strings.

Expression Condition

When you choose to use an Expression condition you define a text expression that returns a boolean value. The value returned determines whether or not the formatting is applied. The configuration panel is shown below:

1	**Field Is**	Field is used to select the type of conditions.
2	**Expression**	This field is used to define an expression that should return a boolean value.

For example, a suitable expression in C #:

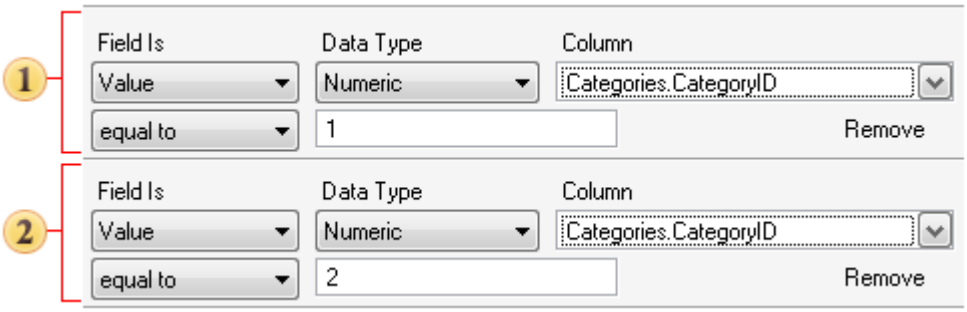
```
Customers.CustomerName == "MyCustomer"
```

If the expression cannot return a boolean value then the report generator will not be able to render the conditional formatting.

Important: The expression MUST return a boolean value or the conditional formatting will fail.

Multi Part Conditions

In some cases, one comparison operation may not be sufficient to define the condition. To allow for this situation Stimulsoft Reports allows you to specify a multi part condition. The picture below shows the condition editor a two level multi part condition:



1	**The first part of the condition**
2	**The second part of the condition**

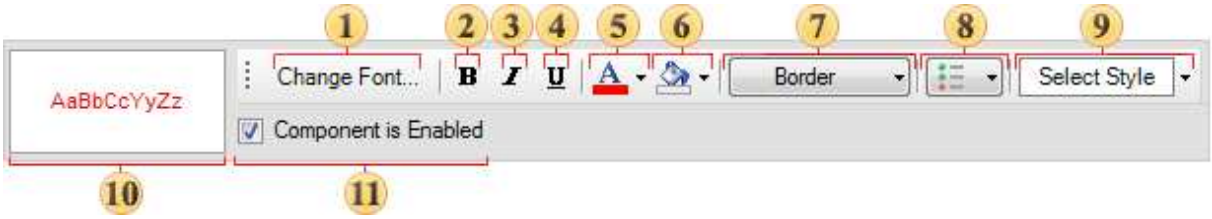
If you were to write this condition in code as a logical expression, it would look like this:

```
(Categories.CategoryID) = 1 or (Categories.CategoryID = 2)
```

It is possible to select the type of logical addition of the various parts of a multi part condition: the **logical AND** or the **Boolean OR**. To define this simply select the appropriate radio button

Defining Formatting

If the condition returns true when evaluated by the report engine the formatting of the component will be changed according to the design settings. Setting is carried out using the formatting panel. The picture below shows the components of the control panel:



1	**Font**	Used to select the font.
2	**Bold button**	Used to define the bold font style.
3	**Italic button**	Used to define the italic font style.
4	**Underlined button**	Used to define the underlined font style.
5	**Font Color Selector**	Used to define the text color.
6	**Background Color Selector**	Used to define the background color.
7	**Border**	Used to set borders.
8	**Control Menu**	Enables/Disables the components of the control panel.
9	**Style button**	This button is used to select a style to be applied.
10	**Pattern**	This shows a preview of how the control will look with the conditional formatting applied.
	Component is Enabled check box	This control lets to control how the result of a condition would affect on the Enabled property of the component.

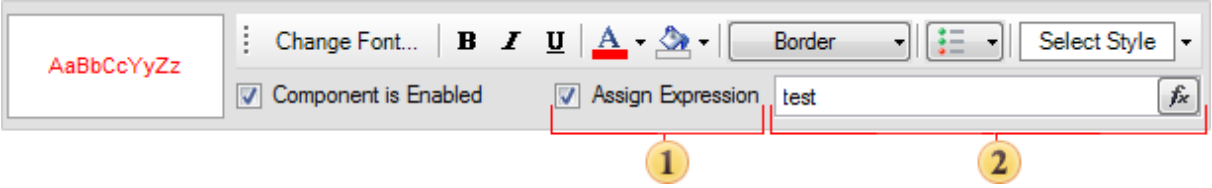
You can enable or disable the accessibility of the component in a report. For example, you can remove a page from a rendered report by setting a condition.

If the condition evaluates to true, then the component appearance will change according to settings made in this panel. If the component does not support the specified appearance (for example, because it has no Font property), the appearance will be automatically deleted.

In addition, you can control the availability of the control within the report using the Component is Enabled check box.

Conditional Formatting and Text Components

The conditions editor of text components has differences from other components. It has additional ability to assign text expression, if the condition is true. On the picture below the panel to edit conditions of the text component is shown.



1	**Assign expression**	This flag controls whether or not a text expression is used in the condition. If it is disabled then the expression is not used.
2	**Text expression**	The text expression that will be assigned to a text component if the condition is true.

Conditional Formatting and Cross-tables

The Cross Table condition editor has several differences from the standard condition editor. In particular there are signification differences when writing expressions within conditions, as it adds some special variables such as: **value, tag, tooltip,** and **hyperlink**.

The **value** variable contains the value of the cross table cell and can be used to calculate a condition:

```
tag > 50
```

In other words, if the value of the cell of a cross table is greater than 50, then the condition is true and formatting that was set in the condition will be applied to the cell.

The **tag**, **tooltip**, and **hyperlink** variables contain the calculated values of the **Tag**, **Tooltip,** and **Hyperlink** properties. For example, you may specify the name of a product in the **Tag** property of the cross table cell:

```
{Products.ProductName}
```

Suppose we wanted to highlight in red the cell of the cross table in which the Coffee product is described. This can be achieved by setting the formatting and using the following condition:

```
tag == "Coffee"
```

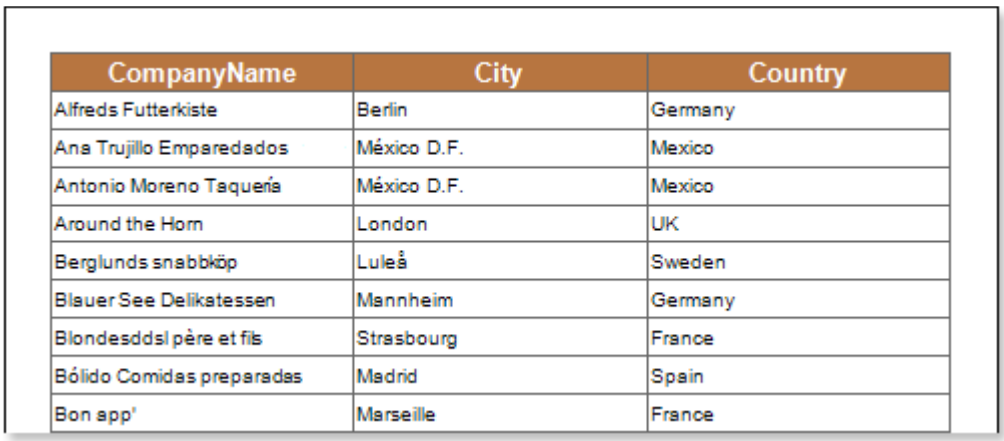
Visual Styles Menu

It is possible to enable/disable visual styles of a component using the conditional formatting. Enabling/disabling visual styles can be done in the visual styles menu. This menu provides the ability to make choice of those visual styles of the component, which will be applied to it for triggering the condition. The picture below shows the menu of visual styles:

1 The **Font Name** menu item. Enabling/Disabling this item provides an opportunity to change/not change the font in the components that match the condition. If this menu item is disabled (not checked), the font will not be changed, i.e. will remain by default;
2 The **Font Size** menu item. Enabling/Disabling this item provides an opportunity to change/not change the font size for components that match the condition. If this menu item is disabled (not checked), then the font size will not be changed, i.e. will remain by default;
3 The **Font Bold** menu item. Enabling of this item provides an opportunity to use bold font for the components that match to the condition. If this menu item is disabled (not checked), the font style will not be changed, i.e. will remain by default;
4 The **Font Italic** menu item. Enabling of this item provides an opportunity to use italic font for the components that match to the condition. If this menu item is disabled (not checked), the font style will not be changed, i.e. will remain by default;
5 The **Font Underline** menu item. Enabling of this item provides an opportunity to use the underlined font for components that match to the condition. If this menu item is disabled (not checked), the font style will not be changed, i.e. will remain by default;
6 The **Text Color** menu. Enabling of this item provides an opportunity to apply the text color for the components which correspond to the condition. If this menu item is disabled (not checked), the color of the text will not be changed, i.e. will remain by default;
7 The **Back Color** menu item. Enabling of this item provides an opportunity to apply the background color for the components that match to the condition. If this menu item is disabled (not checked), then the background color will not be changed, i.e. will remain by default;
8 The **Border** menu item. Enabling of this item provides an opportunity to change the borders of components. If this menu item is disabled (not checked), the borders of a component will not be changed, i.e. will remain by default;

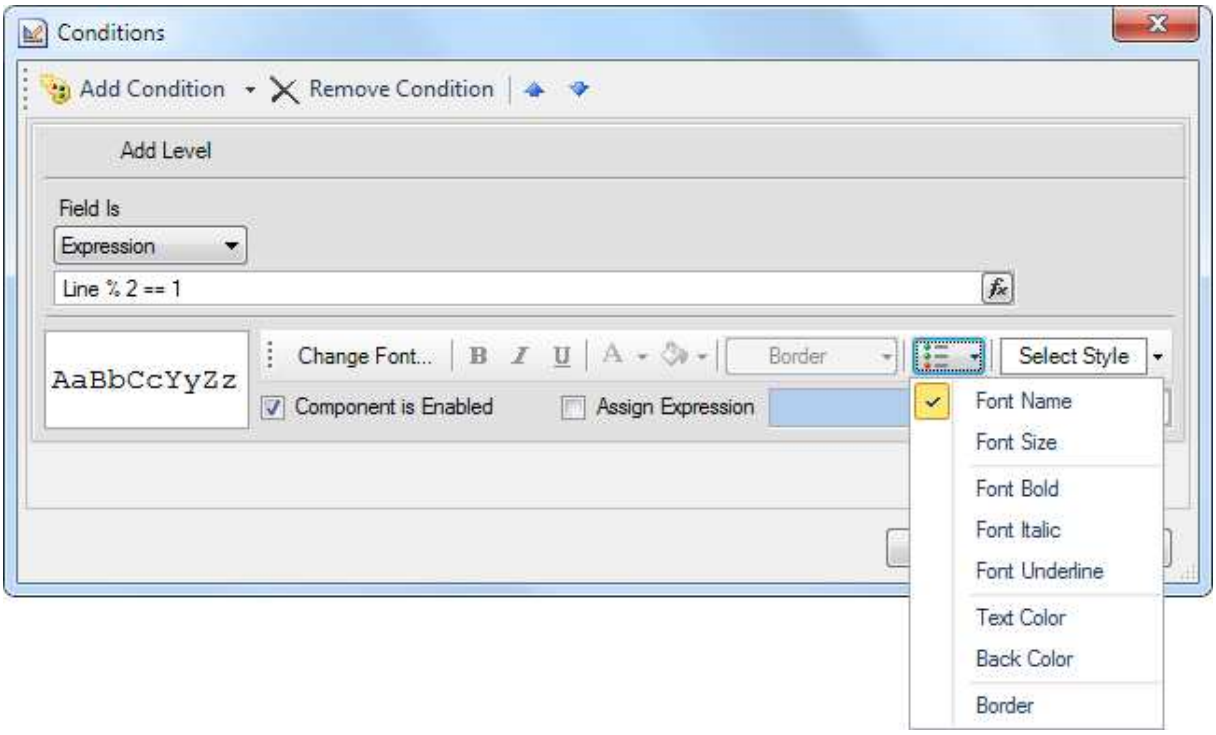
Font Name

Using conditional formatting it is possible to change the font of a text component. The picture below shows a page of a rendered report:

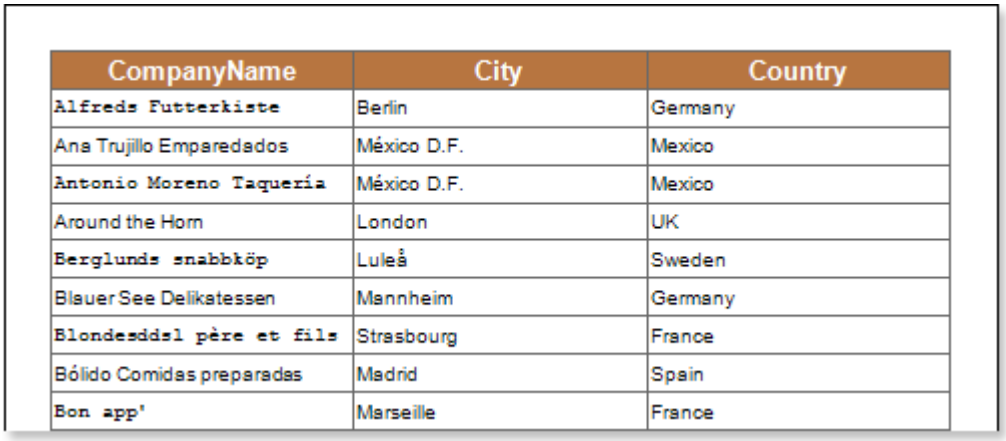


CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

For example, you can use different fonts to display the contents of a text component in the odd and even rows. To do this, select a text component, for example a text component with the **{Customers.CompanyName}** expression, in the **DataBand** and call the **Conditions** editor. Then, you must specify the condition, for example: **Line % 2 == 1**. Change the formatting options, in this case, the Font Name. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the font of the selected text component will be changed, depending on the condition. The picture below shows the page of the rendered report with conditional formatting:



CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

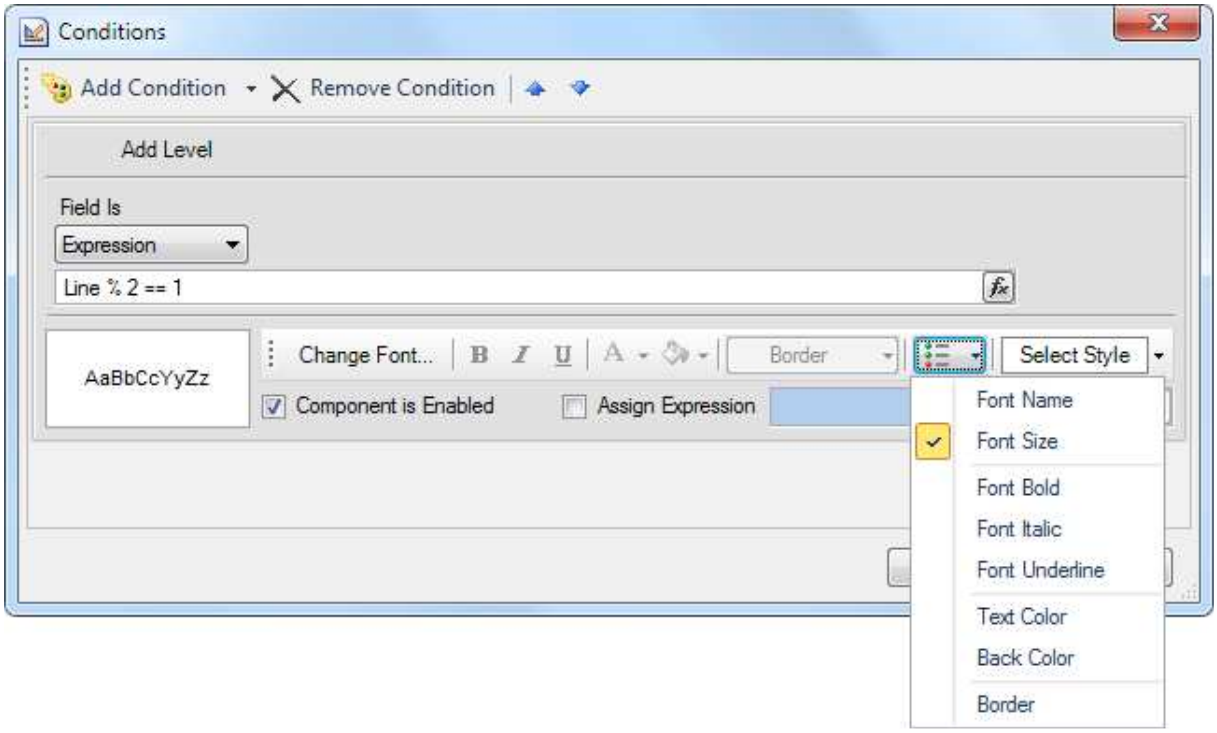
As can be seen in the picture above, the text components of the **CompanyName** column, located in the even and odd lines, use different fonts.

Font Size

Using conditional formatting it is possible to change the font size of a text component. Let us consider in more detail changing the font size of the contents of a text component. The picture below shows a page of a rendered report:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

For example, you can use different font sizes to display the contents of a text component in the odd and even rows. To do this, select a text component, for example a text component with the **{Customers.Country}** expression, in the **DataBand** and call the **Conditions** editor. Then, you must specify the condition, for example: **Line % 2 == 1**. Change the formatting options, in this case, the Font Size. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the font size of the selected text component will be changed, depending on the condition. The picture below shows the page of the rendered report with conditional formatting:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados y helad	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

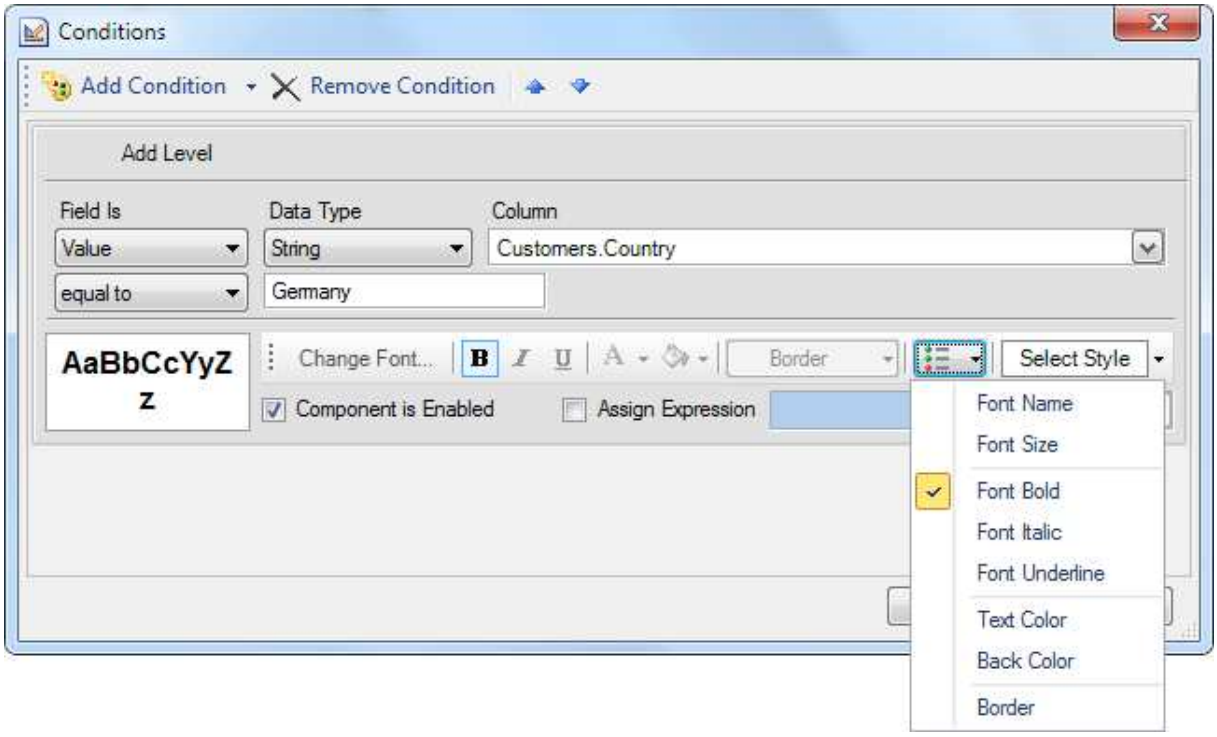
As can be seen in the picture above, the text components of the **Country** column, located in the even and odd lines, use different font sizes.

Font Bold

Using conditional formatting it is possible to apply the bold font for the text component. The picture below shows a page of a rendered report:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

For example, you can make a text bold for components that contain the **Germany** word in the **Country** column. Select a text component with the **{Customers.Country}** expression, in the **DataBand** and call the **Conditions** editor. Then, you should set a condition: select the **Customers.Country** data column, as the first value, and indicate the **Germany** word, as a second value. Also set the **Operation comparison** to the **Containing** value. Change the formatting parameters, in this case, set the font style to bold. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the bold font will be applied for the content of text components that match the specified condition. The picture below shows a page of the rendered report with conditional formatting:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	**Germany**
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	**Germany**
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

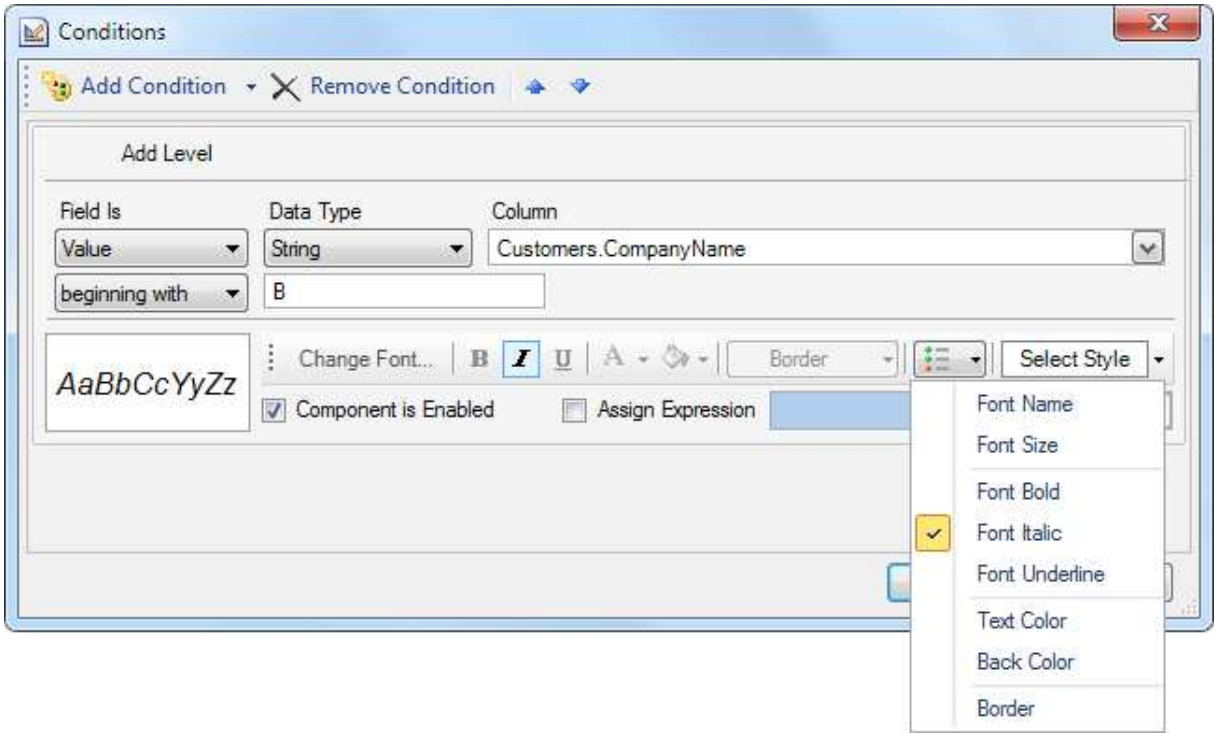
As can be seen in the picture above, lines of text components of the **Country** column which contain a **Germany** word are bold.

Font Italic

Using conditional formatting it is possible to apply the italic font for the text component. The picture below shows a page of a rendered report:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

For example, you can make a text italic for components that contain a **B** letter in the **CompanyName** column. Select a text component with the **{Customers.CompanyName}** expression, in the **DataBand** and call the **Conditions** editor. Then, you should set a condition: select the **Customers.CompanyName** data column, as the first value, and indicate the **B** letter, as a second value. Also set the **Operation comparison** to the **Beginning with** value. Change the formatting parameters, in this case, set the font style to italic. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the italic font will be applied for the content of text components that match the specified condition. The picture below shows a page of the rendered report with conditional formatting:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

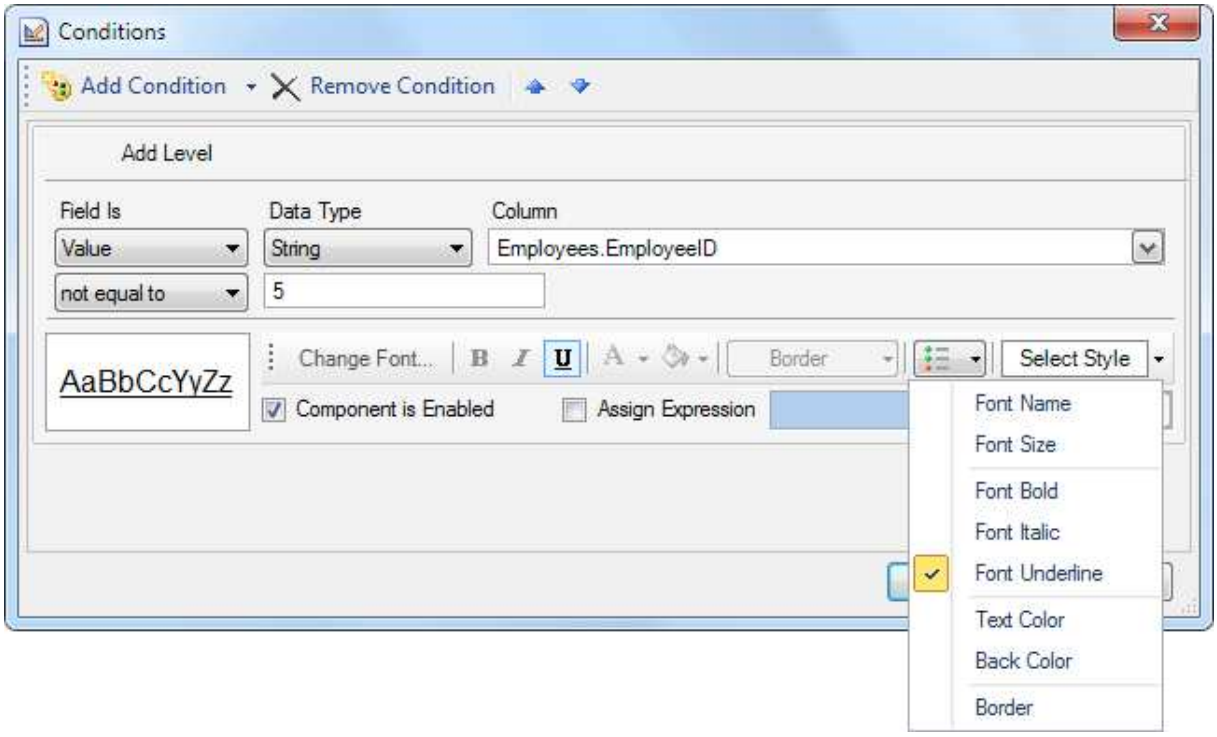
As can be seen in the picture above, lines of text components of the **CompanyName** column which starts with a **B** letter are italic.

Font Underlined

Using conditional formatting it is possible to apply the underlined font for the text component. The picture below shows a page of a rendered report:

EmployeeID	LastName	FirstName	Country
1	Davolio	Nancy	USA
2	Fuller	Andrew	USA
3	Leverling	Janet	USA
4	Peacock	Margaret	USA
5	Buchanan	Steven	UK
6	Suyama	Michael	UK
7	King	Robert	UK
8	Callahan	Laura	USA
9	Dodsworth	Anne	UK

For example, you can make a text underlined for components that contain a **Nancy** word in the **FirstName** column. Select a text component with the **{Employees.LastName}** expression, in the **DataBand** and call the **Conditions** editor. Then, you should set a condition: select the **Employees.FirstName** data column, as the first value, and indicate the **Nancy** letter, as a second value. Also set the **Operation comparison** to the **not equal to** value. Change the formatting parameters, in this case, set the font style to underlined. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the underlined font will be applied for the content of text components that match the specified condition. The picture below shows a page of the rendered report with conditional formatting:

EmployeeID	LastName	FirstName	Country
1	Davolio	Nancy	USA
2	Fuller	Andrew	USA
3	Leverling	Janet	USA
4	Peacock	Margaret	USA
5	Buchanan	Steven	UK
6	Suyama	Michael	UK
7	King	Robert	UK
8	Callahan	Laura	USA
9	Dodsworth	Anne	UK

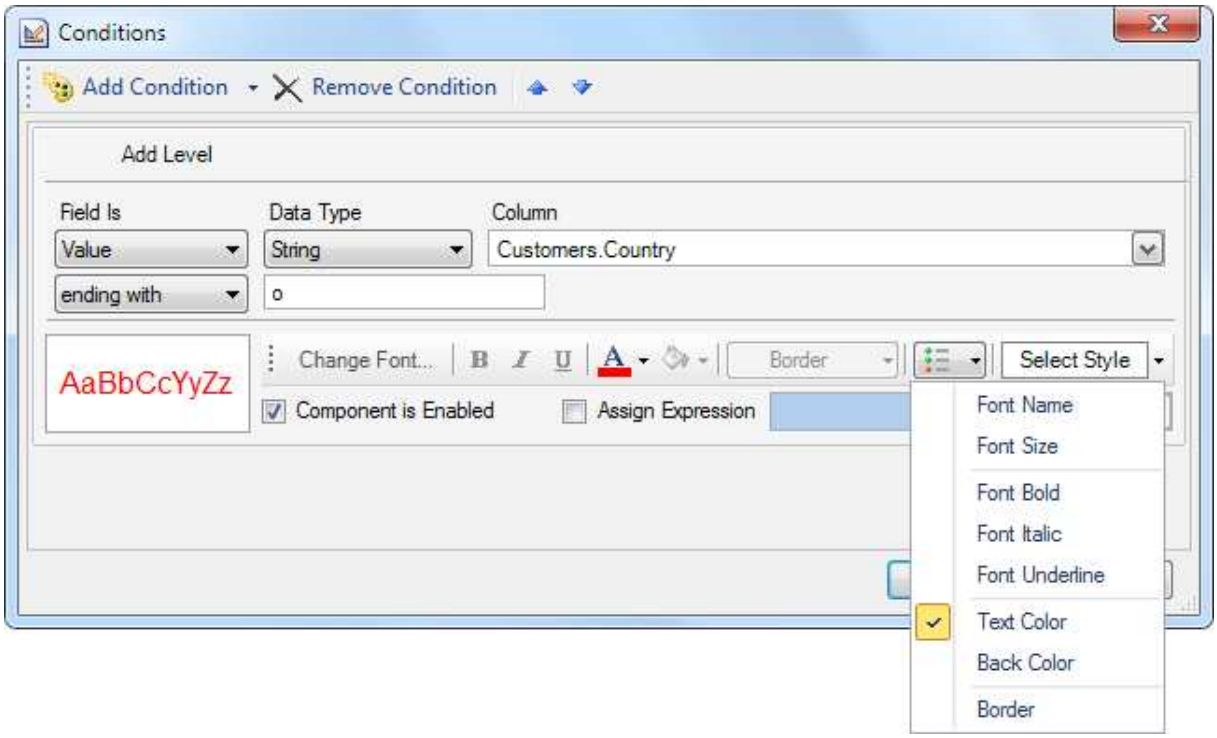
As can be seen in the picture above, lines of text components of the **FirstName** column which starts with the **Nancy** word are underlined.

Text Color

Using conditional formatting it is possible to apply the color for the text component. The picture below shows a page of a rendered report:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

For example, you can change a text color of entries which ends with an **o** letter in the **Country** column. Select a text component with the **{Customers.Country}** expression, in the **DataBand** and call the **Conditions** editor. Then, you should set a condition: select the **Customers.Country** data column, as the first value, and indicate the **o** letter, as a second value. Also set the **Operation comparison** to the **ending with** value. Change the formatting parameters, in this case, change the text color. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the text color will be applied for the content of text components that match the specified condition. The picture below shows a page of the rendered report with conditional formatting:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

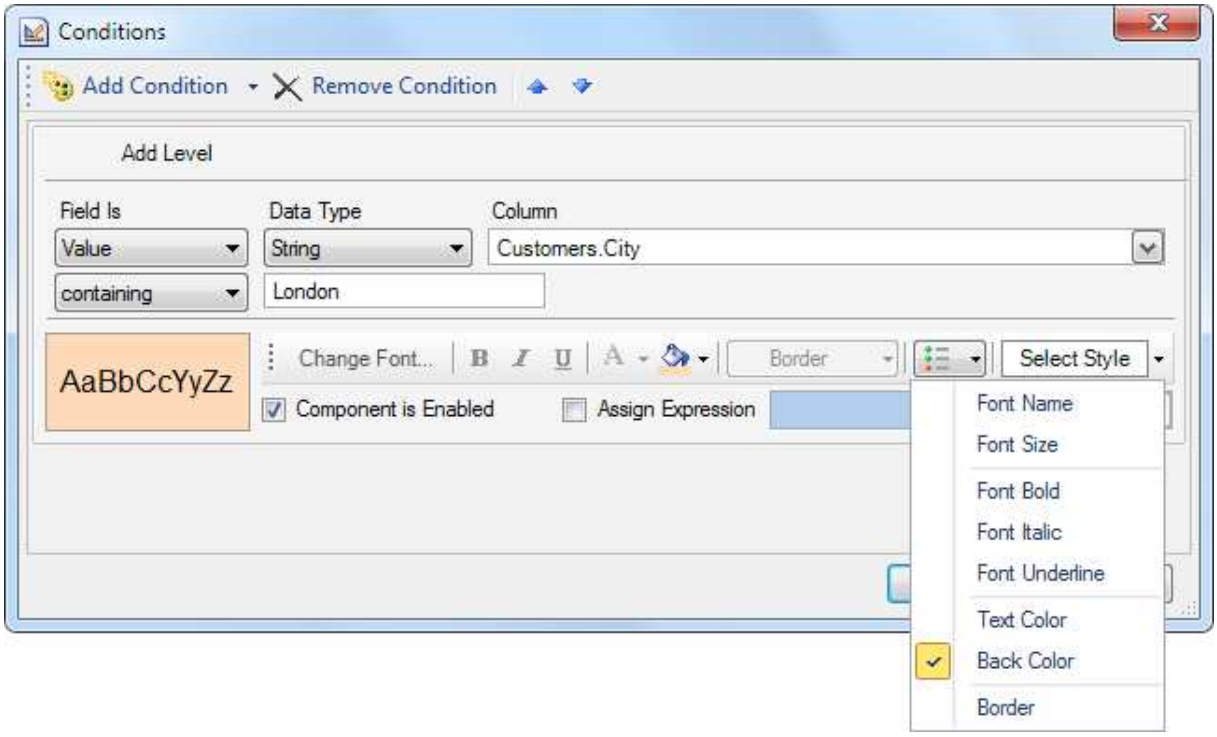
As can be seen in the picture above, lines of text components of the **Country** column which ends with the **o** letter are red.

Back Color

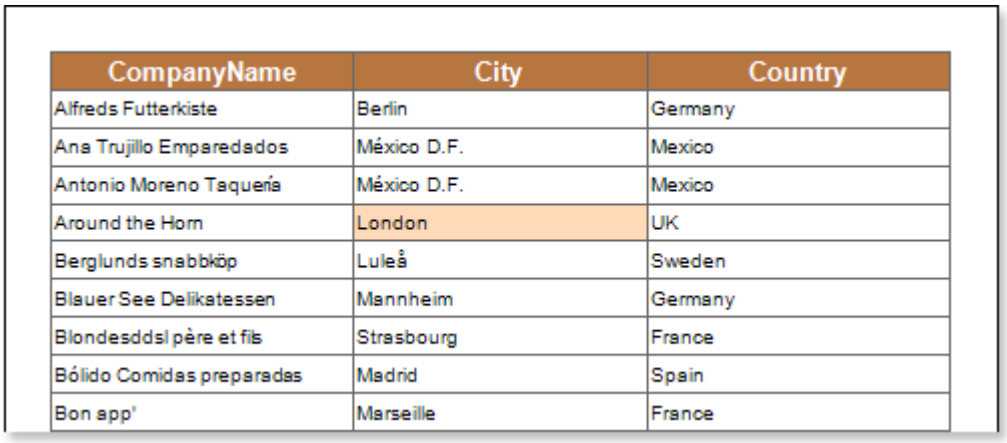
Using conditional formatting it is possible to apply the background color for the text component. The picture below shows a page of a rendered report:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

For example, you can change the background color of text components which contain a **London** word in the **City** column. Select a text component with the **{Customers.City}** expression, in the **DataBand** and call the **Conditions** editor. Then, you should set a condition: select the **Customers.City** data column, as the first value, and indicate the **London** word, as a second value. Also set the **Operation comparison** to the **containing** value. Change the formatting parameters, in this case, change the background color. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the background color will be applied for the content of text components that match the specified condition. The picture below shows a page of the rendered report with conditional formatting:



CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

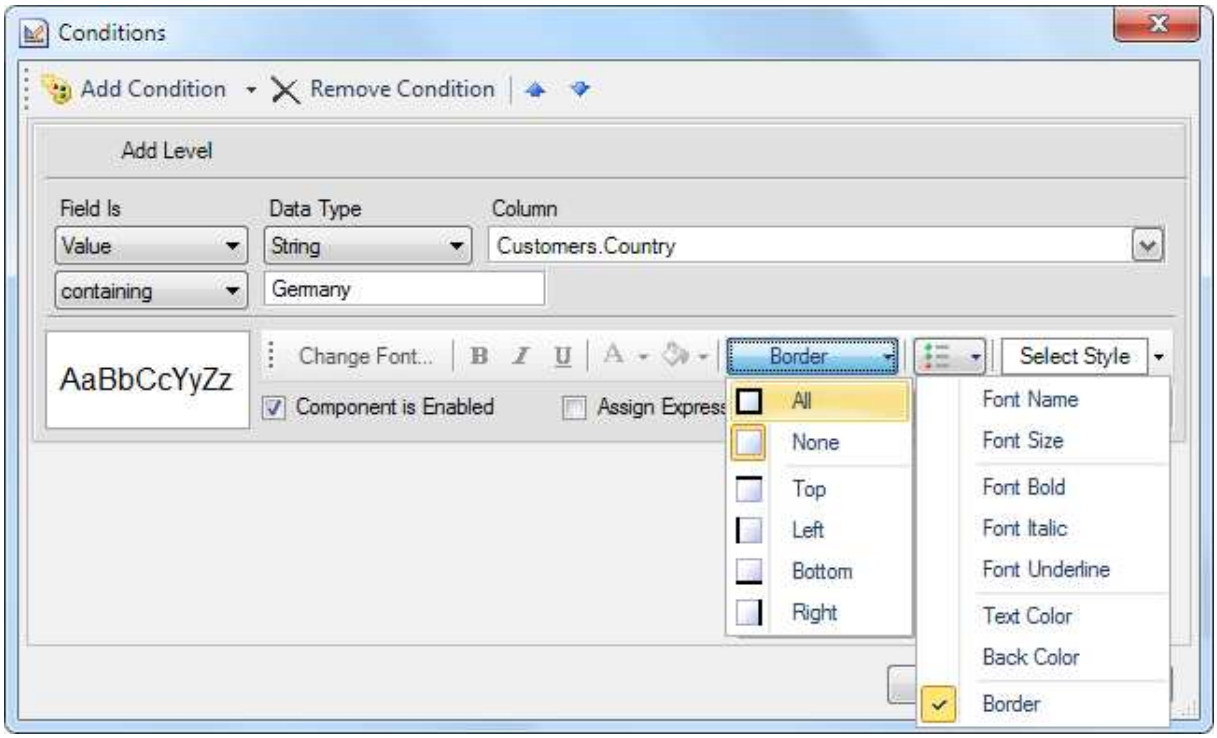
As can be seen in the picture above, background color of text components of the **City** column which contain the **London** word, will be changed.

Borders

Using conditional formatting it is possible to apply borders for the text component. The picture below shows a page of a rendered report:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

For example, you can set borders of text components which contain a **Germany** word in the **Country** column. Select a text component with the **{Customers.Country}** expression, in the **DataBand** and call the **Conditions** editor. Then, you should set a condition: select the **Customers.Country** data column, as the first value, and indicate the **Germany** word, as a second value. Also set the **Operation comparison** to the **containing** value. Change the formatting parameters, in this case, set borders. It is possible to configure showing borders. The following options are available: **All** (show all borders), **None** (Do not show borders), **Top** (show a top border), **Left** (show a left border), **Bottom** (show a bottom border), **Right** (show a right border). The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the borders will be set for the text components that match the specified condition. The picture below shows a page of the rendered report with conditional formatting:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

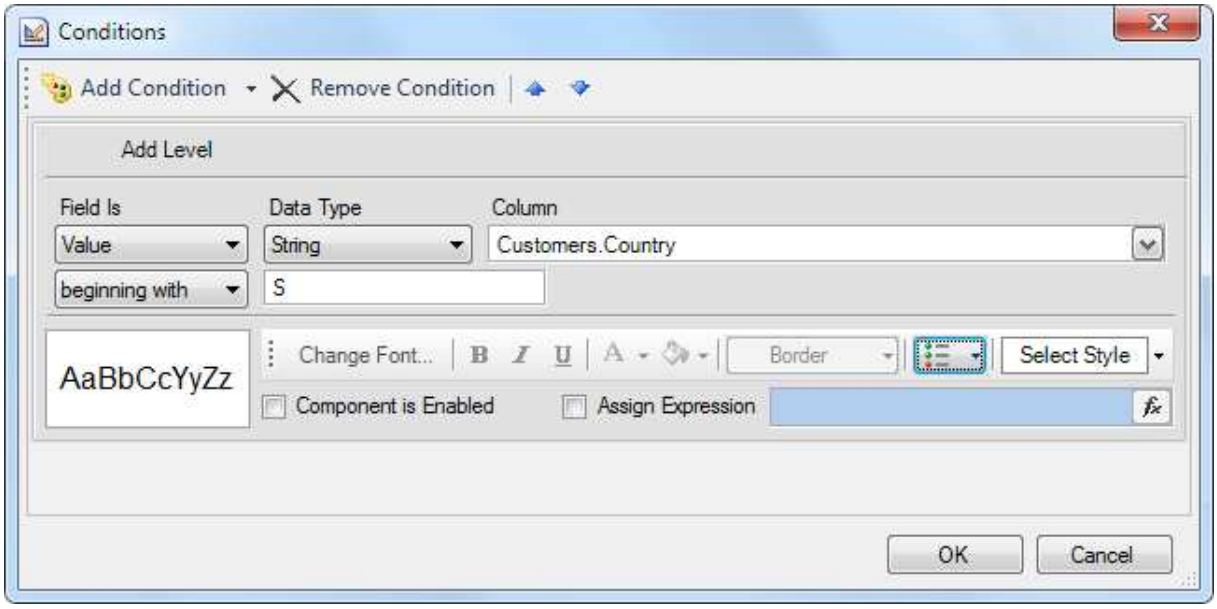
As can be seen in the picture above, borders of text components of the **Country** column which contain the **Germany** word, will be set.

Enabling Component

Using conditional formatting it is possible to show/hide the text component. The picture below shows a page of a rendered report:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France

For example, you can hide the text components which contain a **S** letter in the **Country** column. Select a text component with the **{Customers.Country}** expression, in the **DataBand** and call the **Conditions** editor. Then, you should set a condition: select the **Customers.Country** data column, as the first value, and indicate the **S** letter, as a second value. Also set the **Operation comparison** to the **Beginning with** value. Change the formatting parameters, in this case, uncheck the **Component Is Enabled** check box. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the borders the text components that match the specified condition will be hidden. The picture below shows a page of the rendered report with conditional formatting:

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	
Bon app'	Marseille	France

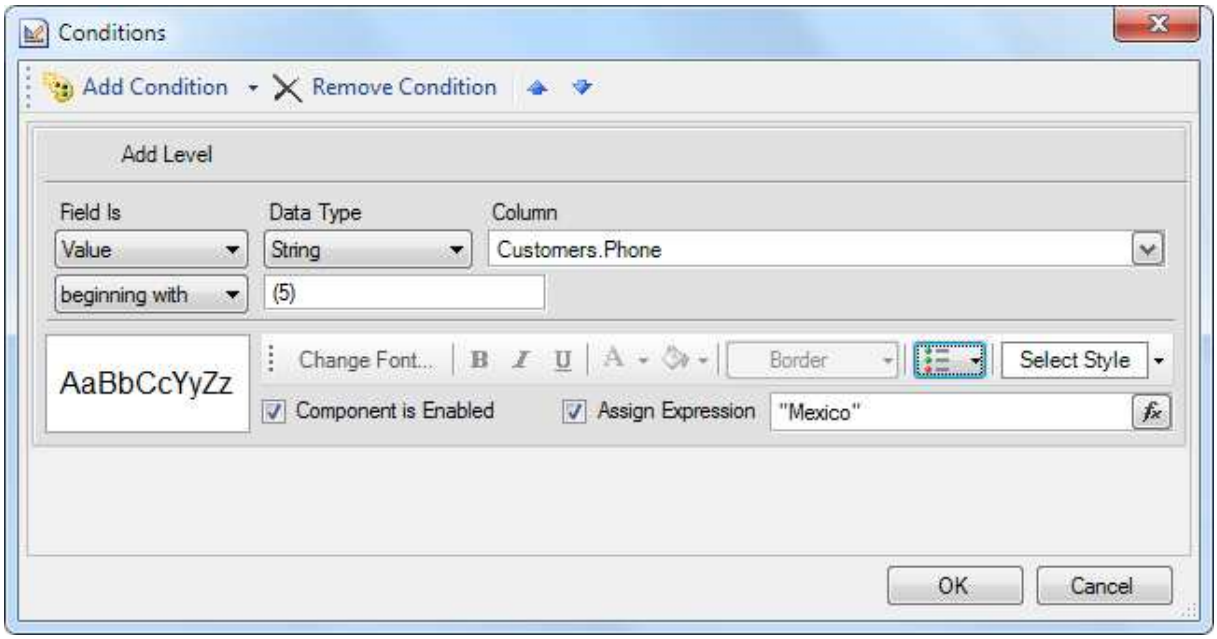
As can be seen in the picture above, the text components of the **Country** column which lines start with the **S** letter are changed.

Assigning Expression

Using conditional formatting it is possible, in a text component, to change the text, i.e. replace its textual expression on a text expression, specified in the condition. The picture below shows a page of a rendered report:

ContactName	Phone	Fax
Maria Anders	030-0074321	030-0076545
Ana Trujillo	(5) 555-4729	(5) 555-3745
Antonio Moreno	(5) 555-3932	
Thomas Hardy	(171) 555-7788	(171) 555-6750
Christina Berglund	0921-12 34 65	0921-12 34 67
Hanna Moos	0621-08460	0621-08924
Frédérique Citeaux	88.60.15.31	88.60.15.32
Martín Sommer	(91) 555 22 82	(91) 555 91 99
Laurence Lebihan	91.24.45.40	91.24.45.41
Elizabeth Lincoln	(604) 555-4729	(604) 555-3745

For example, it is necessary to assign an expression to all text components, which entries in the **Phone** column will start with the (**5**) characters. Select a text component with the **{Customers.Phone}** expression in the **DataBand** and call the **Conditions** editor. Then, you should set a condition: select the **Customers.Phone** column data, as the first value, and specify the (**5**) character, as a second value. Also set the **Operation comparison** to the **Beginning with** value. Change the formatting options, in this case, enable the **Assign Expression** and specify an expression to which it will be replaced on. For example, specify the "Mexico" expression. The picture below shows the **Conditions** editor dialog box:



After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, assigning of the text expression in the text components that match the specified condition will be done. The picture below shows a page of the rendered report with conditional formatting:

ContactName	Phone	Fax
Maria Anders	030-0074321	030-0076545
Ana Trujillo	Mexico	(5) 555-3745
Antonio Moreno	Mexico	
Thomas Hardy	(171) 555-7788	(171) 555-6750
Christina Berglund	0921-12 34 65	0921-12 34 67
Hanna Moos	0621-08460	0621-08924
Frédérique Citeaux	88.60.15.31	88.60.15.32
Martín Sommer	(91) 555 22 82	(91) 555 91 99
Laurence Lebihan	91.24.45.40	91.24.45.41
Elizabeth Lincoln	(604) 555-4729	(604) 555-3745

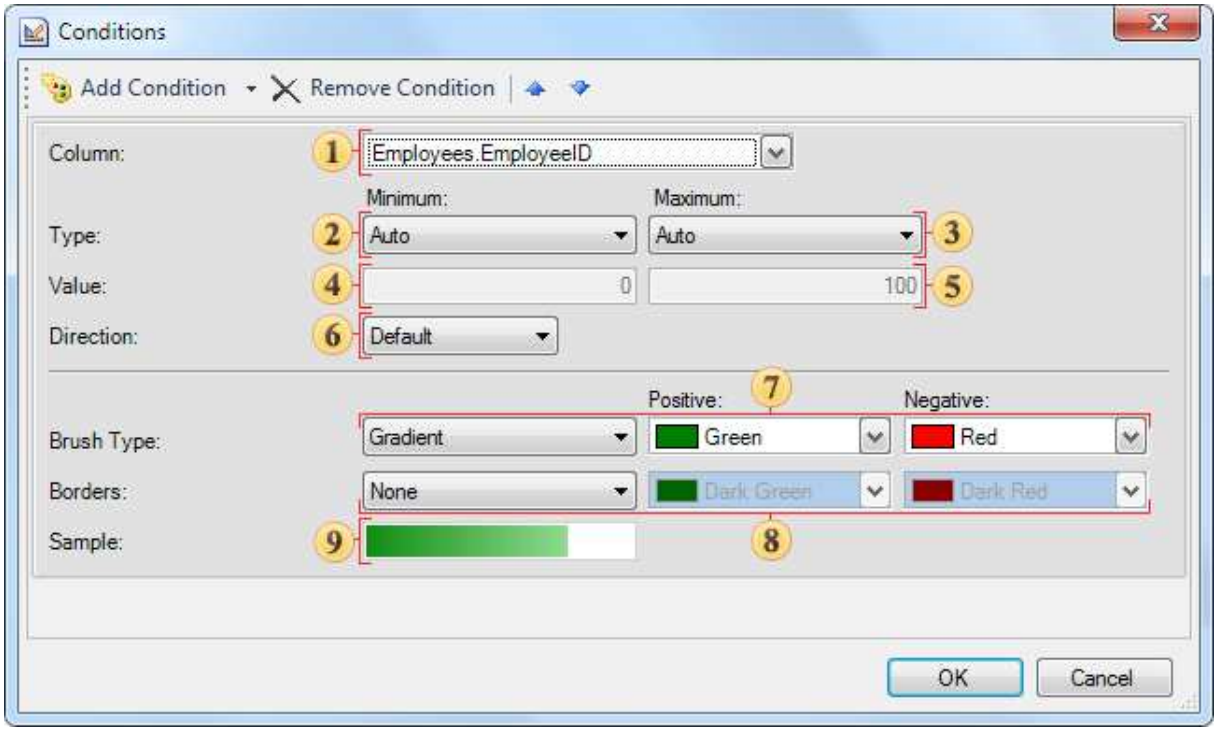
As can be seen in the picture above, assigning of expressions in the text components of the **Phone** column which entries start with the (**5**) character will be done.

Data Bar Condition

The **Data Bar** condition provides an opportunity to visually display the dynamics of changing values of a data column. The **Data Bar** condition works following principles described below. All the values in the selected data column are analyzed, the minimum and maximum values are determined. Minimum corresponds to 0 percent, maximum - 100 percent. When drawing each component, to which this condition is applied, a value from the selected data column will be specified. Then, the percentage of this value is calculated from the minimum to maximum range. Depending on the percentage, the **Data bar** is rendered. If the value is close to the maximum, the greater length a data bar would be. If the value is close to or equal to a minimum value, the data bar will be almost unfilled. The picture below shows a page of the report:

EmployeeID	LastName	FirstName	Country
1	Davolio	Nancy	USA
2	Fuller	Andrew	USA
3	Leverling	Janet	USA
4	Peacock	Margaret	USA
5	Buchanan	Steven	UK
6	Suyama	Michael	UK
7	King	Robert	UK
8	Callahan	Laura	USA
9	Dodsworth	Anne	UK

Add the **Data Bar** condition. To do this, select a text component, for example a text component with the **{Employees.EmployeeID}** expression. Add the **Data Bar** expression. Change parameters of the condition. The picture below shows the **Conditions** dialog box:



1 The **Column** field. This field indicates the data column from which values will be taken for drawing the Data Bar.
2 The **Type** field is used to change the type of a minimum value. The following types are available: **Auto** defines the minimum value in the selected data column, and if it is greater than zero, then reset to zero; **Percentage** is used to specify a minimum value as a percentage, the **Value** provides an opportunity to specify a minimum value as a numerical value, **Minimum** defines the minimum value in the selected data column and resets it to null.
3 The **Type** field is used to change the type of a maximum value. The following types are available: **Auto** defines the minimum value in the selected data column, and if it is less than zero, then reset to zero; **Percentage** is used to specify a maximum value as a percentage, the **Value** provides an opportunity to specify a maximum value as a numerical value, **Maximum** defines the maximum value in the selected data column and resets it to null.
4 The **Value** field for a minimum value.
5 The **Value** field for a maximum value.
6 The **Direction** field is used to change the direction of drawing the **Data Bar**. The following directions are available: **Left to Right**, **Right to Left**, **Default** defines the direction of the Data Bar, depending on the **Right to Left** property of the text component.
7 The Data Bar parameters include: the **Brush Type** is used to choose the brush type (gradient or solid); the **Positive** field is used to change the color a Data Bar for positive values; the **Negative** field is used to change the color a Data Bar for negative values.
8 The Borders parameter include: the **Borders** field is used to choose the type of a border (none or solid); the **Positive** field is used to change the border color a Data Bar for positive values; the **Negative** field is used to change the border color a Data Bar for negative values.
9 The **Sample** field shows an example of a Data Bar.

After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. The picture below shows a page of the rendered report with conditional formatting:

EmployeeID	LastName	FirstName	Country
1	Davolio	Nancy	USA
2	Fuller	Andrew	USA
3	Leverling	Janet	USA
4	Peacock	Margaret	USA
5	Buchanan	Steven	UK
6	Suyama	Michael	UK
7	King	Robert	UK
8	Callahan	Laura	USA
9	Dodsworth	Anne	UK

As can be seen from the picture above, the EmployeeID value includes the numbers from 1 to 9, where 1 is the minimum value, and 9 is the maximum one. And according to the changing dynamics of values a data bar will be drawn.

Negative values

In the column of data from which values are taken when displaying the histogram may be found both positive and negative values. In this case, analysis of all the values in the selected column of data, determined the minimum and maximum values. The minimum value is 0 percent, maximum - 100 percent. Next, we determine a zero, ie beginning from which a histogram of negative and positive values. For example, the minimum value is -1, while the maximum is three, ie percentage of negative values in the absolute values of band reception is 25 percent and 75 percent positive. Hence the beginning, from which will be constructed histogram is 25 per cent of the length of the component from its left border and 75 percent of the length of the component from its right boundary (at the direction of the histogram from left to right). Histogram of negative values will be rendered in a color that is selected in the Negative (Negative), and the histogram of positive values of a color that is selected in the Positive (Positive). The figure below shows an example of a rendered report with negative and positive values:

ProductName	QuantityPerUnit	UnitPrice
Geitost	2,5	62
Guaraná Fantástica	4,5	-30
Konbu	6	-26
Filo Mix	7	-12
Tourtière	7,45	-29
Rhönbräu Klosterbier	7,75	75
Tunnbröd	9	11
Teatime Chocolate Biscuits	9,2	-25
Zaanse koeken	9,5	-14
Rogede sild	9,5	-45
Jack's New England Clam Chowder	9,65	35

Also of note: if the parameter direction (Direction) is set to Left to Right (Left to right) will be constructed from the start of drawing to the left edge of the component, ie from right to left, if the parameter direction (Direction) is set to Right to Left (Right to left), the histogram is built from start to draw the right edge of the component, ie, left to right. The figure below shows an example of a rendered report with negative and positive values:

ProductName	QuantityPerUnit	UnitPrice
Geitost	2,5	62
Guaraná Fantástica	4,5	-30
Konbu	6	-26
Filo Mix	7	-12
Tourtière	7,45	-29
Rhönbräu Klosterbier	7,75	75
Tunnbröd	9	11
Teatime Chocolate Biscuits	9,2	-25
Zaanse koeken	9,5	-14
Rogede sild	9,5	-45
Jack's New England Clam Chowder	9,65	35

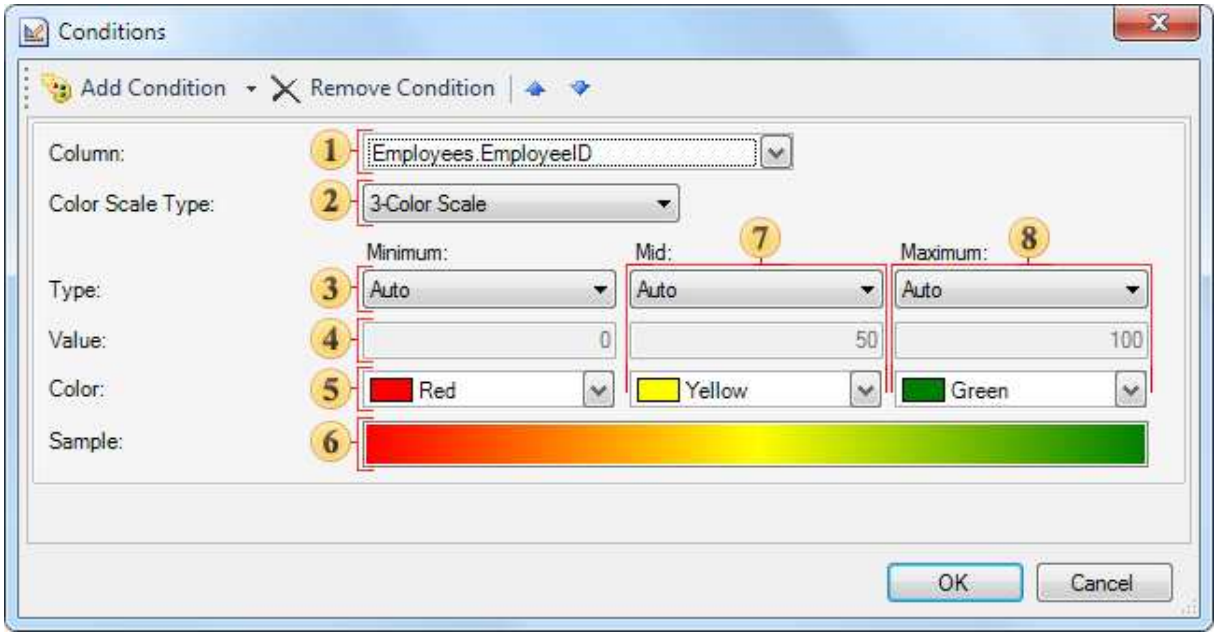
As can be seen in the picture above, the background color depending on the value in a color scale is changed in text components.

Color Scale Condition

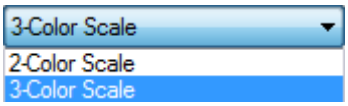
The **Color Scale Condition** allows selecting a component with a color in the rendered report, to which will this condition corresponds. The **Color Scale Condition** is working according to the the following principle: if the color scale consists of 2 colors (minimum and maximum), then the minimum and maximum values for selected data columns are specified. Values that correspond to the maximum and minimum values are indicated with colors. For other values, which are taken from selected data columns, the location in the color scale is calculated.Depending on location in color scale, the color is assigned to this value, so the color is assigned to the component. If the minimum value is equal to or less than the specified minimum in the condition, that means it will be a boundary minimum value and will use the color, chosen for the minimum value. If the maximum value in the data column is equal to or greater than the specified maximum in the condition, then it will be a maximum boundary value, and will use the color selected for a maximum value. If the value is in the middle between the minimum and maximum value, then the background color of a component with this value will be an interpolated color for minimum and maximum values. If the color scale consists of 3 values (low, medium, high), then the minimum, medium and maximum values are defined. For each value, which is taken from the selected data column, the position in the color scale is calculated depending on the location of the value and the color is assigned. So the color of the component is changed. The color scale represents a smooth transition between the three colors: the color from minimum to medium, and the color from medium to maximum. The background color of a component with a value that is strictly in the middle between the minimum and average value will be an interpolated color of minimum and medium values. The background color of a component with a value that is strictly in the middle between the average and maximum value will be an interpolated color of medium to maximum values. The picture shows a page of the report:

EmployeeID	LastName	FirstName	Country
1	Davolio	Nancy	USA
2	Fuller	Andrew	USA
3	Leverling	Janet	USA
4	Peacock	Margaret	USA
5	Buchanan	Steven	UK
6	Suyama	Michael	UK
7	King	Robert	UK
8	Callahan	Laura	USA
9	Dodsworth	Anne	UK

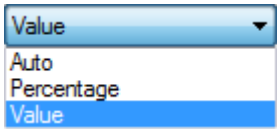
Add the **Color Scale Condition**. To do this, select a text component, for example a component with the **{Employees.EmployeeID}** expression. Add a **Color Scale Condition**. Change the parameters of the condition. The picture below shows the **Conditions** dialog:



1 The **Column** field. This field indicates the data column from which the value for the condition will be taken;
2 The **Color Scale Type** fields provides an opportunity to choose the type of color scheme: 2-color scales, or 3-color scales. The picture below shows the menu to select the type a of color scale:



3 The **Type** field provides an opportunity to change the type of a value that will be specified in the **Value** field for a minimum color scale. The picture below shows the menu to select the type of a value:



4 The **Value** field. Used for a minimum color scale;
5 The **Color** filed. Used for a minimum color scale;
6 The **Sample** field. Shows a color scale in the report how it will look like from minimum to medium and from medium to maximum. If you select the color scale 2-color scales, then in this field a color gradient from minimum to maximum will be displayed;

7 A group of parameters (Type, Value, Color) of the medium color scale;
8 A group of parameters (Type, Value, Color) with a maximum color scale.

After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, depending on the value of the component, the background of a text component will be changed. The picture below shows a rendered page of the report with conditional formatting:

EmployeeID	LastName	FirstName	Country
1	Davolio	Nancy	USA
2	Fuller	Andrew	USA
3	Leverling	Janet	USA
4	Peacock	Margaret	USA
5	Buchanan	Steven	UK
6	Suyama	Michael	UK
7	King	Robert	UK
8	Callahan	Laura	USA
9	Dodsworth	Anne	UK

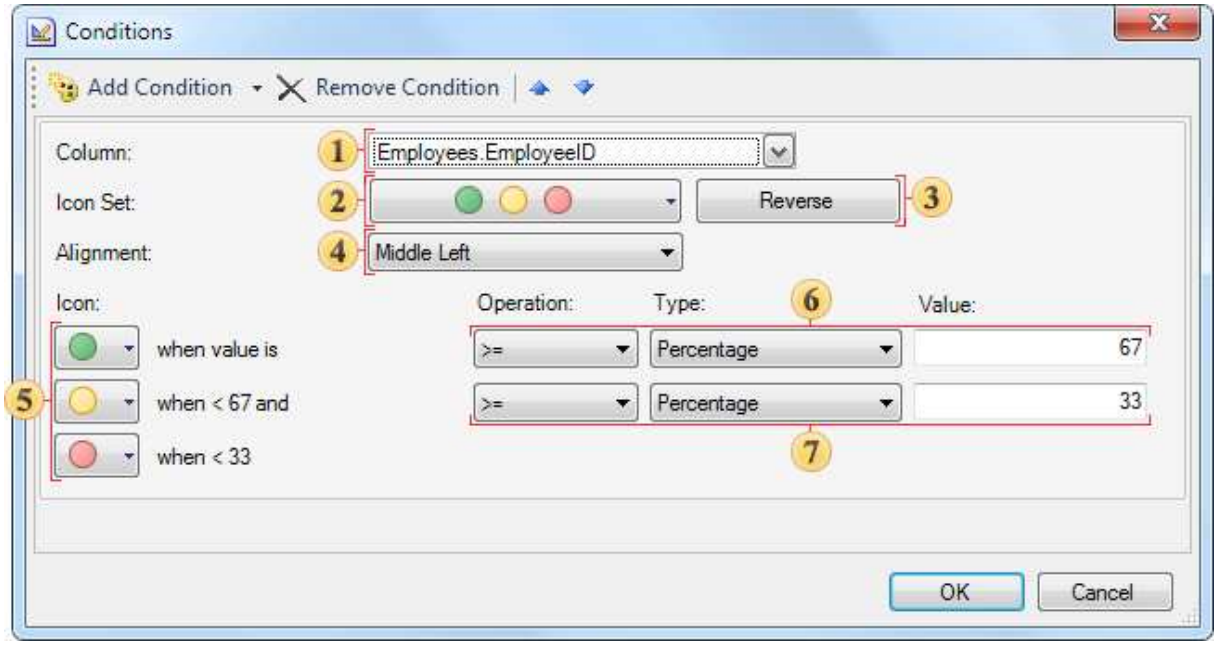
As can be seen in the picture above, the background color depending on the value in a color scale is changed in text components.

Icon Set Condition

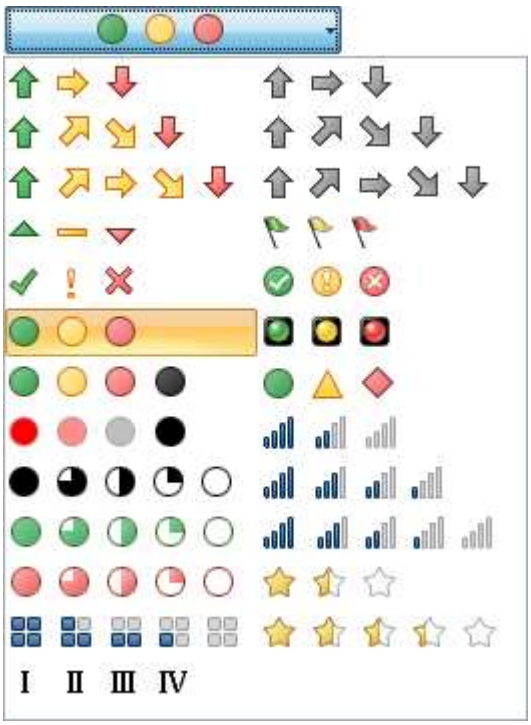
The **Icon Set** condition is used to identify the component with an icon to which a condition is applied. The **Icon Set** works the following way. The minimum and maximum values for all values in the selected data column are defined first. All calculated values are in the range from 0 to 100 percent. A group of icons is selected. Then, the condition and boundary values (for example 33 per cent and 67 per cent) for each icon are set. If, for example, a group of three icons is selected, each of these selected icons have a subrange. In this case, each of the icons has subrange in 33 percent (from 0 to 33, from 33 to 67, from 67 to 100). This allows you to mark a component with an appropriate icon depending on the value. The picture below shows a page of the report:

EmployeeID	LastName	FirstName	Country
1	Davolio	Nancy	USA
2	Fuller	Andrew	USA
3	Leverling	Janet	USA
4	Peacock	Margaret	USA
5	Buchanan	Steven	UK
6	Suyama	Michael	UK
7	King	Robert	UK
8	Callahan	Laura	USA
9	Dodsworth	Anne	UK

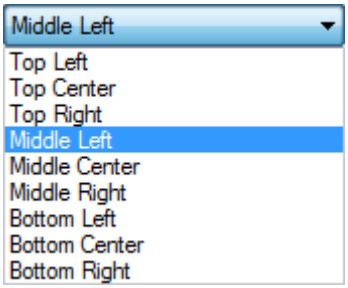
Add an **Icon Set** condition. To do this, select a text component, for example a component with the **{Employees.Country}** expression. Add the **Icon Set** condition. Change the parameters of the condition. The picture below shows the **Conditions** dialog:



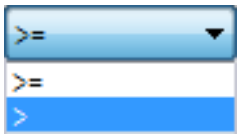
1 The **Column** field. This field is used to choose a data column from which values for the condition will be taken. For example, choose the **{Employees.EmployeeID}** data column;
2 A menu used for selecting a group of icons. The picture below shows the menu of selecting icons:



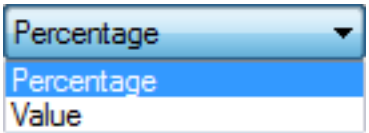
3 The **Reverse** button is used to change the location of icons in reverse order. The order of the icons is displayed in the 5 **Icon** field.
4 The **Alignment** field is used to align icons in text components. The picture below shows the **Alignment** menu options:



5 The **Icon** field shows the order of icons, and provides an opportunity to change the icon for each value in the report;
6 A sub-condition, includes: the **Operation**, **Type**, and **Value** fields. In this case, this is the first sub-condition. The **Operation** field is used to change the type of operation of the first sub-condition. The picture below shows the operations menu:



The **Type** field is used to change the type of a value of the first sub-condition. There are two values: **Percentage** and **Value**. The picture below shows the menu to select the type of a value:



In the **Value** filed the value of a sub-condition is indicated.

7 A sub-condition includes: the **Operation**, **Type**, and **Value** fields. In this case, it is the second sub-condition.

After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the appropriate icon for a text component will be applied. The picture below shows a page of the rendered report with conditional formatting:

EmployeeID	LastName	FirstName	Country
1	Davolio	Nancy	USA
2	Fuller	Andrew	USA
3	Leverling	Janet	USA
4	Peacock	Margaret	USA
5	Buchanan	Steven	UK
6	Suyama	Michael	UK
7	King	Robert	UK
8	Callahan	Laura	USA
9	Dodsworth	Anne	UK

As can be seen in the picture above, the icon depending on the value of a condition will be applied to each text component.

Output Text Parameters

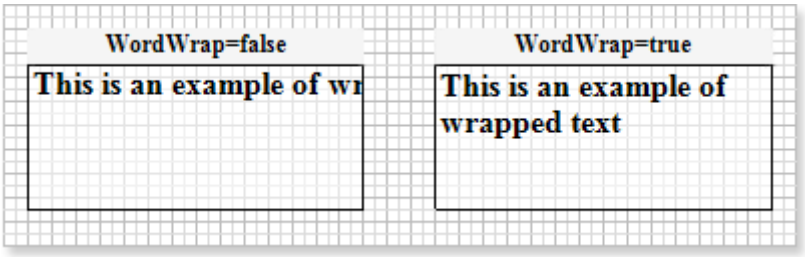
Stimulsoft Reports has a number of ways for handling, processing and showing a text.

The following components to display the text are:
1. **Text** is the basic component to output text in the report. The component supports a large number of different settings, processing and displaying text;
2. **RichText** is a component used to output of an **RTF** text;
3. **Text in Cells** is a special component to output a text in a cell.

The **Text** component abilities will reviewed in next articles.

Multiline Text

If the text cannot be put on one line it will be trimmed by default. If it is required to put a text on some lines, then you should set the word wrap. You should set the **TextOptions.WordWrap** property of the **Text** component to **true**. When the text is wrapped on a new line, vertical and horizontal alignments are used.



Trimming in the End of Text Line

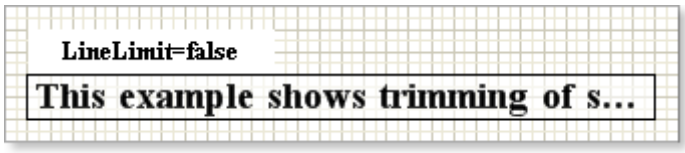
If there is not enough space to put whole text line in the text component, then, using the **TextOptions.Trimming** property, it is possible to customize text trimming. It has the following values:

None - the text is trimmed strictly by the edge of a text component or, if it is a multiline text, by the last visible word;

Character - the line is trimmed after the last visible character;

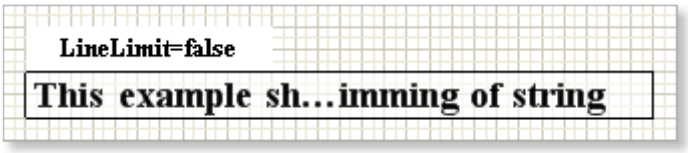
Word - the line is trimmed by the last visible word;

Ellipsis Character – last characters of a word are changed on omission points;



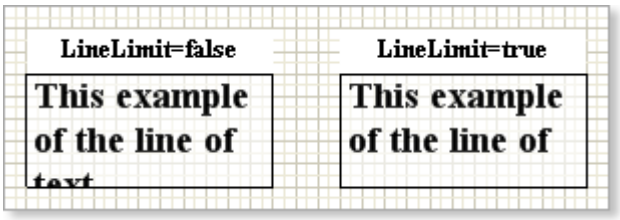
Ellipsis Word - omission points are added after the last visible word;

Ellipsis Path - the middle of a line is changed to dots so as the beginning and the end of a text line can be visible.



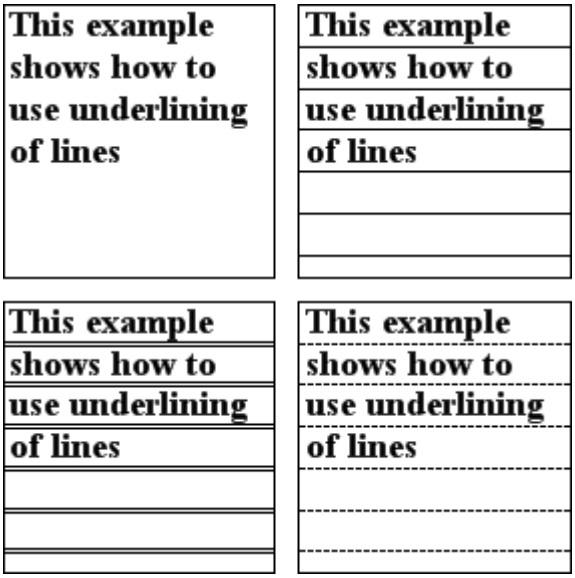
Prevent Showing Incompletely Visible Lines

Often it is necessary to output text and do not show vertically trimmed lines on the bottom of a component. If to set the **LineLimit** property to **true**, then only full lines will be output. Absence of additional line may change the word wrap.



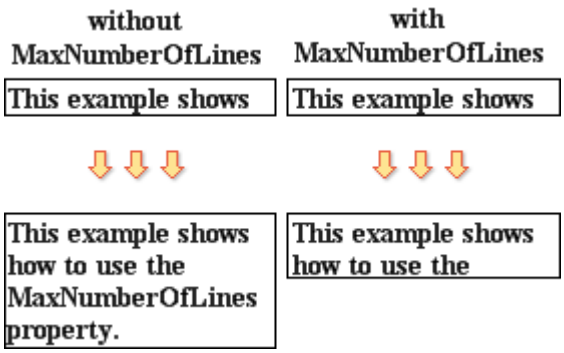
Lines of Underlining

If it is necessary to underline the **Text** component with horizontal lines, then it is possible to use the **LinesOfUnderline** property of the text component. Using this property it is possible to select style of underlining. If to select the **None** style**,** then there will not be any underlining.



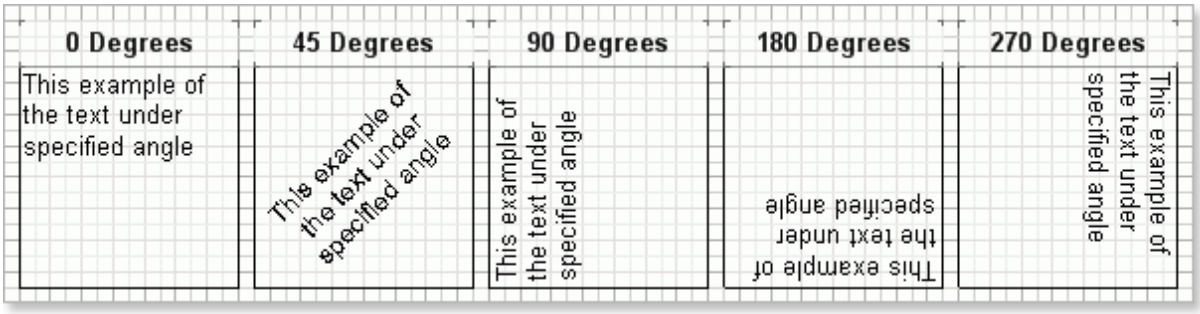
Maximal Number of Lines

How to make the **Text** component, when increasing the vertical size, increase it on the maximal number of horizontal lines? Use the **MaxNumberOfLines** property. By default, this property is equal in zero and the component will be increased vertically. The component increasing is limited in page size. If you set the value of this property in 5, then, when increasing the vertical size, it will be increased in 5 horizontal lines.



Text Rotation

Set the angle of the text rotation using the **Angle** property of the **Text** component. The angle of the text is given in degrees anticlockwise.



Processing Duplicates

In many reports there is a necessity to join a few **Text** components in one which contain duplicated values. The **ProcessingDuplicates** property is used for this. It should be set to **true**.

See the picture below how repeated text values are joined.

In many reports, If these components contain duplicate values, then it is necessary to combine some **Text** components in one. To combine duplicate values it is necessary to use the **ProcessingDuplicates** property.

The picture below shows an example of duplicate text values.

Beverages	Chai	10 boxes x 20 bag	18,00p.	39,00
	Chang	24 - 12 oz bottles	19,00p.	17,00
	Chartreuse verte	750 cc per bottle	18,00p.	69,00
	Côte de Blaye	12 - 75 cl bottles	263,50p.	17,00
	Guaraná Fantástica	12 - 355 ml cans	4,50p.	20,00
	Ipoh Coffee	16 - 500 g tins	46,00p.	17,00
	Lakkalikööri	500 ml	18,00p.	57,00
	Laughing Lumberjack Lager	24 - 12 oz bottles	14,00p.	52,00
	Outback Lager	24 - 355 ml bottles	15,00p.	15,00
	Rhönbräu Klosterbier	24 - 0.5 l bottles	7,75p.	125,00
	Sasquatch Ale	24 - 12 oz bottles	14,00p.	111,00
	Steeleye Stout	24 - 12 oz bottles	18,00p.	20,00
Condiments	Aniseed Syrup	12 - 550 ml bottles	10,00p.	13,00
	Chef Anton's Cajun Seasoning	48 - 6 oz jars	22,00p.	53,00
	Chef Anton's Gumbo Mix	36 boxes	21,35p.	0,00
	Genen Shouyu	24 - 250 ml bottles	15,50p.	39,00
	Grandma's Boysenberry Spread	12 - 8 oz jars	25,00p.	120,00
	Gula Malacca	20 - 2 kg bags	19,45p.	27,00
	Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21,05p.	76,00
	Louisiana Hot Spiced Okra	24 - 8 oz jars	17,00p.	4,00
	Northwoods Cranberry Sauce	12 - 12 oz jars	40,00p.	6,00
	Original Frankfurter grüne Soße	12 boxes	13,00p.	32,00
	Sirop d'érable	24 - 500 ml bottles	28,50p.	113,00
	Vegie-spread	15 - 625 g jars	43,90p.	24,00
Confections	Chocolade	10 pkgs.	12,75p.	15,00
	Gumbär Gummibärchen	100 - 250 g bags	31,23p.	15,00
	Maxilaku	24 - 50 g pkgs.	20,00p.	10,00
	NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	14,00p.	76,00
	Pavlova	32 - 500 g boxes	17,45p.	29,00
	Schoggi Schokolade	100 - 100 g pieces	43,90p.	49,00
	Scottish Longbreads	10 boxes x 8 piece	12,50p.	6,00
	Sir Rodney's Marmalade	30 gift boxes	81,00p.	40,00
	Sir Rodney's Scones	24 pkgs. x 4 piece	10,00p.	3,00
	Tarte au sucre	48 pies	49,30p.	17,00
	Teatime Chocolate Biscuits	10 boxes x 12 pie	9,20p.	25,00
	Valkoinen suklaa	12 - 100 g bars	16,25p.	65,00
	Zaanse koeken	10 - 4 oz boxes	9,50p.	36,00
Dairy Products	Camembert Pierrot	15 - 300 g rounds	34,00p.	19,00
	Flotemysost	10 - 500 g pkgs.	21,50p.	26,00
	Geitost	500 g	2,50p.	112,00
	Gorgonzola Telino	12 - 100 g pkgs	12,50p.	0,00
	Gudbrandsdalsost	10 kg pkg.	36,00p.	26,00
	Mascarpone Fabioli	24 - 200 g pkgs.	32,00p.	9,00
	Mozzarella di Giovanni	24 - 200 g pkgs.	34,80p.	14,00
	Queso Cabrales	1 kg pkg.	21,00p.	22,00
	Queso Manchego La Pastora	10 - 500 g pkgs.	38,00p.	86,00

The **ProcessingDuplicates** property makes it possible to combine duplicate values as follows: **Merge, Hide, RemoveText, GlobalMerge, GlobalHide, GlobalRemoveText**. Next, look at examples of this property.

Merge - In this mode, the text components with identical values are merged into a single text component.

Assistant Sales Agent	Assistant Sales Agent
Assistant Sales Agent	
Assistant Sales Represent	Assistant Sales Represent
Marketing Assistant	Marketing Assistant
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	

Hide - In this mode, the first text component remains on its place without changing the size. The rest of the text components are removed from the report.

Assistant Sales Agent	Assistant Sales Agent
Assistant Sales Agent	
Assistant Sales Represent	Assistant Sales Represent
Marketing Assistant	Marketing Assistant
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	

Hide - In this mode, the first text component remains in place without changing the size. The rest of the text components to remain in their seats, but they removed the text content.

Assistant Sales Agent	Assistant Sales Agent
Assistant Sales Agent	
Assistant Sales Represent	Assistant Sales Represent
Marketing Assistant	Marketing Assistant
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	
Marketing Assistant	

Combining the components with the same value is taken into account in the name of the components of a report template. If suddenly one of the other two will be exactly the same text component with the same text values, but they will have different names, then those components will not be merged. To avoid this limitation you need to use the **GlobalMerge, GlobalHide, GlobalRemoveText**. They worked the same way as described above regimes, but it does not take into account the names of the components.

Ignoring Null Values

Often, when the numerical information is printed then it is required to ignore the zero values. In other words it is necessary do not show print them at all. The **HideZeros** property is used for this. It is necessary to set this property to **true**, and the **Text** component will not print zero values. The picture below shows an example without using this property (**left picture**) and using the property (**right picture**).

HideZeros = false

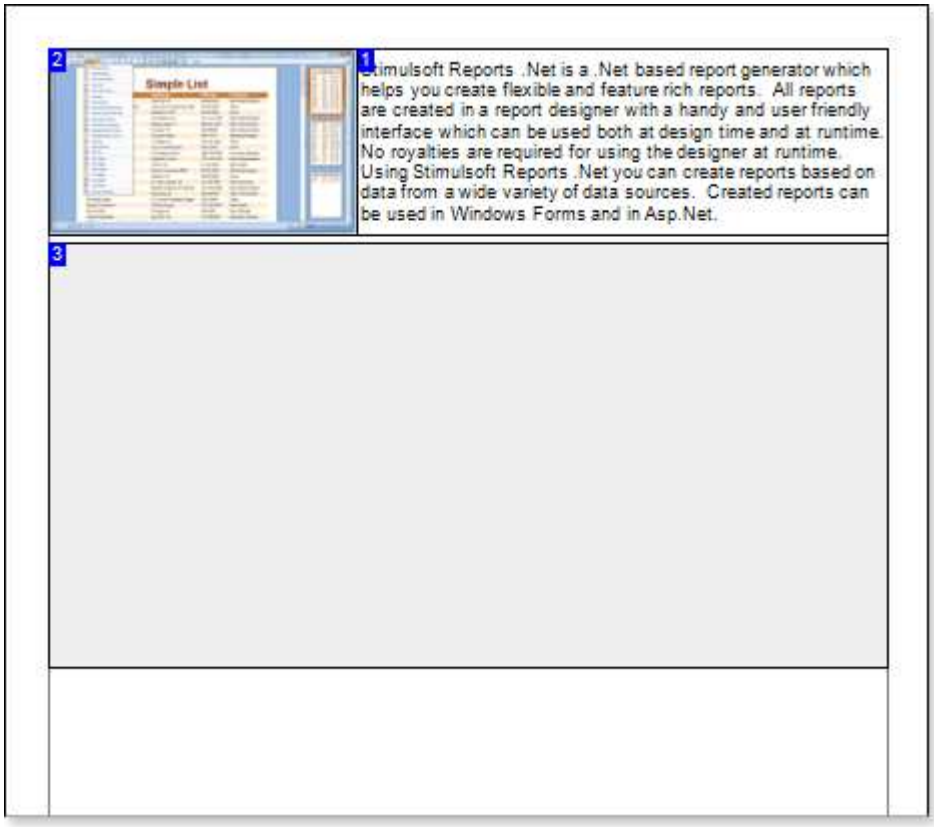
9,00$	61,00
33,25$	22,00
39,00$	0,00
97,00$	29,00
24,00$	115,00
32,80$	0,00
123,79$	0,00

HideZeros = true

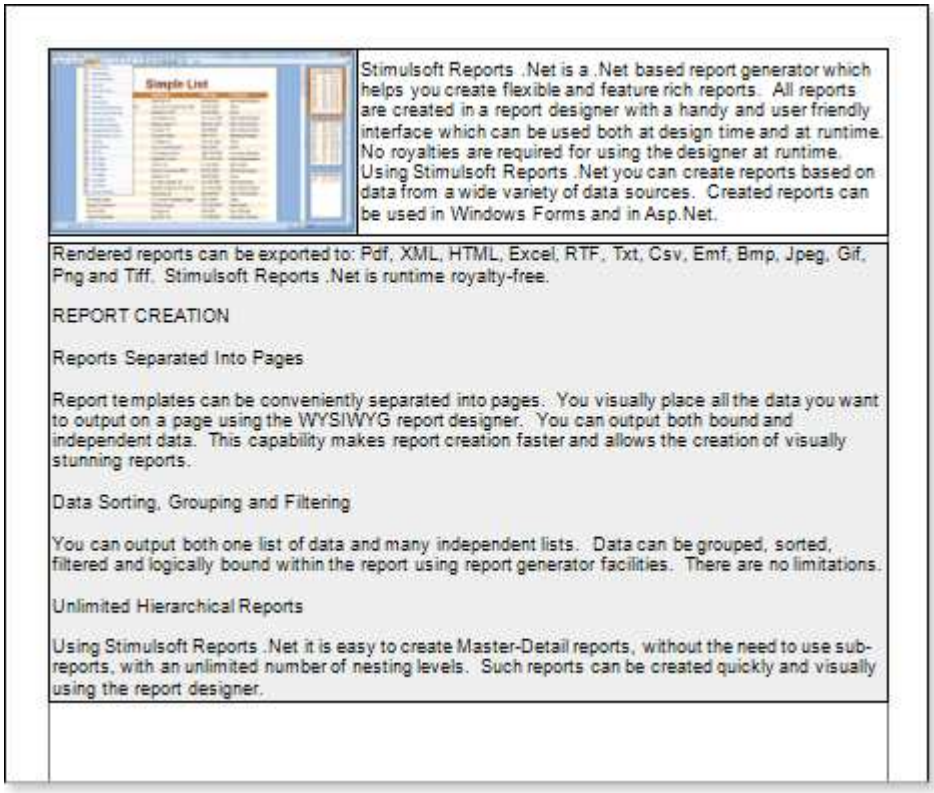
9,00$	61,00
33,25$	22,00
39,00$	
97,00$	29,00
24,00$	115,00
32,80$	
123,79$	

ReportTo Property

The **ReportTo** property of the **Text** component is used for synchronous output of a message in two text components. The message is specified in the first text component. Then, in this text component, in the **ReportTo** property, the second text component, on which message output will be continued, is specified. If the space in the first component is not enough for the message output, then this message will be continuing to output in the second component. You should consider, that in the first component, whole number of vertical visible lines will be output. In the second component the message will be continuing to output starting with the end of the message of the first component. You should know that for the correct work of this function you have to create the first component and then the second one. If there was another order of creation of components you may use commands of components order.



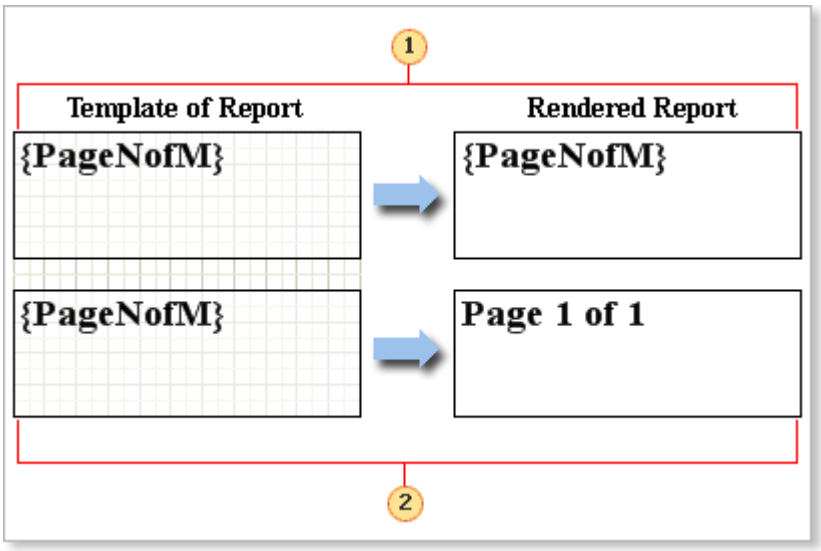
The result can be seen on the picture below.



The **ReportTo** property makes it possible to work only with components that are located on one level - such as a bands.

Output Text Only without Taking Expressions into Consideration

How to get an expression to be output "as is", without code processing? Set the **TextOnly** property to **true**, and all the expressions will be output as a text. No calculations will be made.



1 The **TextOnly** property is set to **true**. The text is output "as is", without processing of expressions.
2 The **TextOnly** property is set to **false**. The text is output with processing of expressions.

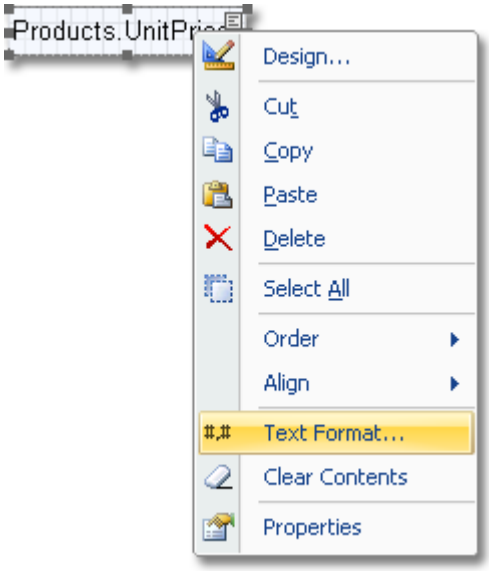
Expression Processing in the End of Report Rendering

By default, the report generator immediately processes all expressions which are met in the text. But sometimes it is necessary to process expressions after the report rendering. For example, while report rendering, the calculation of a variable is in process. The result of calculation will be known right after the report rendering, and the result of calculation is to be output on every page of a report. To do this, set the value of the **ProcessAtEnd** property of the **Text** component to **true**.

Important! When the content of the text component is processed in the end of the report rendering, the report generator cannot define the true size of the component when it is output. Therefore, auto change of the component size will work with failure.

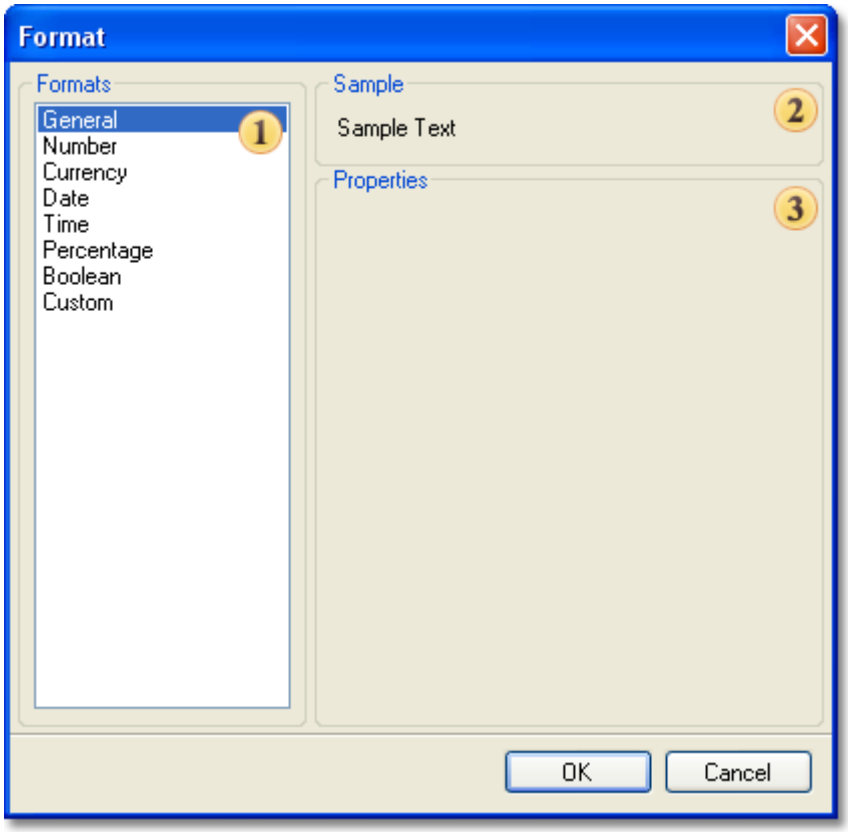
Text Formatting

The Text format is a representation of information in the special form (grouping and data output, in order to the specified pattern). Stimulsoft Report contains all necessary instruments required for formatting of all information that will be output. The **Text Format** is the basic tool for formatting a text before output. This tool is a dialog box, which allows setting parameters of format. Text format dialog box is called from the context menu, that appears when right-clicked on the text components, which supports formatting.



Also, using **TextFormat** properties, the dialog box can be called.

The Format window is divided into three parts:

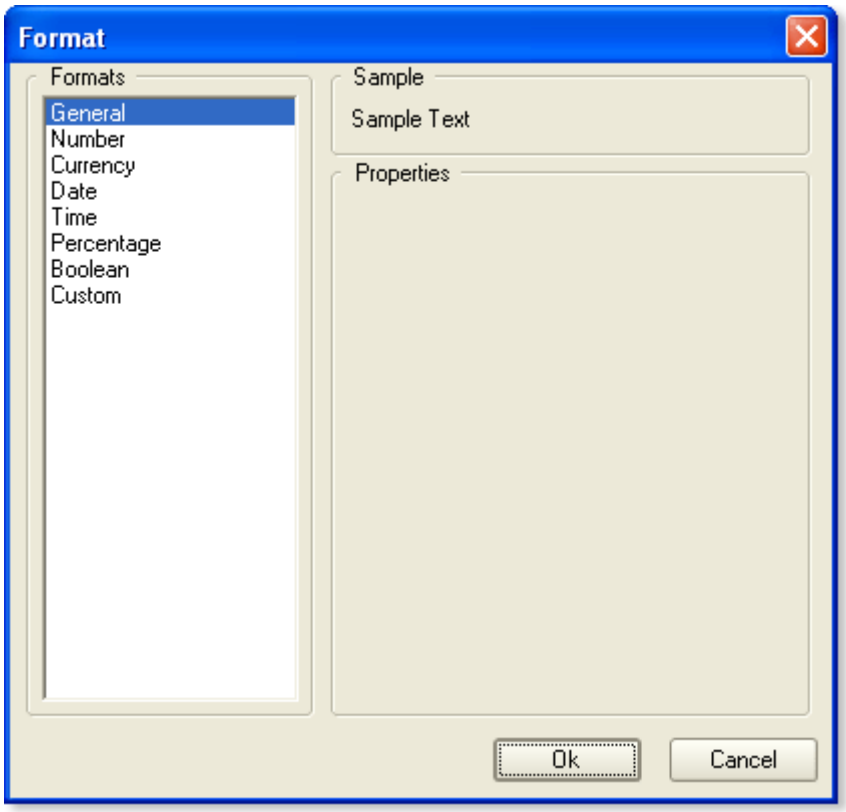


1 A section where the formatting type can be chosen.
There are some types of showing a text:
Standard - output data without specific number format;
Number — this format is used for general display of numbers;
Currency — this format is used for general monetary values;
Date — this format is used to display date values;
Time — this format is used to display time values;
Percent — this format is used to display a result in percent symbol;
Boolean — this format is used to display boolean values;

Custom — custom data formatting.
2 Shows how the formatted text will look like;
3 Shows the format settings.

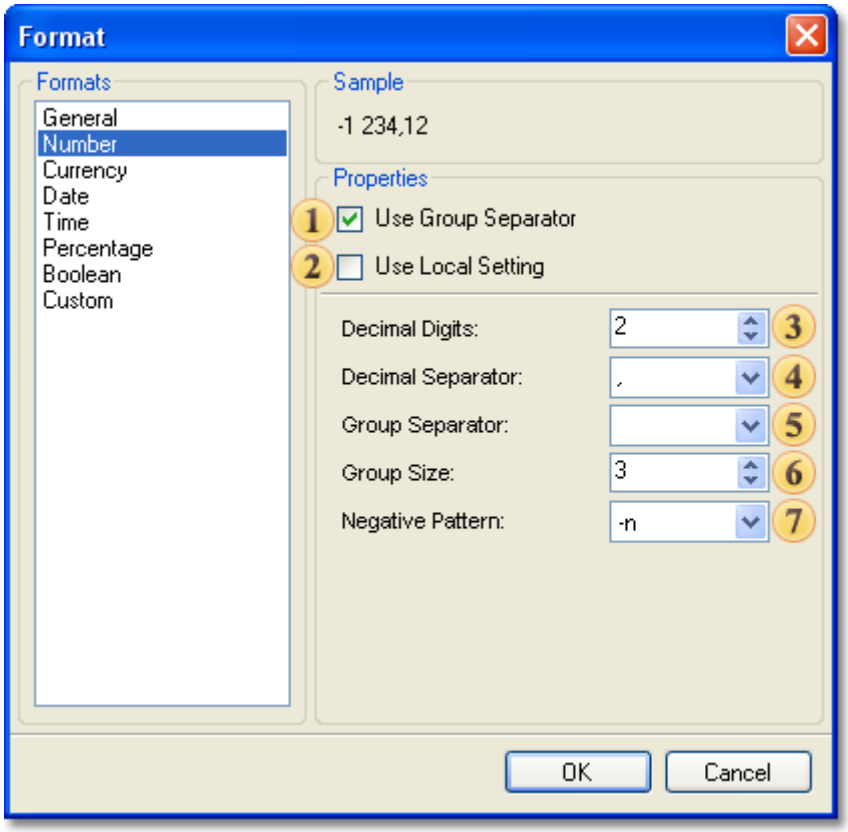
Standard Formatting

The **Standard** format is used to show text and numerical values of any type. No formatting is done in this case.



Numerical Formatting

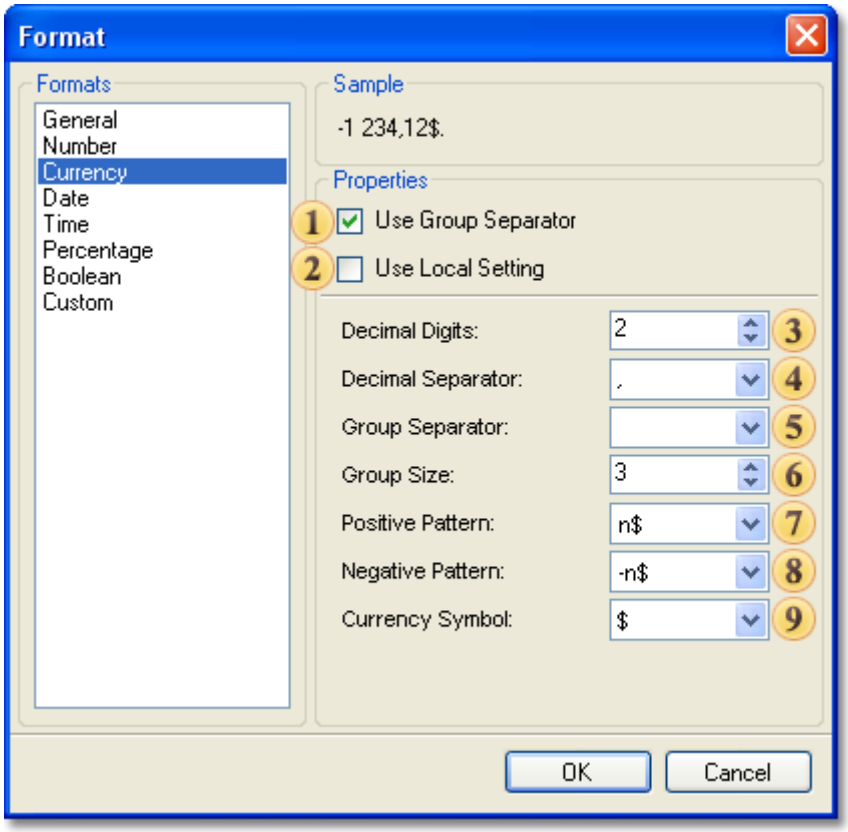
It is recommended to use the numerical format to show numerical values.



1 **Group separator**
When the Group Separator is used then number will be separated into number positions.
2 **Local setting**
When using the Local settings, numerical values are formatted according to the current OS installations.
3 **Decimal digits**
Number of decimal digits, which are used to format numerical values.
4 **Decimal separator**
Used as a decimal separator to separate numerical values in formatting.
5 **Group separator**
Used as a group separator when numerical values formatting.
6 **Group size**
The number of digits in each group in currency values formatting.
7 **Negative pattern**
This pattern is used to format negative values.

Currency Formatting

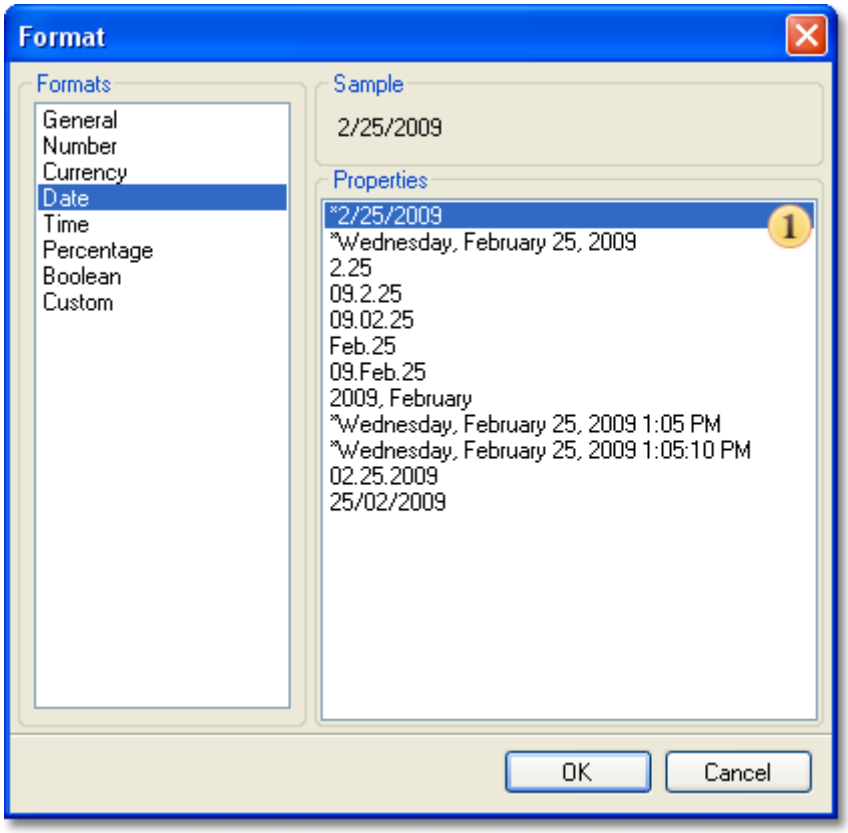
It is used for format currency values. Currency format can be used to output other numbers.



1 **Group separator**
When the Group Separator is used then currency values will be separated into number positions.
2 **Local setting**
When using the Local settings, currency values are formatted according to the current OS installations.
3 **Decimal digits**
Number of decimal digits, which are used to format currency values.
4 **Decimal separator**
Used as a decimal separator to separate currency values in formatting.
5 **Group separator**
Used as a group separator when currency values formatting.
6 **Group size**
The number of digits in each group in currency values formatting.
7 **Positive pattern**
This pattern is used to format positive values.
8 **Negative pattern**
This pattern is used to format negative values.
9 **Currency symbol**
This symbol is used to define the currency name.

Date Formatting

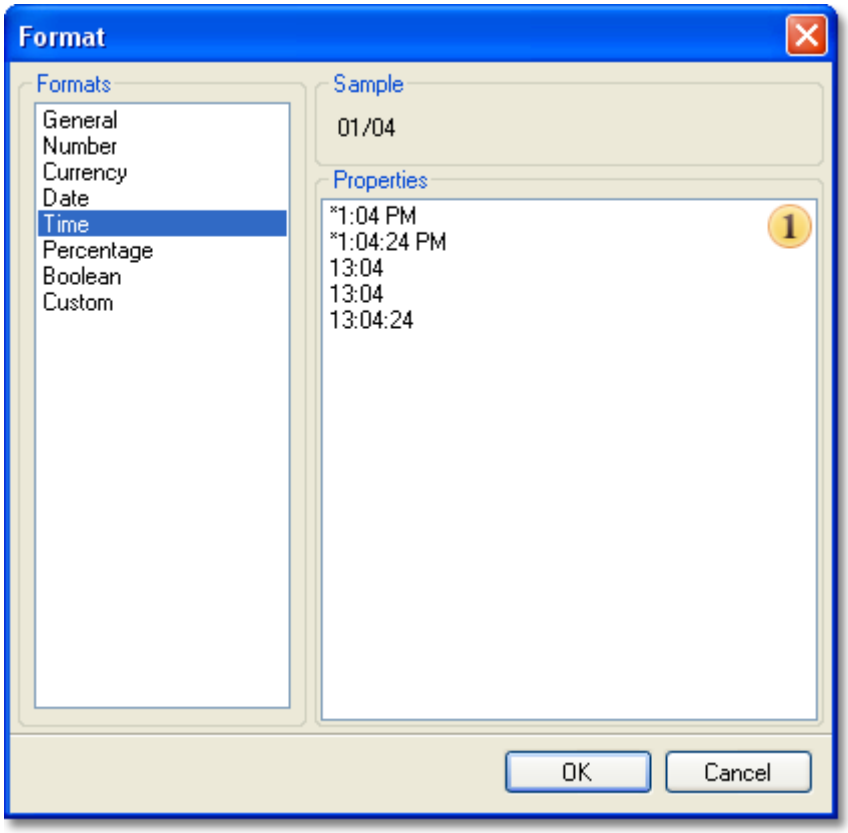
The **Date** format is used to show a date. The **Date** format is selected from the set of formats: short date format, extended date format etc. In all formats, except ones which are marked with the (*) symbol, the order of elements is not changed.



1 **Date format**
The list format types.

Time Formatting

The **Time** format is used to show time. The **Time** format is selected from the set of formats: short date format and extended date format (with seconds).

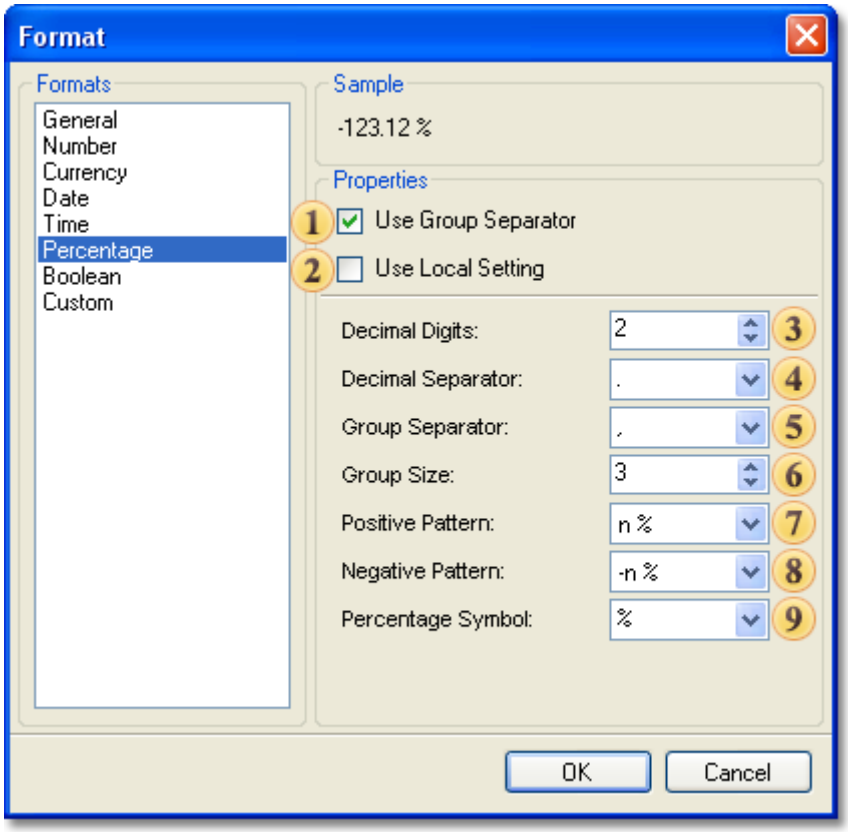


Time format

1 The list of format types

Percentage Data Formatting

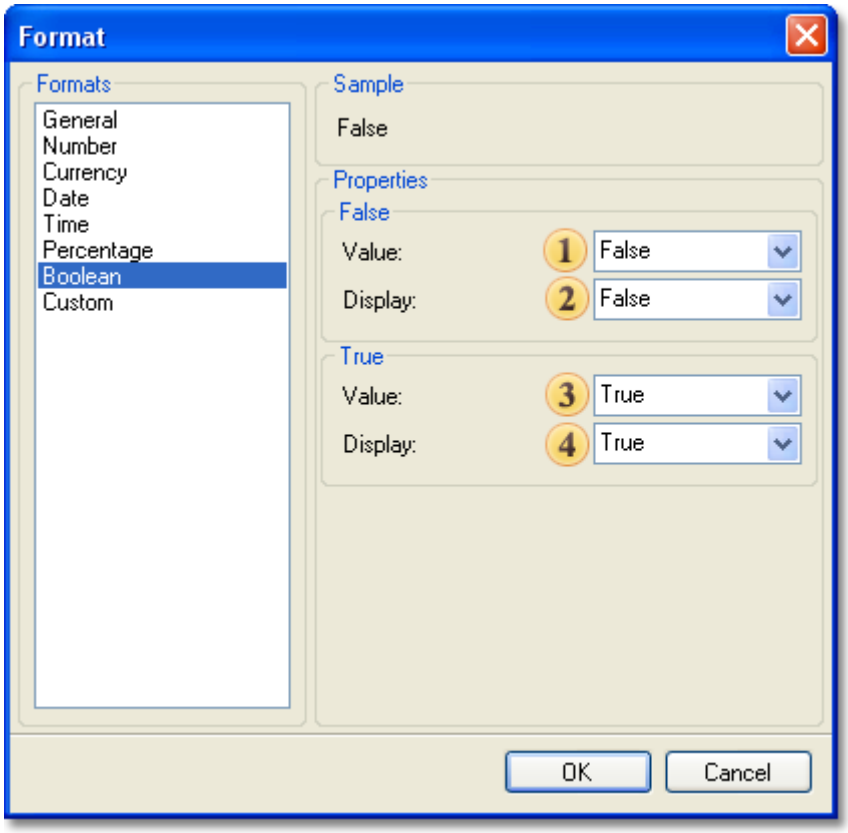
This format is used to show percent values. When formatting, the value is multiplied by 100 and is output with the percent sign.



1 **Group separator**
When the Group Separator is used then currency values will be separated into number positions.
2 **Use local setting**
When using the Local settings, numerical values are formatted according to the current OS installations.
3 **Decimal digits**
Number of decimal digits, which are used to format numerical values.
4 **Decimal separator**
Used as a decimal separator to separate numerical values in formatting.
5 **Group separator**
Used as a group separator when numerical values formatting.
6 **Group size**
The number of digits in each group in currency values formatting.
7 **Positive pattern**
This pattern is used to format positive values.
8 **Negative pattern**
This pattern is used to format negative values.
9 **Percentage symbol**
The symbol will used as a percent sign.

Boolean Values Formatting

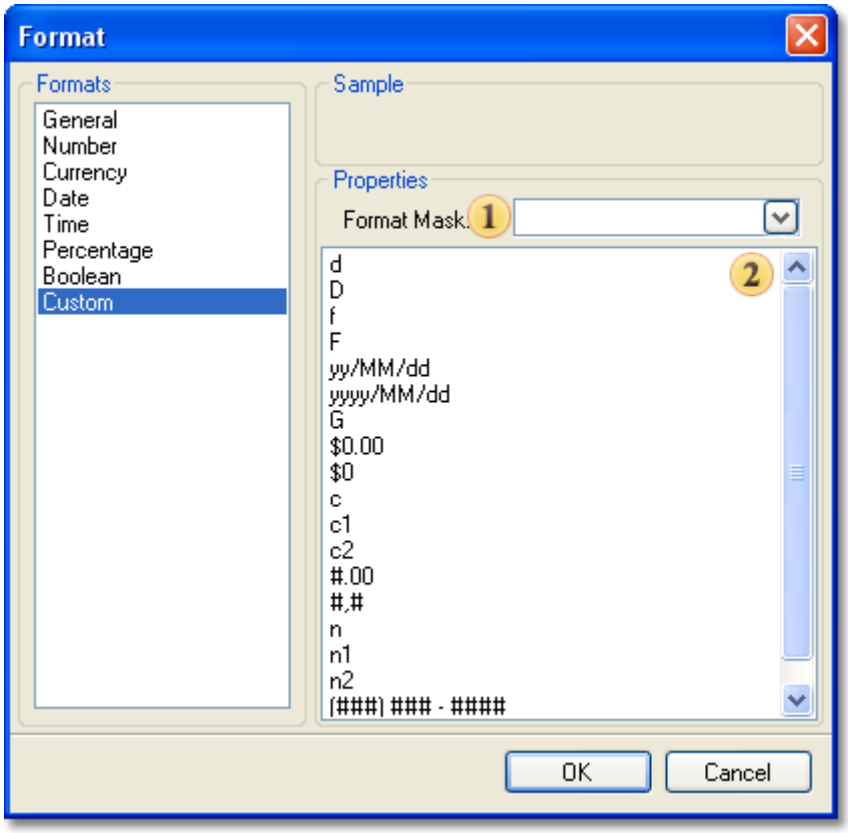
This format is used to format values of the boolean type.



1. The string value to identify boolean values as **false**;
2. The string value to represent boolean value as **false**;
3. The string value to represent boolean value as **true**;
4. The string value to represent the boolean value as **true**.

Custom Formatting

This type is used to show values according to custom requirements. This type allows data formatting in the **Format Mask**.



1 **Mask**
A string or an expression that set formatting mask.
2 **Predefined values**
The list of predefined values to format a string.

Formatting in Text

The **Text Format** tool allows values formatting using a lot of parameters and options. But this tool has one weak point. Formatting is applied on the whole text object. For example, if the text component is used to output data, then it is easy to format. But to do if it is required to format only one value from an expression? Or what to do if it is required to format two or more values of an expression? In this case it is recommended to use use the **string.Format** method. This method is used to make almost the same kind of formatting as if you use the **Text Format** tool. But the **string.Format** method is more flexible. For example, to format the value as a **currency** the **C** specificator is used:

```
Currency values: {string.Format("{0:C}", Value) }
```

if **Value** is 123.12, then after formatting the line will be:

```
Currency values: $123.12
```

The **string.Format** method may have more than one parameter of formatting, for example:

```
Currency values: {string.Format("value1 - {0:C}, value2 - {0: 1}", Value1, Value2) }
```

Please read MSDN to get more information about **string.Format**.

HTML Tags

Stimulsoft Reports has the ability to format text using standard HTML formatting tags.

Important: Only a limited range of HTML tags are supported - for example you cannot use Ordered Lists (OL) or Unordered Lists (UL). If you need to achieve bullet points or numbers within your text your choices are to enter them manually or to use the RTF text editor component.

Sometimes it is necessary to make part of a text expression look Bold, Italic, or Underlined. For example you may wish to achieve something like this:

The fifth word is **bold**

HTML tags can help achieve this. The output shown above could be generated using the following expression:

```
The fifth word is <b>bold</b>
```

It is possible to get a similar result without using HTML by using the Rich text component, but there are some difficulties and the Rich text component works very slowly, so using HTML tags is often the best way to achieve the desired result.

HTML tags can be included only in the text part of expression, in other words their use is possible only in the **Text** property of the **Text** component.

Important: HTML tags can be included only in the text part of an expression.

For example, the following expressions are correct:

```
This is a simple <i>expression {1+2}</i>

This is a simple <i>expression</i> {1+2}

This is a simple expression <i>{1+2}</i>
```

These expressions however are incorrect:

```
The is a simple <i>expression {1</i>+2}

The is a simple <i>expression {1+2</i>}

The is a simple expression {<i>1+2}</i>
```

In the examples above the HTML tags are placed within the body of an expression that will be calculated by C# or VB.Net, shown by the curly braces, so they are impossible to process.

Important: Do **NOT** place HTML tags inside the curly braces of any expression or the expression will fail.

Available Tags

There are few limitations - most valid HTML style tags can be inserted, with the exception of ordered list and unordered list tags. If you need to generate such lists you can use the Rich Text control or create the layout manually.

Important: You cannot use Ordered and Unordered List tags within expressions.

HTML tags can be nested to an unlimited depth. For example:

```
This is a <b>simple <i>expression {1+2}</i></b>
```

If a tag is not closed, then the tag works to the end of the text line.

If HTML tags are used in a text expression then any line breaks in that expression are ignored. If you need to enforce a line break in your text, use the <br> tag.

Note. Use the <br> tag to break a line when using HTML tags.

Activating HTML Tags

It is important to know that by default HTML tags in expressions are simply ignored. To allow the use of HTML tags it is necessary to set the **AllowHtmlTags** property of the Text component to true.

Important: Set the **AllowHtmlTags** property to true to allow the use of HTML tags in the text expression.

HTML <font> Tag

The tag is used to add style, size, and color to a text expression. If there is no closing tag then all changed font characteristics will be applied from the beginning of the tag and to the end of the text.

Syntax:

```
<font face="FontName" color="#rrggbb" size="n"> </font>
```

Parameters:

color Defines the color of the text.
face Defines the font of the text.
size Defines the size of the text.

Not all of these parameters have to be used. The default value is set within the parameters of the text component, so ifthe font size of the text component is 8 points and the **size** parameter is not used in the tag, then the text will be output at 8 points. The same rule works for the other parameters.

Example:

If you enter the following expression:

```
Test <font color="red" face="Courier" size="18">Test</font> Test
```

then after calculation the result appearing in the report will be:

Test Test Test

Color Parameter

The color parameter defines the color of the text in the font element. The color can be set in two ways:

1. By Name

You can define the colour by name - a collection of 147 color names is supported. If the report generator is not able to identify the color set, then it ignores the**color**parameter. For example:

```
<font color="red" ...>

<font color="black" ...>

<font color="white" ...>
```

2.By Hex Value

You can also specify the color using a hex (hexadecimal) value like"#ff0000". It is very important to add the hash symbol '#' before the hexadecimal notation.

The color is a combination of Red, Green and Blue values (#rrggbb). Each of the three colors may have hex values from 00 through to FF. The first two**rr**symbols indicate the red part of the color,**gg**symbols indicate the green part of the color, and**bb**symbols indicate the blue part. A color can be set in a short form using one symbol for each color. For example:

```
<font color="#FF0000" ...>

<font color="#F00" ...>

<font color="#FF0000" ...>

<font color="#998877" ...>

<font color="#FF00FF" ...>
```

Important: If the color value set is not recognized or is invalid, then the color specified in the Text component or in the tag is used.

Alternative Tags

The tag or the tag can also be used to define the text color.For example:

```
<font-color="red">

<color="red">
```

Face Parameter

The face parameter defines the name of the font of the text within the font element. To use this parameter you must specify the font name. If the font is not found, then the font of the text component or the previous font specified in thetag is used.

The sample below shows how to use the **face** parameter:

```
<font face="Arial" ...>
```

Alternative Parameters

Instead of the "**face**" parameter the parameters "**name**" and "**family**" can be used. All these attributes are identical. For example:

```
<font face="Courier" ...>

<font name="Courier" ...>

<font family="Courier" ...>
```

All the text expressions above specify the same font.

Alterative Tags

The tag is the same as the tag with the **face** parameter. For example:

```
<font-face="Arial">
```

Size Parameter

The size parameter defines the size of the text in the font element in points. For example:

```
<font size="14" ...>
```

If the expression is incorrectly formulated then the parameter is ignored.

Alternative Tags

The font size can also defined separately using the tag. For example:

```
<font-size="14">
```

HTML Tags to Change Font Style

The report generator supports nine tags for changing a font style: ****, **<i>**, **<u>**, **<s>**, **<sup>, <sub>, , <p>,
**. These HTML tags are called formatting tags. These formatting tags can make text bold, italic, sub/superscripted, and more.

The example below shows how the **** tag works in a text expression. If you enter the following expression:

```
This <b>text</b> is bold.
```

then after calculation the result appearing in the report will be:

This **text** is bold.

Note that the word 'text' is enclosed within the opening and closing and tags.

Formatting tags can be used in combination with other formatting tags to changing the text style. For example, if you enter the following expression:

```
This <i><b>text</b></i> is bold italic.
```

then after calculation the result appearing in the report will be:

This ***text*** is bold italic.

Style intersection is not allowed, i.e. formatting tags may not be nested partly inside and partly outside another formatting tag. For example:

```
<b>This <i>text</b> is bold</i> italic.   // This will fail
```

The available formatting tags are discussed in detail in the following topics.

HTML Tag

The **** tag is used to define bold text. The tag can be used in combination with other tags to change the text style. For example, if you enter the following expression:

```
Test <b>Test</b> Test
```

then after calculation the result appearing in the report will be:

```
Test **Test** Test
```

If a tag is not closed then the formatting is applied to from the opening tag to the end of the text expression.

HTML <i> Tag

The **<i>** tag is used to define italic text. The tag can be used in combination with other tags to change the text style. For example, if you enter the following expression:

```
Test <i>Test</i> Test
```

then after calculation the result appearing in the report will be:

```
Test *Test* Test
```

If a tag is not closed then the formatting is applied to from the opening tag to the end of the text expression.

HTML Tag

The <**em**> tag is used for indicating emphasis. The text inside this tag is more important than flat text. The text displayed using the <**em**> tag looks italic. The example below shows how the <**em**> tag works:

```
Emphasis <em>Emphasis</em> Emphasis
```

then after calculation the result appearing in the report will be:

```
Emphasis *Emphasis* Emphasis
```

If a tag is not closed then the formatting is applied to from the opening tag to the end of the text expression.

HTML <u> Tag

The **<u>** tag is used to define underlined text. The tag can be used in combination with other tags to change the text style. For example, if you enter the following expression:

```
Test <u>Test</u> Test
```

then after calculation the result appearing in the report will be:

```
Test Test Test
```

If a tag is not closed then the formatting is applied to from the opening tag to the end of the text expression.

HTML <s> Tag

The **<s>** tag is used to define strikethrough text, that is text with a horizontal line through the center. The tag can be used in combination with other tags to change the text style. For example, if you enter the following expression:

```
Test <u>Test</u> Test
```

then after calculation the result appearing in the report will be:

```
Test Test Test
```

If a tag is not closed then the formatting is applied to from the opening tag to the end of the text expression.

HTML <sup> Tag

The **<sup>** tag is used to define a superscripted text. Superscript text appears half a character above the baseline. The tag can be used in combination with other tags to change the text style. For example, if you enter the following expression:

```
Test <sup>Test</sup> Test
```

then after calculation the result appearing in the report will be:

```
      Test
Test        Test
```

If a tag is not closed then the formatting is applied to from the opening tag to the end of the text expression.

HTML <sub> Tag

The <**sub**> tag defines a subscripted text. A subscripted text appears half a character below the baseline. The example below shows how the **<sub>** tag works:

```
Test <sub>Test</sub> Test
```

The result of output:

Test $_{\text{Test}}$ Test

HTML Tag

The <**strong**> tag indicates strong emphasis. It has an end tag. A text within this tag is more important than a flat text. It is usually rendered in bold font style. The example below shows how the <**strong**> tag works:

```
Text <strong>Text</strong> Text
```

The result of output:

Text **Text** Text

HTML <p> Tag

The <**p**> tag defines a paragraph. It has an end tag. The example below shows how the <**p**> tag works:

```
<p>This is a text in a paragraph.</p>
This is a text after the paragraph.
```

The result of output:

This is a text in a paragraph.

This is a text after the paragraph.

HTML
 Tag

The <**br**> tag inserts a single line break. It has no end tag. The example below shows how the <**br**> tag works:

```
How it<br> works.
```

The result of output:

```
How it
works.
```

HTML <background-color> Tag

The **<background-color>** tag is used to change the background color of a text element. By default the background color is set the same as the color specified in thetag, or in the text component properties if no font has been specified.

However, if you place text between a pair of start and end background color tags, then the specified background color will be applied to that text. For example, if you enter the following expression:

```
Test Test Test
```

then after calculation the result appearing in the report will be:

```
TestTestTest
```

HTML <text-align> Tag

The <text-align> tag specifies the horizontal alignment of an element with respect to the surrounding context in the text component. The tag supports four modes of alignment: **left**, **right**, **center,** and **justify**. For example, if you enter the following expression:

```
Test<br>
<text-align="right">Test</text-align><br>
Test<br>
```

then after calculation the result appearing in the report will be:

Test
Test
Test

HTML <letter-spacing> Tag

The <letter-spacing> tag is used to define the space between letters. The value of this tag can be set in any units, and the value can be negative, so it is very important to make sure that a text is readable after applying this tag. By default the value of this tag is 0.

For example, if you enter the following expression:

```
Test<br><letter-spacing="0.5">Test</letter-spacing>
```

then after calculation the result appearing in the report will be:

Test
T e s t

HTML <word-spacing> Tag

Using the **<word-spacing>** tag it is possible to define the space between each words. If the **<text-align>** tag with the **"justify"** value is used, then the **<word-spacing>** tag is ignored. This happens because the interval between words is already specified and a line of a text is aligned by both left and right sides. The example below shows how the **<word-spacing>** tag works:

```
Test <word-spacing="2"> Test </word-spacing>Test
```

The result of output:

Test Test Test

HTML <line-height> Tag

The **<line-height>** tag sets the height of the text line. The tag is set as the multiplier for the basic line height. By default the value if the **<line-height>** tag is 1. The example below shows how this tag works:

```
Test<line-height="1.5"><br></line-height>Test<line-height="0.7"><br></line-height>Test
```

The result of output:

Test

Test
Test

Special Characters

Sometimes it is necessary to use a phrase, for example, in French or German on the website page or to display an example of HTML code on the page. For this purpose, the braces characters, opening "<" and closing ">" are used. They define the first and last character of the tag. For example, in order to display the "greater-than" sign or the opening "<" brace, the "**&lt**;" character is used. Each character has its **&-ASCII** code, which has a specific **&#****** format, where **** is a numeric character. Pointing a **&-ASCII** code, the appropriate symbol will be output on the page. Also, some characters have **&-Name** codes, which have the **&****** formats where **** is an alphabetic names of characters. Below are the tables with the most frequently used characters:

Special Characters

Common Name	ISO Latin-1 Numeric Entity	&-ASCII	&-Name
Quotation mark	"	"	"
Ampersand	&	&	&
Non-breaking space			
Inverted exclamation point	¡	¡	&ixcl;
Cent	¢	¢	¢
Pound sterling	£	£	£
General currency	¤	¤	¤
Yen sign	¥	¥	¥
Broken vertical bar	¦	¦	¦
Section sign	§	§	§
Dieresis	¨	¨	¨
Copyright	©	©	©
Feminine ordinal	ª	ª	ª
Left guillemot	«	«	«
Not sig	¬	¬	¬
Soft hyphen	-		
Registered trademark	®	®	®
Macron	¯	¯	¯
Degree sign	°	°	°
Plus or minus	±	±	±
Superscript 2	²	²	²
Superscript 3	³	³	³
Acute accent	´	´	&acuate;

Mu	µ	µ	µ
Pilcrow	¶	¶	¶
Middle dot	·	·	·
Cedilla	¸	¸	¸
Superscript 1	¹	¹	¹
Masculine ordinal	º	º	º
Right guillemot	»	»	»
Fraction one-fourth	¼	¼	¼
Fraction one-half	½	½	½
Fraction three-fourths	¾	¾	¾
Inverted question mark	¿	¿	¿

UPPERCASE LATIN-1 CHARACTERS

Name	Character	&-ASCII	&-Name
Capital A, grave accent	À	À	À
Capital A, acute accent	Á	Á	Á
Capital A, circumflex accent	Â	Â	Â
Capital A, tilde	Ã	Ã	Ã
Capital A, dieresis	Ä	Ä	Ä
Capital A, ring	Å	Å	Å
Capital AE diphthong	Æ	Æ	Æ
Capital C, cedilla	Ç	Ç	Ç
Capital E, grave accent	È	È	È
Capital E, acute accent	É	É	É
Capital E, circumflex accent	Ê	Ê	Ê
Capital E, dieresis	Ë	Ë	Ë
Capital I, grave accent	Ì	Ì	Ì
Capital I, acute accent	Í	Í	Í
Capital I, circumflex accent	Î	Î	Î
Capital I, dieresis	Ï	Ï	Ï
Capital Eth	Ð	Ð	Ð
Capital N, tilde	Ñ	Ñ	Ñ
Capital O, grave accent	Ò	Ò	Ò
Capital O, acute accent	Ó	Ó	Ó
Capital O, circumflex accent	Ô	Ô	Ô
Capital O, tilde	Õ	Õ	Õ
Capital O, dieresis	Ö	Ö	Ö

Multiply sign	×	×	×
Capital O, slash	Ø	Ø	Ø
Capital U, grave accent	Ù	Ù	Ù
Capital U, acute accent	Ú	Ú	Ú
Capital U, circumflex accent	Û	Û	Û
Capital U, dieresis	Ü	Ü	Ü
Capital Y, acute accent	Ý	Ý	Ý
Capital Thorn	Þ	Þ	Þ
German sz ligature	ß	ß	ß

LOWERCASE LATIN-1 CHARACTERS

Name	Character	&-ASCII	&-Name
Lowercase a, grave accent	à	à	à
Lowercase a, acute accent	á	á	á
Lowercase a, circumflex accent	â	â	â
Lowercase a, tilde	ã	ã	ã
Lowercase a, dieresis	ä	ä	ä
Lowercase a, ring	å	å	å
Lowercase ae ligature	æ	æ	æ
Lowercase c, cedilla	ç	ç	ç
Lowercase e, grave accent	è	è	è
Lowercase e, acute accent	é	é	é
Lowercase e, circumflex accent	ê	ê	ê
Lowercase e, dieresis	ë	ë	ë
Lowercase i, grave accent	ì	ì	ì
Lowercase i, acute accent	í	í	í
Lowercase i, circumflex accent	î	î	î
Lowercase i, dieresis	ï	ï	ï
Lowercase eth	ð	ð	ð
Lowercase n, tilde	ñ	ñ	ñ
Lowercase o, grave accent	ò	ò	ò
Lowercase o, acute accent	ó	ó	ó
Lowercase o, circumflex accent	ô	ô	ô
Lowercase o, tilde	õ	õ	õ
Lowercase o, dieresis	ö	ö	ö
Division sign	÷	÷	÷
Lowercase o, slash	ø	ø	ø

Lowercase u, grave accent	ù	ù	ù
Lowercase u, acute accent	ú	ú	ú
Lowercase u, circumflex accent	û	û	û
Lowercase u, dieresis	ü	ü	ü
Lowercase y, acute accent	ý	ý	ý
Lowercase thorn	þ	þ	þ
Lowercase y, dieresis	ÿ	ÿ	ÿ

Rich Text Output

Stimulsoft Reports allows users to include **Rich Text** formatted (**RTF**) text in reports, without any limitations.

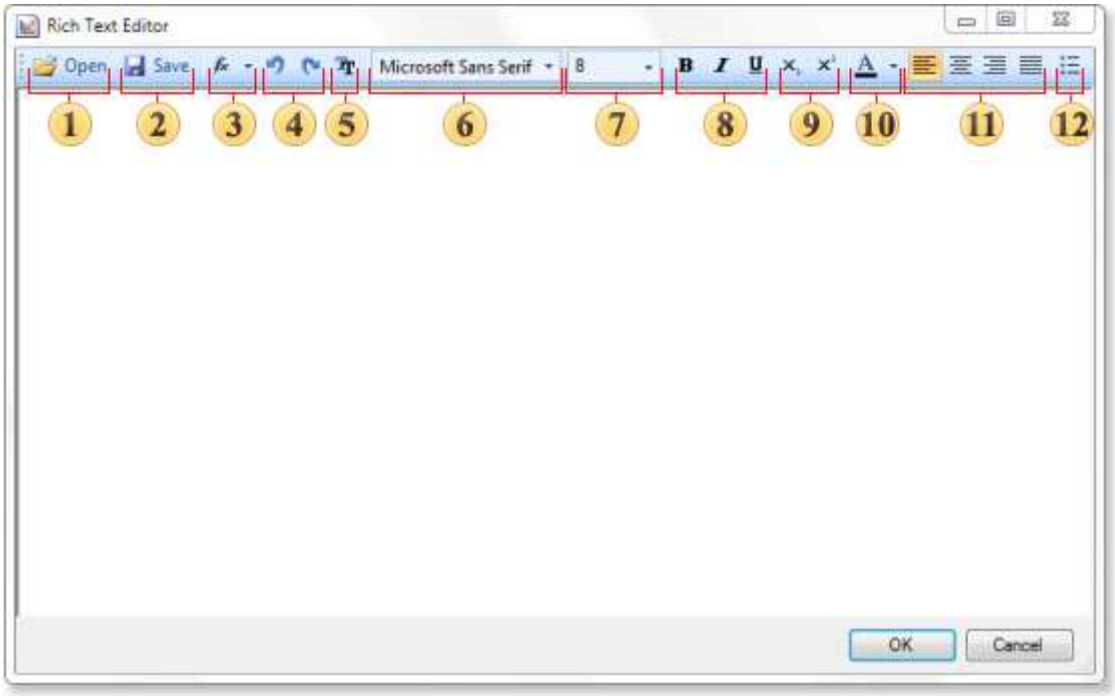
The **RichText** component is designed for working with rich text, and can automatically change its size depending on the size of the **RTF** text within it. It can process expressions, and supports a wide variety of styles, processing at the end of report rendering, etc.

Rich Text Editor

A special editor provided as part of the **RichText** component is used to edit RTF text. This editor is able to load and save RTF text in files, change a font, change font size, change text color, insert expressions etc. No other editor is required to edit RTF text, everything you need is provided within the editor.

The editor is displayed automatically when you insert a **RichText** component, and can be re-opened at any time by double-clicking on the component.

The editor and its controls look something like this:



1	The **Open** button	Displays a standard File Open dialog to allow the content of an existing ***.rtf** file to be loaded into the component.
2	The **Save** button	Displays a standard File Save dialog to allow the component RTF text to be saved to an external ***.rtf** file.
3	The **Insert** button	Displays options allowing you to Insert an expression, function or variable into the component at the current cursor position.
4	**Undo** and **Redo** buttons	**Undo** erases the most recent change to a report reverting it to the previous state. The **Redo** command does the opposite of undo.
5	The **Font** button	Displays a standard Font dialog to allow you to set options such as the font family, style, size etc.
6	The **Font Face** combo	Displays the name of the current font. The font of currently selected text can be changed by selecting a new font from the drop down list of font faces.
7	The **Font Size** combo	Displays the size of the current font. The size of currently selected text can be changed by selecting a new size from the drop down list of font faces.
8	**Bold**,**Italic**,**Underline**buttons	The Font style buttons display the style of the current font. The style of currently selected text can be changed by clicking these buttons to apply or remove styles as required. It is possible to have a font style that combines any number of style aspects, so for example you could have bold underlined text if required.
9	**Subscript** and **Superscript**buttons	Displays the **Subscript** and **Superscript** font styles of the currently selected text. These attributes can be changed by clicking the buttons to apply the required attribute.**Subscript**is text is positioned slightly lower than the remaining text on a line whilst **Superscript** is positioned slightly higher. For example, a footnote or endnote number reference is an example of superscript, and a scientific formula might use subscript text. The **Subscript** and **Superscript** styles are mutually exclusive, so selecting one will automatically deselect the other.
10	The **Color** button	Displays a standard Color dialog to allow the color of the currently selected text to be changed.
	Text Alignment buttons	Displays the alignment of the currently selected text, which can be **Left Align**, **Center**, **Right Align**, or **Justify**. The alignment of the currently selected text can be changed by clicking the buttons to apply the required attribute. The **Text Alignments** are mutually exclusive, so selecting one will automatically deselect the other.

	The **Bullets** button	Displays the bullet status of the currently selected text. The bullet style of currently selected text can be changed by clicking this button to apply or remove bullets as required.

Expressions in Rich Text

The RTF text is an expression in the **RichText** component. There are no significant differences between working with expressions in the **RichText** component and other text components.

The syntax and use of expressions is similar to the syntax and use of expressions in text components, but there is one particular issue to consider - any applied formatting must be applied to the full code insertion and not just part of it.

Suppose that you want the calculated value in the RTF text to be a specific color. It is vital that the colour attribute is applied to the full expression from the opening brace "{" to the closing brace "}" including those symbols. For example:

```
Category:{Categories.CategoryName}
```

Formatting is fully applied to the expression. This expression will work correctly.

```
Category:{Categories.CategoryName}
```

Formatting is applied to only part of the expression. This expression will not work.

```
Category:{Categories.CategoryName}
```

Formatting is fully applied to the expression, but the braces are not included. This expression will not work.

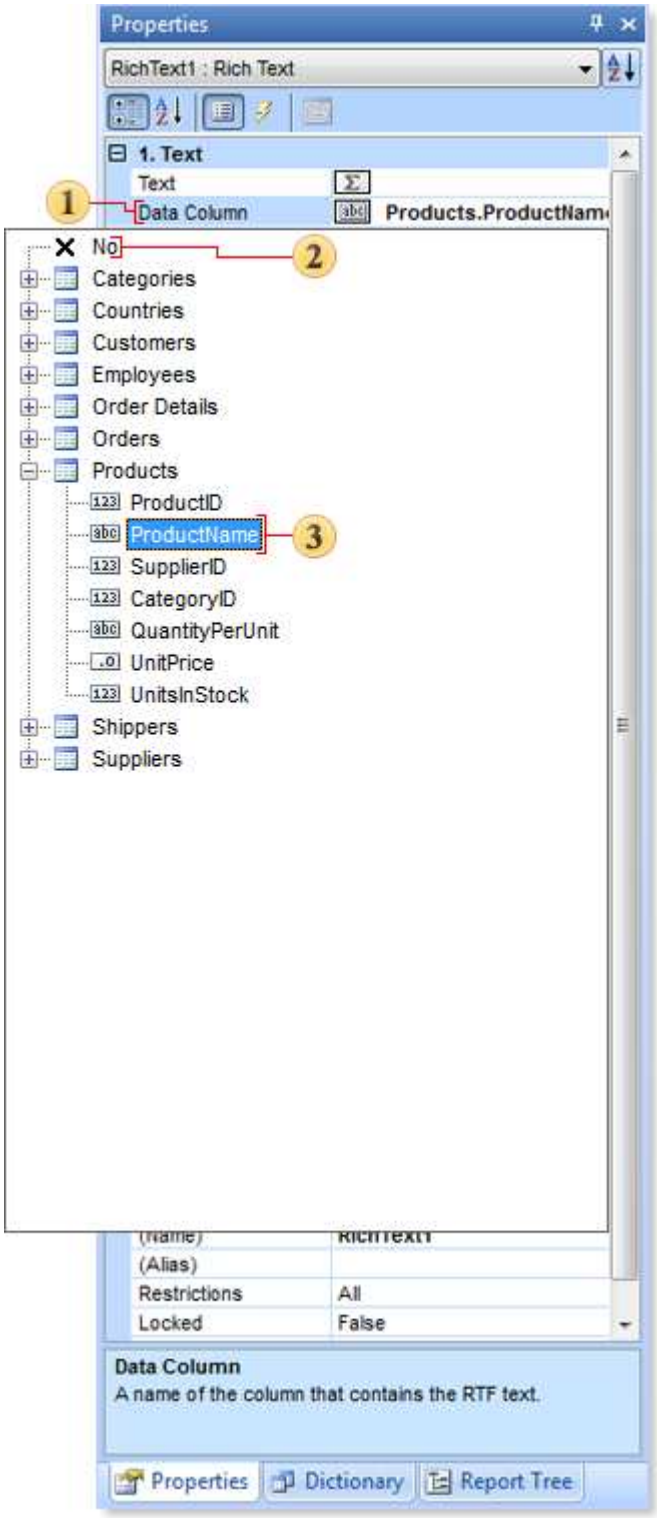
```
Category:{Categories.CategoryName}
```

Formatting does not include the opening brace. This expression will not work.

You should know that in the expressions of the RichText component only plain text can be inserted this way (without formatting commands). So it is not possible to insert the RTF text. You can only assign all of its properties with the help of the DataColumn.

Loading Rich Text From Data Field

The **RichText** component can load the RTF text from the data field using the **DataColumn** property. To load the RTF text simply select a field from the data dictionary tree. When rendering the report generator will automatically load the RTF text for you.



1	**The DataColumn property**	This property is used to indicate from which data field the RTF text should be loaded. Click the button beside to select the relevant column.
2	**Null node**	Selecting this node means that the RTF text is not loaded from a data field.
3	**Selected field**	The Data field from which the RTF text will be loaded .

Sometimes it is necessary to add images to reports. They can be photos of goods, images of colleagues etc. Sometimes it is necessary to place a company logo. The **Image** component is used to output images. This component supports the following types of images: **BMP, JPEG, TIFF, GIF, PNG, ICO, EMF,** and **WMF**.

Images

Country name	Flag	Country name	Flag
Afghanistan		Albania	
Algeria		American Samoa	
Andorra		Angola	
Anguilla		Antigua and Barbuda	
Argentina		Armenia	
Aruba		Ashmore and Cartier Islands	
Australia		Austria	
Azerbaijan		Bahamas	
Bahrain		Baker Island	
Bangladesh		Barbados	
Bassas da India		Belarus	
Belgium		Belize	

(c) 2003-2005 Stimulsoft

Page 1 of 10

Loading Images

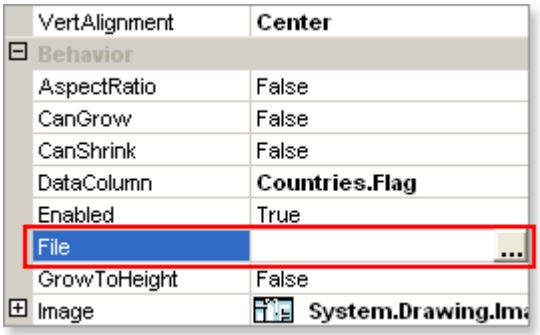
To print an image it is necessary to use the **Image** component. But an image should be loaded first. There are three ways:

1. Load an image from a file;
2. Load an image from the report code;
3. Load an image from the data field.

The below topics describe all these ways.

Loading an image from a file

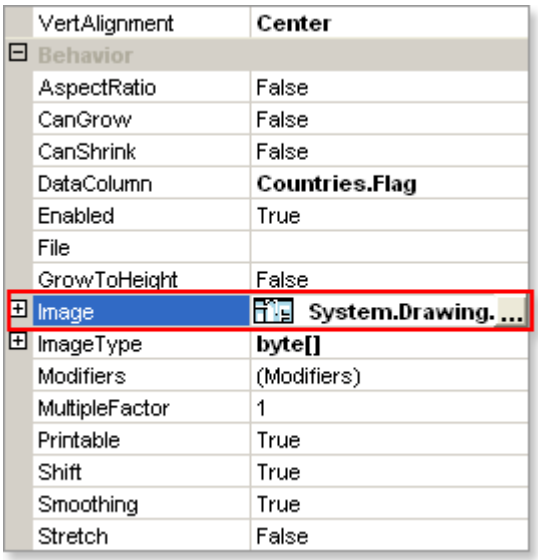
An image can be loaded from a file. Using the **File** property it is necessary specify the file path that contains an image. When report rendering, the report generator will check whether such a file does exist and contains an image. Then the image will be printed.



Loading an image from a report code

Sometimes it is not convenient to store images for report rendering in files. The report generator can save it in the report code. Using the Image property it is possible to load an image from the report code. After loading the image will be saved in the report code.

Important! Do not use this way to output images with the size >100kb. This can be critical for speed of working with the report designer.



Loading an image from a data field

All it is required to load images from a data field is to specify the data field, from what the image will be loaded. The **DataColumn** property is used for this.

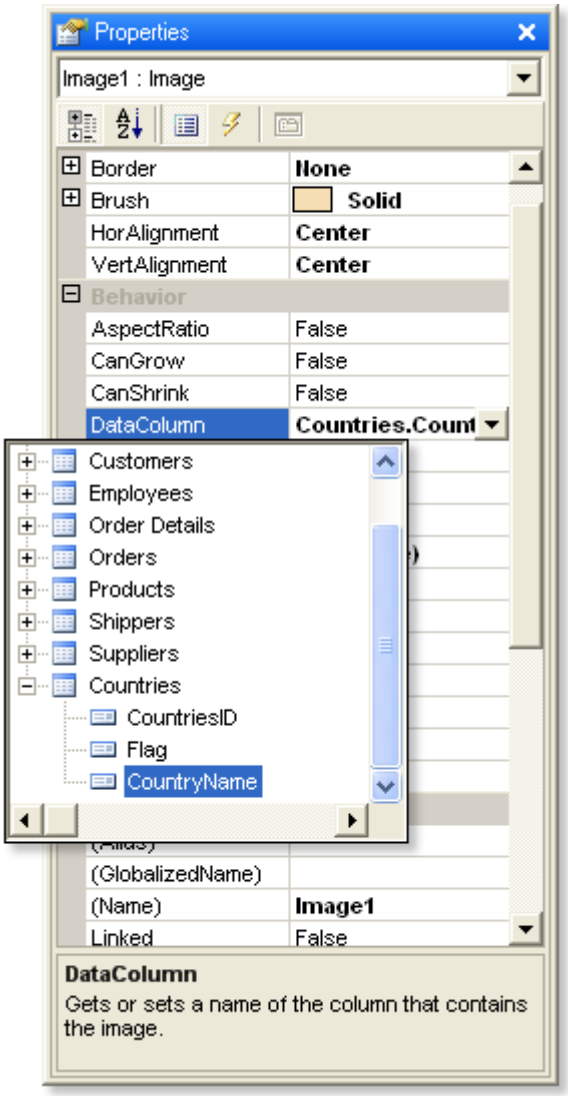
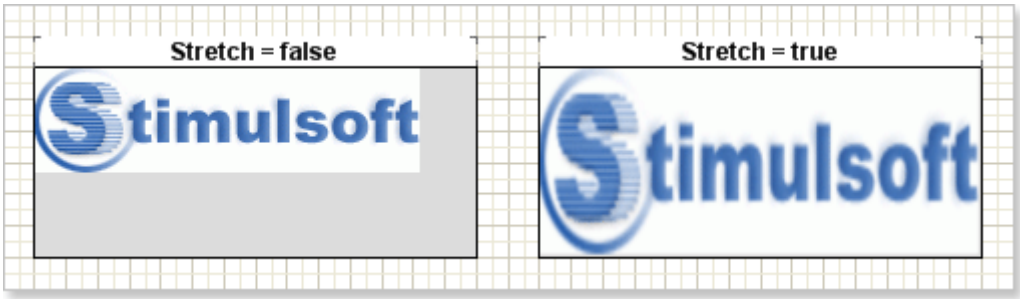
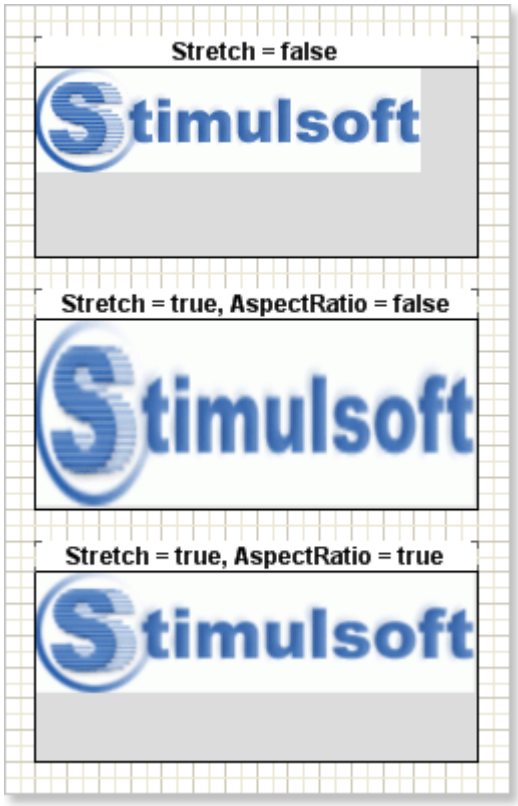


Image Stretching

Often image size does not fit to the component size. In this case free space can be found in a component. Sometimes an image size is bigger that the component size. In such situations it is necessary to stretch images to fill the component with the image. For this, it is necessary to put the **Stretch** property of the Image component to **true**.



After setting the **Stretch** property to **true** the image will fill all free space of the component. When stretching, the image its proportions can be broken. To stretch an image and keep its proportions it is necessary to set the **AspectRatio** property to **true**. And the **Image** component will always keep proportions of images.

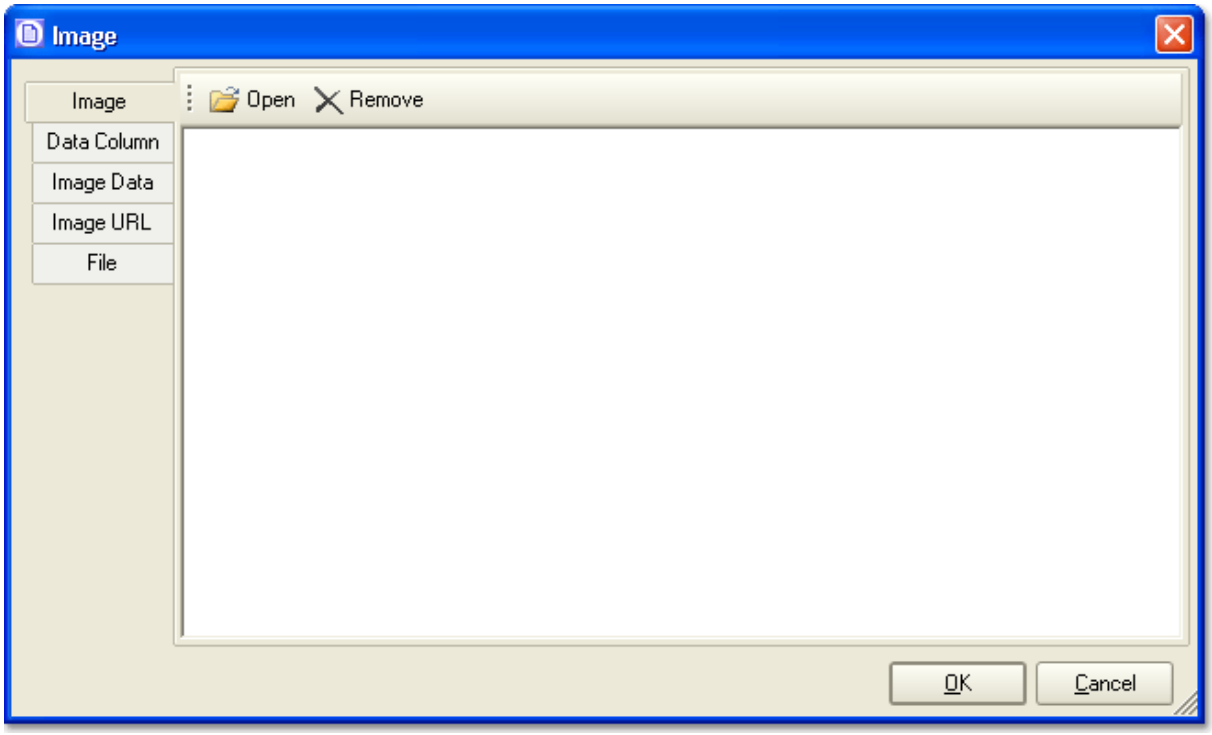


Important! The **AspectRatio** property is in process only when the image stretching is enabled.

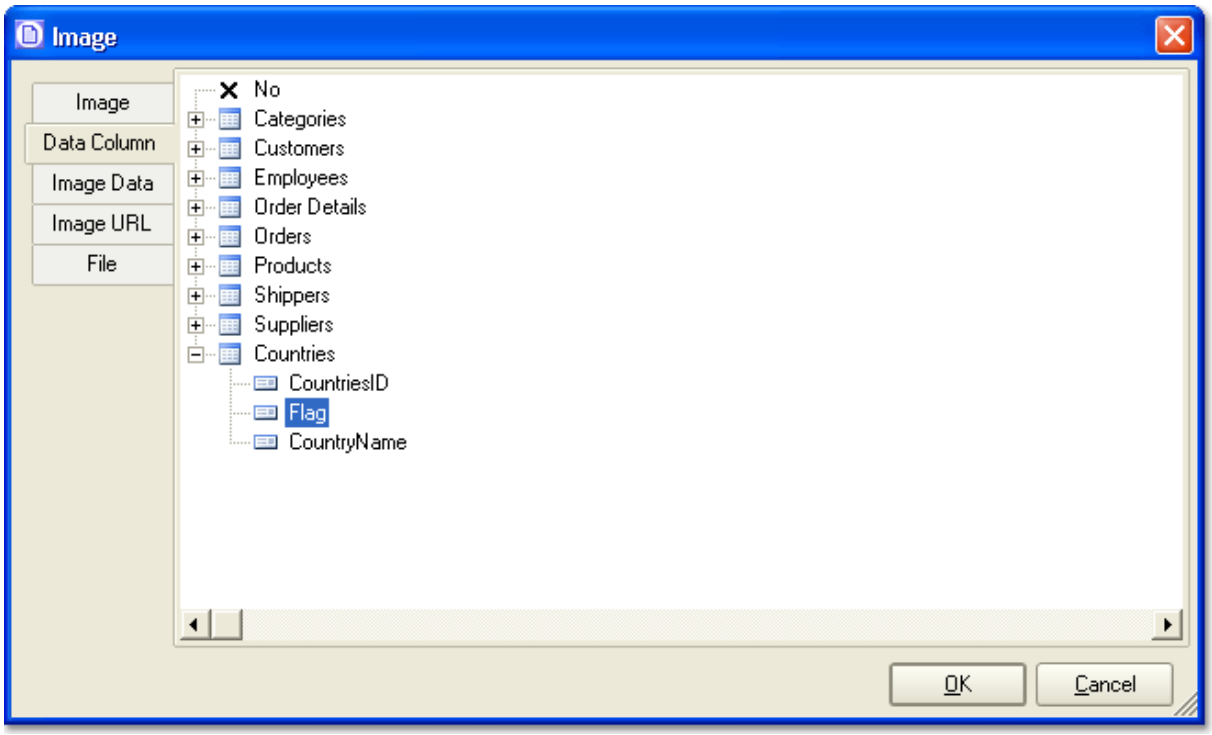
Resources of Images

The **Image** component is used to output images in a report. The way of loading an image can be selected using the the editor of properties of components. This editor is called by double clicking on the **Text** component or clicking the button for calling a designer of **Image** property in the **Property Editor**.
There are the following ways to load images to the **Image** component:

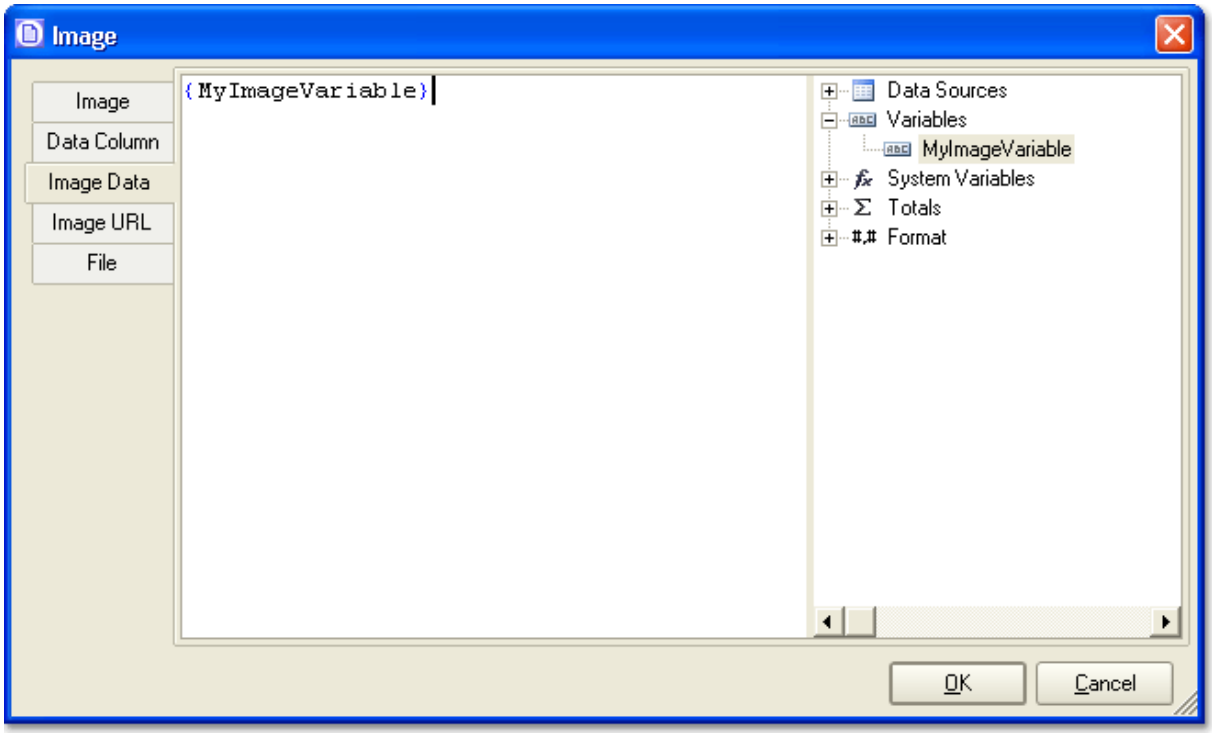
1. Load an image when creating this component in the designer. The **Open** button is used to load an image to the **Text** component in the designer. When saving the file, the image will be saved in the report code. The **Remove** button allows deleting selected components from the designer.



2. From the field of the data base. Select the field of the data base and, in runtime, the image will be loaded to the **Image** object of a report.

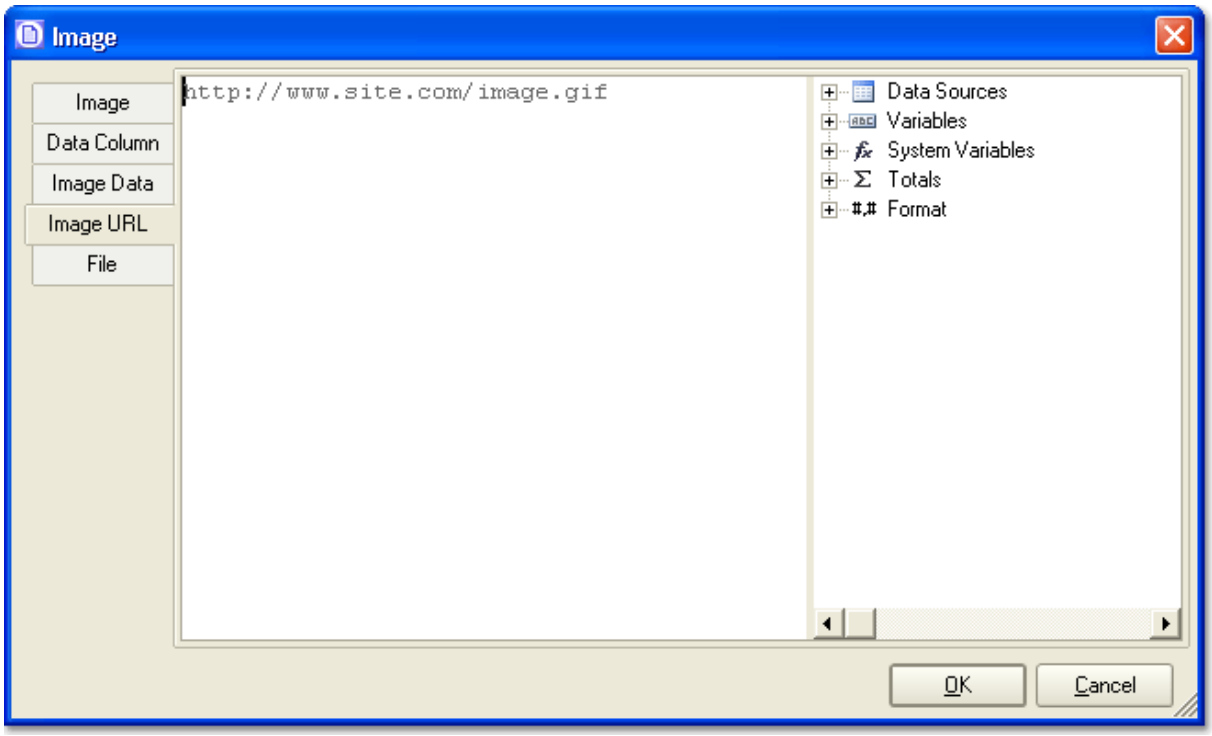


3. From variable. Create a variable in the data dictionary with the **Image** type. This variable can be used in runtime for working with images.

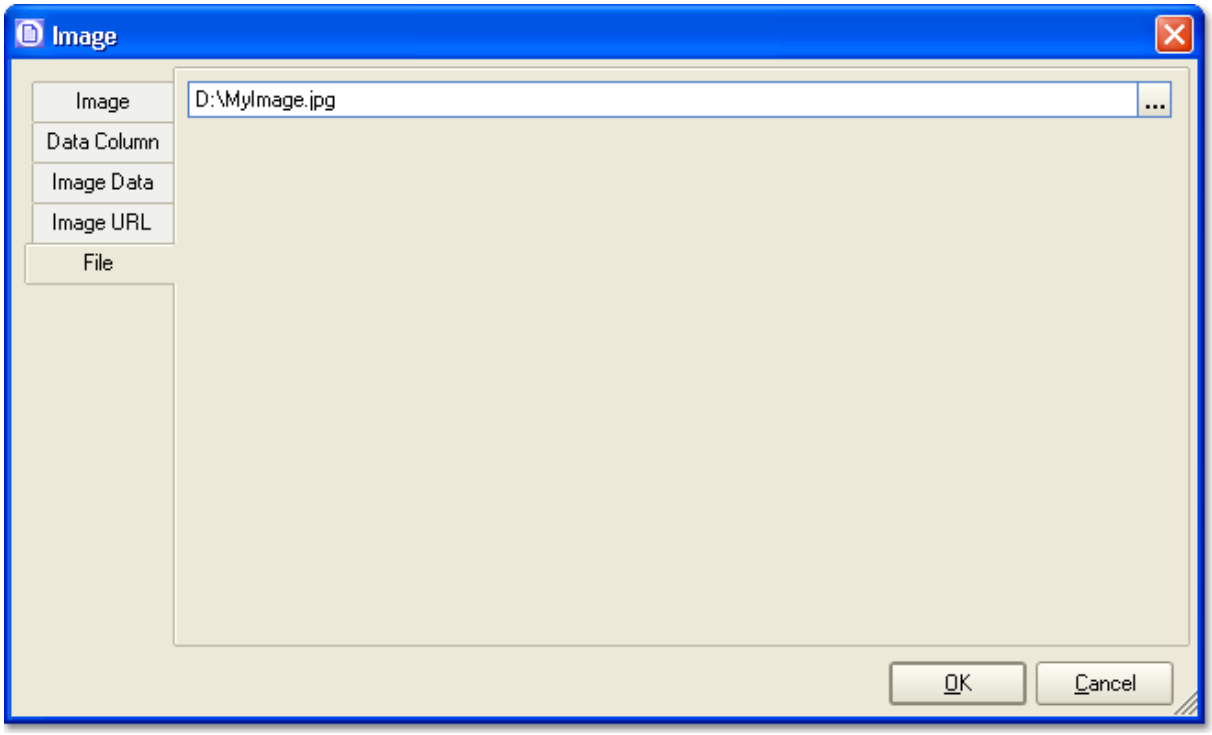


Notice. Instead of variable the expression can be used. This expression will return an image.

4. By the internet link. This way works at runtime.



5. From file. File selection allows loading images from image files. Images of the standard Bitmap type and *.emf and *.wmf metafiles are supported. Images are not serialized in the code.



Also it is possible to load images from a code of application before report rendering:

```
C#

StiReport report = new StiReport();
report.Load("D:\\myimage.mrt");
StiImage image = new StiImage();
image.Image = Image.FromFile("d:\\SomeImage.jpg");
report.Compile();
report["MyImage"] = image;
report.Show();
```

Autosize

Automatic resizing of components is controlled by two properties available in report components: CanGrow and CanShrink.

Can Grow

If the CanGrow property is set to true the component can automatically increase its size if the information contained within it does not fit in the space available. If it is set to false the information will be cropped to the component size, as in the examples below:

Can Shrink

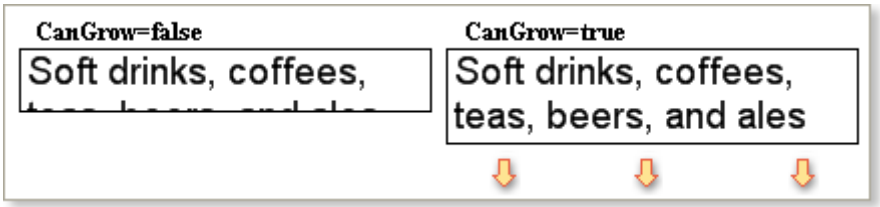
If the CanShrink property is set to true the component can automatically reduce its size so that it fits exactly to the size of the text or image being displayed. If it is set to false the component remains the same size leaving unused space around the information it contains, as in the examples below.

Using this property will help you to prevent wasted space on report pages

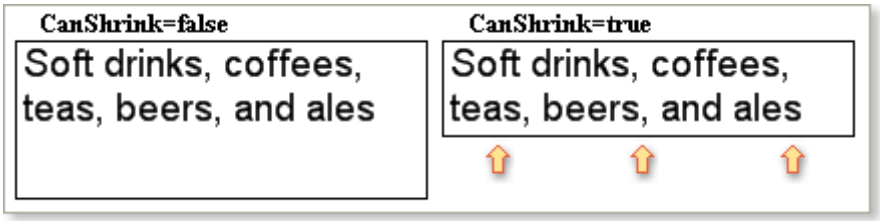
The report generator allows you to set both CanGrow and CanShrink properties. If you set both properties to true the component will automatically increase or decrease in size whenever appropriate. The example below shows an image component that is not large enough to support the height of the image but is too wide for the image width. By setting the CanGrow and CanShrink properties to true the size of the component changes automatically and exactly matches the size of the image.

Automatically Resizing Text

The automatic resizing of text behaves differently from other components. The **CanGrow** and **CanShrink** properties affect only the height of a text component and not the width. The example below shows an example of the **CanGrow** property causing the text height to change:



The **CanShrink** property works in the opposite way, so if it is set to true and there is more space than is needed for the text the report generator will automatically decrease the height of the text component.



As with other components it is possible to set both properties to true. In this case, the height will automatically increase or decrease depending on the size of a text.

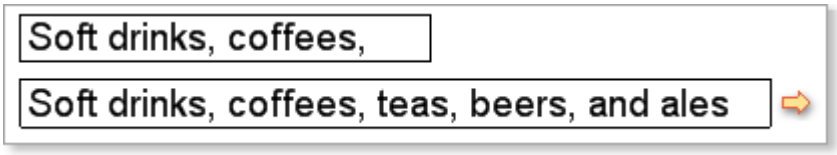
Word Wrap Property

The **WordWrap** property controls whether or not the text in the control automatically wraps when it becomes too long to fit in a single line. If the **WordWrap** property is set to false then the text is cropped at the border of the component, but when set to true new lines are created until all the text is displayed on multiple lines.

When automatically resizing a text component with the **WordWrap** property set to false the report generator will calculate the new size based on the height of a single line only. If you want the report generator to increase the height of the component based on all the text lines then the value of the **WordWrap** property should be set to true so that the text automatically wraps and the calculation can be based on the combined height of all the text lines.

The AutoWidth property

In addition to the **CanGrow** and **CanShrink** properties the **AutoWidth** property can affect the way a text component changes size. If the **AutoWidth** property is set to true then the text component will automatically change its width to match the width of the text. The **CanGrow**, **CanShrink**, and **AutoWidth** properties can be used simultaneously.



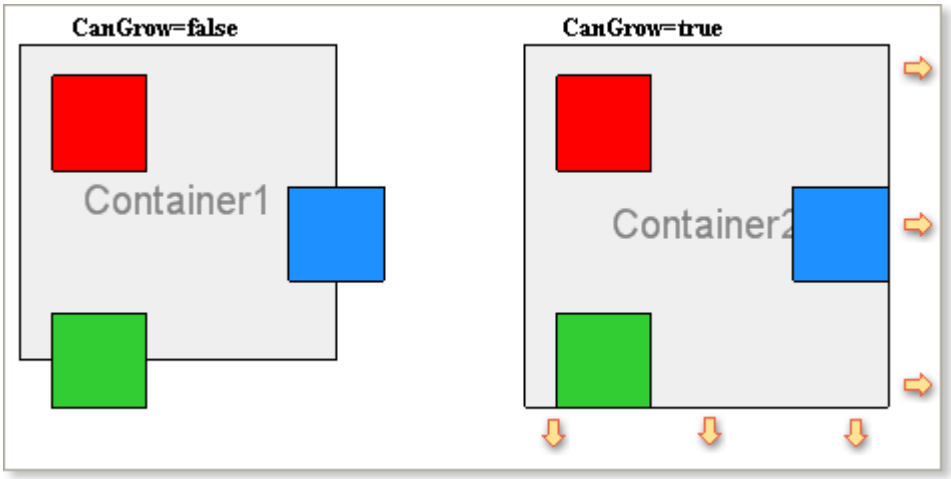
If the **WordWrap** property is set to false, then the height of the text depends on settings of the **CanGrow** and **CanShrink** properties. If the **WordWrap** property is set to false, then the width will be automatically changed.

Important: If the **WordWrap** property is set to false then the height of the text depends on the **CanGrow** and **CanShrink** properties. If the **WordWrap** property is set to false then it will change the width of the text.

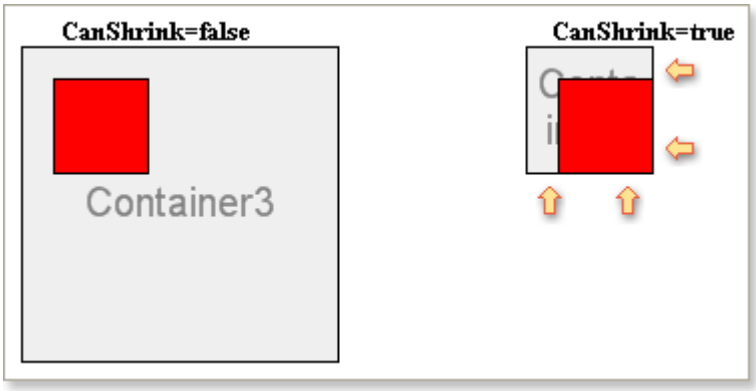
Automatically Resizing Panels

Because **Panels** are only containers and output no visual information in the report it may seem that the **CanGrow** and **CanShrink** properties have no relevance, but this is not the case.

Panel components may contain other components which have specified sizes and positions. If some of the component positions mean that their boundaries cross the border of the panel then setting the **CanGrow** property to true will cause the panel container to be automatically resized so that the child components are wholly enclosed within it. The picture below shows how the **CanGrow** property works:



If the **CanShrink** property is set to true and the bounds of the combination of all the components contained within it are less than the bounds of the panels the panel size will automatically reduce to match the overall size of all components.



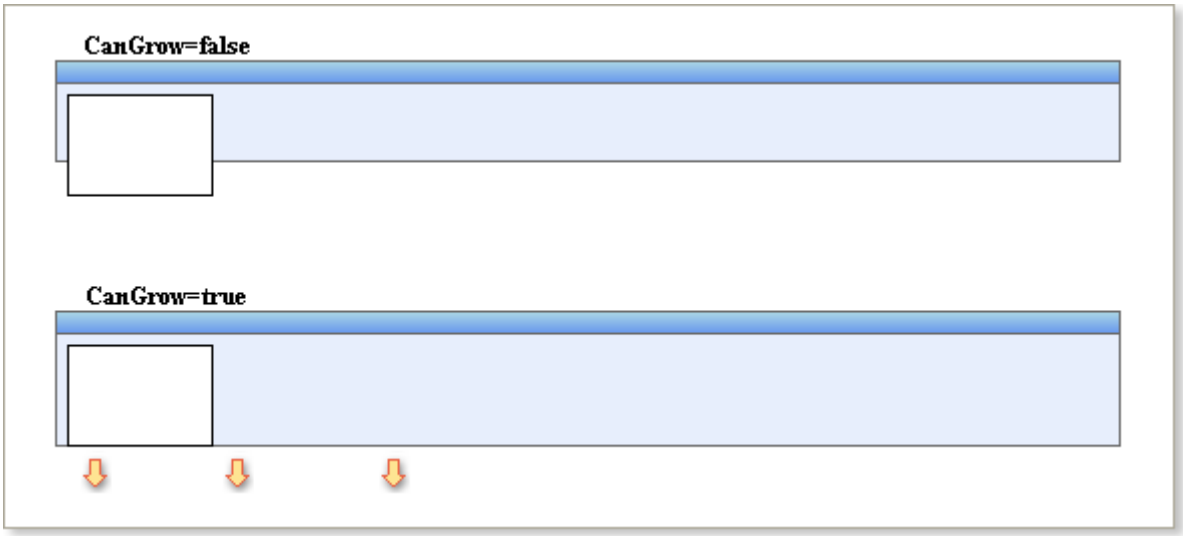
Automatically Resizing Bands

Because bands are inherited from **Panels**, they change their size in the same way. The size of the **Band** can be automatically changed depending on the size of components positioned on the band.

CanGrow Property

It should be noted that most types of band can only automatically change their height - the exception is cross-bands which change their width.

For example, if there is a component on the band which crosses the lower boundary and you set the **CanGrow** property of the band to true, the band height will be automatically increased until the entire component is contained within the band:

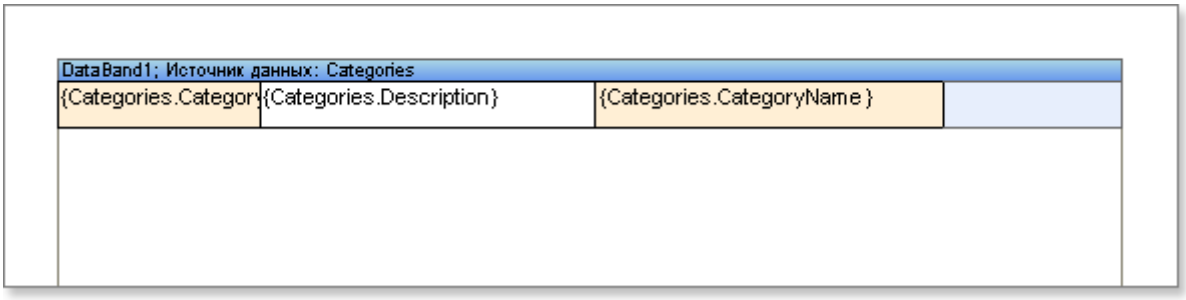


CanShrink Property

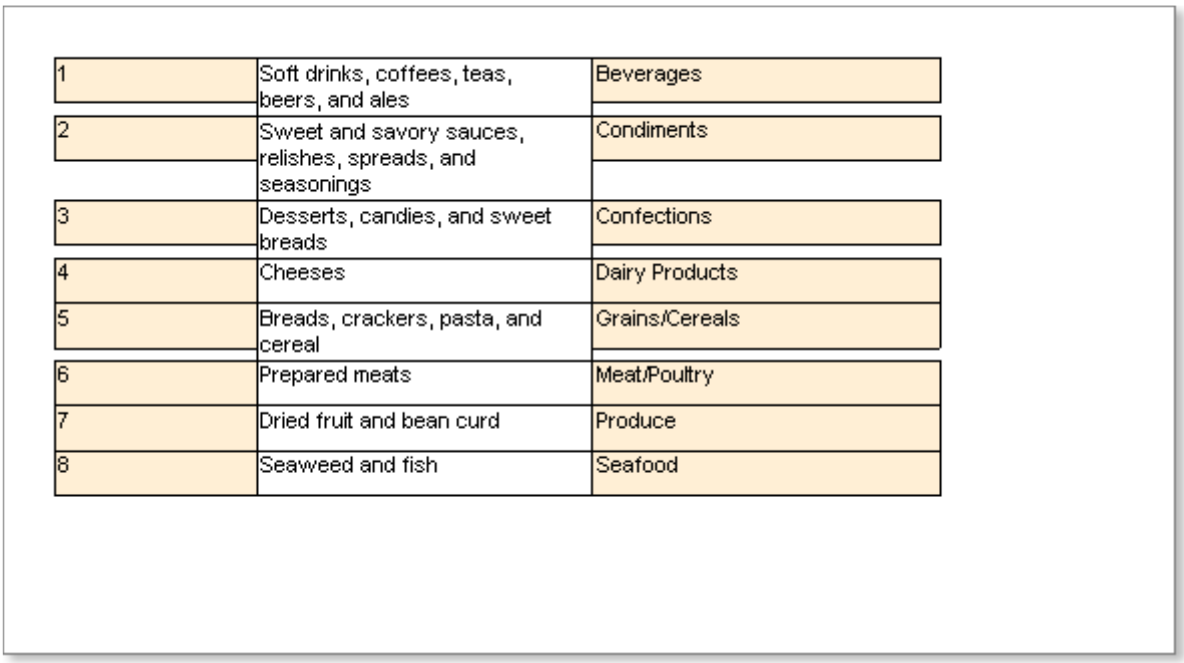
Similarly if there is free space between the boundary of a band and the lower border of the tallest component that it contains and you set the **CanShrink** property to true, the height of the band will automatically be reduced until it matches the lowest point of the lowest contained component:

Binding Bottom Border of Component

Typically there will be more than one component on a band, as in the example shown below:



When rendering a report the height of some of the components may be changed automatically to suit the size of their contents which can result in unwanted breaks in the layout as shown below:



1	Soft drinks, coffees, teas, beers, and ales	Beverages
2	Sweet and savory sauces, relishes, spreads, and seasonings	Condiments
3	Desserts, candies, and sweet breads	Confections
4	Cheeses	Dairy Products
5	Breads, crackers, pasta, and cereal	Grains/Cereals
6	Prepared meats	Meat/Poultry
7	Dried fruit and bean curd	Produce
8	Seaweed and fish	Seafood

In order to prevent this occurring you can bind the bottom border of a component to the lower border of the container in which the component is placed. This binding is done using the **GrowToHeight** property.

GrowToHeight Property

If you set the **GrowToHeight** property to true all components that do not change their size will have their bottom borders bound to the bottom border of the container.

Note: The **GrowToHeight** property binds the bottom border of the component to that of its container whether that container is a **Band** or a **Panel** component.

This will give a consistent and much better looking result as shown below:

1	Soft drinks, coffees, teas, beers, and ales	Beverages
2	Sweet and savory sauces, relishes, spreads, and seasonings	Condiments
3	Desserts, candies, and sweet breads	Confections
4	Cheeses	Dairy Products
5	Breads, crackers, pasta, and cereal	Grains/Cereals
6	Prepared meats	Meat/Poultry
7	Dried fruit and bean curd	Produce
8	Seaweed and fish	Seafood

By default, the **GrowToHeight** property is set to false.

Handling Multiple Components

If there are multiple components on one band that can automatically change their size it is possible set the **GrowToHeight** property for all these components to true. This will cause the height of these components to be automatically adjusted based on the height of the tallest component.

Note: The **GrowToHeight** property can be set for components which automatically change their size as well as those that do not. In this case, if the bottom border is not matched to the bottom border of its container the size of this component will be automatically adjusted to suit.

Automatically Shifting Components

Automatically changing the size of components can lead to a problem when rendering reports - what happens when a change in the size of one component has an adverse effect on another component in the report? For example, if the height of the first component is increased it could overlap a component placed below it.

To prevent this problem the **ShiftMode** property is used.

ShiftMode Property

The **ShiftMode** property allows all components with top borders situated below the top border of an automatically modified component to be automatically shifted down the report so that they maintain the same relative position.

The property has three flag values each of which can be set to **True** or **False**:

- **IncreasingSize**
- **DecreasingSize**
- **OnlyInWidthOfComponent.**

These work as follows:

1. IncreasingSize

If this flag is set to true then any increase in the height of the components located above the specified component causes the component to shift down vertically by the same amount. If the flag is set to false then any increase in the height of the higher components is simply ignored, as shown in the example below:

By default this flag is set to true.

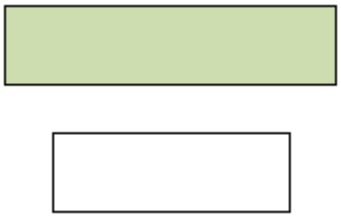
2. DecreasingSize

If this flag is set to true then any decrease the height of the components located above the specified component causes the component to shift up vertically by the same amount. If the flag is set to false then any decrease in the height of the higher components is simply ignored, as shown in the example below:

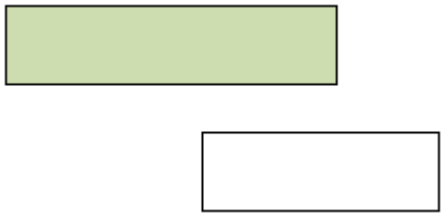
By default, this flag is set to false.

3. OnlyInWidthOfComponent

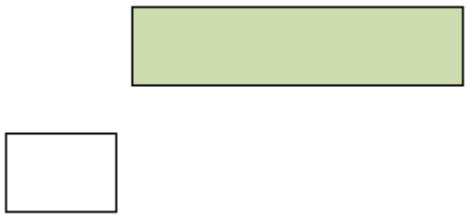
If the flag is set to true, it takes into account changes only to those components that have their left boundary less than the left border of the specified component, and the right border more than the left border of this component as in the examples below:



Or:



If this flag is disabled, the location of the left border of this component is ignored. For example:



By default this flag is disabled.

Barcodes

A barcode is an optical machine-readable representation of data typically made up of parallel bars, varying in width, spacing, or height, which are read by barcode readers. In some cases a line of digits can be placed under a barcode which represent in human readable form the data contained in the barcode.

1D Barcodes

Most commonly barcodes represent their data in the widths and spacings of printed parallel lines which is why they are called linear or 1D (one-dimensional) barcodes or symbolics. Linear barcodes are read in one direction (horizontally). The following linear barcodes are commonly used:

- EAN;
- UPC;
- Code39;
- Code128;
- Codabar;
- Interleaved 2 of 5.

Linear symbolics allow the coding of small amounts of information content (a maximum of 20-30 digits or symbols) and the devices that read them are considered to be simple scanners.

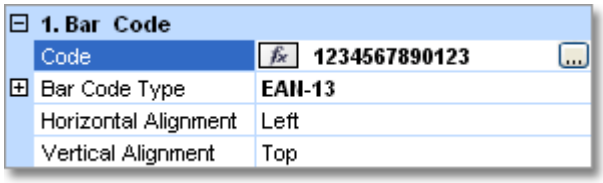
2D Barcodes

2D (two-dimensional) barcodes or symbolics are used for coding large amounts of information in a bar code, potentially up to several pages worth. Such a barcode would consist of square cells, dots, hexagons, and other geometrical figures. Special 2D barcode scanners are required to read the barcodes which decode in two dimensions (horizontal and vertical). The following 2D barcodes are the most common:

- **PDF417**;
- **Datamatrix.**

Setting Barcode Data

The Code property of the Barcode component is used to specify the code of the bar-code.



This property is an expression so can be defined either as a literal string or a code calculation that can generate the barcode based on the content of a data field or any other calculation that may be applicable. For example, the Code below is set as a string:

```
1234567890123
```

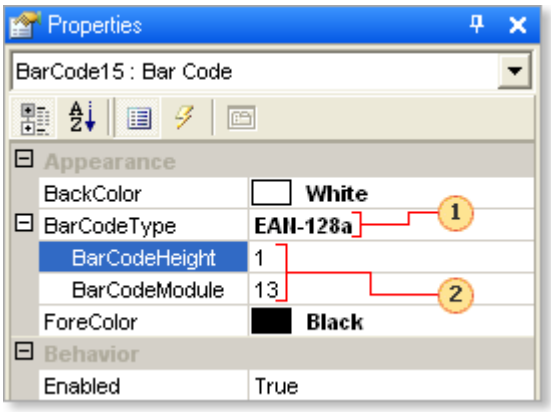
The Code read from a data field:

```
{Items.Code}
```

Using Barcode Components

When using the Barcode components is it important to remember that changing the sizes of those components within the designer does not lead to a change in the printed or displayed size of the barcodes.

All barcodes have to meet a specified standard or it would not be possible to read their data. In many barcodes changing the size of the code is either not allowed or has some limitations. For this reason the size of a barcode is set using special properties. All these properties can be found in the Properties panel of the barcode. For example, on the picture below the Properties panel of the EAN-128a barcode is shown. This particular barcode allows the user to set the BarcodeHeight and the BarCodeModules.



1 The bar-code type.
2 The bar-code properties.

Important: Changing the size of a barcode component does not change the size of the printed barcode. Additional properties are used to change the size and then only on barcode types where changing the size is permitted.

Bar-codes Size

Barcode sizes are very important if they are to read successfully by scanners. Each type of barcode is defined using the following size parameters:

Density

A mil is used to specify the barcode density.

1 mil = 1/1000 inch

Module

Module parameter ("Module", sometimes referred to as the "X dimension") indicates the narrowest bar of a barcode. This parameter is connected with the printing resolution of a barcode and the barcode density. For example, if the narrowest bar is 10 mils it is said that the barcode is printed with 10 mil resolution or that the density of the barcode is 10 mil.

Density
There are two elements of density - the graphics density and information density of a barcode.

Information Density
The information density is the number of characters that can be encoded per inch given a certain X value. The smaller the value of X, the more characters can be encoded in an inch and, thus, the density rises. The information density of a bar-code depends on the character encoding. The less the number of bars and spaces required to encode one symbol the higher the information density of the bar-code.

Graphics Density
The graphics density of the barcode is connected with the bar-code size. The classification of graphic linear barcodes is shown in the table below:

-

Graphics density	Printing resolution
Very high density	< 4 mils
High density	4 mils .. 6 mils
Medium density	7 mils .. 13 mils
Low density	14 mils .. 20 mils
Very low density	> 20 mils

Width
The barcode width depends on the graphic and information density. The density is limited by the resolution of the printer and scanner, but the barcode width depends on the information density of the symbolic. Different symbolics may have different barcode widths even if their graphic density is the same.

Height
The height of the barcode is needed only to allow scanners to easily read it. Usually the best barcode length is based on the ratio of height to width of around 1:5-6.

Spaces
This is a very important attribute, especially for linear barcodes. Spacing is the light regions at the start and the end of the bar-code. They are required for the scanner to identify the bar-code measurements.

Linear Bar-codes

There are a great many linear barcode specifications available, including many that are based on the EAN/UPC specification.

EAN/UPC Based

EAN/UPC barcodes are based on the EAN.UCC system which was created in the USA in 1973 by the Uniform Product Code Council company, now known as Uniform Code Council, Inc. (UCC).

UPC
Initially, UCC developed a 12-digit ID and the **UPC** (Uniform Product Code) barcode. The first **UPC** code was scanned in 1974.

EAN

After successful implementation of the **UPC** system in 1977 the European Article Numbering Association format was created as a superset of the UCC system and uses 13-digit identification numbers but the same data structures as UPC barcodes.

Today global compatibility is reached by using the 14-digit GTIN format. This provides unique identification of goods all over the world.

In this section details of the **UPC-A**, **UPC-E**, **EAN-8**, **EAN-13**, **EAN-128**, **ITF-14** barcodes of "General EAN.UCC Specifications" and based on those the **JAN-8**, **JAN-13**, **ISBN-10**, **ISBN-13** barcodes are displayed.

What is EAN.UCC System?

The EAN.UCC system appeared in the USA and was created in 1973 by the Uniform Product Code Council company. Now this company is known as Uniform Code Council, Inc. (UCC). Initially, the UCC was developed 12-digit ID and appropriate the **UPC** bar-code (Uniform Product Code). The first **UPC** code was scanned in 1974. After successful implementation of the **UPC** system in 1977 the European Article Numbering Association was created. The **EAN** system was created as superset of the UCC system and uses the 13-digit identification numbers but the same structures of data as bar-codes. So the EAN.UCC system was extended. Today the complete global compatibility is reached by using the 14-digit GTIN format. This provides unique goods ID all over the world.

In this section **UPC-A**, **UPC-E**, **EAN-8**, **EAN-13**, **EAN-128**, **ITF-14** barcodes of "General EAN.UCC Specifications" and based on them **JAN-8**, **JAN-13**, **ISBN-10**, **ISBN-13** barcodes are viewed.

UPC-A

UPC-A was the first barcode, created by Uniform Code Council, Inc. in 1973. The **UPC-A** barcode is an unbroken code with a fixed length and high density. It is used for tracking trade items in stores, and otherwise marking goods.

Valid symbols:	0123456789
Length:	fixed, 12 characters
Check digit:	one, modulo-10 algorithm

UPC-A barcodes consist of 11 data digits and one check digit. The first digit is a number system digit that normally represents the type of product being identified. The following 5 digits are a manufacturers code and the next 5 digits are used to identify a specific product.

The barcode contains the following elements:

- 1 digit - system number.
- 5 digits - manufacturer code.
- 5 digits - product code.
- 1 digit - check digit.

The barcode does not contain any information about characteristics of a product, but only a unique number relating to an entry in the International data base where all information about the particular product is stored. An example barcode in **UPC-A** format:

UPC-A Barcode

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "123456789012" is the number encoded in the barcode.

UPC-E

A **UPC-E** is a smaller seven digit UPC symbology for number system 0. For **UPC-E** barcodes, normally 6 digits are specified and the barcode calculates the seventh check digit.

Valid symbols:	0123456789
Length:	fixed, 8 characters
Check digit:	one, modulo-10 algorithm

Before the Middle guard bars, a binary 1 is indicated by a bar, while a 0 is indicated by a space. After the Middle guard bars, however, the patterns are optically inverted. In other words, a 1 is now indicated by a space, and a 0 is now indicated by a bar. It has the same basic structure as the **UPC-A** barcode.

A "UPC-E" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "1234567" is the number encoded in the barcode.

EAN-13

The **EAN-13** barcode was created based on the UPC-A barcode as an extension of the EAN.UCC system used outside the USA. EAN-13 is the European version of UPC-A.

Valid symbols:	0123456789
Length:	fixed, 13 characters
Check digit:	one, modulo-10 algorithm

The structure of EAN-13 barcode is the same as UPC-A. Each bar-code character consist of 2 bars and 2 spaces, which may have a width from 1 to 4 modules. The first digit is always placed outside the symbol, additionally the right quiet zone indicator (>) is used to indicate the Quiet Zones that are necessary for barcode scanners to work properly.

The bar-code contains the following elements:

- 2 (3) digits - country code.
- 5 (4) digits - manufacturer code.
- 5 digits - product code.
- 1 digit - check digit.

The barcode does not contain any information about characteristics of a product, but only a unique number relating to an entry in the International data base where all information about the particular product is stored. An example barcode in EAN-13 format:

An "EAN-13" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "123456789012" is the number encoded in the barcode.

EAN-8

The **EAN-8** bar-code was developed for use on small packages. It is used instead of the EAN-13 bar-code where an EAN-13 barcode would be too large, for example on packets of gum.

Valid symbols:	0123456789
Length:	fixed, 8 characters
Check digit:	one, modulo-10 algorithm

The structure of the **EAN-8** bar-code is in the same as the structure of the **EAN-13** bar-code. The check digit is calculated automatically irrespective of input data.

The bar-code contains the following elements:

- 3 digits - a prefix of the national organization.
- 4 digits - product code.
- 1 digit - check digit.

This bar-code does not contain the code of the producer and has only 4 digits. As a result there can only be 10000 specimen products per organization, so the **EAN-8** bar-code is provided only to those organizations which really need it.

An "EAN-8" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "12345670" is the number encoded in the barcode.

Add-On Symbols

Add-on Symbols (barcodes) can be used in some applications together with the EAN-13, UPC-A, and UPC-E bar-codes. Add-on Symbols may contain 2 or 5 additional digits and are usually placed to the right of the barcode.

Valid symbols:	0123456789
Length:	fixed, 2 or 5 characters
Check digit:	no

The "UPC-E" bar-code with the "02" Add-On Symbols

The "EAN-13" bar-code with the "00321" Add-on Symbols

EAN-128

The **EAN-128** bar-code is a subset of the Code128 barcode which uses a variable length, high density, alphanumeric symbology. It allows the output of 128 characters of ASCII and is effective for digits. There are actually four subcodes,

which can be mixed within a single barcode: **EAN-128a**, **EAN-128b**, **EAN-128c**, and **EAN-128auto** (will automatically switch between code sets to encode the ASCII values).

Valid symbols:	EAN128a: ASCII character 0 to 95 EAN128b: ASCII character 32 to 127 EAN128c: pairs of digits from 00 to 99
Length:	Variable
Check digit:	one, modulo-103 algorithm

The structure of the **EAN-128** bar-code is the same as for the **Code128** bar-code. Elements of the bar-code consist of three bars and three spaces. Bars and spaces have module construction and their width consists of either one or four modules. The width of an element consists of eleven modules.

To difference between the **EAN-128** bar-code and the **Code128** bar-code is that the FNC1 is placed after the start character. This character is reserved for the EAN.UCC system.

An "EAN-128c" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "0123456789012345" is the number encoded in the barcode.

ITF-14

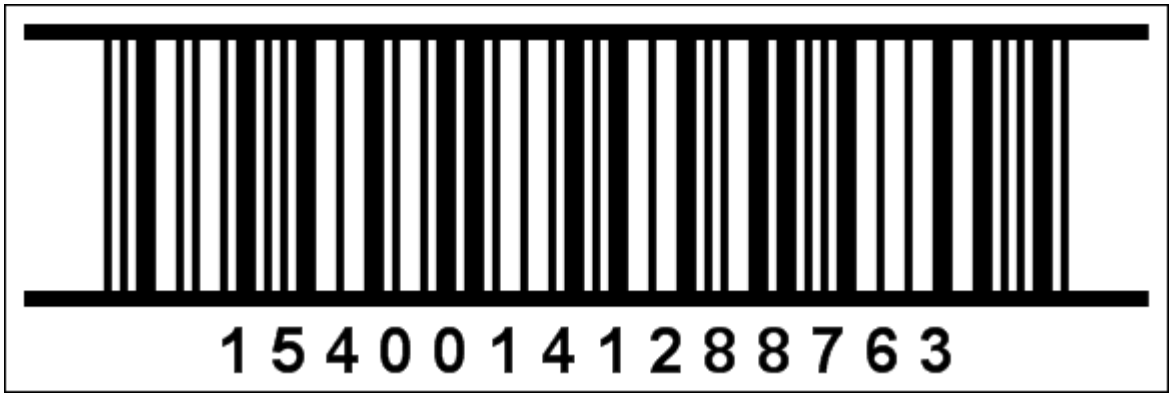
The **ITF-14** bar-code was developed to encode a Global Trade Item Number. The ITF bar-code has the nominal size of (152*44mm) and low requirements to the printing surface. Therefore, it can be printed not only on a label but directly onto a packing carton.

Valid symbols:	0123456789
Length:	fixed, 14 characters
Check digit:	one, modulo-10 algorithm

Each character is encoded using two broad and three narrow bars/spaces. The ITF-14 will always encode 14 digits.

The bar-code contains the following elements:

- 1 digit - logic.
- 3 digits - Global Trade prefix.
- 6 digits - Producer code.
- 3 digits - Product code.
- 1 digit - Check digit.



An "ITF-14" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "15400141288763" is the number encoded in the barcode.

JAN-13

A **JAN-13** barcode is another name for an EAN-13 barcode dedicated for use only in Japan. The first two digits should be 45 or 49 which indicate Japan.

A "JAN-13" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "4901234567894" is the number encoded in the barcode.

JAN-8

A **JAN-8** barcode is another name for an EAN-8 barcode dedicated for use only in Japan. The first two digits of the barcode should be 45 or 49 to indicate Japan.

A "JAN-8" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "49123456" is a number encoded in the barcode.

ISBN-10

ISBN is the abbreviation of International Standard Book Number - a unique, numeric commercial book identifier. Based upon the 9-digit Standard Book Numbering (SBN) code introduced in 1966, 10-digit **ISBN** format was developed in 1970 and became the international standard.

Valid symbols:	0123456789
Length:	Not variable, 10 symbols
Check digit:	One

The **ISBN**, assigned to books of 2006 contained 10 digits length and consist of four fields of variable length:

- For a 13 digit ISBN, a GS1 prefix: 978 or 979.
- The group identifier, (language-sharing country group).
- The publisher code.
- The item number.
- A checksum character or check digit.

An "ISBN-10" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "80-902734-1-6" is the number encoded in the barcode.

ISBN-13

ISBN is the abbreviation of International Standard Book Number - a unique, numeric commercial book identifier. The ISBN-13 specification was Issued from January 2007, and describes how the 13-digit **ISBN** check digit is calculated.

Valid symbols:	0123456789
Length:	fixed, 13 symbols
Check digit:	one, algorithm modulo-10

The **ISBN** assigned to books after 2006 contained 13 digits length and consist of four fields of variable length:

- prefix: 978 or 979.

- The group identifier, (language-sharing country group).
- The publisher code.
- The item number.
- A checksum character or check digit.

A "ISBN-13" barcode.

Note the 'human readable' digits at the foot which can be used by operators if the label becomes damaged or will not scan for some reason - "978-0-306-40615-7" is a number encoded in the barcode.

Other Barcodes

Pharmacode

A **Pharmacode** barcode is used in the pharmaceutical industry as a packing control system. The Pharmacode barcode is placed on the package.

Valid symbols:	A whole number from 3 to 131070
Length:	Variable, 1..6 characters of a digit
Check digit:	No

A **Pharmacode** barcode can represent only a single integer from 3 to 131070. All digits in the specified range make correct barcodes, but some of these barcodes can be unreadable because all barcodes are identical. So, the following digits should not be used:

3, 6, 7, 14, 15, 30, 31, 62, 63, 126, 127, 254, 255, 510, 511, 1022, 1023, 2046, 2047, 4094, 4095, 8190, 8191, 16382, 16383, 32766, 32767, 65534, 65535, and 131070.

A "Pharmacode" barcode. "12345" is a number encoded in the barcode.

Plessey

A **Plessey** barcode was created by **Plessey** company in England on March 1971. The Plessey barcode is widely used in libraries, supermarkets, and production environments. A variant of the barcode known as Anker Code and appropriate scanners were provided by the ADS company.

Encoding technology of the **Plessey** barcode was used by MSE Data Corporation. This company used it to create an MSI barcode that sometimes is called 'modified Plessey'.

This bar-code is now obsolete and new scanners cannot read it.

Valid symbols:	0123456789ABCDEF
Length:	Variable
Check digit:	No, one or two; Algorithm modulo-10 or modulo-11

Plessey is a variable length, numeric-only symbology. It allows to output digits 0..9 and letters A, B, C, D, E, F but more frequently only digits are used. Check digits calculated using the modulo-10 or modulo-11 algorithm can be used.

Each character of the bar-code consist of 4 elements. An element consists of a bar and a space and has 3 modules width. If the element is the binary 0 then the bar-code has 1 module width and a space has 2 modules. If the element is the binary 1 the the bar has 2 module width and a space has 1 module. So, each character has 12 modules length. Therefore, this bar-code has very low data density.

A "Plessey" barcode. "1234567890" is a number encoded in the barcode.

Msi

The **Msi** bar-code developed by the MSI Data Corporation. It is based on the original Plessey symbology. Sometimes the **Msi** bar-code is called the **Modified Plessey**. The basic implementation of the **Msi** bar-code is used for warehouse shelves and inventory.

Valid symbols:	0123456789
Length:	Variable
Check digit:	none, one or two; algorithm modulo-10 or modulo-11

Msi is a variable length, numeric-only symbology and allows to output digits 0..9. One or two check digits calculated by **modulo-10** or **modulo-11** can be used. Each character of the bar-code consist of 4 elements. If the element is the binary 0 then the bar-code has the 1 module width and a space has 2 modules. If the element is the binary 1 the the bar the 2 module width and a space has 1 modules. So, each character has 12 modules length. Therefore, this bar-code has very low data density.

A "Msi" barcode. "1234567890" is a number encoded in the barcode.

2of5

The **2of5** bar-code was developed 40 years ago. This is a low density variable length numeric. This barcode is used in manufacture and is known as Code 25, Code 25 Standard or Code 25 Industrial. It is very seldom used these days.

Valid symbols:	0123456789
Length:	Variable
Check digit:	no

A "2of5 Standard" barcode. "1234567890" is a number encoded in the barcode.

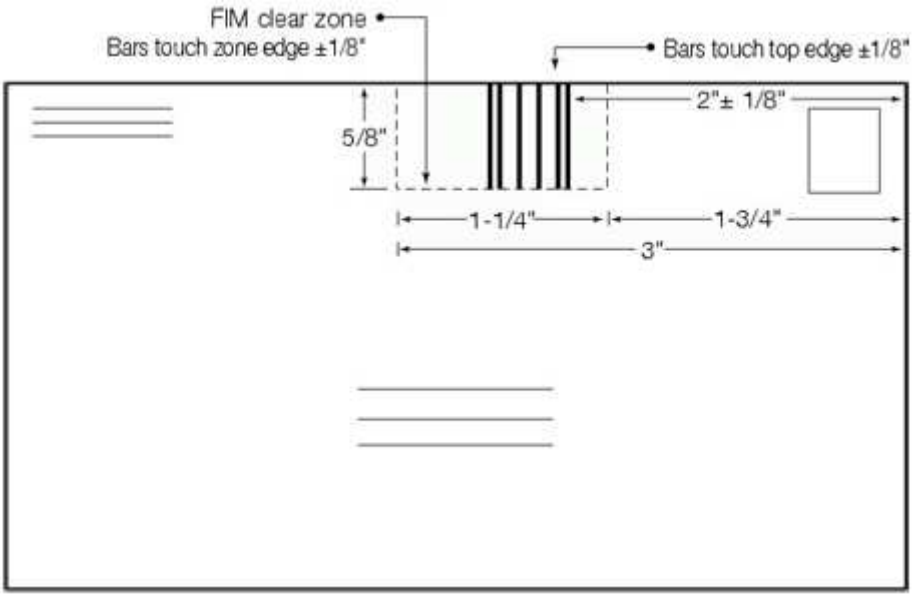
The **2of5 Interleaved** barcode is a high density variable length numeric only symbology that encodes digit pairs in an interleaved manner. This bar-code is developed of the Code 25 Standard. It is usually used in the industrial.

Valid symbols:	0123456789
Length:	Variable, even
Check digit:	No

A "2of5 Interleaved" barcode. "1234567890" is a number encoded in the barcode.

FIM

Facing Identification Mark (**FIM**) is the type of postal bar code used in automated mail processing by the U.S. Postal Service. FIM is a set of vertical bars. FIM patterns are placed in the upper right corner along the top edge and two inches in from the right edge of letters and cards.



The FIM bar-code on a card

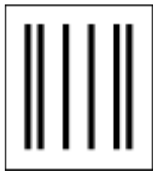
The table below shows basic parameters of the **FIM** bar-code.

Valid symbols:	ABCD
Length:	Fixed, 1 symbol
Check digit:	No

The **FIM** bar-code consists of nine elements. Each element can be 1 (vertical bar) or 0 (space). Four bar-codes are used:

FIM A: 110010011
FIM B: 101101101
FIM C: 110101011
FIM D: 111010111

So the data row should contain 1 of 4 available characters: A, B, C, D.



A "FIM C" barcode

Codabar

The **Codabar** is a linear barcode symbology developed in 1972. It can be called as NW-7, USD-4, Code 2 of 7 (2 values of a bar length, 7 elements). It is frequently used in medicine (for example, blood bank forms).

Valid symbols:	0123456789 - $: / . + ABCD (only as start/stop symbols)

Length:	Variable
Check digit:	no

Two bars and three spaces are used for encoding. The bar-code has four different sets of start/stop characters: A, B, C, D. These characters are used only as start/stop characters and should not be appeared in the bar-code.

A "Codabar" barcode. "A12345678A" is a number encoded in the barcode.

Postnet

The **POSTNET** (POSTal Numeric Encoding Technique) bar-code was developed by the United States Postal Service for encoding ZIP-codes and correct sorting using BCSs. It can encode ZIP, ZIP+4, and ZIP+4+2 postal codes.

Valid symbols:	0123456789
Length:	Fixed, 5, 9 or 11 characters
Check digit:	One, algorithm modulo-10

The Postnet bar-code can encode 0-9 digits. The bar-code consist of short and long bars. Each symbol of data is encoded using five bars. This bar-code contains only one check symbol, that is calculated by the modulo-10 algorithm.

A "Postnet" barcode. "11387975204" is a number encoded in the barcode.

Australia Post 4-state

The **Australia Post 4-Stage** bar-code is used in Australia for the purposes of sorting and directing mail.

Valid symbols:	0123456789
Length:	FCC - fixed, 2 characters, DPID - fixed, 8 characters, CustomerInfo variable
Check digit:	Four, ReedSolomon algorithm

The barcode consists of 4 elements (4 conditions), each has its own name, value. Each element consists of two bars and two spaces. Each barcode contains 4 check symbols, calculated by the ReedSolomon algorithm. The value of these symbols are usually printed after the text of the barcode.

The string may contain the following parts:

- FCC ("Format Control Code"), 2 digits. May have the following values 11, 45, 87, 92, 59, 62, 44.
- DPID ("Delivery Point Identifier" or "Sorting Code"), 8 digits.
- CustomerInfo may contain 0-9, A-Z, a-z, # symbols and space. The maximal length depends on FCC:

Notes:

If FCC = 11, 45, 87, 92 then the CustomerInfo in ignored.
If FCC = 59 then the CustomerInfo may contain 8 digits or 5 letters/digits.
If FCC = 62, 44 then the CustomerInfo may contain 15 digits or 10 letters/digits.

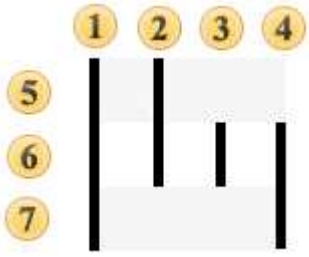
A "Australia Post 4-state" barcode. "1138797520" is a number encoded in the barcode.

Royal TPG Post KIX 4-State

This symbology is used by Royal Dutch TPG Post (Netherlands) for Postal code and automatic mail sorting. It provides information about the address of the receiver. This symbology encodes alpha-numeric characters (0-9, A-Z). The barcode is also known as Royal TNT Post Kix, Dutch KIX 4-State Barcode, Kix Barcode, TPG KIX, Klantenindex Barcode, TPGPOST KIX.

Valid symbols:	0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ
Length:	Variable
Check digit:	none

The barcode consists of four types of bars. The Barcode structure is shown in the picture below:



1 Full bar;
2 Ascender;
3 Tracker;
4 Descender;
5 Ascending Region;
6 Tracking Region;
7 Descending Region.

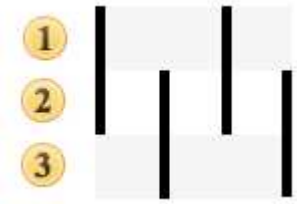
A Royal TPG Post KIX 4-State Barcode. "1234567890123" is a number encoded in the barcode.

Royal Mail 4-state

The **Royal Mail 4-state** is a barcode symbology for use in automated mail sort process. There are 38 valid characters in the entire character set:

Valid symbols:	numeric characters 0-9; alpha characters A-Z
Length:	Variable
Check digit:	none

A barcode consists of four bars, of which two are ascenders and two descenders. The tracking region is present in all bars.



1 Ascending Region;
2 Tracking Region;
3 Descending Region.

A Royal Mail 4-state Barcode. "1234567890123" is a number encoded in the barcode.

Code11

The **Code 11** bar-code was developed by **Intermec** in 1977. It is used in telecommunications.

Valid symbols:	0123456789 -
Length:	Variable
Check digit:	None, one or two; modulo-10 algorithm

This barcode has high density and can encode any length string consisting of the digits 0-9 and the dash character. The **Code 11** uses one or two check digits and two check symbols. Usually, if the length of the string is less than 10 symbols then only one check symbol is used. If the the length of the string is 10 symbols and more then 2 check symbols are used. The value of the check symbol is calculated by the modulo-10 algorithm.

A "Code 11" barcode. "12345-6789" is a number encoded in the barcode.

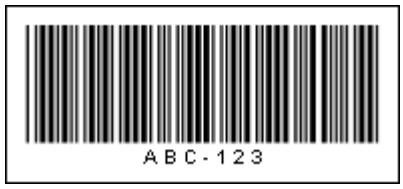
Code39

Code 39 is a variable length symbology that can encode 44 characters. Code 39 is the most popular symbology in the non-retail world and is used extensively in manufacturing, military, and medicine applications.

Valid symbols:	0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ -.$/+% space
Length:	Variable
Check digit:	No, according to the specification; In practice - one, modulo-43 algorithm

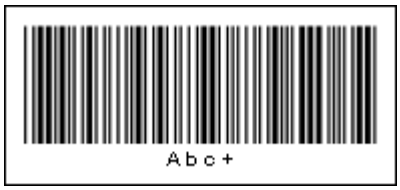
Each Code 39 bar code has a start/stop character represented by an asterisk (*).The bar-code code does not contain the check character but can be added programmatically. Each character starts with a 'dark bar' that consists of 5 dark and 4 blank bars. The ratio between narrow and wide bars may range from 2.2:1 to 3:1.

The Code 39 barcode has low data density. It requires more free space than Code 128, but the Code 39 barcode can be identified by any barcode scanner.



A "Code 39" barcode. "ABC-123" is a number encoded in the barcode.

Code 39 extended is the version of the **Code 39** bar-code which also supports the ASCII set of characters. The 0-9, A-Z, "." and "-" characters are encoded the same as of the **Code 39** bar-code.



A "Code 39 extended" barcode. "Abc+" is a number encoded in the barcode.

Note: Barcode scanners cannot differentiate between the **Code 39** and **Code 39 extended** barcodes. It is necessary to select the correct barcode either by setting a property on the scanner or programmatically.

Code93

The **Code 93** is a variable length symbology that can encode the complete 128 ASCII character set. This barcode was developed as an enhanced version of the Code 39 barcode. It has a higher density than either the Code 39 or the Code 128 bar-code.

Valid symbols:	0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ -.$/+% space
Length:	Variable
Check digit:	Two, algorithm modulo-47

The Code 93 barcode may encode Latin letters (from A to Z), digits (from 0 to 9) and a group of special characters. The barcode always contains two check characters. Each characters consist of nine modules which are joined in 3 groups. Each group has one black bar and one white bar.

A "Code 93" barcode. "ABC-123" is a number encoded in the barcode.

Code 93 extended is a version of the **Code 93** barcode that supports a set of ASCII characters. All additional symbols are encoded as a sequence of two **Code 93** characters. The first character is always one of four special characters. Therefore, scanners can always identify the different versions of the bar-code.

A "Code 93 extended" barcode. "Abc+" is a number encoded in the barcode.

Code128

The **Code128** barcode was developed in 1981. It is a variable length, high density, alphanumeric symbology. It allows the output of 128 characters of ASCII and is effective for digits. There are actually four subcodes, which can be mixed within a single barcode: **Code128a**, **Code128b**, **Code128c**, and **Code128auto** (will automatically switch between code sets to encode the ASCII values).

Valid symbols:	Code128a: ASCII character 0 to 95 Code128b: ASCII character 32 to 127

	Code128c: pairs of digits from 00 to 99
Length:	Variable
Check digit:	One, algorithm modulo-103

The barcode consist of three bars and three spaces. Bars and spaces have module construction and their width consist of one or four modules. The width of an element consist of eleven modules. The "Stop" sign consist of 13 modules and has four bars and three spaces. The check sum is calculated automatically.

A "Code128c" barcode. "0123456789012345" is a number encoded in the barcode.

2D barcodes parameters

The table below shows the list of linear bar-codes supported by Stimulsoft Reports.

-

Type	Length	Check symbols	Checksum algorithm	0-9	A-Z	a-z	other symbols
UPC-A	12	1	modulo-10	+			
UPC-E	8	1	modulo-10	+			
EAN-13	13	1	modulo-10	+			
EAN-8	8	1	modulo-10	+			
EAN-128a	var	1	modulo-103	+	+	ASCII 0 to 95	
EAN-128b	var	1	modulo-103	+	+	+	ASCII 32 to 127
EAN-128c	var	1	modulo-103	+			
ITF-14	14	1	modulo-10	+			
JAN-13	13	1	modulo-10	+			
JAN-8	8	1	modulo-10	+			
ISBN-10	10	1	modulo-10	+			
ISBN-13	13	1	modulo-10	+			
Pharmacode	1..6	-	-	int 3..131070			
Plessey	var	0-2	modulo-10/11	+	A B C D E F		
Msi	var	0-2	modulo-10/11	+			
2of5 Standard	var	-	-	+			
2of5 Interleaved	var	-	-	+			

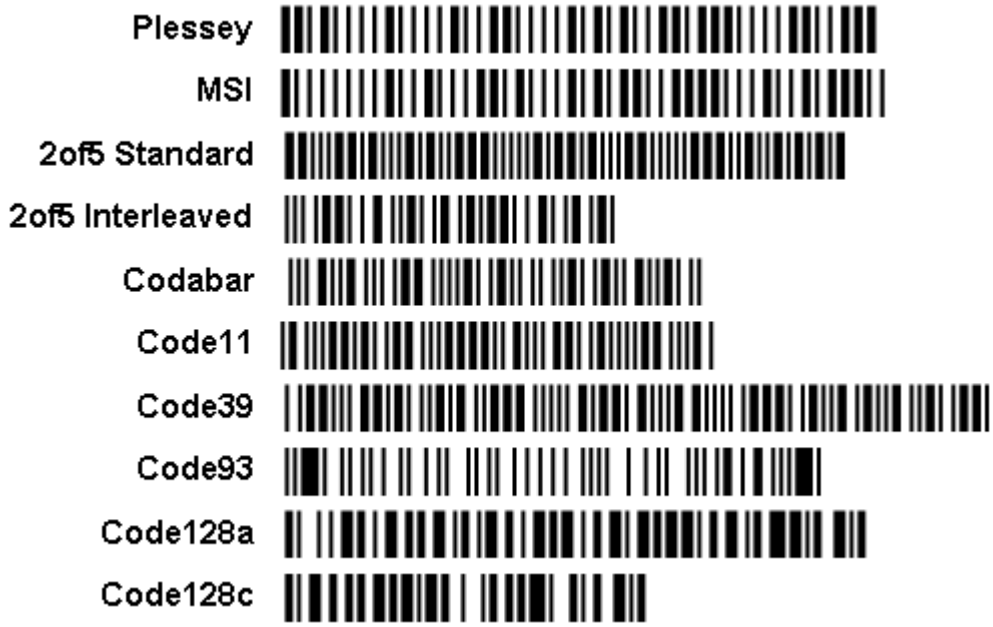
FIM	1	-	-	A B C D			
Codabar	var	-	-	+	- $: / . +		
Postnet	5, 9, 11	1	modulo-10	+			
Australia Post	10[+var]	4	ReedSolomon	+			
Code 11	var	0-2, A	modulo-11	+	-		
Code 39	var	0-1	modulo-43	+	+	- . $ / + % space	
Code 39 ext	var	0-1	modulo-43	+	+	+	full ASCII
Code 93	var	2	modulo-47	+	+	-.$/+% space	
Code 93 ext	var	2	modulo-47	+	+	+	full ASCII
Code128a	var	1	modulo-103	+	+	ASCII 0 to 95	
Code128b	var	1	modulo-103	+	+	+	ASCII 32 to 127
Code128c	var	1	modulo-103	+			

Explanation:

- "Length" - is the data length, it is the number of characters, which can the bar-code can encode; "var" means the variable length.
- "Check symbols" - possible number of check digits; "A" means that number of check digits can be chosen automatically.
- "Checksum algorithm" - the algorithm for calculating check digits.
- "0-9", "A-Z", "a-z" - ranges of symbols; + means that the bar-code can encode characters of this range.
- "other symbols" - other symbols which the bar-code can encode.

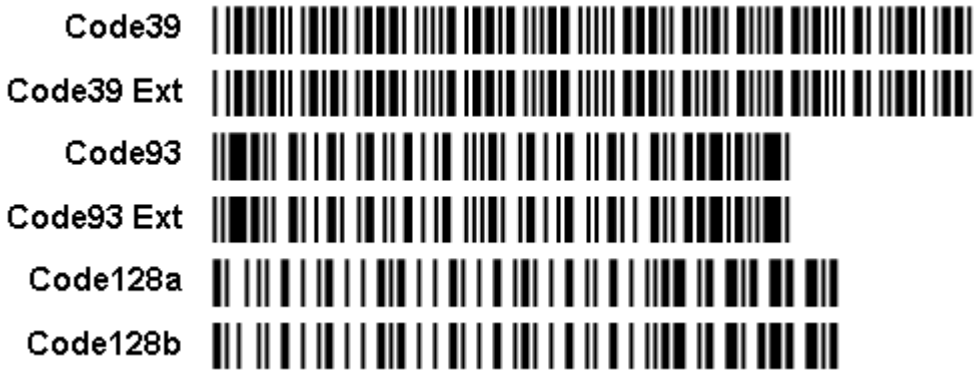
Barcode Sizes

Below is a comparison of bar-codes of variable length, which can encode the numbers 0 to 9. All bar-codes have the same input data - the row has "ABCDEFGHIJK", and the same module 20, other parameters set by default.



Coding English Uppercase Letters

Below is a comparison of the barcodes of variable length which can encode uppercase English letters. All barcodes have the same input data - the row has "ABCDEFGHIJK", and the same module 20, other parameters set by default.



Coding English lowercase letters

Below is a comparison of the barcodes of variable length, which can encode lowercase English letters. All bar-codes have the same input data - the row has "abcdefghijk", and the same module 20, other parameters set by default.

2D Barcodes

A matrix code, also known as a 2D barcode or simply a 2D code, is a two-dimensional way of representing information. It is similar to a linear (1-dimensional) barcode, but has more data representation capability.

Today a lot different symbolics of 2D bar-codes are available. Stimulsoft Reports supports three most popular bar-codes: **PDF417 Datamatrix**, and **QR Code**.

PDF417

The **PDF417** barcode was developed by Symbol Technologies in 1991. The name of the barcode consist of 2 parts. The PDF comes from Portable Data File. The 417 comes from the structure of the bar-code: each bar-code character consists of 17 modules, each of which consists of 4 bars and 1 space.

PDF417 is a high density 2 dimensional bar code symbology that consists of a stacked set of smaller bar codes. Any ASCII characters can be encoded in this bar-code. The length of data depends on the encoding mode and can reach 1100 bytes, or 1800 text characters, or 2600 digits.

The bar-code contains from 3 to 90 rows each of which is like a small linear bar code. Each row has:

- A quiet zone.
- A start pattern which identifies the type of symbol as PDF417.
- A "row left" codeword containing information about the row.

- A "row right" codeword with more information about the row.
- A stop pattern.
- A quiet zone.

The bar-code may have any number of rows and columns (patterns in the data row), although the total number of patterns should not be greater then 928. The number of rows and columns can be set using the **DataRows** and **DataColumns** properties. If the **AutoDataRows** and **AutoDataColumns** properties are set to false, then the bar-code size will be fixed. If one of these properties is set to true, then the bar-code size can increased and decreased in this direction depending on data. If both of these properties are set to true, then the size of the bar-code is set automatically, considering the "AspectRatio" parameters (the ratio of the bar-code width to the bar-code height) and **RatioY** (the height of the code word in modules, from 2 to 5).

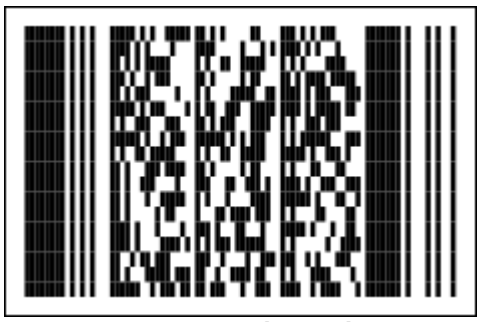
It is possible to select one of three modes of data encoding depending on the type of encoded information. Each mode allows encoding has its own set of characters and its own rate of compression.

Encoding mode	Valid symbols	Compression
Byte	ASCII 0 to 255	1,2 bytes per word
Text	ASCII 9,10,13 & 32-127	2 characters per word
Numeric	0123456789	2,9 digits per word

The bar-code contains levels of error corrections: even if the bar-code is damaged, it will be read. There are 9 levels of error corrections shown in the table below:.

Level of error correction	Number of codewords
0	2
1	4
2	8
3	16
4	32
5	64
6	128
7	256
8	512

To set the level of correction the **ErrorsCorrectionLevel** property can be used. This property can be set to "Auto", in which case the level will be set automatically.



A "PDF417" barcode.

Datamatrix

The **DataMatrix** barcode was created by the CiMatrix company. Every DataMatrix is composed of two solid adjacent borders in an "L" shape (called the "finder pattern") and two other borders consisting of alternating dark and light "cells" or modules (called the "timing pattern"). Symbol sizes vary from 8×8 to 144×144. The **DataMatrix** is used to mark small products.

Data Matrix symbols are rectangular in shape and usually square, they are made of cells: little elements that represent individual bits.

The bar-code contains error correction codes so the bar-code can be read even if it is partially damaged. There are two main versions of this barcode: the first version is called ECC-000 or ECC-140. The second version is described as ECC-200 version, and uses the Reed-Solomon method for error correction. In Stimulsoft Reports the second version of this bar-code is used.

The bar-code consist of black and white square elements, which are joined into square or rectangular regions. Symbol sizes vary from 8×8 to 144×144. All available combinations of sizes is shown on the table below:

Bar-code size	Length, bites	Bar-code size	Length, bites
10 × 10	3	32 × 32	62
12 × 12	5	36 × 36	86
8 × 18	5	40 × 40	114
14 × 14	8	44 × 44	144
8 × 32	10	48 × 48	174
16 × 16	12	52 × 52	204
12 × 26	16	64 × 64	280
18 × 18	18	72 × 72	368
20 × 20	22	80 × 80	456
12 × 36	22	88 × 88	576
22 × 22	30	96 × 96	696
16 × 36	32	104 × 104	816
24 × 24	36	120 × 120	1050
26 × 26	44	132 × 132	1304
16 × 48	49	144 × 144	1558

The bar-code size can be set using the **MatrixSize** property. If this property is used to specify the specific size of the bar-code, then the barcode will be of that fixed size. If this property is set to **Automatic** (the default), then the minimal size that is necessary to encode the data will be selected from the list. There are 6 types of the bar-code. If it is required to get a square bar-code in the **Automatic** mode, then the **UseRectangularSymbols** property should be set to **false** (the default). If the property is set to true, then square and rectangular forms are used.

There are several modes of data encoding. Which is used depends on the type of the encoded information. Each mode allows encoding their own set of characters and their own rate of compression.

-

Encoding mode	Valid symbols	Bits per symbol
ASCII	ASCII character 0 to 127	8

	ASCII character 128 to 255 ASCII numeric	16 4
C40	Upper-case alphanumeric Lower-case letters and punctuation	5,33 10,66
TEXT	Lower-case alphanumeric Upper-case letters and punctuation	5,33 10,66
X12	ANSI X12	5,33
EDIFACT	ASCII character 32 to 94	6
BASE 256	ASCII character 0 to 255	8

The ASCII is the universal mode of data encoding (the default). It allows encoding any characters, but pairs of digits are compressed the best and the ASCII values (128-255) are compressed the worst.

A "DataMatrix" bar-code.

QR Code

A **QR Code** (QR is the abbreviation for Quick Response) is a two-dimensional code, readable by QR scanners, mobile phones with a camera, and smartphones. It was created by Toyota subsidiary Denso-Wave in 1994.
QR Code is capable of handling all types of data (see a table below):

Numeric mode:	0123456789	Maximum 7089 characters
Alphanumeric mode:	ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789 $ % * + - . / : space	Maximum 4296 characters
Binary mode (8 bits byte data):	JIS 8-bit (Latin and Kana)	Maximum 2953 bytes
Kanji mode:	Shift JIS (8140H-9FFCH and E040H-EBBFH)	Maximum 1817 characters

The **QR Code** characteristics:
The bar-code size (not including quiet zone): Versions 1 to 40 (21*21 modules to 177*177 modules, increasing in steps of 4 modules per side)

Four levels of error correction allowing recovery of:

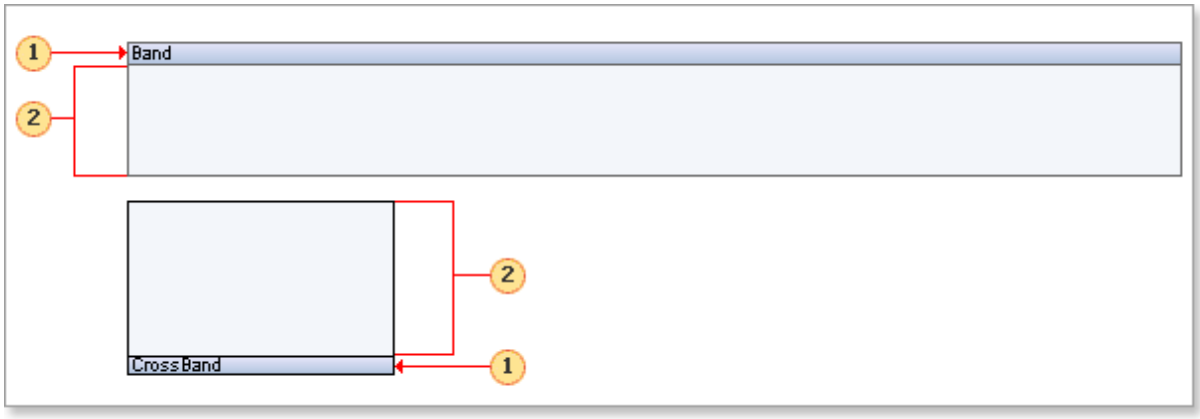
Correction Level	Percentage of the recovered information
L	7%
M	15%
Q	25%
H	30%

The higher the level of correction is, the bigger percentage of information of the corrupted bar-code can be recovered, but fewer information can be encoded in the bar-code of the same size.

A "QR Code" barcode.

Bands

Stimulsoft Reports builds its reports using bands (sometimes bands are called sections in other products). A band consist of two parts: the band header and the working area. On the band header the name of the band is shown, and other information and controls can be displayed. Every band is a container and may contain other components.



1 The band header;
2 The band working area.

Bands do not appear in the rendered report, only the calculated content of the bands is displayed. The properties of the band control only control its position within the rendered report.

Usually a report will consist of many bands with text and images on them. When a report is rendered, bands are copied as many times as necessary to complete the report. For example, the Header band is output once before data, then the Data band is output once for each record.

Band Types

There are many bands in Stimulsoft Reports. Each type of band has its own unique capabilities. All bands fall into one of two categories: standard bands or cross bands.

Standard Bands

Standard bands are rendered top-down. They are usually placed directly on a page. Also they can be placed on a panel.

Cross Bands

Cross-bands are rendered from left to right. Usually they are placed on standard bands.

There is one special category of band, the Child Band, which whilst it is a standard band is typically used to extend a Data band.

Standard Bands

Standard bands are the basic elements of any report. The table below shows all the standard bands.

Icon	Band Name	Description
	ReportTitle	This band is printed in the beginning of a report
	ReportSummary	This band is printed in the end of a report
	PageHeader	This band is printed on the top of each page
	PageFooter	This band is printed on the bottom of each page
	GroupHeader	This band is printed in the beginning of a group
	GroupFooter	This band is printed in the end of a group
	Header	This band is printed before data
	Footer	This band is printed after data
	ColumnHeader	This band is printed before a column is output
	ColumnFooter	This band is printed after a column is output
	Data	This band is printed as many times as there are rows in the data source
	Hierarchical Data	This band is printed as many times as there are rows in the data source. Data items are output as a tree
	Child	This band is printed only once, after the band beneath which it is placed
	Empty Data	Fills the free space at the bottom of a page
	Overlay	This band is printed on the background of a page. It does not effect on other bands.

To make the structure of reports easier to understand and to make a report template look clearer each type of band has its own color:

Cross-bands

Cross bands must be placed on a simple band, so they cannot be placed directly on a page or a container. They are used to permit the rendering if complicated cross-reports.

Important! Cross-bands can be placed only on simple bands. It is impossible to put them on a page or container.

The list below shows types of cross-bands:

Icon	Name	Description
	CrossGroupHeader	This band is printed in the beginning of a group
	CrossGroupFooter	This band is printed in the end of a group
	CrossHeader	This band is printed before data
	CrossFooter	This band is printed after data
	CrossData	This band is printed as many times as there are rows in the data source

Unlike simple bands, the cross-bands header is displayed at the bottom of a band.

Output Order of Bands

When rendering a report there is a definite order in which the bands are generated because every band has specific functionality. For example, for a table output you should use three bands: Header, Data, Footer. The Header band is used to place column headers, the Data values are placed on the Data band, and the totals are placed on the Footer band.

The following table describes the bands and their output order:

Order	Name	Description
1	PageHeaderBand	On each page. Output on the first page is optional.
2	ReportT	Once at the beginning of a report. The ReportTitleBand can be output before the PageHeaderBand if the

	itleBand	TitleBeforeHeader property of the page on which both bands are placed is set to true.
3	Header Band, Column Header	Once before data output (for the ColumnHeader - once for every column). Output on each new page is optional.
4	GroupHeaderBand	At the beginning of each group. Output on each new page is optional.
5	DataBand	Once for every row of data.
6	GroupFooterBand	At the end of each group.
7	FooterBand, Column Footer	After all data has been output (for the ColumnFooter - once for every column). Output on each new page is optional.
8	ReportSummary Band	Once at the end of a report.
9	PageFooterBand	On every page. Output on first page is optional.

The Child band is not specified in the preceding table. This band is always printed immediately after the band it follows when placed on a page.

Note: Components placed directly on the page (i.e. not on any band) are printed first, followed by the bands.

In a report it is possible to use as many bands of one type as you wish. For example, you can use two Header bands.

If there is more than one band of one type then they will be output in the order of their position on a page. In other words the band furthest up the page will be printed first followed by the other bands of the same type. This is also true with cross-bands except that the band on the left will be printed first.

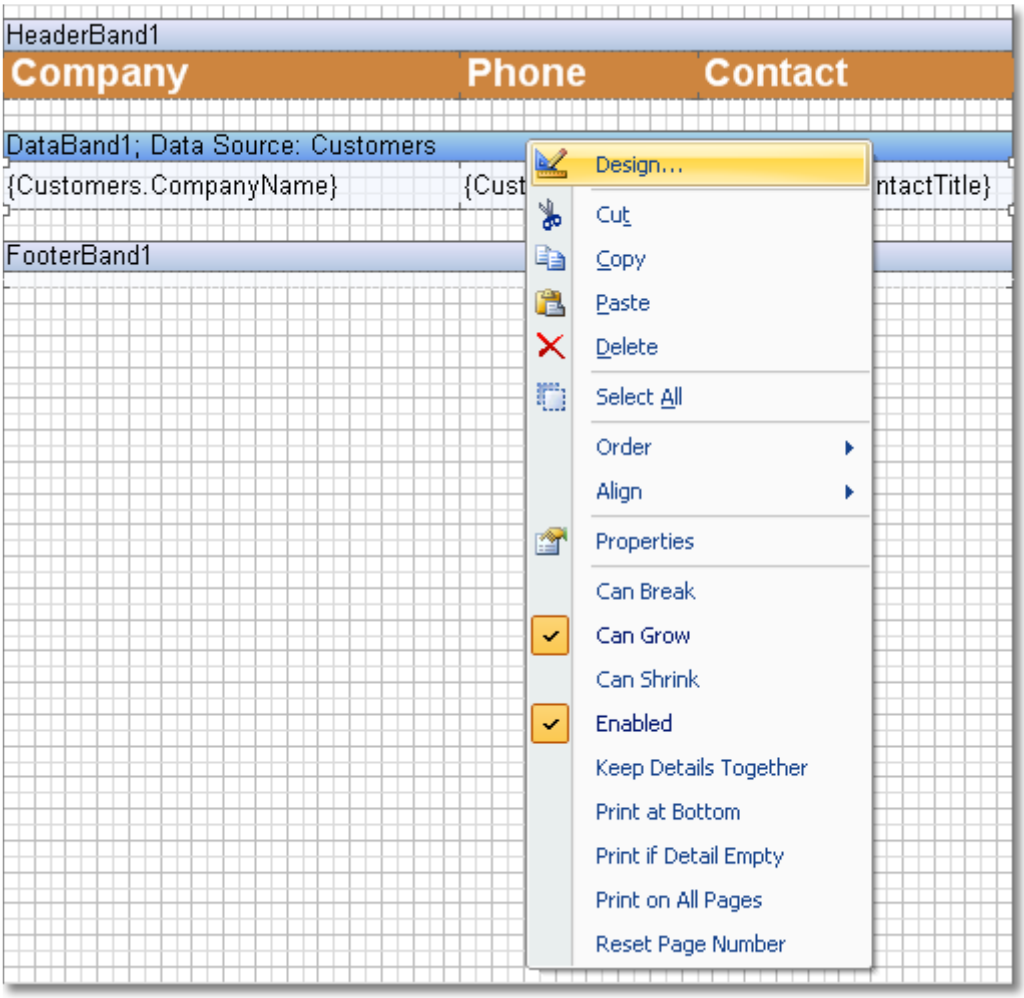
The order in which bands of the same type are output can be changed by simply moving the bands on the page. You can drag one of the bands with the mouse or you can use the Move Forward and Move Backward commands from the context (right click) menu or you can use the buttons on the Layout toolbar to change the band order.

Creating Lists

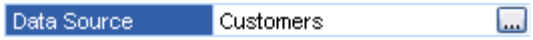
Lists in a report can be output using three bands: **Header**, **Footer**, and **Data**. Data are output using these bands. The basic band is the **Data** band. A data source is specified to each **Data** band. The data source is a table. Each data source has data fields. It is possible to output a table by placing text components with references to these fields. One data source can specify previously unknown number of rows with data. The **Data** band is output as many times as there are rows in the specified data source. For example, if there are 100 rows in the data source, then the **Data** bad will be output 100 times. If it is not enough space on one page, the second page will be generated and printing will be continued. Using the **Header** band, headers will be added to the table that is output using the **Data** band. Correspondingly, the **Footer** band is used to output different totals by the output table.

Data Source of Data Band

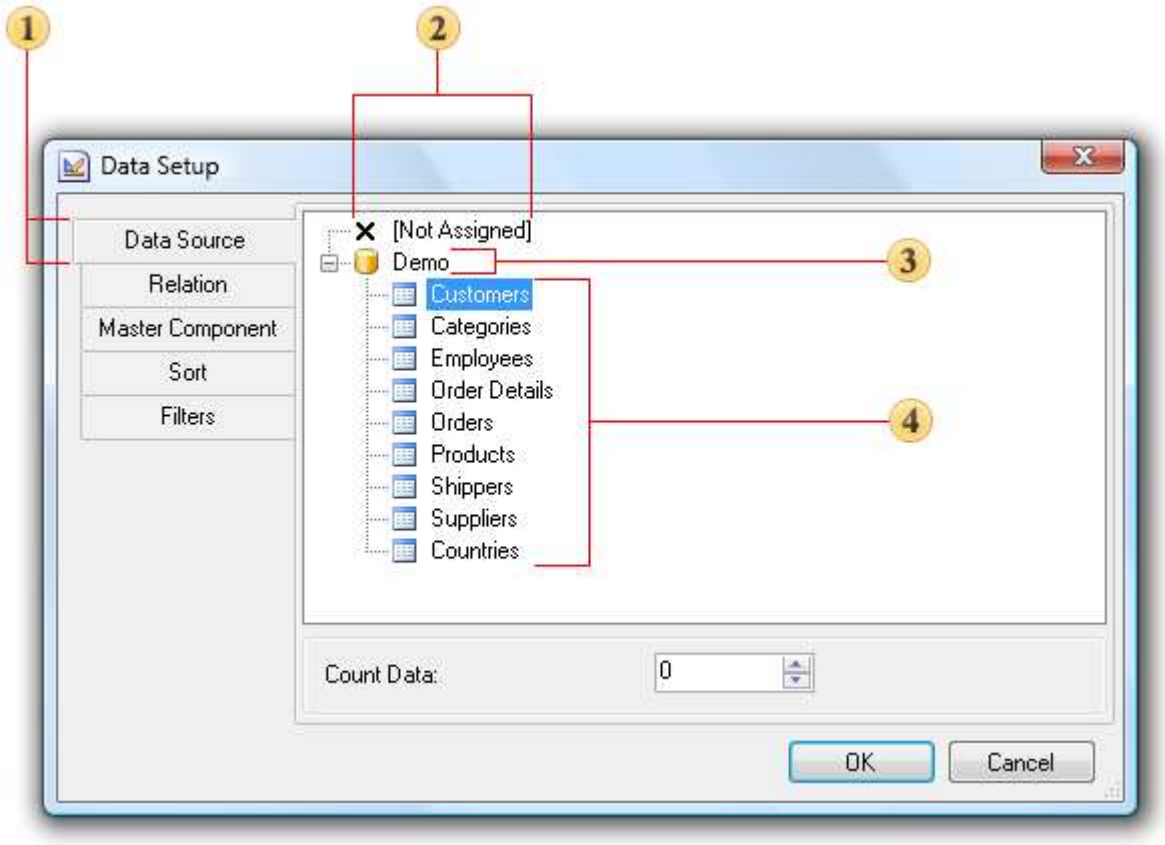
It is necessary to specify what data source will be used when you output lists in the **Data** band. It is important because report generator should know how many times the **Data** band must be printed. Therefore, the reference to the **Data** band is specified. This can be done with several ways. First, it is possible to use the **Data** band editor. To call the editor it is enough double-click on the **Data** band. Also it is possible to call the editor from the context menu. See below an example of this menu.



Also the editor can be called using the **DataSource** property of the **Data** band.

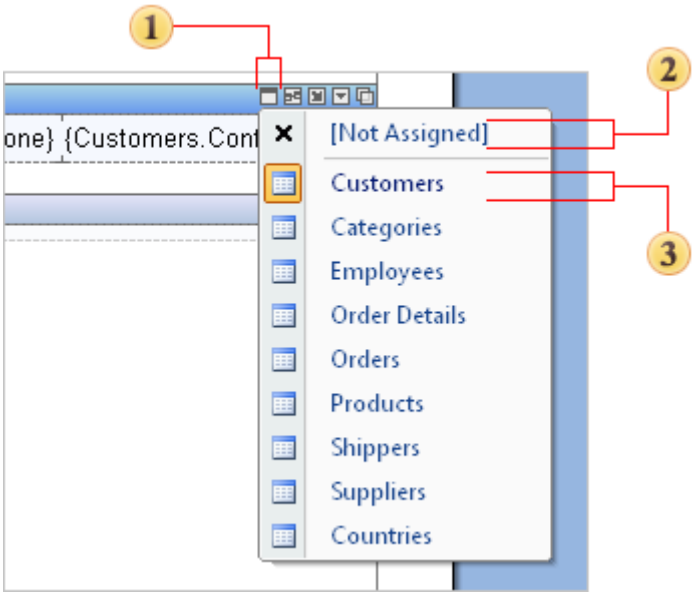


Data band editor allows quickly selecting data source. Data source is selected on the first bookmark of the **Data** band editor. All data sources are grouped in categories. Each category is one data connection with data in the Dictionary of Data. The picture below shows data in the **Data** band editor.



1. Select data source bookmark of the **Data** band.
2. Select this node if there is no need to specify any data source.
3. The "Demo" category of data.
4. The "Demo" category data source .

Second, it is possible to use quick button on the **Data** band and select data source from menu. Basic elements of menu are represented on the picture below.

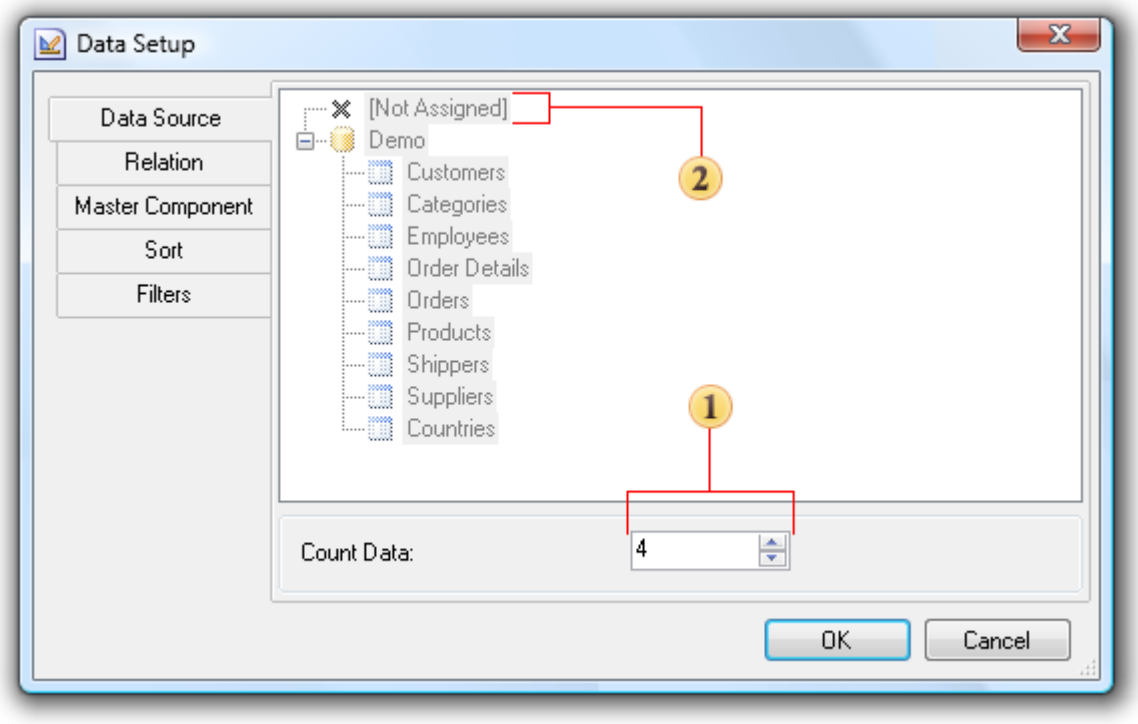


1. Quick button the select data source.
2. This menu item is used to reset data source selection.
3. The **Customers** data source is selected.

Virtual Data Band

Sometimes it is necessary to print a **Data** band several times without specifying a data source. The **CountData** property is used for this purpose.

It is possible to specify number of elements in the **Data** band editor. On the picture below the **Data** editor is shown.



1 The field in what number of elements for the **Data** band can be specified.
2 A data source is not specified.

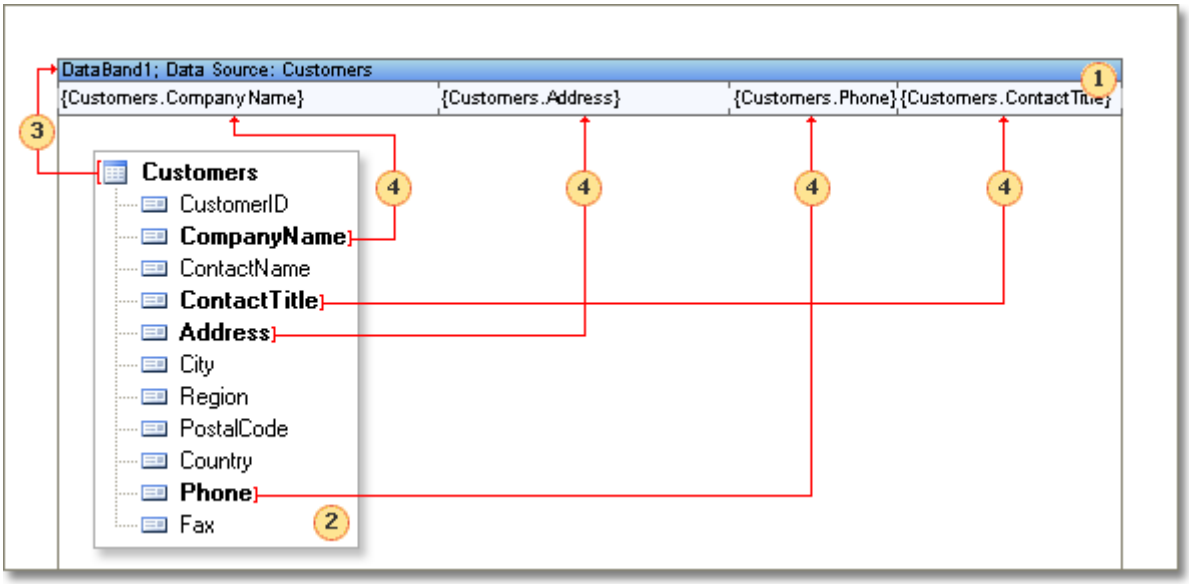
By default the **CountData** property is 0. But if to set it to 4, then the **Data** band will be printed 4 times. This can be used to print empty columns. It is important to remember that in this case data source is not specified.

List Output

Render a report that prints a list. Put one **Data** band on a page. Using the **DataSource** property assign a data source to the band. Put **Text** components on the band. Make a reference to data fields in each component. For example:

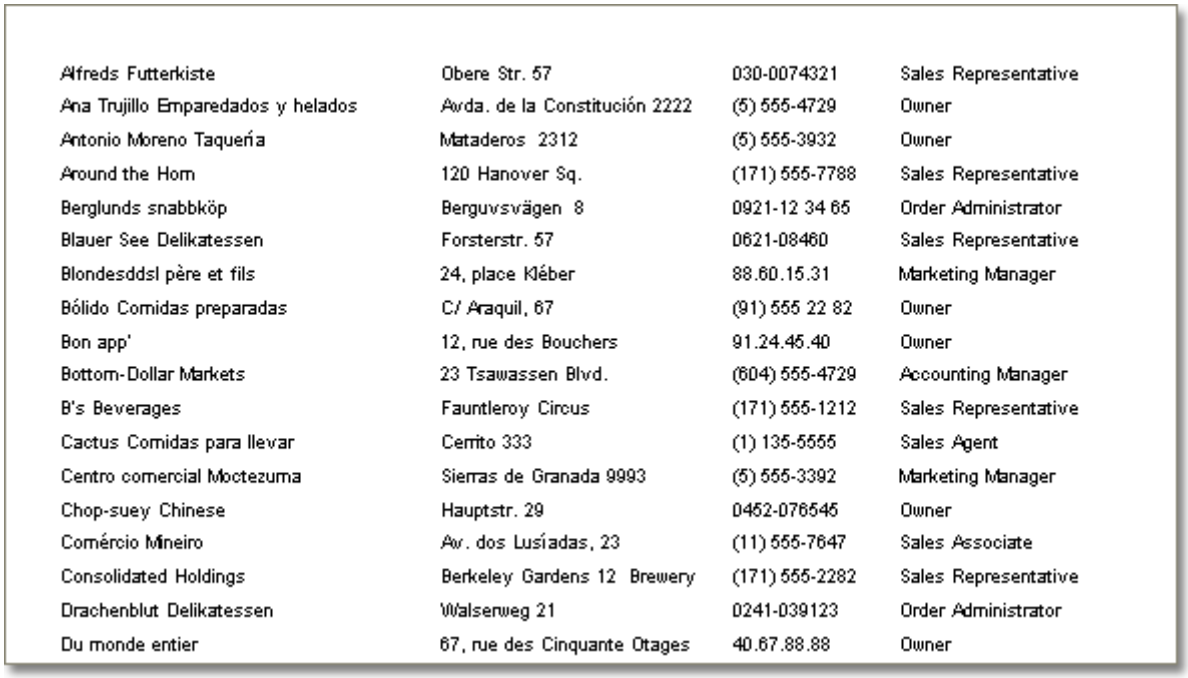
```
{Customers.CompanyName}
```

The report template will have the following view.



1 **Data** band that outputs a table.
2 The data source that is used to get data rows.
3 Reference to the data source. It is necessary to specify data source to the **Data** band.
4 Reference to the data source. **Text** components are placed on the **Data** band. References to data sources fields are created. When rendering, all references will be changed on data.

After report rendering all references to data fields will be changed with data from specified fields. Data will be taken from the data source, that was specified for this band. Number of copies of the **Data** band in the rendered report will be equal to the number of rows in the data source. As a result, all fields were output as a list. The picture below shows a rendered report.



Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative
Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative
Blondesddsl père et fils	24, place Kléber	88.60.15.31	Marketing Manager
Bólido Comidas preparadas	C/ Araquil, 67	(91) 555 22 82	Owner
Bon app'	12, rue des Bouchers	91.24.45.40	Owner
Bottom-Dollar Markets	23 Tsawassen Blvd.	(604) 555-4729	Accounting Manager
B's Beverages	Fauntleroy Circus	(171) 555-1212	Sales Representative
Cactus Comidas para llevar	Cerrito 333	(1) 135-5555	Sales Agent
Centro comercial Moctezuma	Sierras de Granada 9993	(5) 555-3392	Marketing Manager
Chop-suey Chinese	Hauptstr. 29	0452-076545	Owner
Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647	Sales Associate
Consolidated Holdings	Berkeley Gardens 12 Brewery	(171) 555-2282	Sales Representative
Drachenblut Delikatessen	Walserweg 21	0241-039123	Order Administrator
Du monde entier	67, rue des Cinquante Otages	40.67.88.88	Owner

If all lists cannot be placed on one page, then the report generator will add additional pages.

List with Header

Usually, a name of a column is output over each column. To output data name or other information before data the special **Header** band is used. It is placed on a page before the **Data** band. There should not be any headers between the **Data** band and the **Header** band. On the picture below a sample of a report with one **Header** band and one **Data** band is shown.

Create a new report. Put a data band on a page. Add the **Header** band to a report. Put text components on a band. Specify data name, which are output on the **Data** band, in these text components. Increase the font size, make it bold. Change the text components background on the **Header** band. Render a report. The picture below shows the result of report rendering.

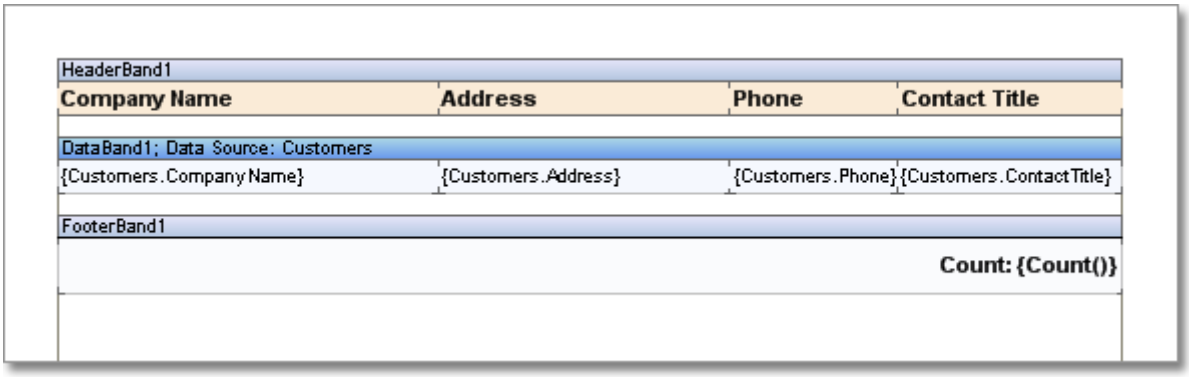
Company Name	Address	Phone	Contact Title
Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative
Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative
Blondesddsl père et fils	24, place Kléber	88.60.15.31	Marketing Manager
Bólido Comidas preparadas	C/ Araquil, 67	(91) 555 22 82	Owner
Bon app'	12, rue des Bouchers	91.24.45.40	Owner
Bottom-Dollar Markets	23 Tsawassen Blvd.	(604) 555-4729	Accounting Manager
B's Beverages	Fauntleroy Circus	(171) 555-1212	Sales Representative
Cactus Comidas para llevar	Cerrito 333	(1) 135-5555	Sales Agent
Centro comercial Moctezuma	Sierras de Granada 9993	(5) 555-3392	Marketing Manager
Chop-suey Chinese	Hauptstr. 29	0452-076545	Owner
Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647	Sales Associate
Consolidated Holdings	Berkeley Gardens 12 Brewery	(171) 555-2282	Sales Representative
Drachenblut Delikatessen	Walserweg 21	0241-039123	Order Administrator
Du monde entier	67, rue des Cinquante Otages	40.67.88.88	Owner

When report rendering for one **Data** band, it is possible to create more than one **Header** band. For example, one **Header** band can be output only in the beginning of data. And the second one can be output in the beginning of data and on other pages of a report. **Header** bands are output in the same order as they are placed on a page.

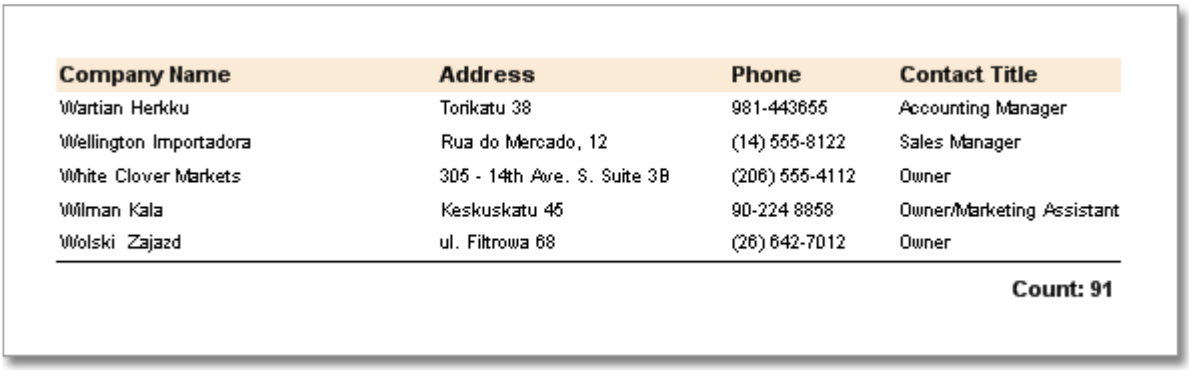
Notice. For one **Data** band unlimited number of **Header** bands can be created.

List with Footer

Besides **Data** bands and **Headers** bands, **Footer** bands can be used. These bands are used to output total of data. The **Footer** band is placed after data are output. Different information is output in the band. For example, totals of a list, data, additional information. On the picture below a report template with the **Footer** band is shown.



As a result of report rendering with the **Footer** band, the report generator will output total after all data will be output. For example:



Company Name	Address	Phone	Contact Title
Wartian Herkku	Torikatu 38	981-443655	Accounting Manager
Wellington Importadora	Rua do Mercado, 12	(14) 555-8122	Sales Manager
White Clover Markets	305 - 14th Ave. S. Suite 3B	(206) 555-4112	Owner
Wilman Kala	Keskuskatu 45	90-224 8858	Owner/Marketing Assistant
Wolski Zajazd	ul. Filtrowa 68	(26) 642-7012	Owner

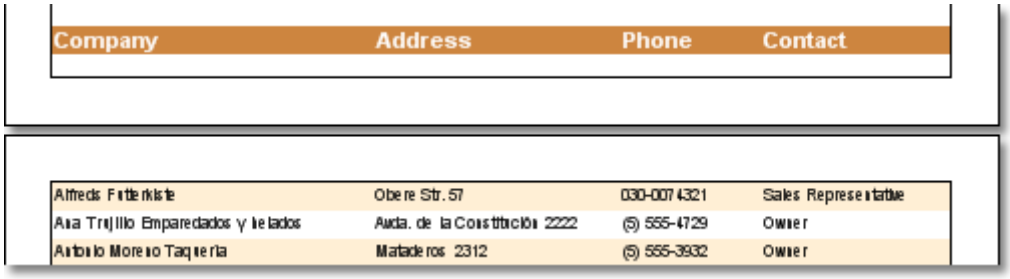
Count: 91

The **Data** band may have unlimited number of bands. Bands of totals will be output in the same order as they are placed on a page.

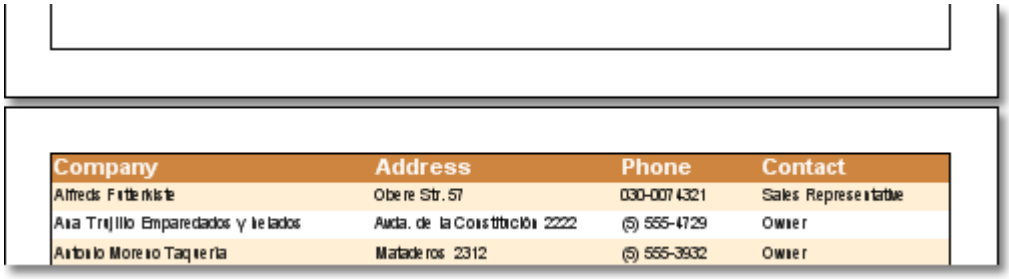
Notice. For one **Data** band unlimited number of **Footer** bands can be created.

KeepHeaderTogether Property

Sometimes, when printing lists, a header will be printed on one page, and the first row of data on another. To escape this visual gap of data the **KeepHeaderTogether** property of the **Header** band can be used. If the property is **true**, then headers will be printed together with data. In other words as minimum one row with data will be output. If there is no enough free space for a header with data row, then they will be carried over on the next page. See a sample of a rendered report with the **KeepHeaderTogether** property set to **false**.



As the same report with keeping header together with the first data row.

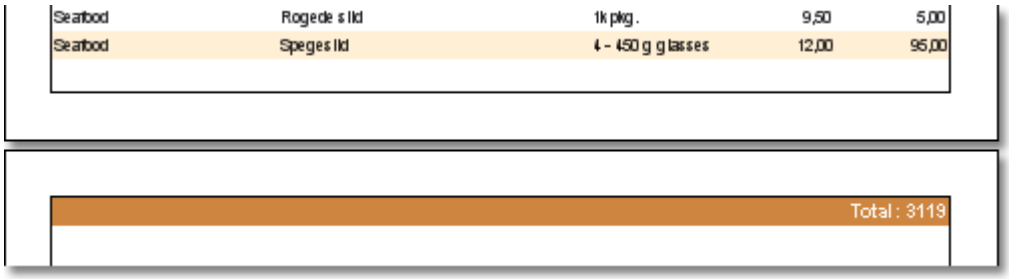


Company	Address	Phone	Contact
Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner

By default, the **KeepHeaderTogether** property is set to **true.** So headers will be kept together with the first row of data.

KeepFooterTogether Property

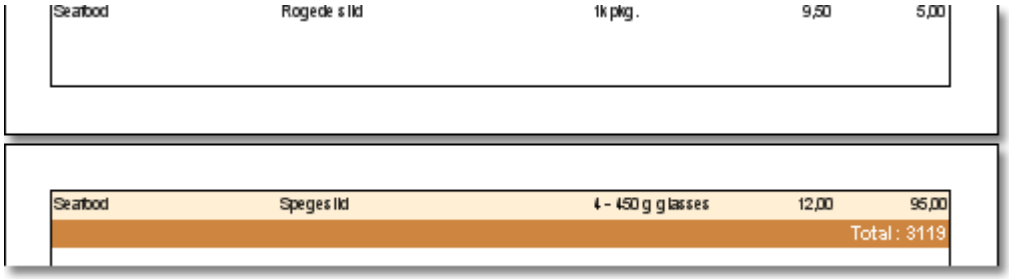
The **KeepFooterTogether** property is used to print a list so that to output data row together with totals of data. If the property is **true,** then totals will be printed with the last row of data. If total cannot be placed after the last page printing, then it is output on the current page. If there is no enough free space to output totals, then it is carried over on the next page. On picture below a sample of a report with the **KeepFooterTogether** property set to **false** is shown.



Seafood	Rogede sild	1kg pkg.	9,50	5,00
Seafood	Spegesild	4 - 450 g glasses	12,00	95,00

Total : 3119

And the same report with keeping footer together with the last row of data.



Seafood	Rogede sild	1kg pkg.	9,50	5,00
Seafood	Spegesild	4 - 450 g glasses	12,00	95,00

Total : 3119

By default, the **KeepFooterTogether** property is set to **true**, so totals of data will be kept together with last row of data.

Enumeration in Lists

Sometimes it is necessary to number lists. It is more convenient to work with an enumerated list. On the picture below an enumerated list is shown.

1.Chai	10 boxes x 20 bags	39,00
2.Chang	24 - 12 oz bottles	17,00
3.Chartreuse verte	750 cc per bottle	69,00
4.Côte de Blaye	12 - 75 cl bottles	17,00
5.Guaraná Fantástica	12 - 355 ml cans	20,00
6.Ipoh Coffee	16 - 500 g tins	17,00
7.Lakkalikööri	500 ml	57,00
8.Laughing Lumberjack Lager	24 - 12 oz bottles	52,00
9.Outback Lager	24 - 355 ml bottles	15,00
10.Rhönbräu Klosterbier	24 - 0.5 l bottles	125,00
11.Sasquatch Ale	24 - 12 oz bottles	111,00
12.Steeleye Stout	24 - 12 oz bottles	20,00

To add a number of a row into an expression it is possible to use the **Line** system variable. For example, the following expression can be used to get the result as is shown on the picture above:

{**Line**}.{Products.ProductName}

The **Line** system variable returns the number of the current row. Numeration starts with 1. In other words the system variable returns 1 for the first row, 2 for the second one and etc. This system variable has the **Int64** type. The **Line** system variable may also be used in arithmetic expressions. If you need to start numeration from 0, it is necessary to use the following expression:

{**Line - 1**}.{Products.ProductName}

In addition to the **Line**, **LineABC** and **LineRoman** system variables can also be used for the list enumeration. The **LineABC** system variable returns the alphabetical index instead of a number of a row. The **LineRoman** system variable returns Roman numerals of a number of a row. For example, a report where the **LineABC** system variable is used is shown on the picture below:

A.Chai	10 boxes x 20 bags	39,00
B.Chang	24 - 12 oz bottles	17,00
C.Chartreuse verte	750 cc per bottle	69,00
D.Côte de Blaye	12 - 75 cl bottles	17,00
E.Guaraná Fantástica	12 - 355 ml cans	20,00
F.Ipoh Coffee	16 - 500 g tins	17,00
G.Lakkalikööri	500 ml	57,00
H.Laughing Lumberjack Lager	24 - 12 oz bottles	52,00
I.Outback Lager	24 - 355 ml bottles	15,00
J.Rhönbräu Klosterbier	24 - 0.5 l bottles	125,00
K.Sasquatch Ale	24 - 12 oz bottles	111,00
L.Steeleye Stout	24 - 12 oz bottles	20,00

A report where the **LineRoman** system variable is used is shown on the picture below:

I.Chai	10 boxes x 20 bags	39,00
II.Chang	24 - 12 oz bottles	17,00
III.Chartreuse verte	750 cc per bottle	69,00
IV.Côte de Blaye	12 - 75 cl bottles	17,00
V.Guaraná Fantástica	12 - 355 ml cans	20,00
VI.Ipoh Coffee	16 - 500 g tins	17,00
VII.Lakkalikööri	500 ml	57,00
VIII.Laughing Lumberjack Lager	24 - 12 oz bottles	52,00
IX.Outback Lager	24 - 355 ml bottles	15,00
X.Rhönbräu Klosterbier	24 - 0.5 l bottles	125,00
XI.Sasquatch Ale	24 - 12 oz bottles	111,00
XII.Steeleye Stout	24 - 12 oz bottles	20,00

LineABC and **LineRoman** system variables, unlike the **Line** system variable, return numbers as strings. For example, to enumerate a list with letters in the lower case, it is possible to use the following expression:

```
{Line.ToLower()}.{Products.ProductName}
```

Selecting Rows One After Another

To make a report look better and for much convenient work with rows it is recommended to alternate rows filled with different colors. This will make your report look professional. There are two ways in the report generator to make such filling: 1. using highlight conditions; 2. using special properties of the **Data** band styles.
The first way - using the **Data** band highlight condition. Open a report that has a list. An example of such a report is shown on the picture below.

Simple List

Company	Address	Phone	Contact
Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative
Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative

All rows have the same background color. Add highlight condition to the Data band. The **Conditions** property of the band is used for this. Add a new condition in the editor, change background color on another color to fill odd rows, change text color (it is red by default) and set the highlight condition. The **Line** system variable is used to specify whether this row is odd or even. For example:

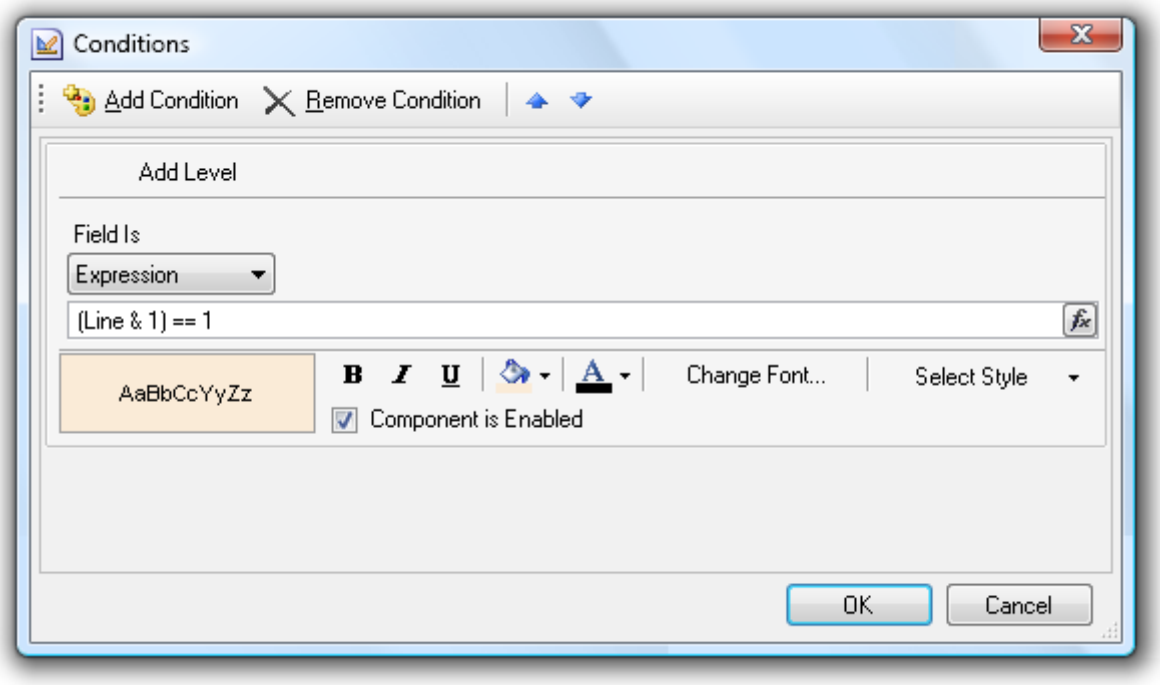
C#:

```
(Line & 1) == 1
```

VB.Net

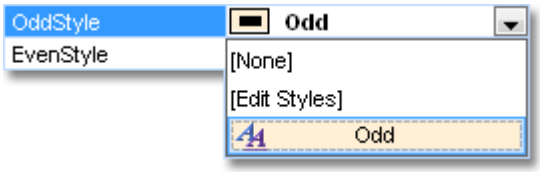
```
(Line And 1) = 1
```

In other words for odd rows this condition is true. On the picture below the Conditions editor is shown.



After adding a condition to the data band a report will look as it shown on the picture below.

The second way - using properties of styles. The **Data** band has two special properties - **OddStyle** and **EvenStyle**. To add highlight condition to rows it is enough to specify a style in one of these properties. For example, the collection of styles has **OddStyle**. Select this style in the **OddStyle** property.



The report looks the same as the one where the first way was used.

Events and Data Band

Except standard event for all components the **Data** band has three special events: **BeginRenderEvent**, **EndRenderEvent** and **RenderingEvent**. The **Data** band must be created for each data row of the specified data source. For example, if there are 10 rows in the data source, then the **Data** band will be created 10 times. The **BeginRenderEvent** event is called before the data is rendered. In other words when data rows are not output. The event can be used for initialization some data ans variables, calling some actions. The **EndRenderEvent** event is called after the **Data** band is rendered, when all data rows will be output. In this event data processing, totals calculation processing is done. **RenderingEvent** event is called when rendering one data row. The event is called before the **Data** band is printed. If these are 10 data rows, then the **RenderingEvent** event will be output 10 times.

Calculate number of elements in the data source. Write the following code in the **BeginRenderEvent**:

```
myvariable = 0;
```

Also it is necessary to create the **myvariable** variable in the data dictionary. Write the following code in the **RenderingEvent**:

```
myvariable = myvariable + 1;
```

And the **EndRenderEvent** is not used in this case. As a result of calculation the **myvariable** will store the value that equal to number of elements in the data source. To output this value in the **Text** component the following expression will be used:

```
{myvariable}
```

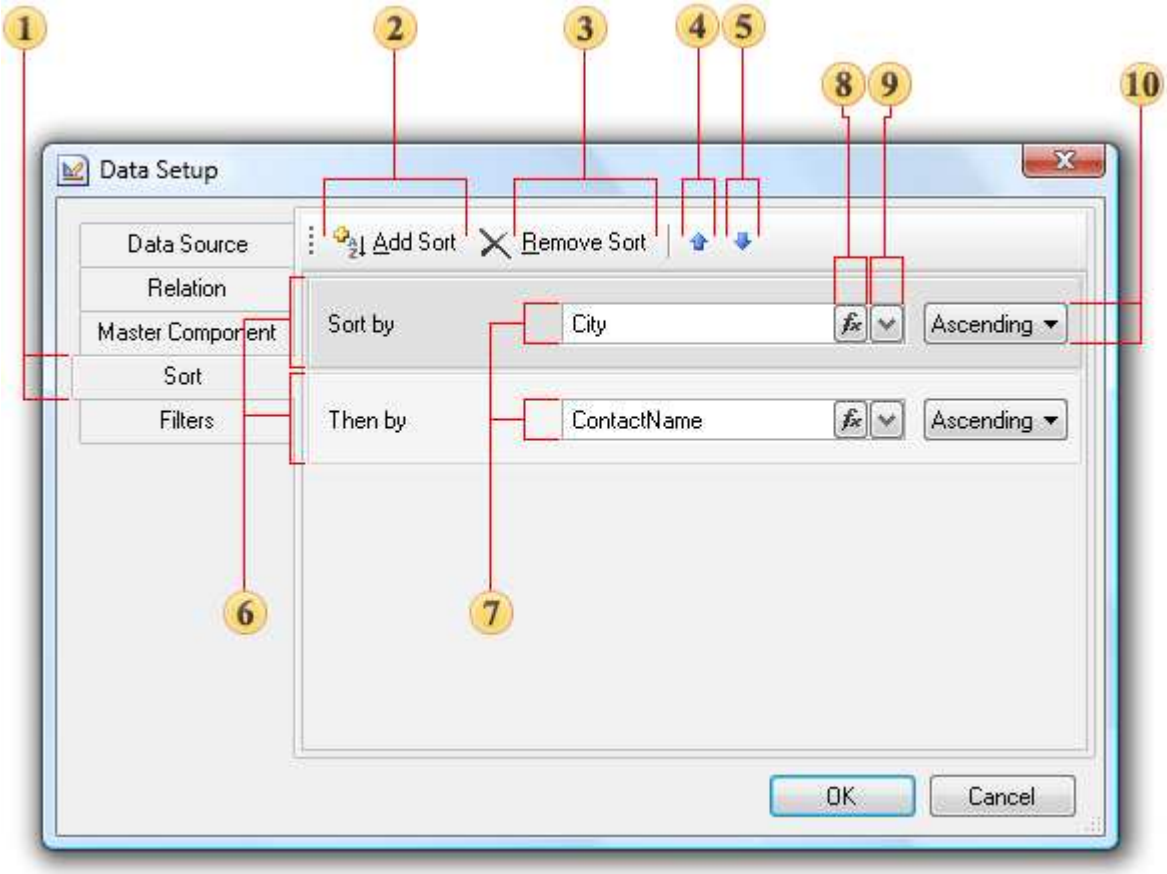
Also it is necessary to set the **ProcessAtEnd** property of the **Text** component to **true**. It is necessary to output calculated value in the **myvariable.**

Data Sorting

Frequently data, which are used for the report rendering, are sorted in order that does not to meet your requirements. In this case, it is possible to sort data using by abilities of Stimulsoft Reports. Sorting can be set for each **Data** band separately. To set sorting it is necessary to use the **Sort** property of the **Data** band. Using this property it is possible to call the editor of the **Data** band.

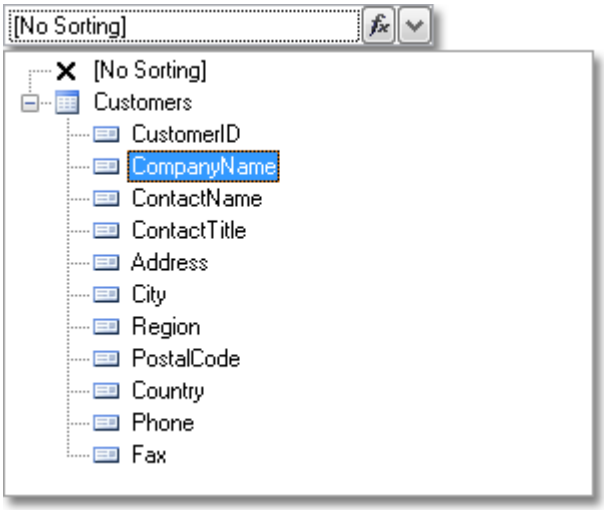
Sort	**[No Sorting]**

Also it is possible to call the editor by double-click on the band. The **Sort** bookmark is responsible for sorting in the band editor. The picture below shows structure of the bookmark of sorting.



1 The Sort bookmark;
2 The button to add a new level of sorting;
3 The button to remove the selected level of sorting;
4 Move the selected level of sorting upwards;
5 Move the selected level of sorting downwards;
6 Level of sorting;
7 The column or expression which are used for sorting;
8 The button to add or edit expressions of the sorting level;
9 The button the select a column for sorting;
10 Direction of sorting.

Each sorting consist of several levels. For example, the first list can be sorted by one column, then by the second column, then by the third column. On the picture above bookmark sorting, sorting levels are marked with figure 6. Number of levels of sorting is unlimited. Each level of sorting has the sort order. It is possible to sort in ascending order and in descending order. By default, sorting is set in ascending order. In addition to the sort order in each level of sorting the column (figure 9 on the picture above) is set or expression (figure 8 on the picture above) is set, which is used to obtain the values by which sorting will be done.

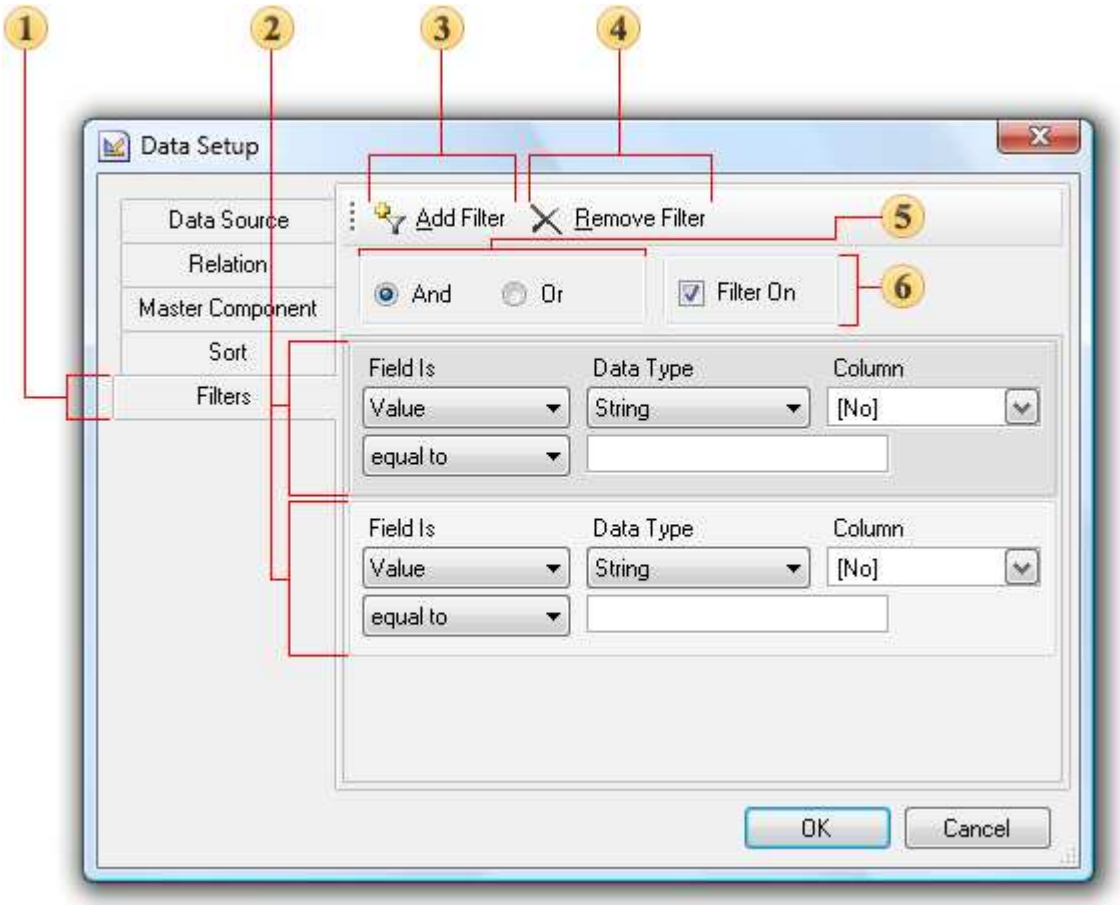


Data Filtration

When rendering a report, sometimes it is necessary to print rows of the data source which correspond to the definite condition. To select the necessary rows the data filtration is used. Data filtration is set using the **Filters** property of the **Data** band. In addition to the **Filters** property the **FilterOn** property can also be used. This property controls filter activity.

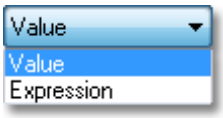
Filter On	True
Filters	**[No Filters]**

How does the filter work? In each filter the condition is set. If the condition is set to **true**, this means that the result of its calculation is **true**, then this data row will be output. If the result of calculation is set to **false**, then this row will be ignored. Each band may contain more than one filter. For example it is necessary to check one of columns of the data source on the equality to the string constant and simultaneously the value of this column should start with the definite character. The filtration is setup in the window of the **Data** band setup (the Filters bookmark). On the picture below such a window is shown.

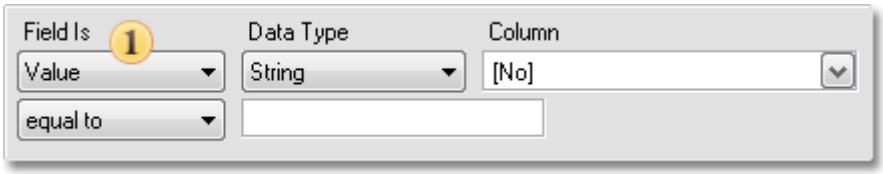


1 The **Filters** bookmark;
2 Filter panels. Each **Data** band may contain one or more filters;
3 The button to select a new filter;
4 The button to delete the selected filter;
5 The type of logical operation, according to what filters will be formed. This field is available if the **Data** band contains more than one filter. There are two options: a logical **And** and logical **Or**. If you select the logical **And**, then data row will be output, if all filters are set to **true**. If you select the logical **Or**, then the data row will be output, if at least one of the filters is set to **true**;
6 The **Filter On** flag is used to enable/disable filters of the data band.

Each filter is a condition for data row processing. There are two ways set a condition:
1. **Value.** The condition is set using the wizard;
2. **Expression.** The condition is set as an expression.

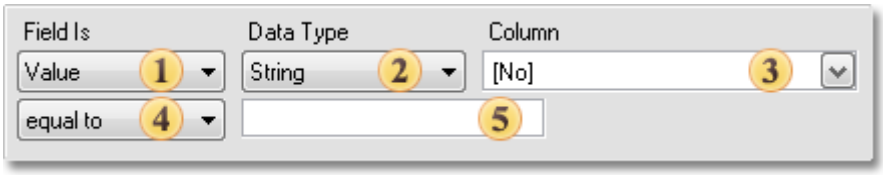


On the picture below, the figure 1 is the field in what the way of calculating condition is indicated.

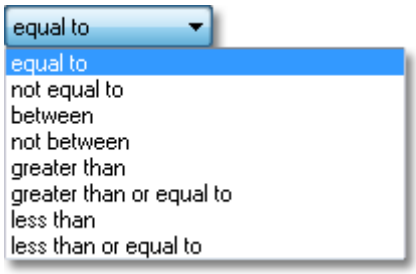


How to set a condition using the wizard

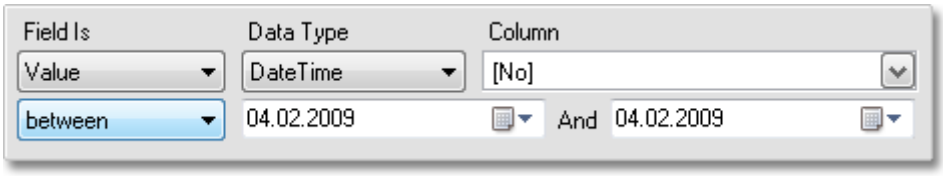
On the picture below the panel of setting a condition using the wizard is shown.



1 The way of selecting a condition;
2 This field specifies the type of data with what the condition will work. There are five types of data: **String**, **Numeric**, **DateTime**, **Boolean**, **Expression**. Data type has affect on how the reporting tool processes a condition. For example, if the data type is a string, then the method of work with strings is used. In addition, depending on the data type the list of available operations of conditions is changed. For example, only for the **String** data type is **Containing** operation is available;
3 The column of the data source is specified in the field. The value from this column will be used as the first value of a condition;
4 The type of operation, using what the calculation of the value of a condition is done. All available types of operation are grouped in the table and shown on the picture below;



5 The second value of a condition of a filter. It is required to specify two values for some operations. For example, for the **between** operation it is required to specify two values.



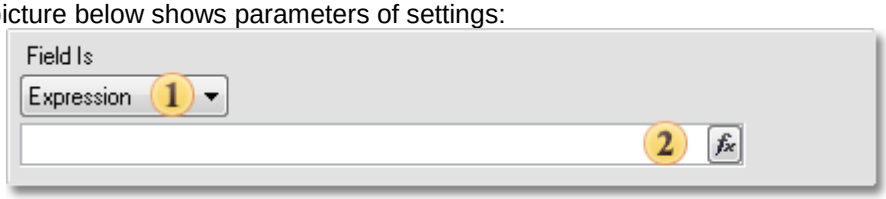
The table below shows operations and their description for each data type.

Name of operation	Types of data					Description
	String	Numeric	Date	Logic	Expression	
equal to	✓	✓	✓	✓	✓	If the first value is equal to the second value, then the condition is true.
not equal to	✓	✓	✓	✓	✓	If the first value is not not equal to the second value, then the condition is true.
between		✓	✓		✓	If the first value is in the range, then the condition is true.
not between		✓	✓		✓	If the first value is not in the range, then the condition is true.
greater than		✓	✓		✓	If the first value is greater than the second value, then the condition is true.

Name of	Types of data					Description
greater than or equal to		✓	✓		✓	If the first value greater than or equal to the second value, then the condition is true.
less than		✓	✓		✓	If the first value is less than the second value, then the condition is true.
less then or equal to		✓	✓		✓	If the first value is less then or equal to the second value, then the condition is true.
containing	✓					If the first value contains the second value, then the condition is true. This operation can be applied only to strings.
not containing	✓					If the first value does not contain the second value, then the condition is true. This operation can be applied only to strings.
beginning with	✓					If the first value begins with the second value, then the condition is true. This operation can be applied only to strings.
ending with	✓					If the first value ends with the second value, then the condition is true. This operation can be applied only to strings.

How to set a condition using as an expression

When using the **Expression** type of a condition, the condition is set as a text expression, that should return the boolean value. The picture below shows parameters of settings:



1 The way to select an expression;
2 The expression is specified in this field. It should return the boolean value. For example, the expression in C#:

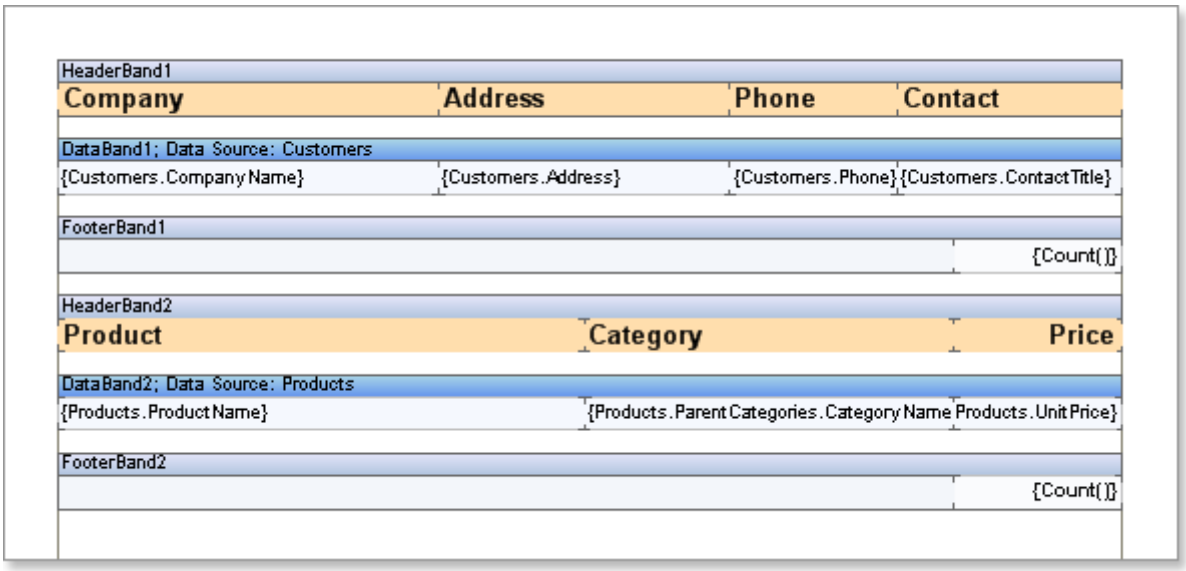
```
Customers.ID == 53447
```

If the expression will return the value of not a boolean type, then the reporting tool will not be able to render an expression of this type.

Lists One After Another

Often it is necessary to output some lists one after another in a report. Stimulsoft Reports has no restrictions on it. All you have to do to render such a report is to place two **Data** bands with headers and footers bands. For example.

Put two **Data** bands on a page, specify them with different data sources. In addition create a header and a footer for the **Data** band. For this, place two **Header** bands and two **Footer** bands. How do you know which header and footer bands belong to the **Data** band? It's very simple. The **Header** band should be placed over the **Data** band. The **Footer** band should be placed under the **Data** band. Thus, the **Header** band or the **Footer** band are considered to belong to this **Data** band, if there are no other **Data** bands between them. For example, two bands of each type are placed on a page. The **HeaderBand1** band is placed over the **DataBand1** and there are no other **Data** bands between them. So it belongs to the **DataBand1**. But if to take the **DataBand2**, then between this band and the **HeaderBand1** band the **DataBand1** is placed. Therefore, the **HeaderBand1** does not belong to the **DataBand2**. The **FooterBand1** is placed under the **DataBand1** band and there are no other **Data** bands between them. So it belongs to the **DataBand1**. But the **FooterBand2** band is placed under the **DataBand1**, and the **DataBand2**. But there is the **DataBand2** in placed between the **DataBand1** and the **FooterBand2**. Therefore, the **FooterBand2** belong the the **DataBand2**. Here is an example of a report template, which outputs several lists one after another.



The first **Data** band will output the first list. When the list will be output the second list will be output. The second band will output on the second list. The number of lists is unlimited. The picture below shows the sample of how to output a report with with two lists.

Company	Address	Phone	Contact
The Cracker Box	55 Grizzly Peak Rd.	(406) 555-5834	Marketing Assistant
Toms Spezialitäten	Luisenstr. 48	0251-031259	Marketing Manager
Tortuga Restaurante	Avda. Azteca 123	(5) 555-2933	Owner
Tradição Hipermercados	Av. Inês de Castro, 414	(11) 555-2167	Sales Representative
Trail's Head Gourmet Provisioners	722 DaVinci Blvd.	(206) 555-8257	Sales Associate
Vaffeljernet	Smagsloget 45	86 21 32 43	Sales Manager
Victuailles en stock	2, rue du Commerce	78.32.54.86	Sales Agent
Vins et alcools Chevalier	59 rue de l'Abbaye	26.47.15.10	Accounting Manager
Wartian Herkku	Torikatu 38	981-443655	Accounting Manager
Wellington Importadora	Rua do Mercado, 12	(14) 555-8122	Sales Manager
White Clover Markets	305 - 14th Ave. S. Suite 3B	(206) 555-4112	Owner
Wilman Kala	Keskuskatu 45	90-224 8858	Owner/Marketing Assistant
Wolski Zajazd	ul. Filtrowa 68	(26) 642-7012	Owner
			91

Product	Category	Price
Alice Mutton	Meat/Poultry	39
Aniseed Syrup	Condiments	10
Boston Crab Meat	Seafood	18,4
Camembert Pierrot	Dairy Products	34
Carnarvon Tigers	Seafood	62,5
Chai	Beverages	18

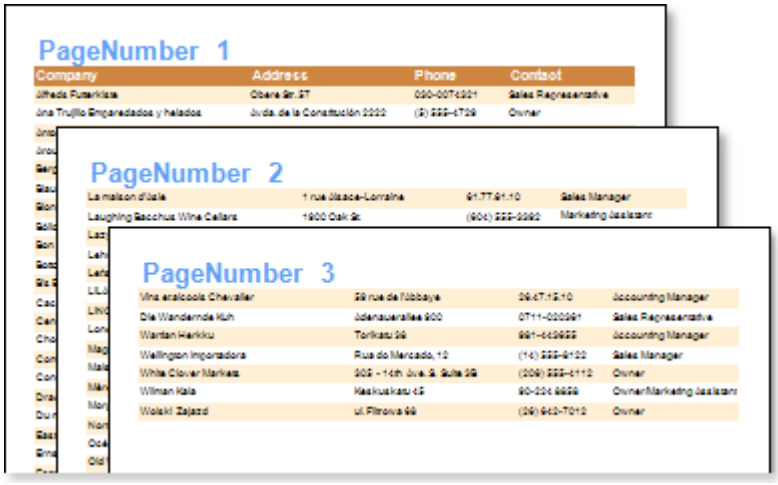
PrintOn Property

The PrintOn property have all components including HeaderBand and FooterBand. This property is used to display a component on report pages according to the value of this property. If the property is set to **All pages**, then components will be shown as usually. If the property is set to any other value then the component will not be showing on the first/last page of a report or on the contrary will be shown on all pages except the first/last ones.

The **PrintOn** property has the following values:

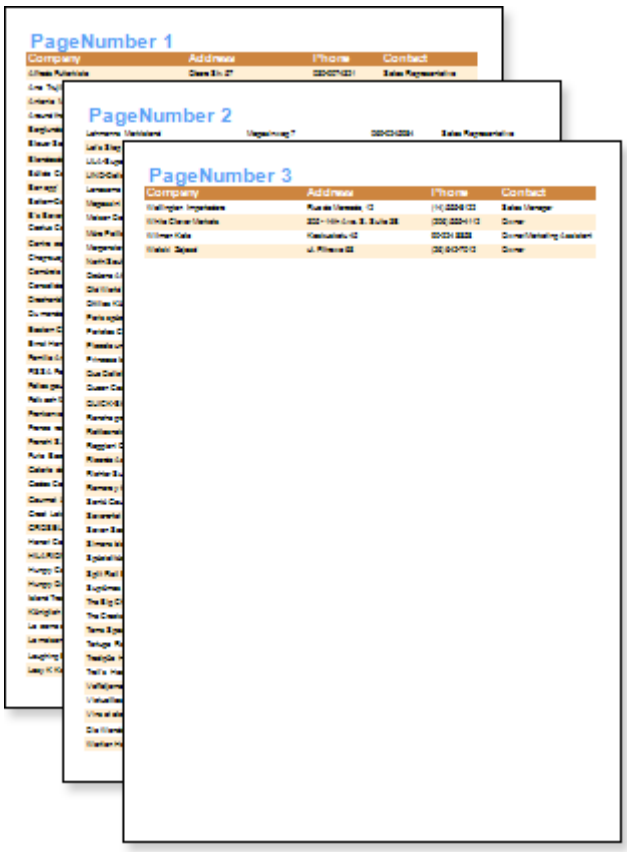
- **All pages**;
- **ExceptFirstPage**;
- **ExceptLastPage**;
- **ExceptFirstAndLastPages**;
- **OnlyFirstPage**;
- **OnlyLastPage**;
- **OnlyFirstAndLastPages**.

The picture below shows a report sample with the **PrintOn** property of the **HeaderBand** set to **OnlyFirstPage**.

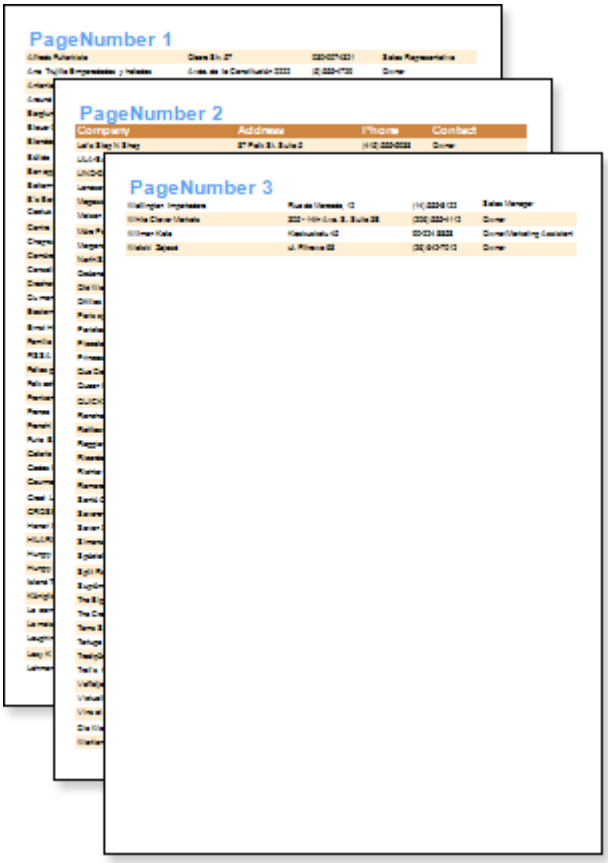


PrintOnEvenOddPages Property

The **PrintOnEvenOddPages** property is used to print headers and footers on even/odd pages, for **HeaderBands** and **FooterBands**.



The picture above shows a sample of a report with the **PrintOnEvenOddPages** property of the **HeaderBand** set to **OddPage**.



The picture above shows a sample of a report with the **PrintOnEvenOddPages** property of the **HeaderBand** set to **EvenPage**.

Three values are available for this property:

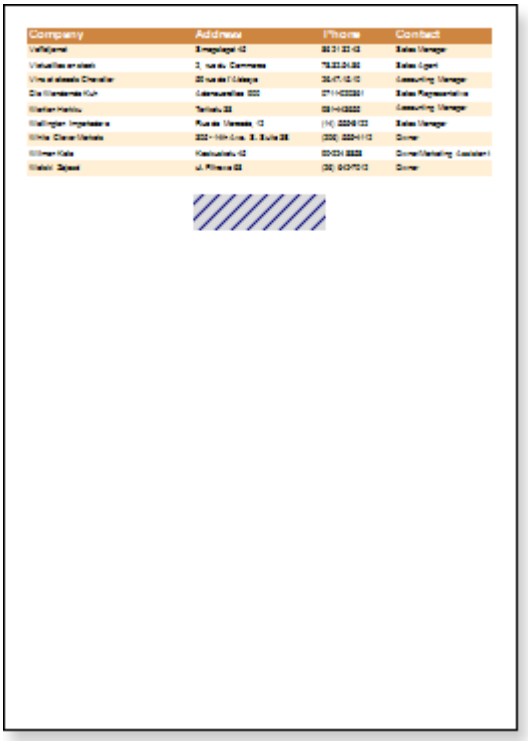
- **Ignore**. Headers and footers are printed on all pages;
- **PrintOnEvenPages**. Headers and footers are printed on even pages;
- **PrintOnOddPage**. Headers and footers are printed on odd pages.

PrintOnAllPages Property

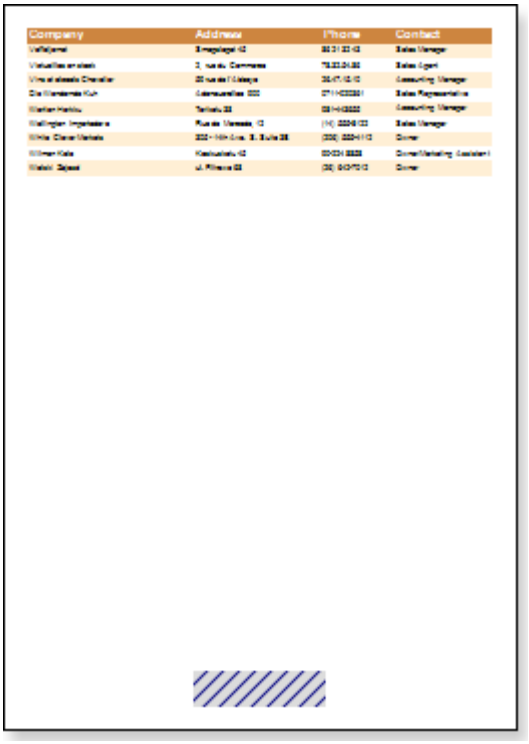
HeaderBand, **FooterBand**, **ColumnHeaderBand**, **ColumnFooterBand**, **GroupHeaderBand** have the **PrintOnAllPages** property, which may have two of the following values: **true** and **false**.
If the property is set to **false**, then bands are printed one time in a report before/after the DataBand to which they are related.
If the property is set to **true**, then these bands are printed only on report pages where a Data Band to which they are related is printed. The bands mentioned above are printed before/after their Data Band.
By default the **PrintOnAllPages** property is set to **true** for **HeaderBand** and **ColumnHeaderBand**. For other bands this property is set to **false**.

PrintAtBottom Property

HeaderBand and **FooterBand** have the **PrintAtBottom** property.
Sometimes data take third part of a page and the data footer will be output right after the data ends.



The picture above shows data footer output after data.
If you want to output the footer on the bottom of the page, then set the **PrintAtBottom** property for the FooterBand to **true**.
The data footer will be displayed at the bottom of the page.



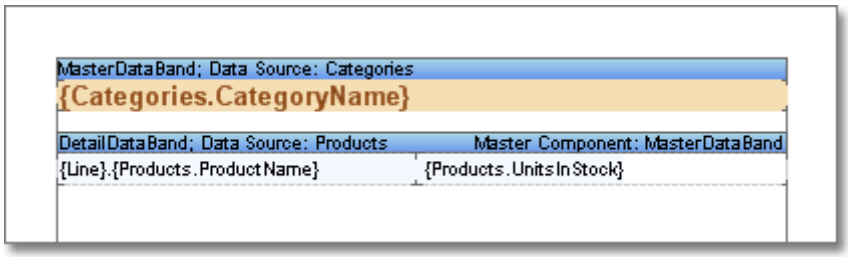
The default value of the property is set to **false**.

Creating Master-Detail Lists

The previous topic describes how to create a report using data as a table. And data are not connected to each other. Three bands were used: **Data**, **Header,** and **Footer**. But sometimes it is required to create reports and output data which are organized in some levels and connected to each other. For example, invoice and a list of goods, clients and goods delivery to them etc. In this case **Master-Detail** reports are used. These are reports in which the output value of the Master data source, corresponds to the number of values (from 0 and greater) from the Detail data source. On the picture below the example of the Master-Detail report is shown:

Beverages	
1.Chai	10 boxes x 20 bags
2.Chang	24 - 12 oz bottles
3.Chartreuse verte	750 cc per bottle
4.Côte de Blaye	12 - 75 cl bottles
5.Guaraná Fantástica	12 - 355 ml cans
6.Ipoh Coffee	16 - 500 g tins
7.Lakkalikööri	500 ml
8.Laughing Lumberjack Lager	24 - 12 oz bottles
9.Outback Lager	24 - 355 ml bottles
10.Rhönbräu Klosterbier	24 - 0.5 l bottles
11.Sasquatch Ale	24 - 12 oz bottles
12.Steeleye Stout	24 - 12 oz bottles
12	
Condiments	
1.Aniseed Syrup	12 - 550 ml bottles
2.Chef Anton's Cajun Seasoning	48 - 6 oz jars
3.Chef Anton's Gumbo Mix	36 boxes
4.Genen Shouyu	24 - 250 ml bottles
5.Grandma's Boysenberry Spread	12 - 8 oz jars
6.Gula Malacca	20 - 2 kg bags
7.Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles
8.Louisiana Hot Spiced Okra	24 - 8 oz jars
9.Northwoods Cranberry Sauce	12 - 12 oz jars
10.Original Frankfurter grüne Soße	12 boxes
11.Sirop d'érable	24 - 500 ml bottles
12.Vegie-spread	15 - 625 g jars
12	

As one can see on the picture, each category of products corresponds to the list of products from this category. An example of the Master-Detail report template is shown on the picture below:



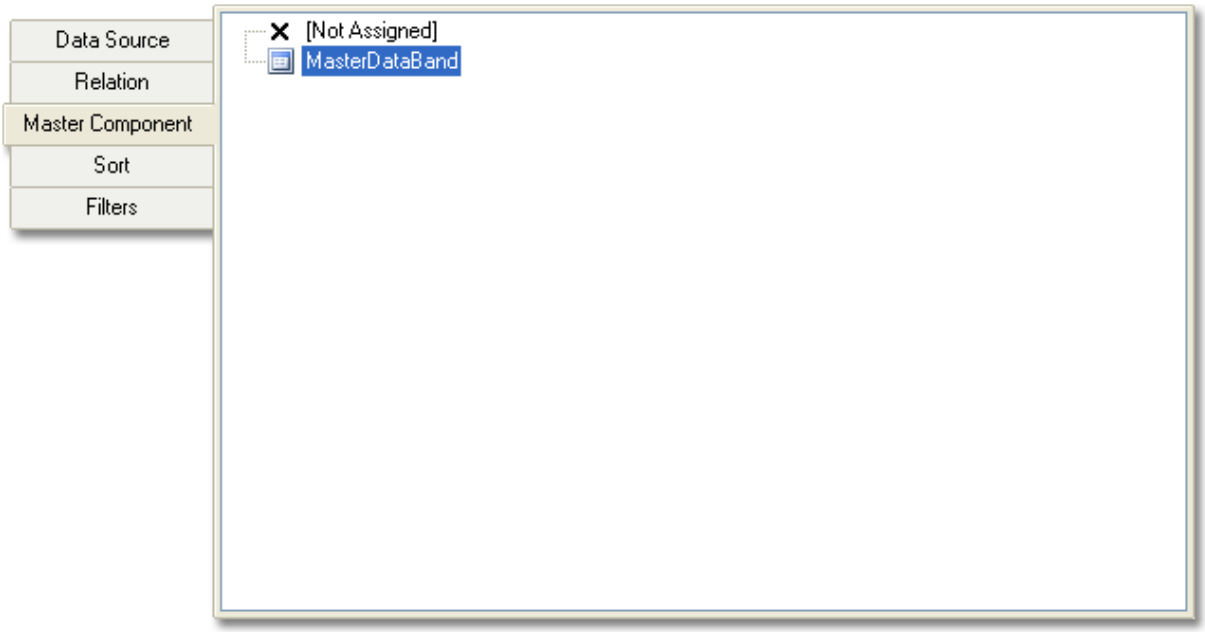
Data are output in the Detail part of the Master-Detail report are nested data. These data are as if nested into one data row of the Master data source. And the number of nesting is called the level of nesting. For example, if in the report the Master-Detail report two lists are output and the second list is connected with the first list, then this report will have two levels of nesting (the first is the Master, and the second is the Detail). And if this detailed list will have an additional list which will detail this list, then this report will have three levels of nesting (the first is the Master, the second is the Detail, and the third is the SubDetail). The number of nesting is unlimited. Usually number of nesting is no more than 3-4 levels.

MasterComponent Property

Put two **Data** bands on a page to start creating the Master-Detail report. Specify the Master data source to the first band (this is the Master band). Specify the Detail data source to the second band (this is the Detail). Then, it is necessary to bind these bands using the **MasterComponent** property of the second band. The Master band should be selected.

MasterComponent	**MasterDataBand**

The selection can be made in the **Data** band editor window.



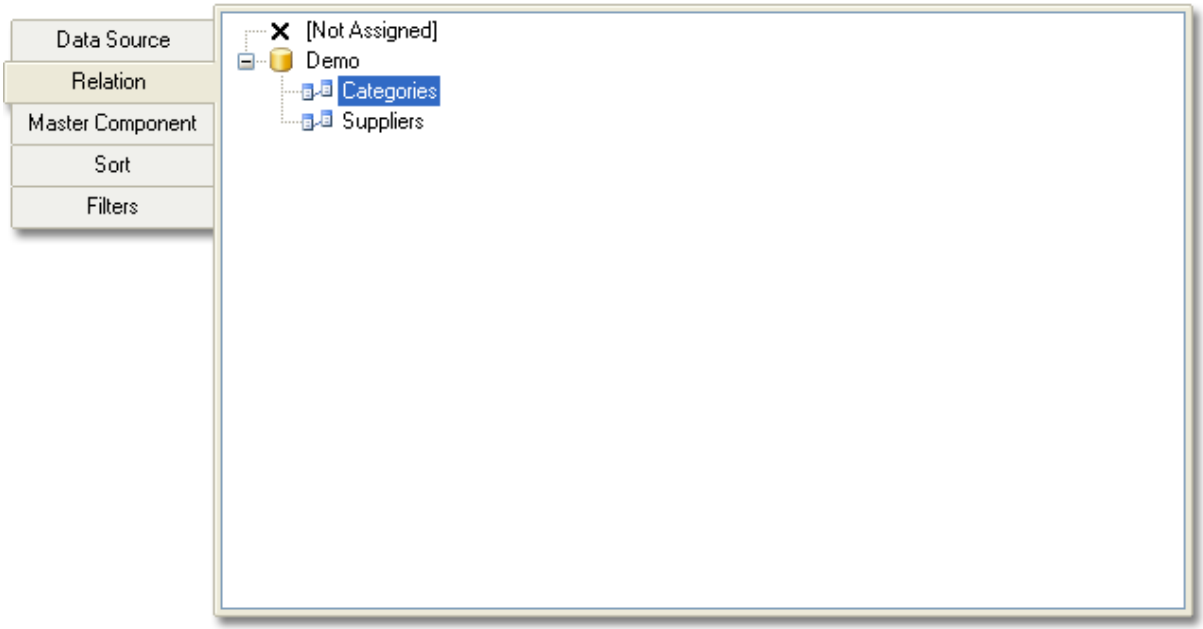
After filling the **MasterComponent** property two bands will be bound to each other. When printing one row of the Master band, all rows of the Detail band will be output. The Detail band will not be printed itself but only in relation to the Master band.

DataRelation Property

After filling the **MasterComponent** property it is necessary to fill the **Data Relation** property of the Detail band. This relation is used to select detailed data only for the specific Master band row. If the relation is not specified, then all Detail band rows will be output for each rows of the Master band.

Data Relation	Categories

Selection of relation occurs using the **Data** band editor, as well as in case with the **MasterComponent** property.



Selection is done between relations which were created between Master and Detail data sources, and in which the Detail data source is subordinate data source. There can be more than one relation (for example, as seen on the picture above). Therefore, it is important to select the correct relation.

Relation

If the **Relation** is not specified in the **Master-Detail** report, then, for each **Master** record, all **Detail** records will be printed. In order to build a **Master-Detail** report, which will print only those **Detail** records that are associated with this **Master** record, you should create a **Relation** between data sources. The **Relation** describes the relationship between data sources such as "master-detail". For example, in the table of the **Categories** data source in the **CategoriesID** data column, may be one record with a unique name **1**, and in the table of the **Products** data source in the **CategoriesID** column data may be many records with the same unique name **1**. The picture below shows an example of data source tables:

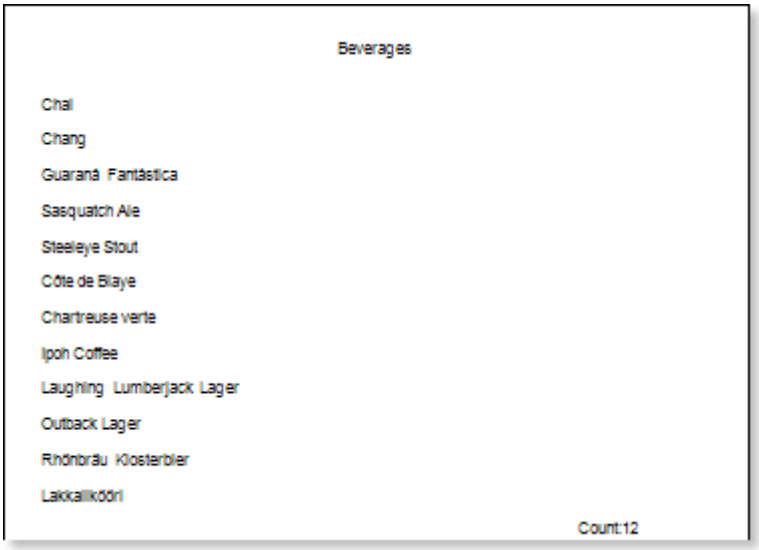
Categories

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beer
2	Condiments	Sweet and savory sauces
3	Confections	Desserts, candies, and sweet
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

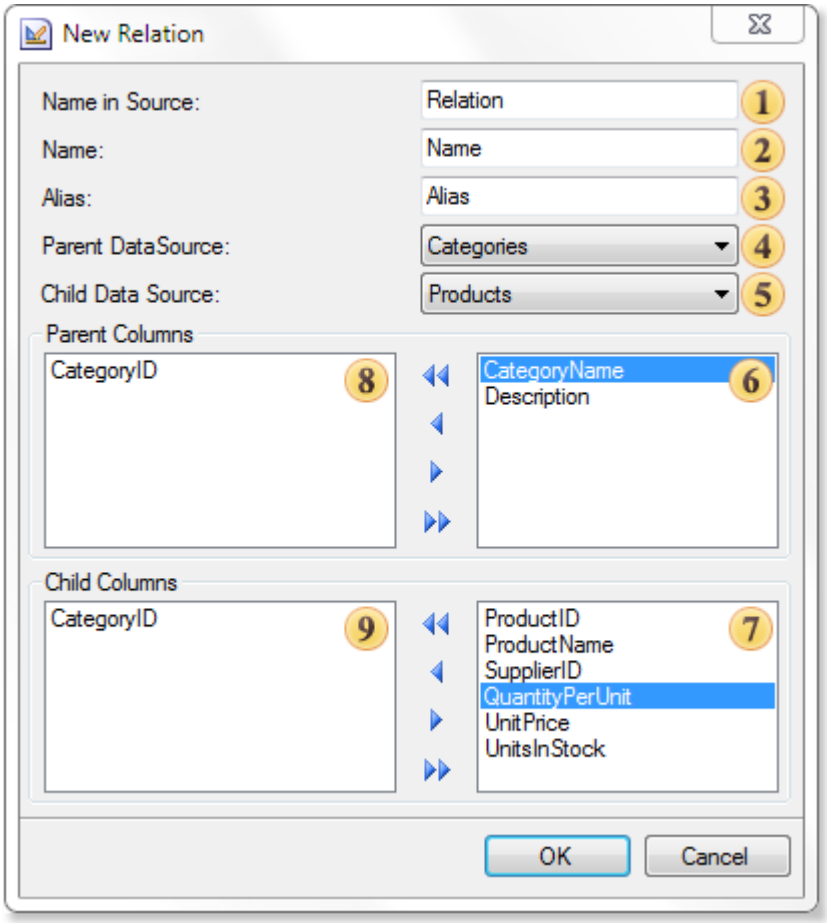
Products

ProductID	ProductName	SupplierID	CategoryID
1	Chai	1	1
2	Chang	1	1
24	Guaraná Fant	10	1
34	Sasquatch Al	16	1
35	Steeleye Sto	16	1
38	Côte de Blay	18	1
39	Chartreuse v	18	1
43	Ipoh Coffee	20	1
67	Laughing Lu	16	1
70	Outback Lag	7	1
75	Rhönbräu Klo	12	1
76	Lakkalikööri	23	1
3	Aniseed Syru	1	2
4	Chef Anton's	2	2

As can be seen from the picture above, one record with the name **1** in the table of the **Categories** data source corresponds to 12 records in the table of the **Products** data source. In other words, if you create a **Relation** by the **CategoriesID** column data between **Categories** and **Products** data tables, then when creating the **Master-Detail** report, the first **Master** record will correspond to **Detail** 12 entries. The picture below shows an example of the rendered **Master-Detail** report by **CategoryName** and **ProductName** columns, where the **Relation** is arranged between the **Product** and **Category** data sources by columns of **CategoryID** data:



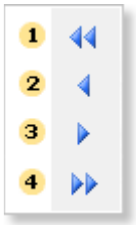
The parameters of relations are specified in the **New Relation** window. To invoke this window, choose the **New Relation** item from the context menu of the data source or click the **New Relation** button form the **Data Setup** window in the **Relation** tab. The picture below shows an example of the **New Relation** window:



As can be seen on the picture above, nine fields, which define the relation parameters:

1 The **Name in Source** field provides an opportunity to change the name of the data source (not in the report), i.e. the name in the original data source, for example, in a database;
2 The **Name** field provides an opportunity to change the name of the relation that is displayed to a user;
3 The **Alias** field provides an opportunity to change the alias of the relation;
4 The **Parent DataSource** field provides an opportunity to change the main data source, i.e. the data source which entries are **Master** entries in the **Master-Detail** report is selected;
5 The **Child Data Source** provides an opportunity to change the child data source, i.e. the data source which entries are **Detail** entries in the **Master-Detail** report is selected;
6 This field displays the column-keys of the master data source;
7 This field displays the column-keys of the child data source;
8 - 9 fields shows the master and child data column-keys, which set the **Relation** between data sources. Column-keys should comply with all rules of creation relations in ADO.NET:
1 It should be the same number of them;
2 Their types should match, i.e. if the master column-key of the **String** type, then the child column-key should be of the **String** type;
3 And so on;

Control panel of data columns in the **New Relation** dialog box is represented by 4 buttons.



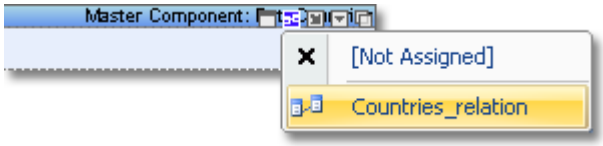
1 The button to move all data columns from the field 6 or 7 in the field 8 or 9, respectively;
2 The button to move the selected data column from the field 6 or 7 in the field 8 or 9, respectively;
3 The button to move the selected data column from the field 8 or 9 in the field 6 or 7, respectively;

4 The button to move all the data columns from the field 8 or 9 in the field 6 or 7, respectively.

Multilevel Nesting

The logic of building Master-Detail reports with more than 2 nesting levels is the same as the logic of building simple Master-Detail reports. For each Detail band the **MasterComponent** and **DataRelation** properties are set. For example, it is necessary to render a report in what there are four nesting levels. The first level is **countries**, the second - **regions**, the third - **cities**, the fourth - **quarters**. In this case one should place **Data** bands one on another on a page for each data source. Set the **MasterComponent** of the second band on the band **countries**. This property for the third band will indicate the **regions** band. For the last band **quarters** - will indicate on the **cities** band.

Then it is necessary to select relations for three bands for the report generator is able to select correct data for each detailed band.

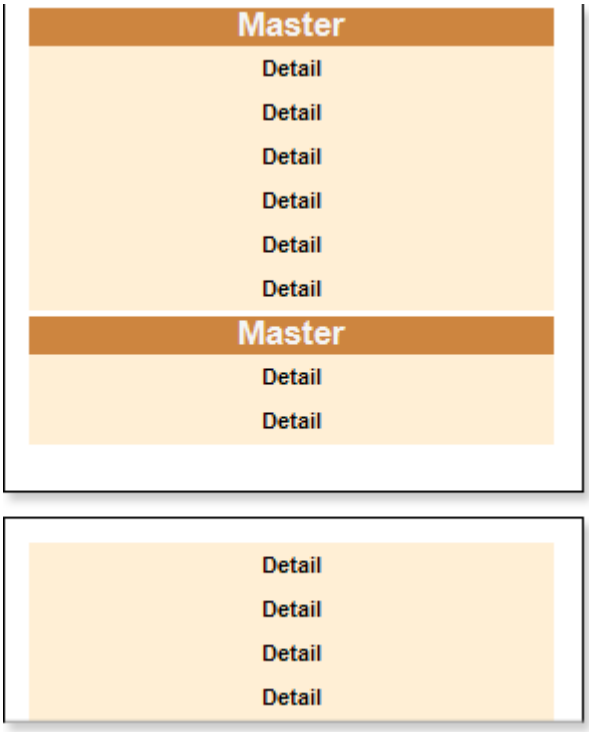


Then this report will be ready for rendering. One **Master** band may have more than one **Detail** band. In other words two, three or four **Detail** bands may refer to it. And each of them may have their own Detail bands. There are no limitations on number of nesting levels in the **Master-Detail** reports.

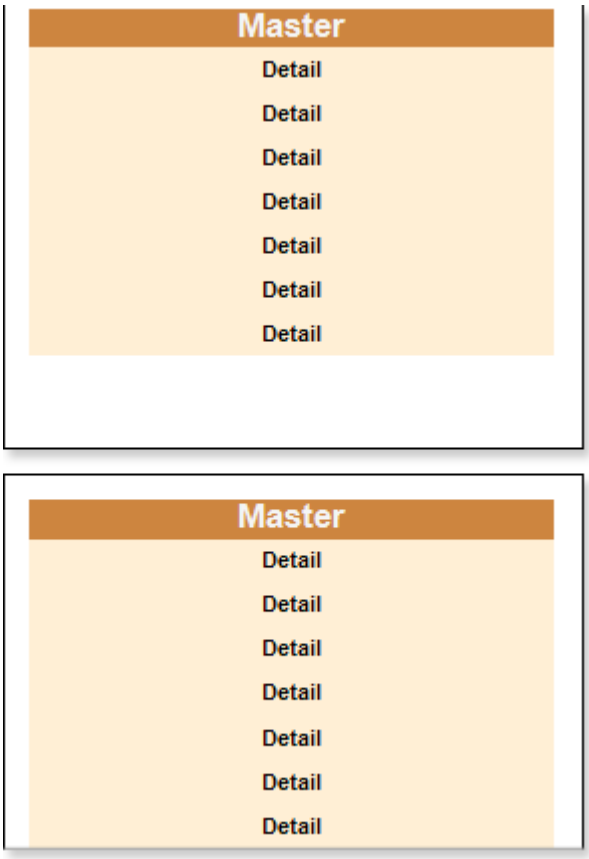
Notice. Number of nesting levels in the **Master-Detail** reports is unlimited.

KeepDetailsTogether Property

Sometimes, when creating **Master-Detail** reports, a part Details (subordinate entries) of the **Master-Detail** band will be on one page, while another part will be moved to the next page. This may happen due to the fact that all the detailed records will not fit one page. In this case, if it is still necessary to output the **Master** along with its details on one page, you can use the **KeepDetailsTogether** property. By default, this property is set to **false**.



The picture above shows a report in what a part of Details is located on one page, while the other part of details has been moved to the next page. If property is set to **true**, then the report generator will try to place the **Master** and **Detail** records on one page. If the report generator cannot do it, the **Master** and **Details** together will be moved to the next page.



The picture above shows an example of a report with the **KeepDetailsTogether** property of the **Master** set to **true**. If it is not possible to put them together, then the data will be forcibly broken and displayed on different pages. In this case, if the **Master** component has many **Detail** records and take a significant part on the page, and the **KeepDetailsTogether** property is set to **true**, then there may be a large empty space at the bottom of each page.

Rows Numbering in Master-Detail Reports

Rows numbering in the Master-Detail reports works the same as in ordinary lists. But there is on difference. If numbering is used in the Detail of the **Data** band, then for each sublist there will be their own numbering. For example, on the picture below the Master-Detail report is shown.

1.Beverages	
1.Chai	10 boxes x 20 bags
2.Chang	24 - 12 oz bottles
3.Chartreuse verte	750 cc per bottle
4.Côte de Blaye	12 - 75 cl bottles
5.Guaraná Fantástica	12 - 355 ml cans
6.Ipoh Coffee	16 - 500 g tins
7.Lakkalikööri	500 ml
8.Laughing Lumberjack Lager	24 - 12 oz bottles
9.Outback Lager	24 - 355 ml bottles
10.Rhönbräu Klosterbier	24 - 0.5 l bottles
11.Sasquatch Ale	24 - 12 oz bottles
12.Steeleye Stout	24 - 12 oz bottles
2.Condiments	
1.Aniseed Syrup	12 - 550 ml bottles
2.Chef Anton's Cajun Seasoning	48 - 6 oz jars
3.Chef Anton's Gumbo Mix	36 boxes
4.Genen Shouyu	24 - 250 ml bottles
5.Grandma's Boysenberry Spread	12 - 8 oz jars
6.Gula Malacca	20 - 2 kg bags
7.Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles
8.Louisiana Hot Spiced Okra	24 - 8 oz jars
9.Northwoods Cranberry Sauce	12 - 12 oz jars
10.Original Frankfurter grüne Soße	12 boxes
11.Sirop d'érable	24 - 500 ml bottles
12.Vegie-spread	15 - 625 g jars

Numbering in the Master list is indicated with the red color. Numbering in the Detail list is indicated with green color. As you can see on the picture, the numbering in the Detail list starts every time after the row from the Master list is output.

Besides using system variables numbering can be done using the **Line** property of the **Data** band. In this case the expression will be as follow:

```
{DetailDataBand1.Line}.{Customers.CompanyName}
```

Why is it necessary? Why not to use the **Line** system variable? The system variable has the visibility zone. For example, you use the **Line** system variable on the Master band. In this case numbering will be output for the Master band. If you use the **Line** system variable on the Detail band, then, in this case, numbering will be output for the Detail band. But what to do if it is necessary to output numbering of two different **Data** bands in one expression? In this case the **Line** property of the **Data** band is used. For example, see the following expression on the Detail band:

```
{DataBand1.Line}.{Line}.{Products.ProductName}
```

this will lead to the following result in a report:

1.Beverages	
1.1.Chai	10 boxes x 20 bags
1.2.Chang	24 - 12 oz bottles
1.3.Chartreuse verte	750 cc per bottle
1.4.Côte de Blaye	12 - 75 cl bottles
1.5.Guaraná Fantástica	12 - 355 ml cans
1.6.Ipoh Coffee	16 - 500 g tins
1.7.Lakkalikööri	500 ml
1.8.Laughing Lumberjack Lager	24 - 12 oz bottles
1.9.Outback Lager	24 - 355 ml bottles
1.10.Rhönbräu Klosterbier	24 - 0.5 l bottles
1.11.Sasquatch Ale	24 - 12 oz bottles
1.12.Steeleye Stout	24 - 12 oz bottles
2.Condiments	
2.1.Aniseed Syrup	12 - 550 ml bottles
2.2.Chef Anton's Cajun Seasoning	48 - 6 oz jars
2.3.Chef Anton's Gumbo Mix	36 boxes
2.4.Genen Shouyu	24 - 250 ml bottles
2.5.Grandma's Boysenberry Spread	12 - 8 oz jars
2.6.Gula Malacca	20 - 2 kg bags
2.7.Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles
2.8.Louisiana Hot Spiced Okra	24 - 8 oz jars
2.9.Northwoods Cranberry Sauce	12 - 12 oz jars
2.10.Original Frankfurter grüne Soße	12 boxes
2.11.Sirop d'érable	24 - 500 ml bottles
2.12.Vegie-spread	15 - 625 g jars

Through Lines Numbering in Master-Detail Reports

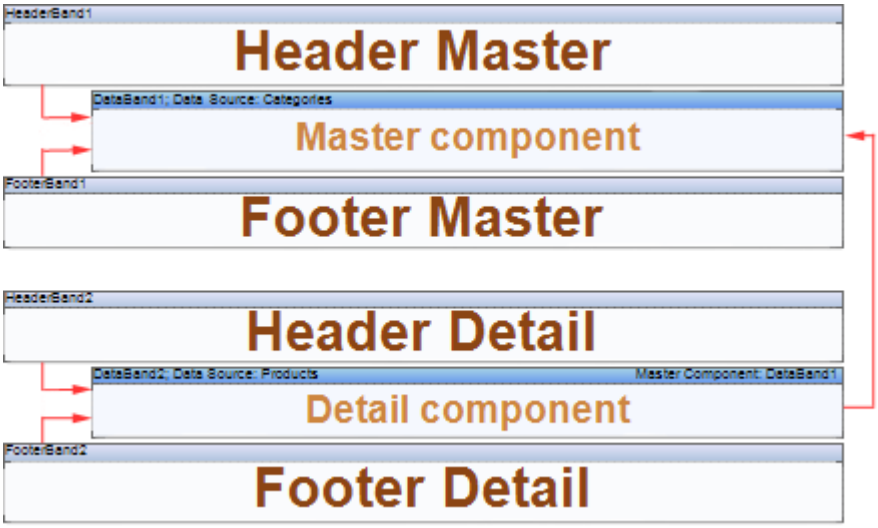
Besides the **Line** system variable, there is also additional **LineThrough** system variable for numbering the **Master-Detail** lists. What is the difference? The **LineThrough** system variable is used to output numbers using the continuous numbering. On the picture below the same report with continuous numbering is shown.

1.Beverages	
1.Chai	10 boxes x 20 bags
2.Chang	24 - 12 oz bottles
3.Chartreuse verte	750 cc per bottle
4.Côte de Blaye	12 - 75 cl bottles
5.Guaraná Fantástica	12 - 355 ml cans
6.Ipoh Coffee	16 - 500 g tins
7.Lakkalikööri	500 ml
8.Laughing Lumberjack Lager	24 - 12 oz bottles
9.Outback Lager	24 - 355 ml bottles
10.Rhönbräu Klosterbier	24 - 0.5 l bottles
11.Sasquatch Ale	24 - 12 oz bottles
12.Steeleye Stout	24 - 12 oz bottles
2.Condiments	
13.Aniseed Syrup	12 - 550 ml bottles
14.Chef Anton's Cajun Seasoning	48 - 6 oz jars
15.Chef Anton's Gumbo Mix	36 boxes
16.Genen Shouyu	24 - 250 ml bottles
17.Grandma's Boysenberry Spread	12 - 8 oz jars
18.Gula Malacca	20 - 2 kg bags
19.Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles
20.Louisiana Hot Spiced Okra	24 - 8 oz jars
21.Northwoods Cranberry Sauce	12 - 12 oz jars
22.Original Frankfurter grüne Soße	12 boxes
23.Sirop d'érable	24 - 500 ml bottles
24.Vegie-spread	15 - 625 g jars

In this case the numbering of the Detail list starts not after the row of the Master list is output but before the first row of the Detail list is output. The system variable starts numbering with 1.

Headers, Footers and Master-Detail Reports

The principle of using **HeaderBands** and **FooterBands** in **Master-Detail** reports is the same as in simple lists. All **HeaderBand1** bands, which are placed above the **DataBand1** bands, up to the next **DataBand2** band, belong to this **DataBand1** band. The **HeaderBand** is placed on the page above the **DataBand**, which outputs data rows. The **HeaderBand** always refers to any particular **DataBand**. Typically, this band is the first **DataBand**, which is located below the **HeaderBand**.
The **FooterBand** is placed below the **DataBand**. And it is meant that the **DataBand**, with what the **HeaderBand** is bind. Each **FooterBand**, refers to any specific **HeaderBand**. Without the **HeaderBand**, the **FooterBand** is not output.



The picture above shows a structure of a **Master-Detail** reports with two **DataBand** bands**.**

PrintifDetailEmpty Property

The **PrintifDetailEmpty** property of the **DataBand** band is used in building **Master-Detail** reports. The picture below shows a template of a **Master-Detail** report.

For example, not all **Master** entries have **Detail** records. Then, if the **PrintIfDetailEmpty** property is set to **false**, then the result shown below is obtained:

Only a part of Master records (in the picture above they are marked with numbers 2 and 6) will be output and the remaining Master records (which have no Detail records) will not be output. To print all Master records, regardless whether they have Detail records, it is necessary to set the **PrintifDetailEmpty** property of the Master band to **true**. An example of a report for this case is shown below below:

1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Aniseed Syrup	10
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
17	Alice Mutton	39
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

As seen on the picture Master records were output (see numbers 1,3,4,5,7,8) i.e. all Master records. Moreover, they are output without Detail records. By default, the property is set to **false**.

Groups

One of the main tasks when rendering reports is grouping the data. Grouping can be used both for the logical separation of data rows and to make a report look better. Two bands are used to create grouped reports: the **GroupHeader** band and the **GroupFooter** band.

The **GroupHeader** band is output in the beginning of each group. The **GroupFooter** band is output in the end of each group. The picture below shows how a report with grouping may look:

A

Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative

Count: 4

B

Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative
Blondesddsl père et fils	24, place Kléber	88.60.15.31	Marketing Manager
Bólido Comidas preparadas	C/ Araquil, 67	(91) 555 22 82	Owner
Bon app'	12, rue des Bouchers	91.24.45.40	Owner
Bottom-Dollar Markets	23 Tsawassen Blvd.	(604) 555-4729	Accounting Manager
B's Beverages	Fauntleroy Circus	(171) 555-1212	Sales Representative

Count: 7

C

Cactus Comidas para llevar	Cerrito 333	(1) 135-5555	Sales Agent
Centro comercial Moctezuma	Sierras de Granada 9993	(5) 555-3392	Marketing Manager
Chop-suey Chinese	Hauptstr. 29	0452-076545	Owner
Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647	Sales Associate
Consolidated Holdings	Berkeley Gardens 12 Brewery	(171) 555-2282	Sales Representative

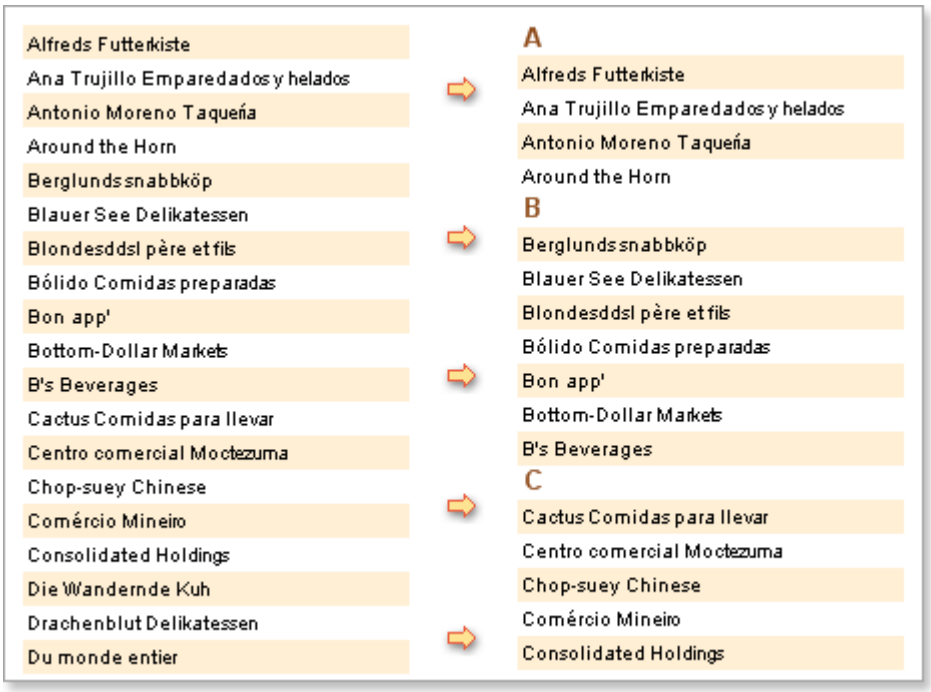
Count: 5

Grouping Conditions

To create a report with grouping it is necessary to define a condition by which the records can be grouped. This condition will be used to divide the data rows into suitable groups, and is set using the Condition property of the GroupHeaderBand.

Important: You **MUST** define a condition for every group, otherwise no grouping will take place in the rendered report

For example, if you create a report that generates a list of companies the results could be grouped in alphabetical order by the first letter of the company name. Companies with names starting with A would be in the first group, companies with names starting with B would be in the second group and so on, as in the example below:



The grouping condition you use can be any valid value. For example, if you wanted the companies to be grouped according to their location you could set the condition to group on a column from the database that contains the necessary location data.

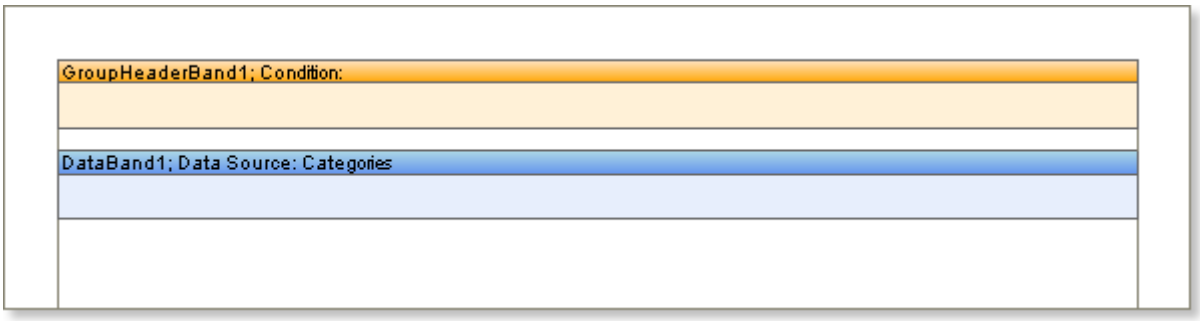
GroupHeaderBand

The Group header is created using the **GroupHeader** band, the basic band for rendering reports that use grouping. It is impossible to generate grouped reports without using a **GroupHeader** band.

The **GroupHeader** band is output once at the beginning of each group and typically contains components that display header information such as a group name, date, grouping condition, etc.

To create groups within a report you must specify a grouping condition using the **GroupHeader** band designer or the **Condition** property of the band.

Note: The Header band is always output before the **GroupHeader** band, regardless of where bands may be positioned on a page in the designer.



When rendering a report the report generator binds the group header to the specified Data band. The **GroupHeader** band is positioned on a page before the **Data** band that outputs data rows. The **GroupHeader** band always belongs to a specific **Data** band, usually the first **Data** band positioned under the **GroupHeader** band.

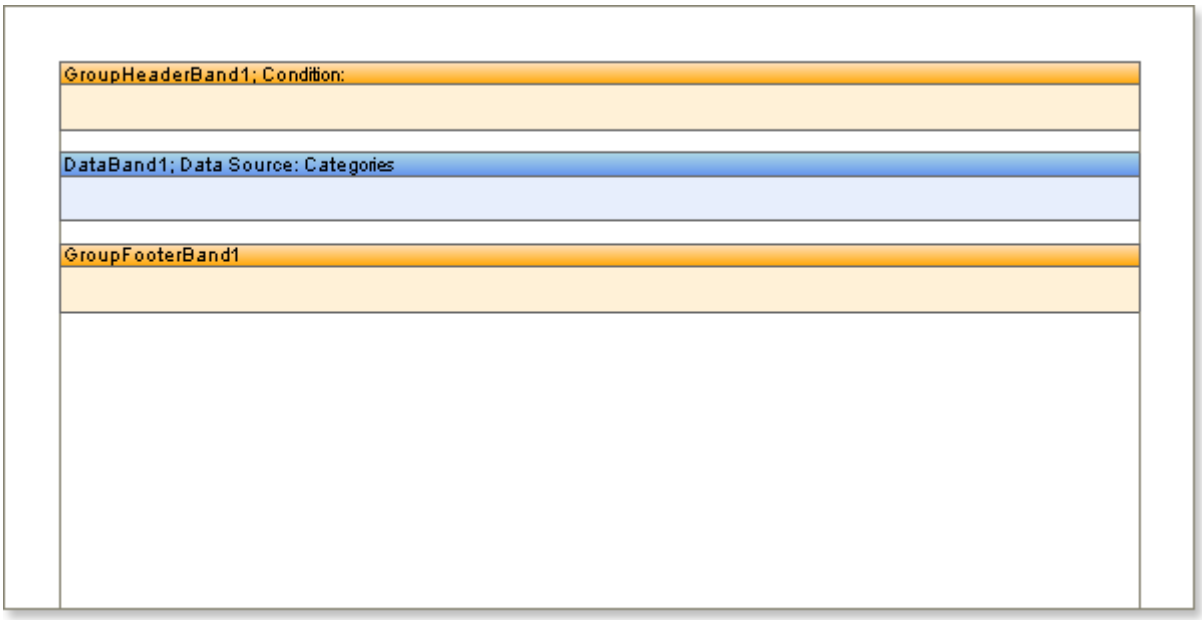
You must have a **Data** band to be able to render grouped reports because data rows are output using this band and because those data rows are the basis of the grouping in the report. In addition you can specify the sorting of rows in the **Data** band which will affect the order in which the groups are rendered.

Important: To render reports with grouping you **MUST** use a **Data** band.

GroupFooterBand

The **GroupFooter** band is commonly used to generate a group footer which is placed after the **Data** band bound to the group and typically contains components that output summary information relating to the group content. Every **GroupFooter** band belongs to the **GroupHeader** band associated with it, and will not be output if there is no associated **GroupHeader** band.

Note. The **GroupFooter** band is always output before the **Footer** band regardless of where bands may be positioned on a page.

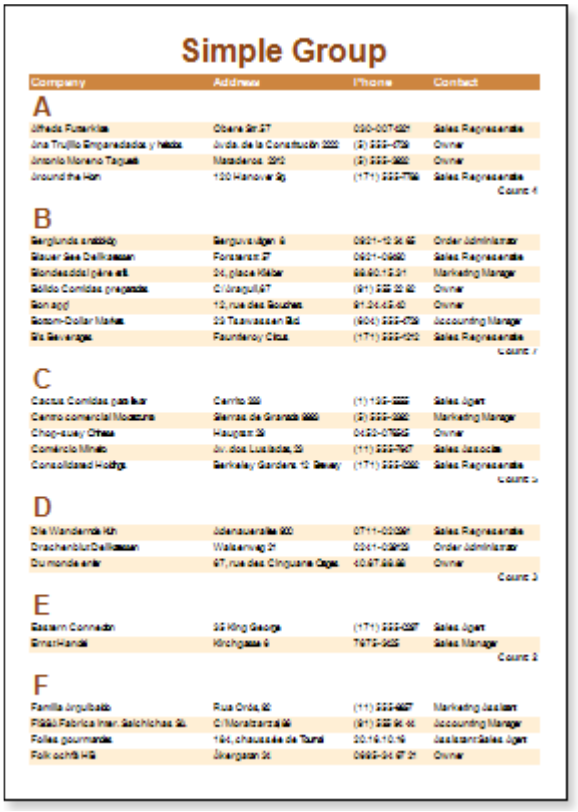


The **GroupFooter** band is used to output information specific to each group. For example, if you want to output the number of rows in a group, it is enough to put a text component on the **GroupFooter** band and assign it the following expression:

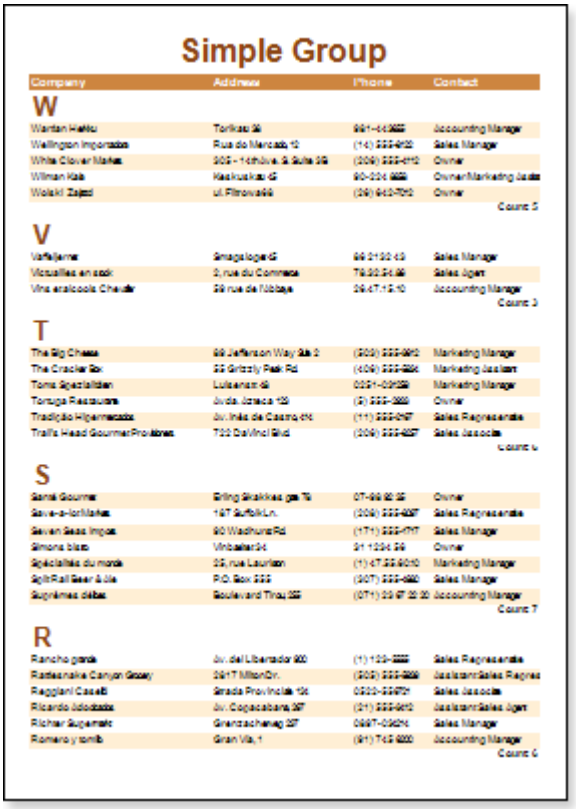
```
{Count()}
```

Data Sorting in Group

Please note that the report generator automatically sorts the rows of data before grouping. By default sorting by ascending order from A to Z is used. Sorting direction can be changed using the **SortDirection** property. This can take three values: **None**, **Ascending**, **Descending**.
1. **None**. The data will be displayed in order they are put in the data source.
2. **Ascending**. Data are displayed in alphabetical order from A to Z. The picture below shows an example of a report where sorting by ascending order:



3. **Descending.** Data are displayed in alphabetical order from Z to A. The picture below shows an example of a report where sorting by descending order:

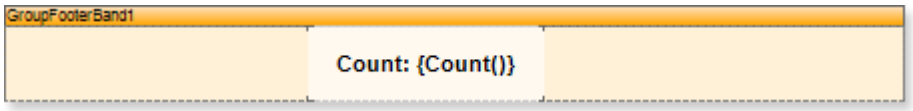


GroupFooter

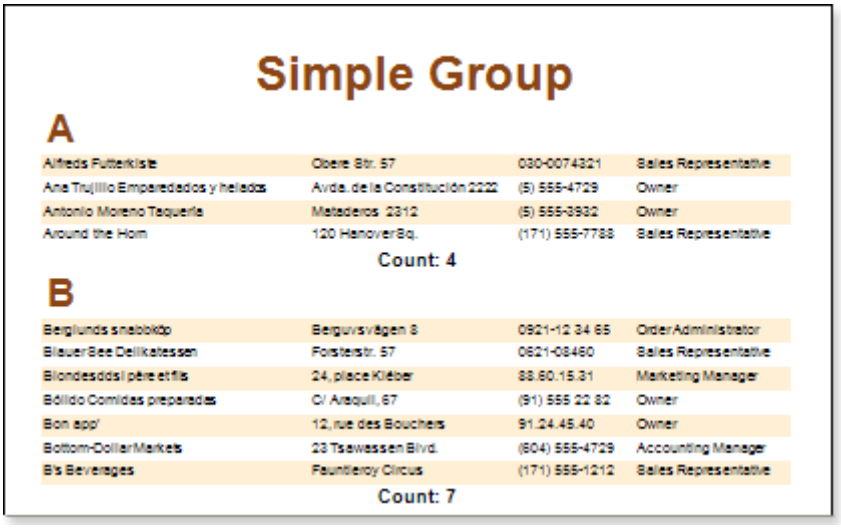
It is enough to place a text component with an aggregate function in a **GroupFooter** to output footer by group. Also, the footer of a group may be placed in a **GroupHeaderBand**. For example, in order to count the number of rows in each group in a **Text** component the following expression can be used:

```
{Count()}
```

A component is placed in the **GroupFooter** band.



After rendering, it is possible to see that in the footer of each group calculation by number of rows is done.



Simple Group

A

Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative

Count: 4

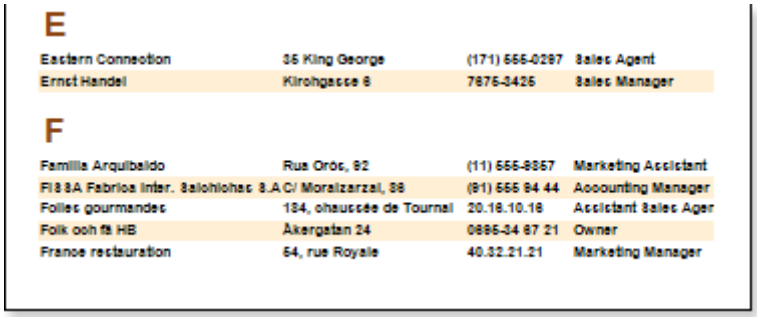
B

Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative
Blondesddsl père et fils	24, place Kléber	88.60.15.31	Marketing Manager
Bólido Comidas preparadas	C/ Araquil, 67	(91) 555 22 82	Owner
Bon app'	12, rue des Bouchers	91.24.45.40	Owner
Bottom-Dollar Markets	23 Tsawassen Blvd.	(604) 555-4729	Accounting Manager
B's Beverages	Fountleroy Circus	(171) 555-1212	Sales Representative

Count: 7

KeepGroupTogether Property

When rendering a report with grouping, a group may not fit to one page. I.e. several lines of group will be output on one page and other part on the next page.

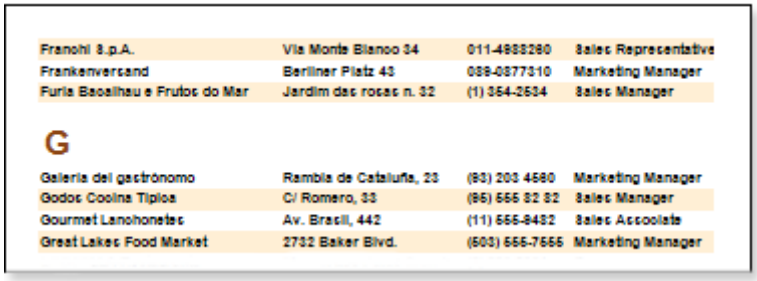


E

Eastern Connection	35 King George	(171) 555-0297	Sales Agent
Ernst Handel	Kirchgasse 6	7675-3425	Sales Manager

F

Familia Arquibaldo	Rua Orós, 92	(11) 555-9857	Marketing Assistant
FISSA Fabrica Inter. Salchichas S.A.	C/ Moralzarzal, 86	(91) 555 94 44	Accounting Manager
Folies gourmandes	184, chaussée de Tournai	20.16.10.16	Assistant Sales Agent
Folk och fä HB	Åkergatan 24	0695-34 67 21	Owner
France restauration	54, rue Royale	40.32.21.21	Marketing Manager

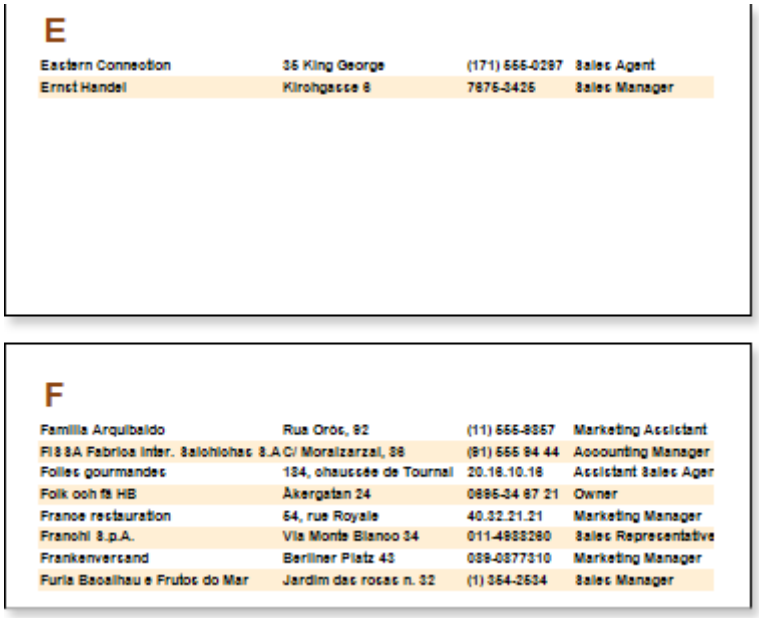


Franchi S.p.A.	Via Monte Bianco 34	011-4988260	Sales Representative
Frankenversand	Berliner Platz 43	089-0877310	Marketing Manager
Furia Bacalhau e Frutos do Mar	Jardim das rosas n. 32	(1) 354-2534	Sales Manager

G

Galería del gastrónomo	Rambla de Cataluña, 23	(93) 203 4560	Marketing Manager
Godos Cocina Típica	C/ Romero, 33	(95) 555 82 82	Sales Manager
Gourmet Lanchonetes	Av. Brasil, 442	(11) 555-9482	Sales Associate
Great Lakes Food Market	2732 Baker Blvd.	(503) 555-7555	Marketing Manager

This can be avoided using the **KeepGroupTogether** property of the **GroupHeaderBand**. If to set this property to **true**, then, if a group cannot be placed on one page, the whole group is moved to the next page. If it is impossible to print a group on the next page then the group will be forcibly broken and output on multiple pages.



E

Eastern Connection	35 King George	(171) 555-0297	Sales Agent
Ernst Handel	Kirchgasse 6	7675-3425	Sales Manager

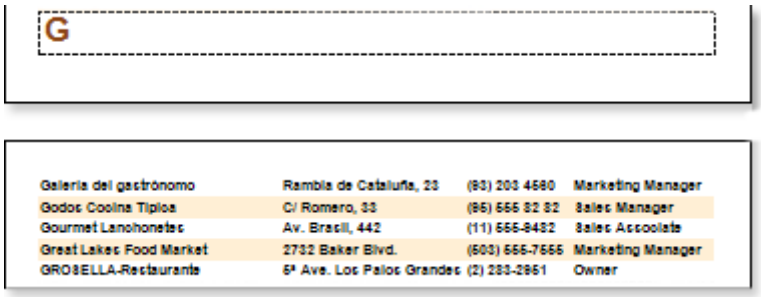
F

Familia Arquibaldo	Rua Orós, 92	(11) 555-9857	Marketing Assistant
FISSA Fabrica Inter. Salchichas S.A.	C/ Moralzarzal, 86	(91) 555 94 44	Accounting Manager
Folies gourmandes	184, chaussée de Tournai	20.16.10.16	Assistant Sales Agen
Folk och fä HB	Åkergatan 24	0695-34 67 21	Owner
France restauration	54, rue Royale	40.32.21.21	Marketing Manager
Franchi S.p.A.	Via Monte Bianco 34	011-4988260	Sales Representative
Frankenversand	Berliner Platz 43	089-0877310	Marketing Manager
Furia Bacalhau e Frutos do Mar	Jardim das rosas n. 32	(1) 354-2534	Sales Manager

Work with this property may lead to empty space on page, if groups contain a large number of rows.

KeepGroupHeaderTogether Property

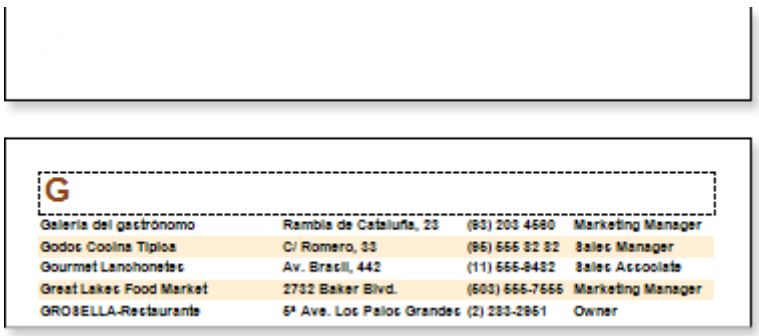
The **GroupHeaderBand** has the **KeepHeaderGroupTogether** property. If the property is set to **false**, then the group header can be displayed on one page, and data of a group to another page. So data will be separated from its header. The picture below shows that the header is on one page, and the data were moved to another.



G

Galería del gastrónomo	Rambla de Cataluña, 23	(93) 203 4560	Marketing Manager
Godos Cocina Típica	C/ Romero, 33	(95) 555 82 82	Sales Manager
Gourmet Lanchonetes	Av. Brasil, 442	(11) 555-9482	Sales Associate
Great Lakes Food Market	2732 Baker Blvd.	(503) 555-7555	Marketing Manager
GROSELLA-Restaurante	5ª Ave. Los Palos Grandes	(2) 283-2951	Owner

If the property is set to **true**, then the group header will be displayed with at least one row of a group. The picture below shows how a group will be output if the **KeepHeaderGroupTogether** property is set to true.



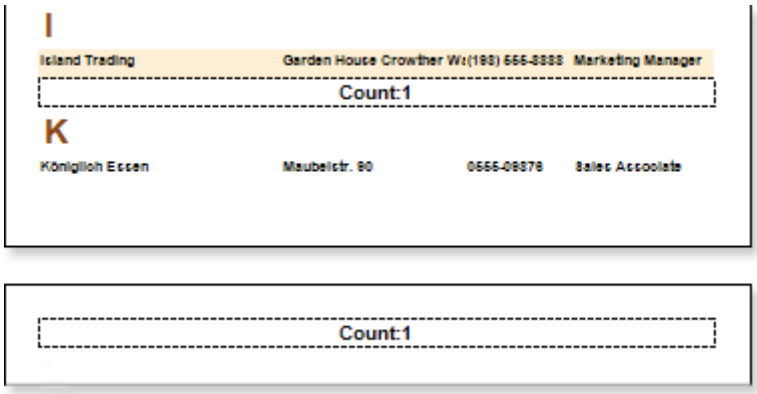
G

Galería del gastrónomo	Rambla de Cataluña, 23	(93) 203 4560	Marketing Manager
Godos Cocina Típica	C/ Romero, 33	(95) 555 82 82	Sales Manager
Gourmet Lanchonetes	Av. Brasil, 442	(11) 555-9482	Sales Associate
Great Lakes Food Market	2732 Baker Blvd.	(503) 555-7555	Marketing Manager
GROSELLA-Restaurante	5ª Ave. Los Palos Grandes	(2) 283-2951	Owner

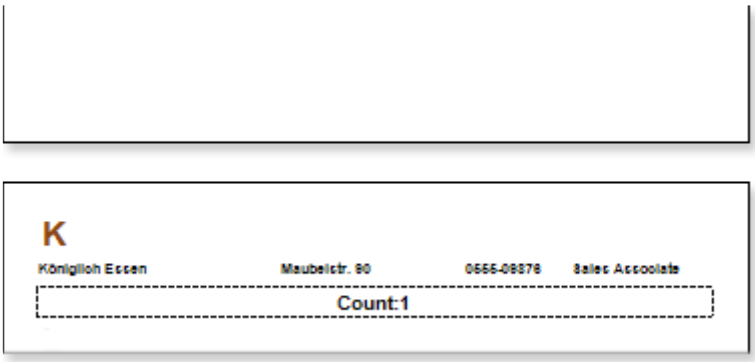
By default the **KeepHeaderGroupTogether** property is set to **true**.

KeepGroupFooterTogether Property

The **GroupFooterBand** has the **KeepGroupFooterTogether** property. If the property is set to **false**, then the data can be placed on one page and the footer of a group on another, and data of groups will be separated from its footer:



If the property is set to **true**, then at least one line of data will be together with the footer of a group:



By default this property is set to **true**.

Events and GroupHeaderBand

Like the **Data** band, the **Group Header** band has three specific events:

- **BeginRenderEvent**,
- **EndRenderEvent** and
- **RenderingEvent**.

BeginRenderEvent

The **BeginRenderEvent** is called before a group is rendered, in other words whenever a new group is output. This event can be used for the initialization of data or variables, or for calling certain actions.

EndRenderEvent

The **EndRenderEvent** is called after the group is output. Usually in the handler for this event data processing and the calculation of totals is done.

RenderingEvent

The **RenderingEvent** is called when the engine is rendering one data row from a group.

Group without GroupHeader

In grouped reports is is usual to display both a group header and a group footer. However, what if you need to output only group footers without group headers?

When creating grouped reports you must use a **GroupHeader** band, but if you do not want it to display it can be hidden by setting the height of the **GroupHeader** band to **0** which will cause the report to be rendered successfully but the **GroupHeader** band will not appear in the output.

Groups without GroupFooter

In grouped reports is is usual to display both a group header and a group footer. However, what if you need to output only group headers without group footers?

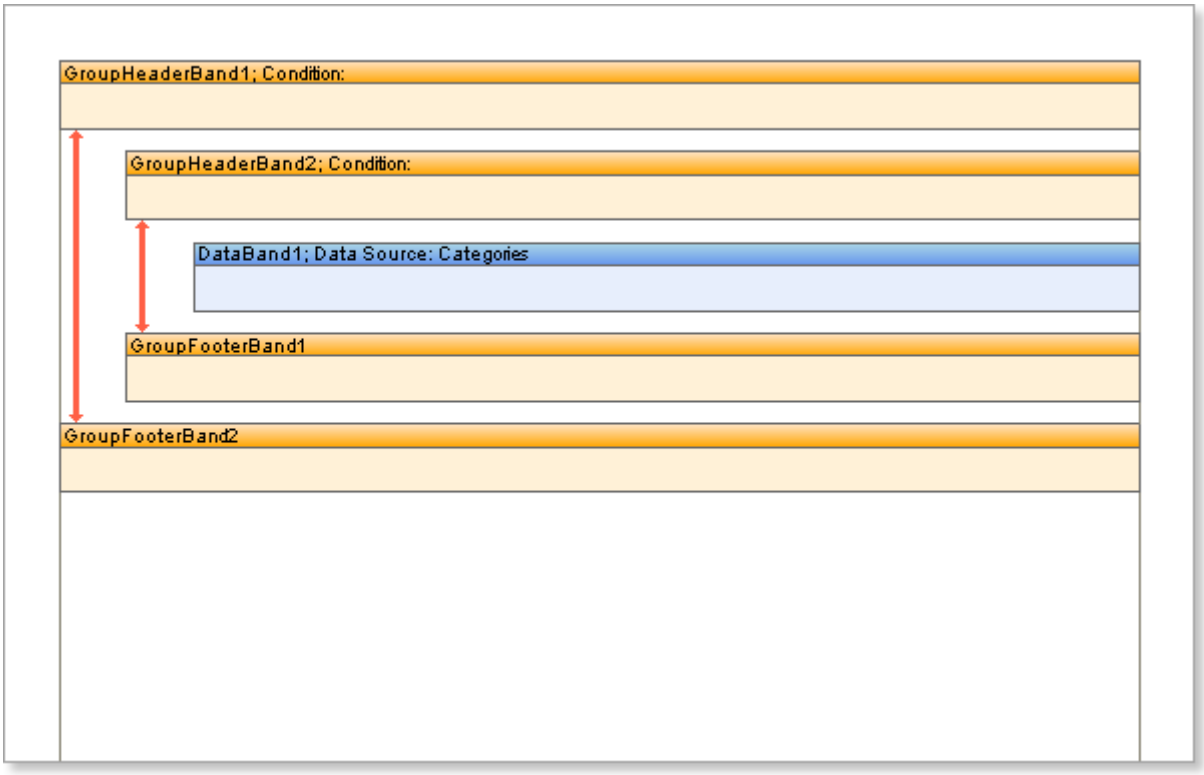
It is possible to simply not include a **GroupFooter**, but this is **NOT** recommended as it can lead to unexpected results particularly if you are working with **Nested** groups. It is therefore recommended that you **ALWAYS** use **GroupHeaders** and **GroupFooters** in pairs.

Important: To render reports with grouping you should always use **GroupHeaders** and **GroupFooters** in pairs to avoid the possibility of unexpected results.

If you do not want the **GroupFooter** to be displayed it can be hidden by setting its height to **0** which will cause the report to be rendered successfully but the band will not appear in the output.

Nested Groups

When rendering grouped reports you may use more than one grouping to achieve the desired output, known as 'nesting'. For example, you might group Customers by location and then sub group them alphabetically. To achieve this style of report you should put the required number of **GroupHeader** bands before the **DataBand** and ideally the same number of **GroupFooter** bands immediately after it:



Although it is possible to leave out unwanted **GroupFooters** it is recommended that you always place equal numbers of **GroupHeader** and **GroupFooter** bands on a report to avoid unexpected results. If the number of **GroupFooter** bands is greater than the number of **GroupHeader** bands then the outer ones will be used and the inner bands ignored. If the number of **GroupFooter** bands is less than the number of **GroupHeader** bands, then the **GroupHeader** bands placed closer to the **Data** band will be output without footers.

Important: It is recommended to have equal number of **GroupHeader** and **GroupFooter** bands in a report.

In each **GroupHeader** band you must specify the grouping criteria. When rendering the report the **GroupHeader** bands are processed in the in which they appear on a page working from the top down i.e. the topmost band is processed first, then the one that is placed directly underneath it and so on. When placing **GroupFooter** bands on a report page it is important remember that the last **GroupFooter** band is always associated with the first **GroupHeader** band.

Numbering Rows in Group

If you wish to display line numbers within a group you should use the **Line** system variable. The reference to this variable should be specified in the expression assigned to a text component placed on the group Data band.

For example, put a text component on the **Data** band and write the following expression in it:

```
{Line}
```

After the report has been rendered there will be a numbered list of rows in each group, the numbers starting 1.

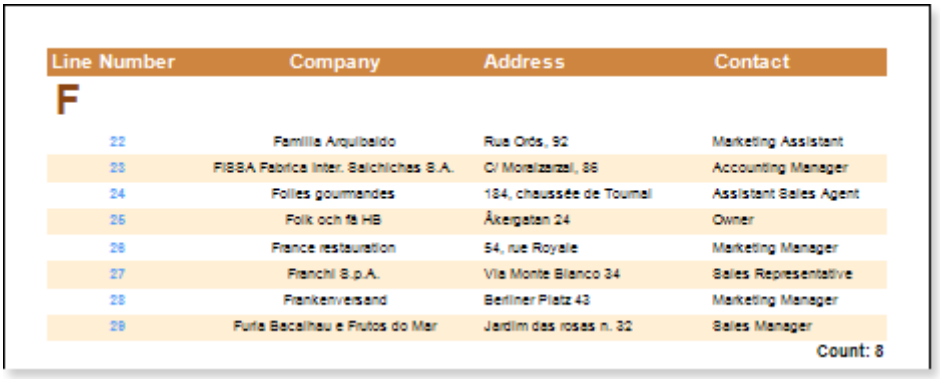
In each new group within a report the numbering starts all over again at 1. If you want the numbers to continue from one group into the next group (known as 'through-numbering') you should use the **LineThrough** system variable instead. For example, write the following expression in the text component:

```
{LineThrough()}
```

As a result the row numbers in the subsequent group will continue from the numbers in the preceding group.

LineThrough System Variable

One of the tasks of lines numbering is through numbering in a group. The numbering starts with number 1. Through numbering of lines in a group is defined by the **LineThrough** system variable.



Line Number	Company	Address	Contact
F			
22	Familia Arquibaldo	Rua Orós, 92	Marketing Assistant
23	FISSA Fabrica Inter. Salchichas S.A.	C/ Moralzarzal, 86	Accounting Manager
24	Folies gourmandes	184, chaussée de Tournai	Assistant Sales Agent
25	Folk och fä HB	Åkergatan 24	Owner
26	France restauration	54, rue Royale	Marketing Manager
27	Franchi S.p.A.	Via Monte Bianco 34	Sales Representative
28	Frankenversand	Berliner Platz 43	Marketing Manager
29	Furia Bacalhau e Frutos do Mar	Jardim das rosas n. 32	Sales Manager
			Count: 8

In other words, when using the **LineThrough** system variable, all rows in the rendered list have an index number and start of printing a new group header does not affect the numbering (numbering does not reset to its initial state equal to 1).

GroupLine System Variable

Numbering of groups in the report generator is defined by the **GroupLine** system variable. Group numbering starts with 1. The picture below shows an example of a report with numbering of groups:

Simple Group

Company	Address	Phone	Contact
1			
Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative
			Count: 4
2			
Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
Blauer See Delikatessen	Forstweg 27	0621-08460	Sales Representative
Blondesddsl père et fils	24, place Kléber	88.60.15.31	Marketing Manager
Bólido Comidas preparadas	C/ Araquil, 67	(91) 555 22 82	Owner
Bon app'	12, rue des Bouchers	91.24.45.40	Owner
Bottom-Dollar Markets	23 Tsawassen Blvd.	(604) 555-4729	Accounting Manager
B's Beverages	Fauntleroy Circus	(171) 555-1212	Sales Representative
			Count: 7
3			
Cactus Comidas para llevar	Cerrito 333	(1) 135-5555	Sales Agent
Centro comercial Moctezuma	Sierras de Granada 9993	(5) 555-3392	Marketing Manager
Chop-suey Chinese	Hauptstr. 29	0452-076545	Owner
Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647	Sales Associate
Consolidated Holdings	Berkeley Gardens 12 Brewery	(171) 555-2282	Sales Representative
			Count: 5
4			
Die Wandernde Kuh	Adenauerallee 900	0711-020361	Sales Representative
DrachenblutDelikatessen	Walserweg 21	0241-039123	Order Administrator
Du monde entier	67, rue des Cinquante Otages	40.67.88.88	Owner
			Count: 3
5			
Eastern Connection	35 King George	(171) 555-0297	Sales Agent
Ernst Handel	Kirchgasse 6	7675-3425	Sales Manager
			Count: 2
6			
Familia Arquibaldo	Rua Orós, 92	(11) 555-9857	Marketing Assistant
FISSA Fabrica Inter. Salchichas S.A.	C/ Moralzarzal, 86	(91) 555 94 44	Accounting Manager
Folies gourmandes	184, chaussée de Tournai	20.16.10.16	Assistant Sales Agent
Folk och fä HB	Åkergatan 24	0695-34 67 21	Owner

A text component with the **GroupLine** system variable can be placed in the **GroupHeaderBand** band, and in the **GroupFooterBand** band.

Combining Groups and Master-Detail Reports

In **Master-Detail** reports it is possible to group both **Master** and **Detail** components. When creating a report, the report generator binds a group header and the **Data** band. The **GroupHeader** is placed on a page above the **Data** band, which outputs data rows. The **GroupHeader** band always refers to a specific **Data** band. Typically, the band is the first **Data** band, which is placed below the **GroupHeader** band. To render a report with the grouping, the **Data** band is required. The **GroupFooter** band is placed below the **Data** band. It is meant that very **Data** band, with what the **GroupHeader** band is bound. Each **GroupFooter** band, refers to a certain **GroupHeader** band. The **GroupFooter** band will not be output if there is no the **GroupHeader** band.

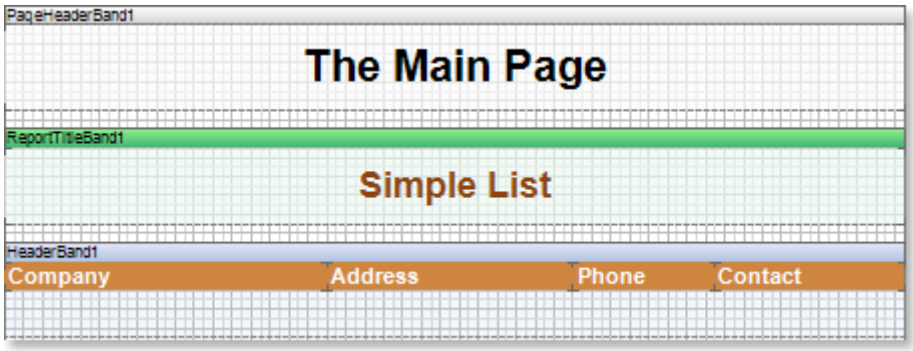
The picture above shows a combination of **GroupHeaderBand** and **GroupFooterBand** bands with Data bands in a Master-Detail report.

Report Bands

There are two report bands in Stimulsoft Reports: the **ReportTitleBand** and the **ReportsSummaryBand**. The ReportTitleBand is output in the beginning of a report and the ReportSummaryBand is output in the end of a report.
The number of **ReportTitleBand** bands and **ReportSummaryBand** bands on a page is unlimited.
The **ReportTitleBand** band and the **ReportSummaryBand** band can be output more than one time and can be used on each page.

ReportTitleBand

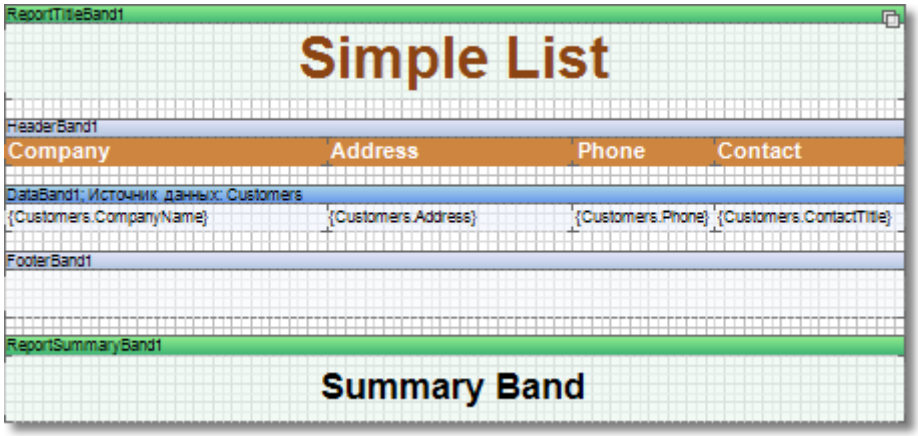
One of the ways to display the report header is the way of using the **ReportTitleBand** band. The report header will be output only once in the beginning of a report. The **ReportTitleBand** band is placed after the **PageHeaderBand** band, **and** before the **HeaderBand** band. The number of **ReportTitleBand** bands on a page is unlimited.



On the picture above shows how bands can be placed on a page. Here one can see top-down the **PageHeaderBand** band, the **ReportTitleBand** band, and the **HeaderBand** band.

ReportSummaryBand

A report summary can be output using the **ReportSummaryBand** band. The number of **ReportSummaryBand** bands placed in a report is unlimited. This band is output on each page as many times as there are pages.



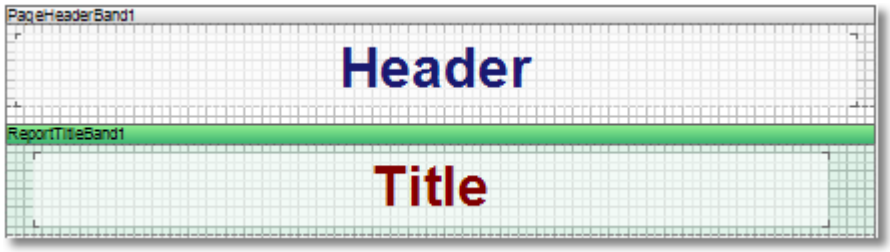
This band is used to output report summary.

On the picture above shows how bands can be placed on a page. Here one can see the top-down order of bands:

- The **ReportTitleBand** band;
- The **HeaderBand)** band;
- The **DataBand)** band;
- The **FooterBand)** band;
- The **ReportSummaryBand)** band.

ReportTitleBand Property

By default, the **PageHeaderBand** band is placed above the **ReportTitleBand** band:

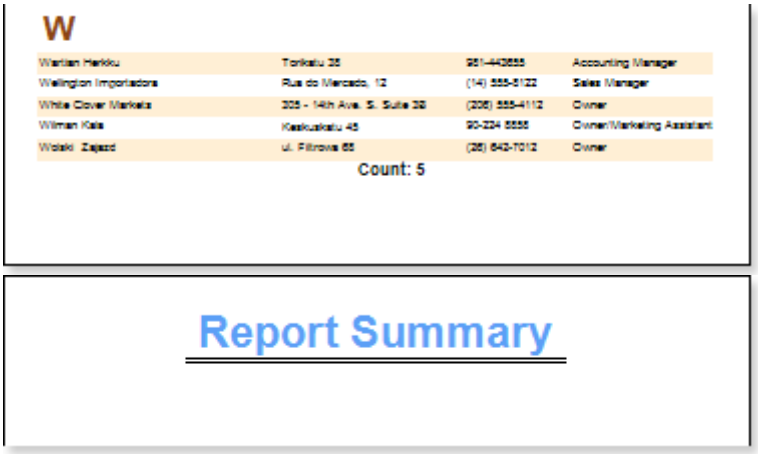


but it is also possible to output the **ReportTitleBand** band before the **PageHeaderBand** band:

By default this property is set to **false**. Set the **TitleBeforeHeader** property to **true** and the **ReportTitleBand** band will be output before the **PageHeaderBand** band.

KeepReportSummaryTogether Property

When printing, sometimes the last data row will be on one page and the report summary on the next one. The report will not look good.

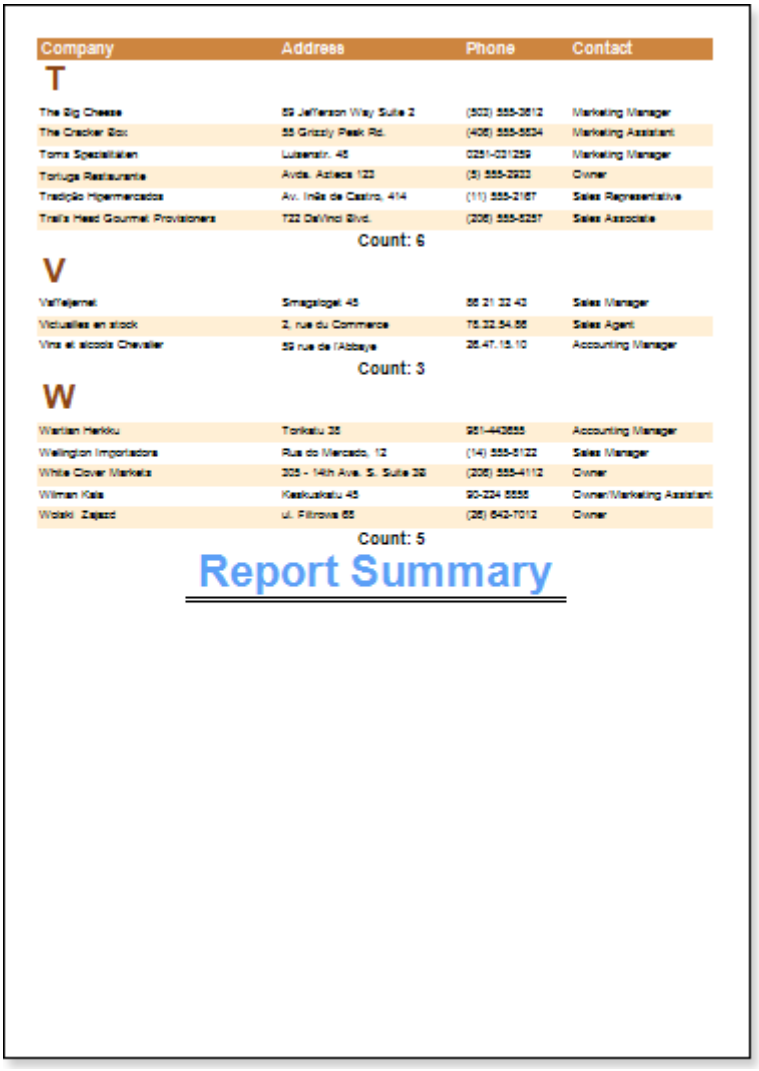


To avoid such unpleasant incidents the **ReportSummaryBand** has the **KeepReportSummaryTogether** property.
If the **KeepReportSummaryTogether** property is set to **true**, then minimum one data row will be printed with the report summary. Thus it is necessary to take into account that after the data row is transferred free space may remain on a fist page. Therefore, one should take this into account when working with this property.

The default value of the property is set to **true**.

PrintAtBottom Property

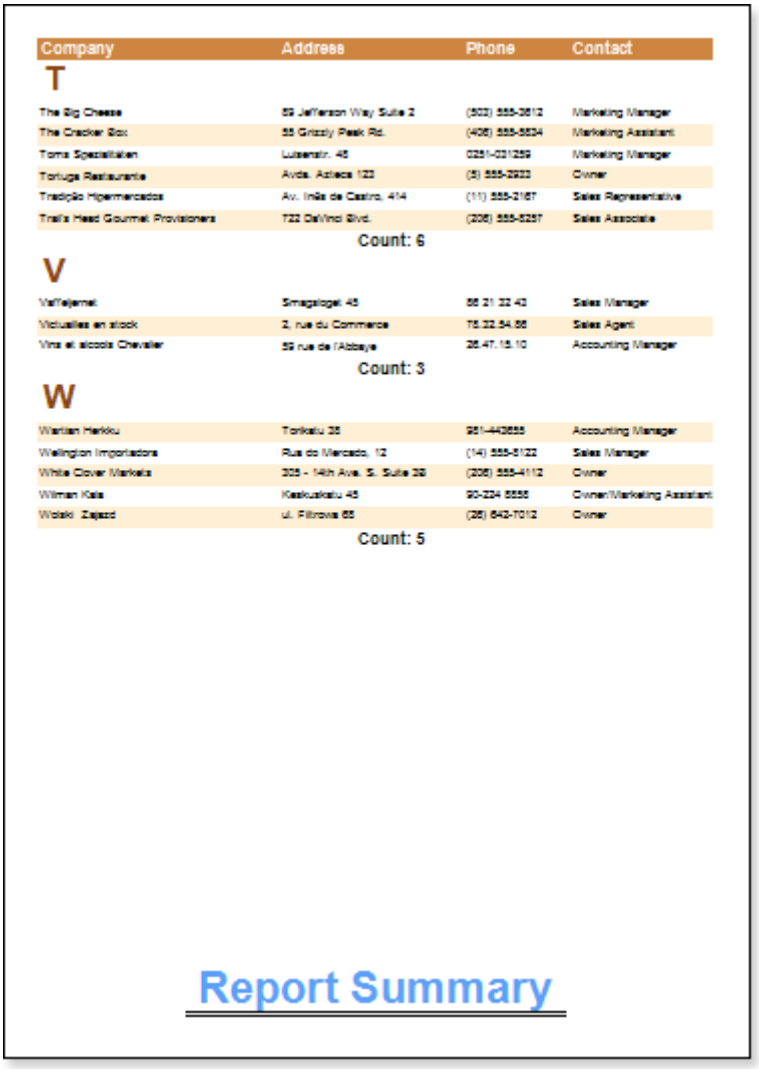
Suppose there is a report in which data covers only one-third of the last page. The report summary is displayed after the data.



Company	Address	Phone	Contact
T			
The Big Cheese	89 Jefferson Way Suite 2	(503) 555-3612	Marketing Manager
The Cracker Box	55 Grizzly Peak Rd.	(406) 555-5834	Marketing Assistant
Toms Spezialitäten	Luisenstr. 48	0251-031259	Marketing Manager
Tortuga Restaurante	Avda. Azteca 123	(5) 555-2933	Owner
Tradição Hipermercados	Av. Inês de Castro, 414	(11) 555-2167	Sales Representative
Trail's Head Gourmet Provisioners	722 DaVinci Blvd.	(206) 555-5257	Sales Associate
	Count: 6		
V			
Vaffeljernet	Smagsloget 45	86 21 32 43	Sales Manager
Victuailles en stock	2, rue du Commerce	75.32.54.56	Sales Agent
Vins et alcools Chevalier	59 rue de l'Abbaye	26.47.15.10	Accounting Manager
	Count: 3		
W			
Wartian Herkku	Torikatu 38	981-443655	Accounting Manager
Wellington Importadora	Rua do Mercado, 12	(14) 555-5122	Sales Manager
White Clover Markets	305 - 14th Ave. S. Suite 3B	(206) 555-4112	Owner
Wilman Kala	Keskuskatu 45	90-224 5555	Owner/Marketing Assistant
Wolski Zajazd	ul. Filtrowa 68	(26) 642-7012	Owner
	Count: 5		

Report Summary

But it is necessary that the report summary should be placed on the bottom of the page. The **ReportSummaryBand** has the **PrintAtBottom** property. By default, the property is set to **false**.
If the **PrintAtBottom** property is set to **true**, then summary will be output on the bottom of the page.

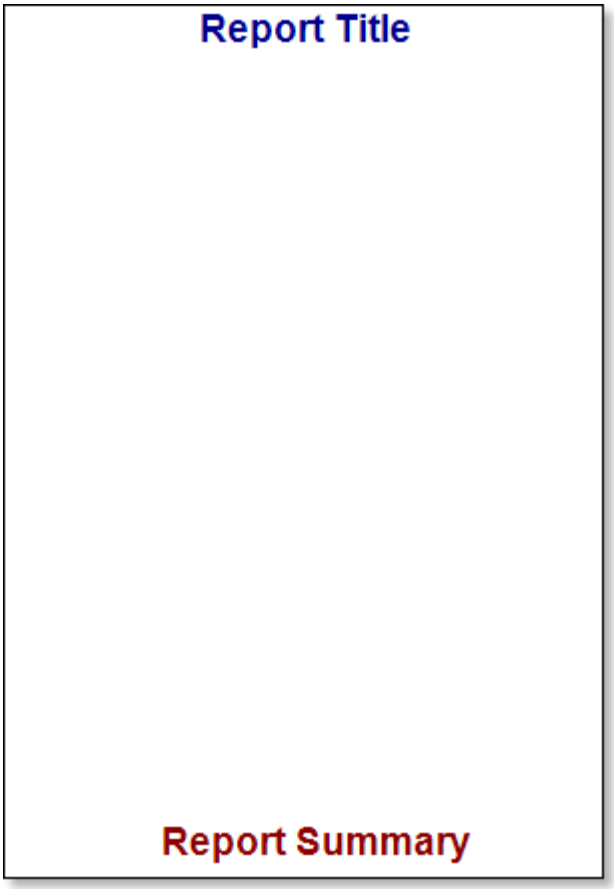


Company	Address	Phone	Contact
T			
The Big Cheese	89 Jefferson Way Suite 2	(503) 555-3612	Marketing Manager
The Cracker Box	55 Grizzly Peak Rd.	(406) 555-5834	Marketing Assistant
Toms Spezialitäten	Luisenstr. 48	0251-031259	Marketing Manager
Tortuga Restaurante	Avda. Azteca 123	(5) 555-2933	Owner
Tradição Hipermercados	Av. Inês de Castro, 414	(11) 555-2167	Sales Representative
Trail's Head Gourmet Provisioners	722 DaVinci Blvd.	(206) 555-5257	Sales Associate
	Count: 6		
V			
Vaffeljernet	Smagsloget 45	86 21 32 43	Sales Manager
Victuailles en stock	2, rue du Commerce	75.32.54.56	Sales Agent
Vins et alcools Chevalier	59 rue de l'Abbaye	26.47.15.10	Accounting Manager
	Count: 3		
W			
Wartian Herkku	Torikatu 38	981-443655	Accounting Manager
Wellington Importadora	Rua do Mercado, 12	(14) 555-5122	Sales Manager
White Clover Markets	305 - 14th Ave. S. Suite 3B	(206) 555-4112	Owner
Wilman Kala	Keskuskatu 45	90-224 5555	Owner/Marketing Assistant
Wolski Zajazd	ul. Filtrowa 68	(26) 642-7012	Owner
	Count: 5		

Report Summary

Print If Empty Property

There is a property in a report generator that allows you to display a report header and/or report footer when the DataBand is not on a page or data of a report. This is the **Print If Empty** property, which have both the **ReportTitleBand**, and the **ReportSummaryBand**.



By default, this property is enabled. If you disable this property for two bands, you get a blank page.

Note that in this example, in addition to the **Print If Empty** property, the **Print At Bottom** property of the **ReportSummaryBand** band is also set.

Page Bands

Page bands are printed at the top or bottom of a page. Usually they are used to output things like page numbering, copyright notices, company address and contact information etc. Stimulsoft Reports supports three types of page bands: **PageHeader**, **PageFooter**, and **EmptyData**.

PageHeader Band

The PageHeader band is used to output information such as page numbers, dates, and company information at the top of a page. The PageHeader band is output at the top of every page of the report. An unlimited number of PageHeader bands can be placed on a page.

Note: The number of PageHeader bands that can be placed on a page is effectively unlimited other than by available space.

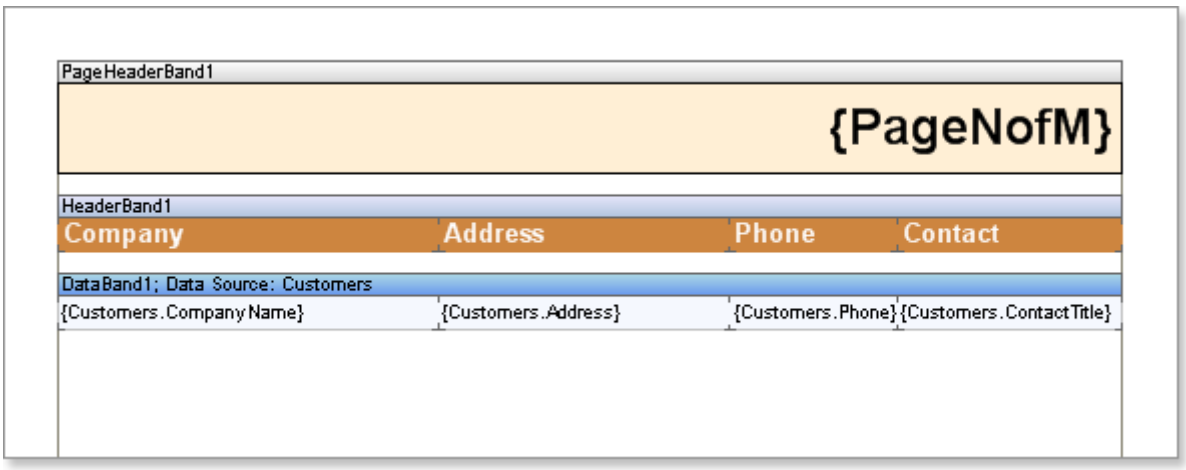
Example

Create a new report and drop three bands on a page: a PageHeader band for the current page number and number of pages in the report, a Data band to output data and a HeaderBand band to output data column headers. Drop a text component on the PageHeader band and enter the following expression in the Text Property Editor:

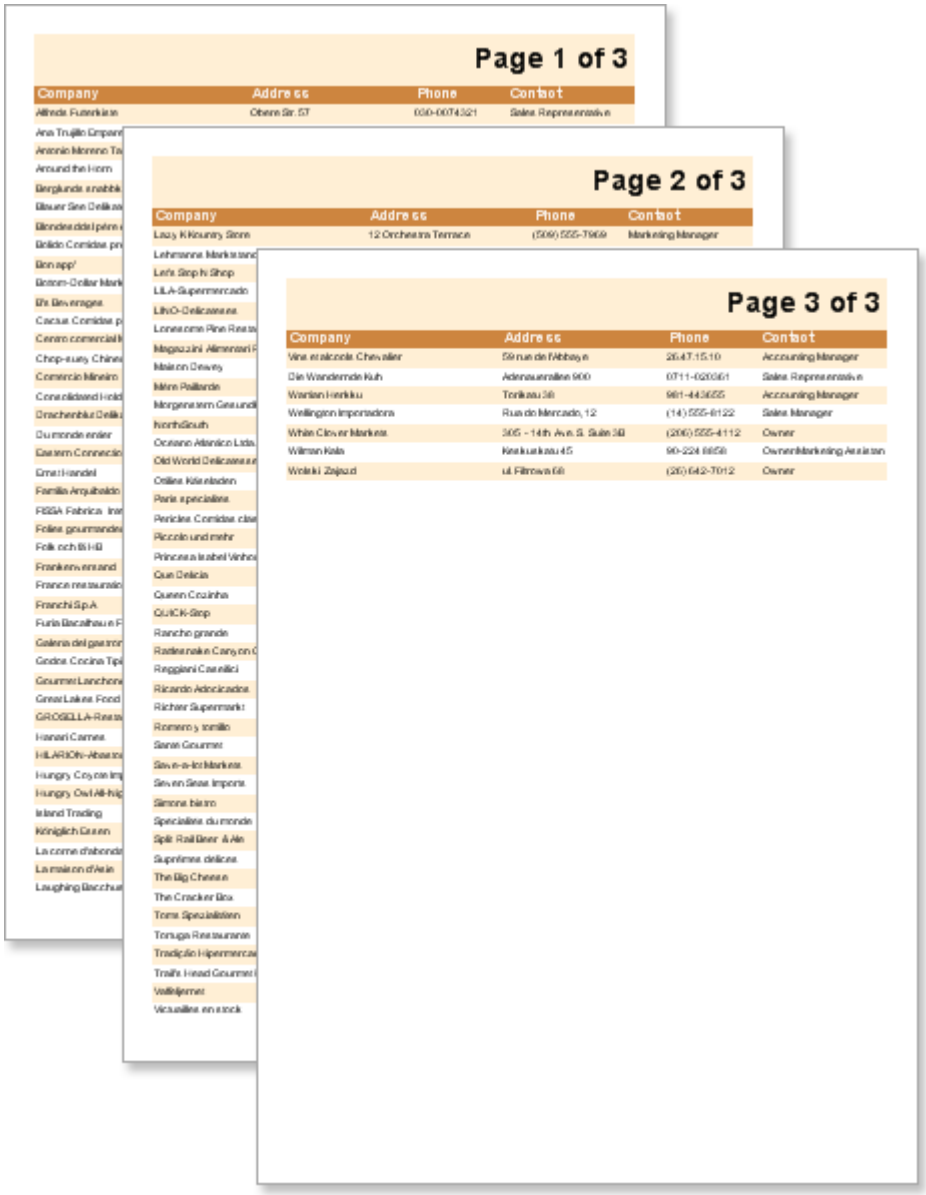
```
{PageNofM}
```

Note: If you prefer instead of typing the expression it is possible to select it from the System Variables in the Expression Editor.

The result should look something like this:



Now run the report, and you will see that the page number is printed at the top of each page.



PageFooter Band

The PageFooter band is used to output information such as page numbers, dates, and company information at the bottom of a page. The PageFooter band is output at the bottom of every page of the report. An unlimited number of PageFooter bands can be placed on a page.

Note: The number of PageFooter bands that can be placed on a page is effectively unlimited other than by available space.

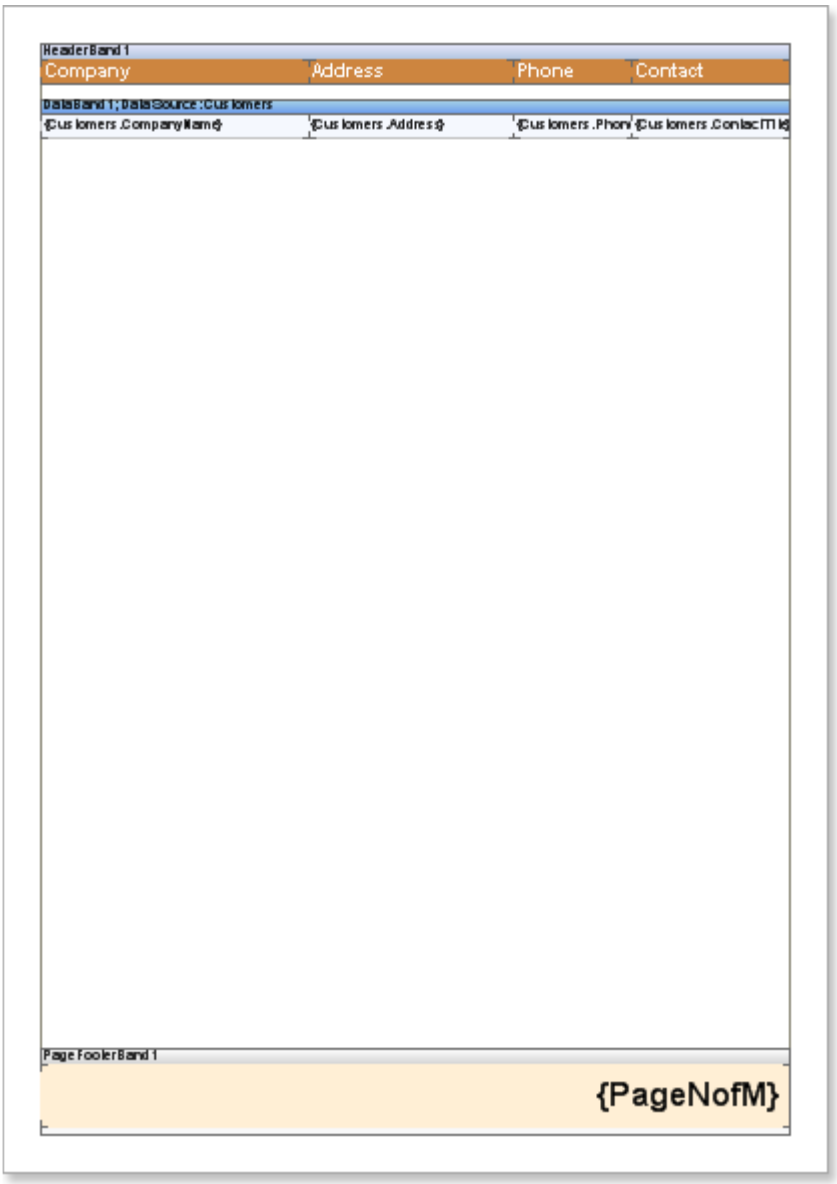
Example

Create a new report and drop three bands on a page: a PageFooter band for the current page number and number of pages in the report, a Data band to output data and a HeaderBand band to output data column headers. Drop a text component on the PageFooter band and enter the following expression in the Text Property Editor:

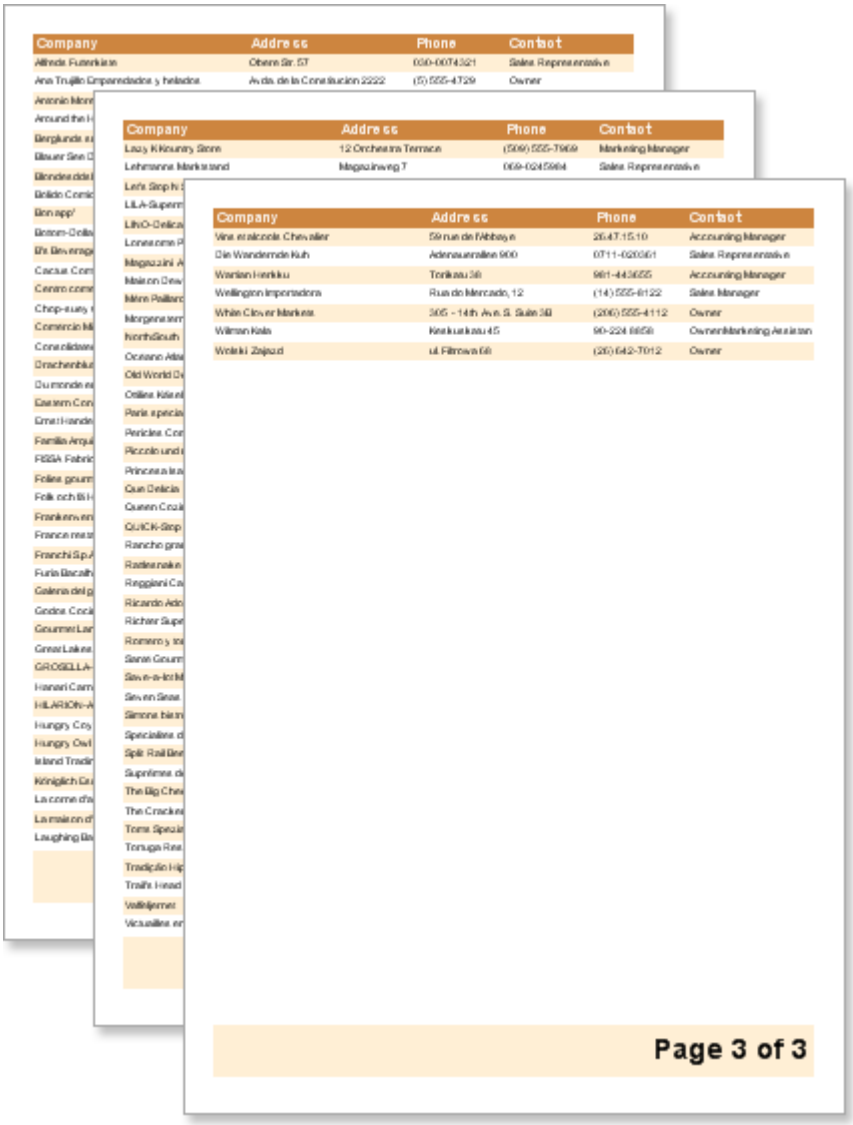
```
{PageNofM}
```

Note: If you prefer instead of typing the expression it is possible to select it from the System Variables in the Expression Editor.

The result should look something like this:



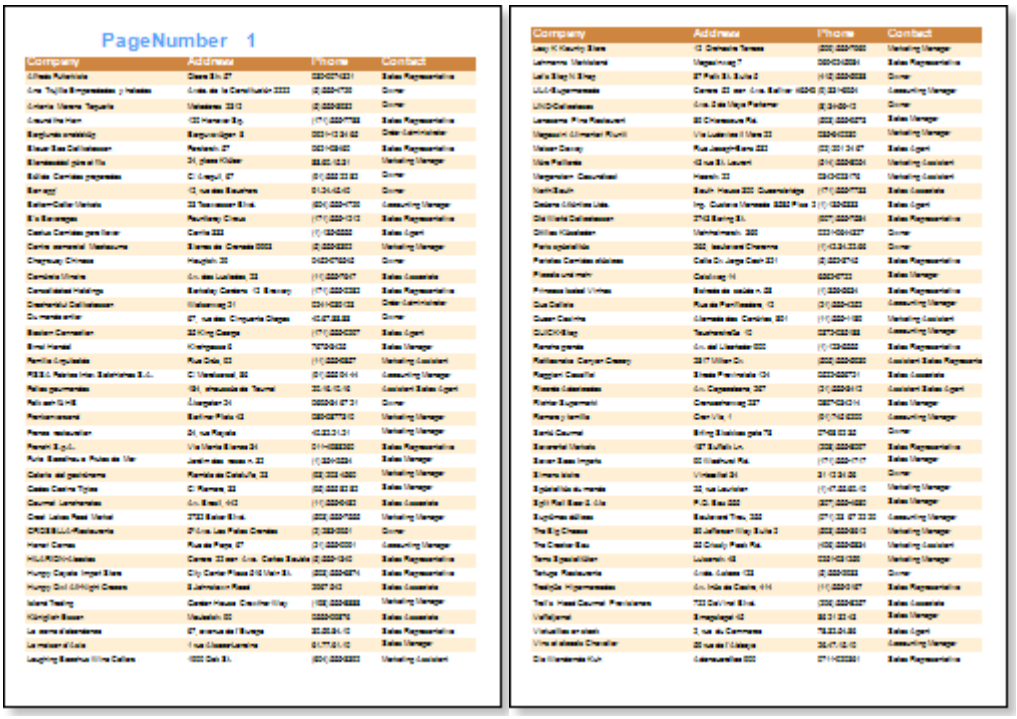
Now run the report, and you will see that the page number is printed at the bottom of each page.



PrintOnEvenOddPages Property

The **PrintOnEvenOddPages** property is used to print headers and footers on even/odd pages, for **PageHeaderBands** and **PageFooterBands**.

The picture above shows a sample of a report with the **PrintOnEvenOddPages** property of the **PageHeaderBand** set to **EvenPage**.



The picture above shows a sample of a report with the **PrintOnEvenOddPages** property of the **PageHeaderBand** set to **OddPage**.

Three values are available for this property:

- **Ignore**. Bands are printed on all pages;
- **PrintOnEvenPages**. Bands are printed on even pages;
- **PrintOnOddPage**. Bands are printed on odd pages.

Columns

Stimulsoft Reports has the ability to group data in columns. Data output in columns can improve the appearance of a report, and also allows more efficient use of page space.
Two types of columns are supported: columns on a Page and columns on a Data band. Columns on a Data band support two modes: Across Then Down and Down Then Across.
Stimulsoft Reports has a full set of tools to allow reports to be rendered with columns.

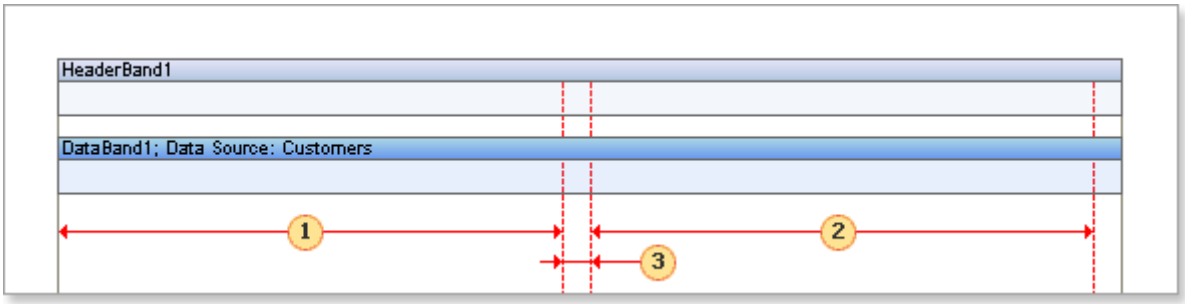
Columns on Page

It is possible to output data on a page in columns using the **Columns** property. By default this property is set to 0. Setting the value to 2 or more will cause the data to be output in columns. You will also need to set the **ColumnWidth** and **ColumnGaps** properties.

3. Columns	
Columns	2
Column Width	9,5
Column Gaps	0
Right to Left	False

The ColumnWidth property is used to set the column width and is applied to all columns which will be output on the page. The ColumnGaps property is used to set the space between the columns.

Important: Three page properties have to be set to output columns on a page. The **Columns** property is used to define the number of columns, the **ColumnWidth** property is used to set the width of each column, and the **ColumnGaps** property is used to set the space between the columns.



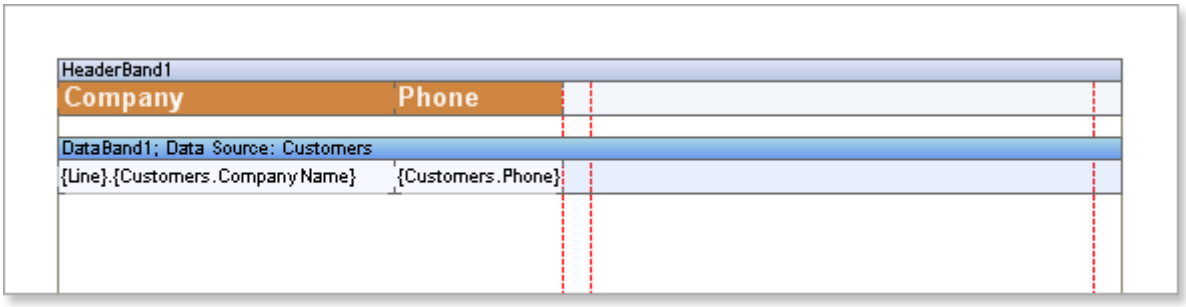
1	The first column width
2	The second column width
3	The space between columns

In columnar output mode the page is separated vertically and the report is logically output in the first column, then in the second etc.

Note: The number of columns on a page is unlimited.

Example:
Suppose that you need a report with two columns. Set the **Columns** property to 2 (this means that two columns will be output on each page). Set the **ColumnWidth** to a suitable width for one column and in the **ColumnGaps** property set the space between columns. Put two bands on a page: a Header band and a Data band. The data headers will be output on the Header band and data itself will be output on the Data band.

Note: Column borders are indicated by the red line.



Run the report. There are two columns on each page and all lines are numbered.

Company	Phone	Company	Phone
1.Alfreds Futterkiste	030-0074321	46.Let's Stop N Shop	(415) 555-5938
2.Ana Trujillo Emparedados y helados	(5) 555-4729	47.LILA-Supermercado	(9) 331-6954
3.Antonio Moreno Taquería	(5) 555-3932	48.LINO-Delicateses	(8) 34-56-12
4.Around the Horn	(171) 555-7788	49.Lonesome Pine Restaurant	(503) 555-9573
5.Berglunds snabbköp	0921-12 34 65	50.Magazzini Alimentari Riuniti	035-640230
6.Blauer See Delikatessen	0621-08460	51.Maison Dewey	(02) 201 24 67
7.Blondesddsl père et fils	88.60.15.31	52.Mère Paillarde	(514) 555-8054
8.Bólido Comidas preparadas	(91) 555 22 82	53.Morgenstern Gesundkost	0342-023176
9.Bon app'	91.24.45.40	54.North/South	(171) 555-7733
10.Bottom-Dollar Markets	(604) 555-4729	55.Océano Atlántico Ltda.	(1) 135-5333
11.B's Beverages	(171) 555-1212	56.Old World Delicatessen	(907) 555-7584
12.Cactus Comidas para llevar	(1) 135-5555	57.Ottilies Käseladen	0221-0644327
13.Centro comercial Moctezuma	(5) 555-3392	58.Paris spécialités	(1) 42.34.22.66
14.Chop-suey Chinese	0452-076545	59.Pericles Comidas clásicas	(5) 552-3745
15.Comércio Mineiro	(11) 555-7647	60.Piccolo und mehr	6562-9722
16.Consolidated Holdings	(171) 555-2282	61.Princesa Isabel Vinhos	(1) 356-5634
17.Die Wandernde Kuh	0711-020361	62.Que Delícia	(21) 555-4252
18.Drachenblut Delikatessen	0241-039123	63.Queen Cozinha	(11) 555-1189
19.Du monde entier	40.67.88.88	64.QUICK-Stop	0372-035188
20.Eastern Connection	(171) 555-0297	65.Rancho grande	(1) 123-5555

The columns are generated automatically - Stimulsoft Reports prints bands until there is no free space left on a page. Then, instead of creating a new page, a new column is added and data is output in a new column until again there is no free space. This is repeated until the required number of columns has been generated, at which point if there is still data to be output a new page is created and the process starts all over again.

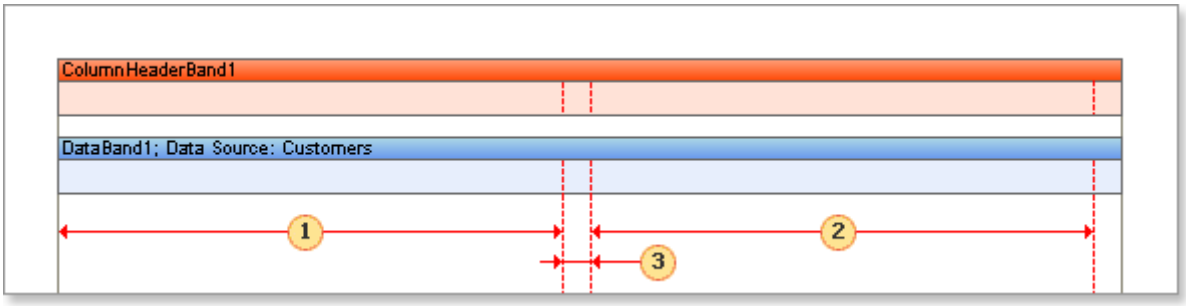
Columns on Data Band

Columns have one disadvantage, which is that there may be situations where the available data is sufficient to fill only one column leaving other columns empty and that part of a page will stay unused. To get around this problem it is possible to output columns using the Data band.

The **Columns** property of the Data band is used to enable the output of data in columns. Set this property to 2 or more to cause the data to be output in a columnar format.

It will also be necessary to set the **ColumnWidth** and **ColumnGaps** properties. The **ColumnWidth** property is used to set the column width and is applied to all columns on the Data band. The **ColumnGaps** property is used to set the space between two columns.

Note: Three data band properties have to be set to output columns on a band. The **Columns** property is used to define the number of columns, the **ColumnWidth** property is used to set the width of each column, and the **ColumnGaps** property is used to set the space between the columns.



1	The first column width
2	The second column width
3	The space between columns

Note: The number of columns on a Data band is unlimited.

There are two output modes for columns on the Data band: AcrossThenDown and DownThenAcross.

AcrossThenDown Mode

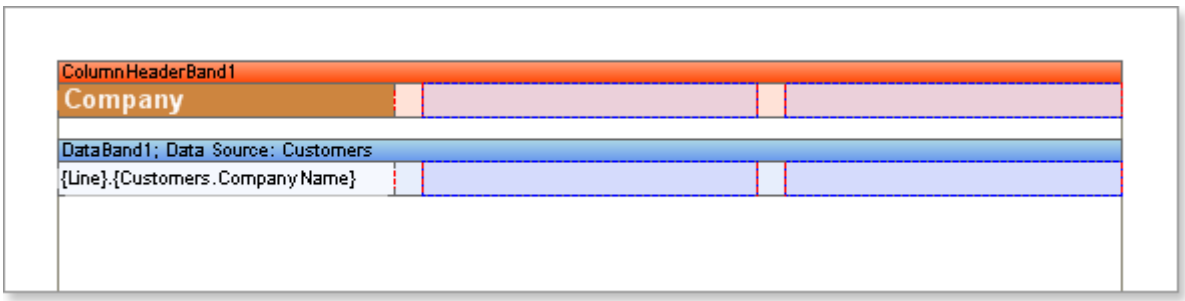
This mode is used to output strings logically from left to right on the Data band. Strings are output one string to one column. When all columns on the Data band have been generated a new Data band will be formed and again all strings in columns will be output. The data will take up as much space in the report as is necessary.

Note: The number of columns on a Data band is unlimited.

Example:
In this example we will build a report with three columns on the Data band. Put two bands on a page: A **ColumnHeader** band and a **Data** band. On the **Data** band set the Column property to 3 (this will create three columns). Set the column width using the **ColumnWidth** property, and the space between columns using the **ColumnGaps** property. Set the **ColumnDirection** property of the Data band to **AcrossThenDown** mode.

Place text components on the **ColumnHeader** band to represent the Column titles.

Note: Column edges are indicated with red vertical lines. All components which are placed on the first column will be automatically repeated in the other columns.



Now run the report. It is very easy to see the direction of data output.

Company	Company	Company
1.Alfreds Futterkiste	2.Ana Trujillo Emparedados y helados	3.Antonio Moreno Taquería
4.Around the Horn	5.Berglunds snabbköp	6.Blauer See Delikatessen
7.Blondesddsl père et fils	8.Bólido Comidas preparadas	9.Bon app'
10.Bottom-Dollar Markets	11.B's Beverages	12.Cactus Comidas para llevar
13.Centro comercial Moctezuma	14.Chop-suey Chinese	15.Comércio Mineiro
16.Consolidated Holdings	17.Die Wandernde Kuh	18.Drachenblut Delikatessen
19.Du monde entier	20.Eastern Connection	21.Ernst Handel
22.Familia Arquibaldo	23.FISSA Fabrica Inter. Salchichas S.A	24.Folies gourmandes
25.Folk och fä HB	26.France restauration	27.Franchi S.p.A.
28.Frankenversand	29.Furia Bacalhau e Frutos do Mar	30.Galería del gastrónomo
31.Godos Cocina Típica	32.Gourmet Lanchonetes	33.Great Lakes Food Market
34.GROSELLA-Restaurante	35.Hanari Carnes	36.HILARION-Abastos
37.Hungry Coyote Import Store	38.Hungry Owl All-Night Grocers	39.Island Trading
40.Königlich Essen	41.La corne d'abondance	42.La maison d'Asie
43.Laughing Bacchus Wine Cellars	44.Lazy K Kountry Store	45.Lehmanns Marktstand
46.Let's Stop N Shop	47.LILA-Supermercado	48.LINO-Delicateses
49.Lonesome Pine Restaurant	50.Magazzini Alimentari Riuniti	51.Maison Dewey
52.Mère Paillarde	53.Morgenstern Gesundkost	54.North/South
55.Océano Atlántico Ltda.	56.Old World Delicatessen	57.Ottilies Käseladen
58.Paris spécialités	59.Pericles Comidas clásicas	60.Piccolo und mehr
61.Princesa Isabel Vinhos	62.Que Delícia	63.Queen Cozinha

DownThenAcross Mode

AcrossThenDown mode has a weakness in that it is not always easy to read information on the page because the content is output from left to right and then down. It is often easier to read when columns are output using the DownThenAcross mode. In this mode the data is displayed in the first column and only when that is full is data shown in the second, and so on.

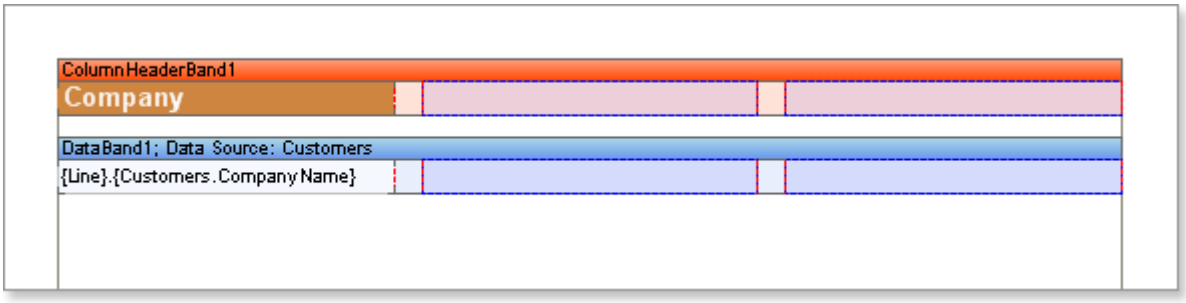
When using **DownThenAcross** mode, the report generator tries to distribute data rows evenly across the columns. When all data rows have distributed between the columns the first column is output. Because the data is evenly distributed the first column may not reach the bottom of a page - the data will take as much space on a page as is required, and it will be represented in convenient readable form (unlike the AcrossThenDown mode).

Note: The number of columns on a Data band is unlimited.

Example:
In this example we will build a report with columns in **DownThenAcross** mode. Put two bands on a page: A **ColumnHeader** band and a **Data** band. On the **Data** band set the Column property to 3 (this will create three columns). Set the column width using the **ColumnWidth** property, and the space between columns using the **ColumnGaps** property. Set the **ColumnDirection** property of the Data band to **DownThenAcross** mode.

Place text components on the **ColumnHeader** band to represent the Column titles.

Note: Column edges are indicated with red vertical lines. All components which are placed on the first column will be automatically repeated in the other columns.

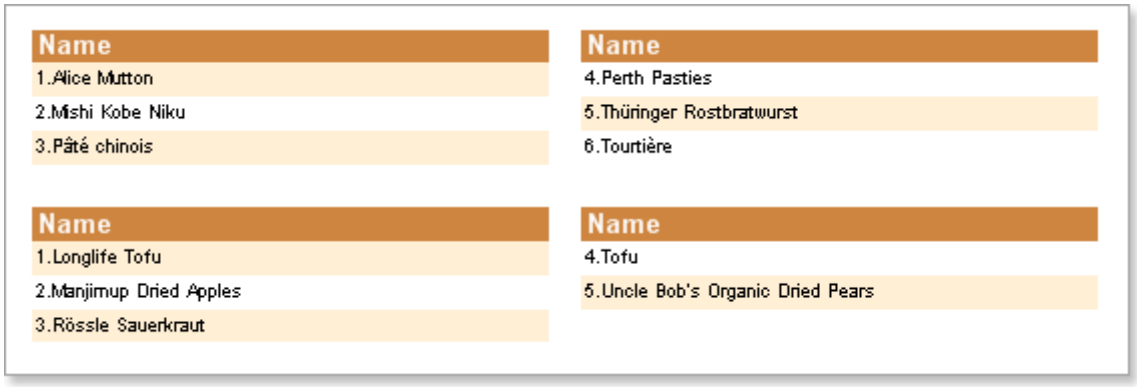


Now run the report. The report generator tried to distribute evenly all data rows between all three columns - using our sample data there are 31 rows in the first column, 31 in the second one, and 29 in the third. All information is readable top-down and from left to right.

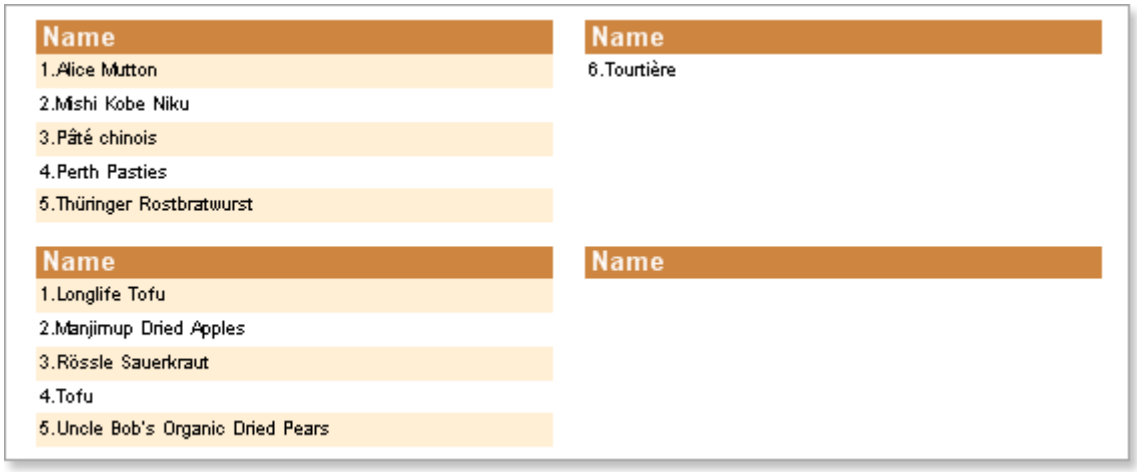
Company	Company	Company
1.Alfreds Futterkiste	32.Gourmet Lanchonetes	63.Queen Cozinha
2.Ana Trujillo Emparedados y helados	33.Great Lakes Food Market	64.QUICK-Stop
3.Antonio Moreno Taquería	34.GROSELLA-Restaurante	65.Rancho grande
4.Around the Horn	35.Hanari Carnes	66.Rattlesnake Canyon Grocery
5.Berglunds snabbköp	36.HILARION-Abastos	67.Reggiani Caseifici
6.Blauer See Delikatessen	37.Hungry Coyote Import Store	68.Ricardo Adocicados
7.Blondesddsl père et fils	38.Hungry Owl All-Night Grocers	69.Richter Supermarkt
8.Bólido Comidas preparadas	39.Island Trading	70.Romero y tomillo
9.Bon app'	40.Königlich Essen	71.Santé Gourmet
10.Bottom-Dollar Markets	41.La corne d'abondance	72.Save-a-lot Markets
11.B's Beverages	42.La maison d'Asie	73.Seven Seas Imports
12.Cactus Comidas para llevar	43.Laughing Bacchus Wine Cellars	74.Simons bistro
13.Centro comercial Moctezuma	44.Lazy K Kountry Store	75.Spécialités du monde
14.Chop-suey Chinese	45.Lehmanns Marktstand	76.Split Rail Beer & Ale
15.Comércio Mineiro	46.Let's Stop N Shop	77.Suprêmes délices
16.Consolidated Holdings	47.LILA-Supermercado	78.The Big Cheese
17.Die Wandernde Kuh	48.LINO-Delicateses	79.The Cracker Box
18.Drachenblut Delikatessen	49.Lonesome Pine Restaurant	80.Toms Spezialitäten
19.Du monde entier	50.Magazzini Alimentari Riuniti	81.Tortuga Restaurante
20.Eastern Connection	51.Maison Dewey	82.Tradição Hipermercados
21.Ernst Handel	52.Mère Paillarde	83.Trail's Head Gourmet Provisioners
22.Familia Arquibaldo	53.Morgenstern Gesundkost	84.Vaffeljernet
23.FISSA Fabrica Inter. Salchichas S.A	54.North/South	85.Victuailles en stock
24.Folies gourmandes	55.Océano Atlántico Ltda.	86.Vins et alcools Chevalier
25.Folk och fä HB	56.Old World Delicatessen	87.Wartian Herkku
26.France restauration	57.Ottilies Käseladen	88.Wellington Importadora
27.Franchi S.p.A.	58.Paris spécialités	89.White Clover Markets
28.Frankenversand	59.Pericles Comidas clásicas	90.Wilman Kala
29.Furia Bacalhau e Frutos do Mar	60.Piccolo und mehr	91.Wolski Zajazd
30.Galería del gastrónomo	61.Princesa Isabel Vinhos	
31.Godos Cocina Típica	62.Que Delícia	

Minimal Number of Lines in Column

When using the Down Then Across column mode a situation could arise where there are too few rows are available to output evenly in a report. In some cases may be necessary not to distribute data rows equally across all columns for better visualization.



The **MinRowsInColumn** property of the Data band can be used to define the minimum permitted number of rows in the first column. By default the value of this property is set to 0 which means that there is no minimum number of data rows. If the value of this property is higher than 0 then no less than specified number of rows will be output in the first column. In the example below the **MinRowsInColumn** property has been set to 5:



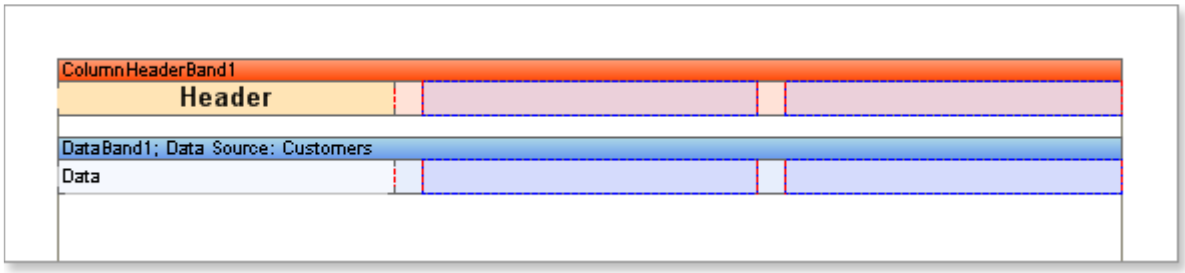
ColumnHeader Band

The Header band is normally used to output data headers, but there is also a special **ColumnHeader** band. The Header band is output once before the Data band and contains only one set of data. The **ColumnHeader** band is also output only once, but the components on this band are repeated above every column. It is used only for the columns positioned on the Data band.

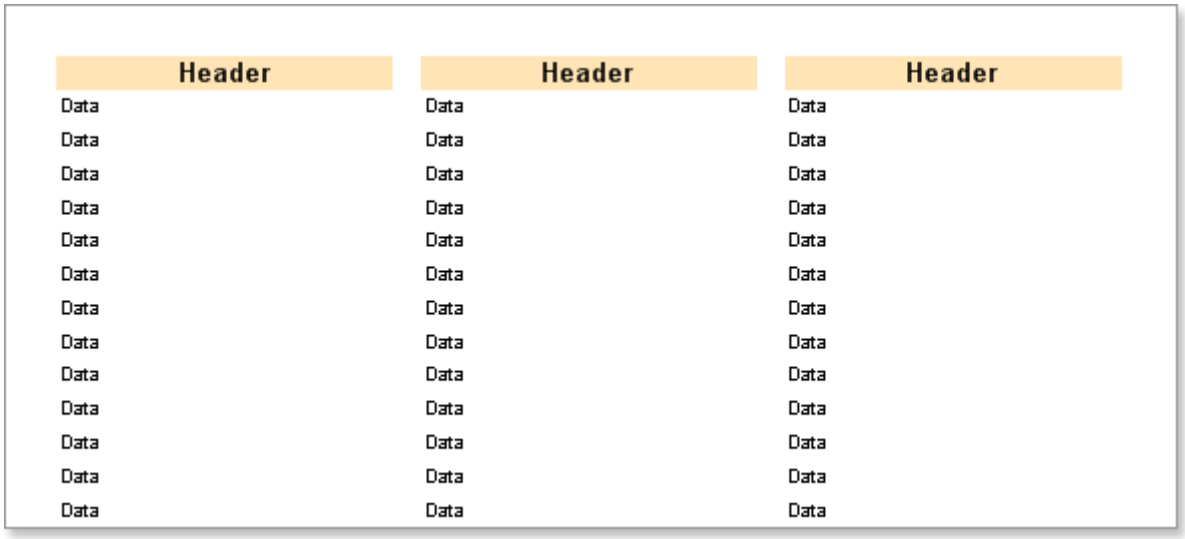
Notice: The **ColumnHeader** band is used for columns placed on the Data band. The Header band for page columns has the same functionality.

Example:
In this example we will build a report using a **ColumnHeader** band. Put two bands on a page: A **ColumnHeader** band and a **Data** band. On the Data band set the Column property to 3 (this will create three columns). Set the column width using the **ColumnWidth** property, and the space between columns using the **ColumnGaps** property. Set the **ColumnDirection** property of the Data band to **DownThenAcross** mode.

Place a text component on the **ColumnHeader** band with the text 'Header'. Then put a text component on the Data band with the text 'DATA'. Do not forget that the red lines are the column edges.

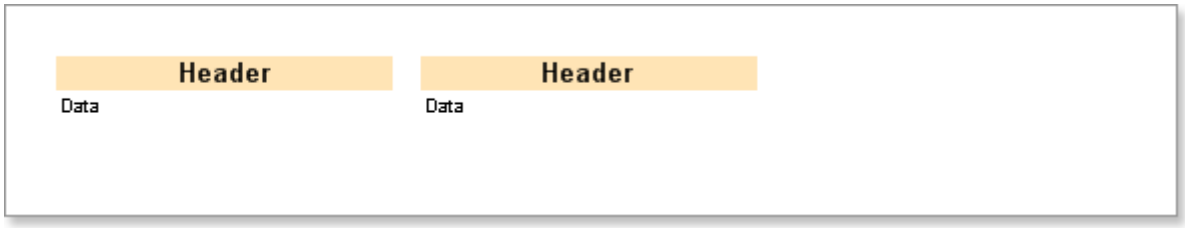


Now run the report and you will see that the word "Header" is shown over every column. You need only create a single column header and it will be automatically printed on each column.



PrintIfEmpty Property

Ugly output can result if the number of data rows is less than number of columns resulting in gaps on the page because the same number of column headers will be output as the number of columns. If there is data sufficient for two columns then only two headers will be output.



If you want to ensure that the same number of column headers are shown as the number of columns on a page without considering the number of strings available you can use the **PrintIfEmpty** property of the **ColumnHeader** band. If you set this property to true, then one header will be output for each column regardless of the amount of available data.

Important: It is important to remember that when the **MinRowsInColumn** property of the **DownThenAcross** mode is used, the report generator is not able to indicate the exact number of rows. Therefore, when using the **MinRowsInColumn** property, set the **PrintIfEmpty** property to true.

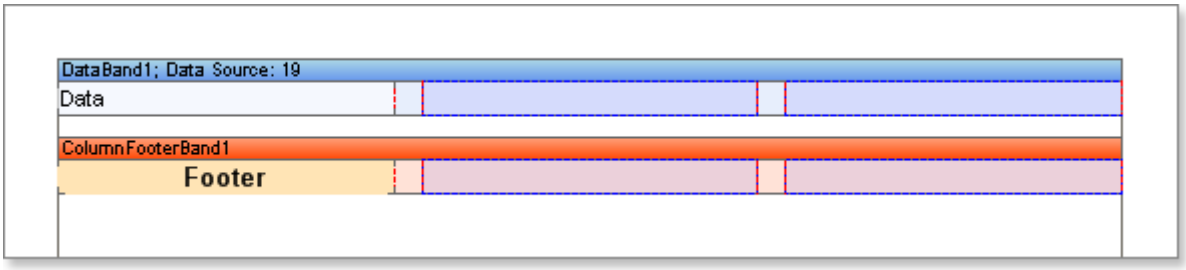
ColumnFooter Band

The Footer band is normally used to output data footers, but there is also a special **ColumnFooter** band. The Footer band is output once after the Data band and contains only one set of data. The **ColumnFooter** band is also output only once, but the components on this band are repeated beneath every column. It is used only for the columns positioned on the Data band.

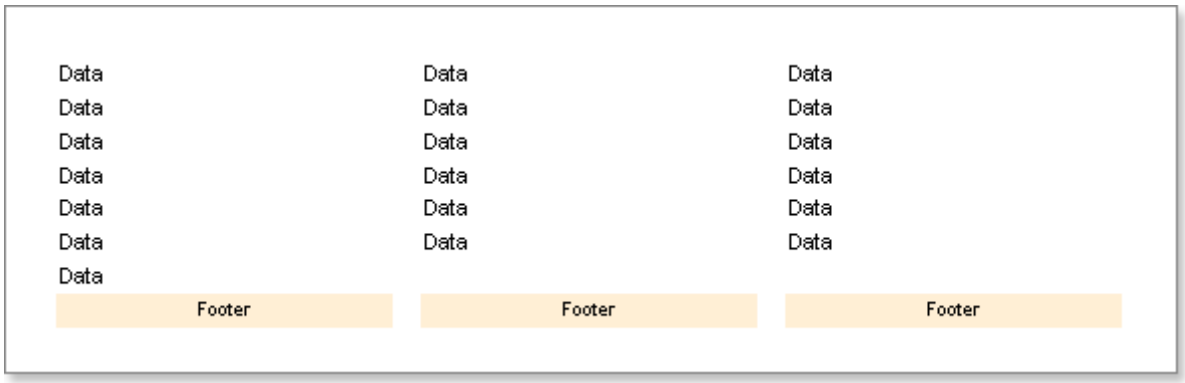
Notice: The **ColumnFooter** band is used for columns placed on the Data band. The Footer band for page columns has the same functionality.

Example:
In this example we will build a report using a **ColumnFooter** band. Put two bands on a page: A **ColumnFooter** band and a **Data** band. On the Data band set the Column property to 3 (this will create three columns). Set the column width using the **ColumnWidth** property, and the space between columns using the **ColumnGaps** property. Set the **ColumnDirection** property of the Data band to **DownThenAcross** mode.

Place a text component on the **ColumnFooter** band with the text 'Footer'. Then put a text component on the Data band with the text 'DATA'. . Do not forget that the red lines are the column edges.



Now run the report and you will see that the word "Footer" is shown under every column. You need only create a single column footer and it will be automatically printed on each column.



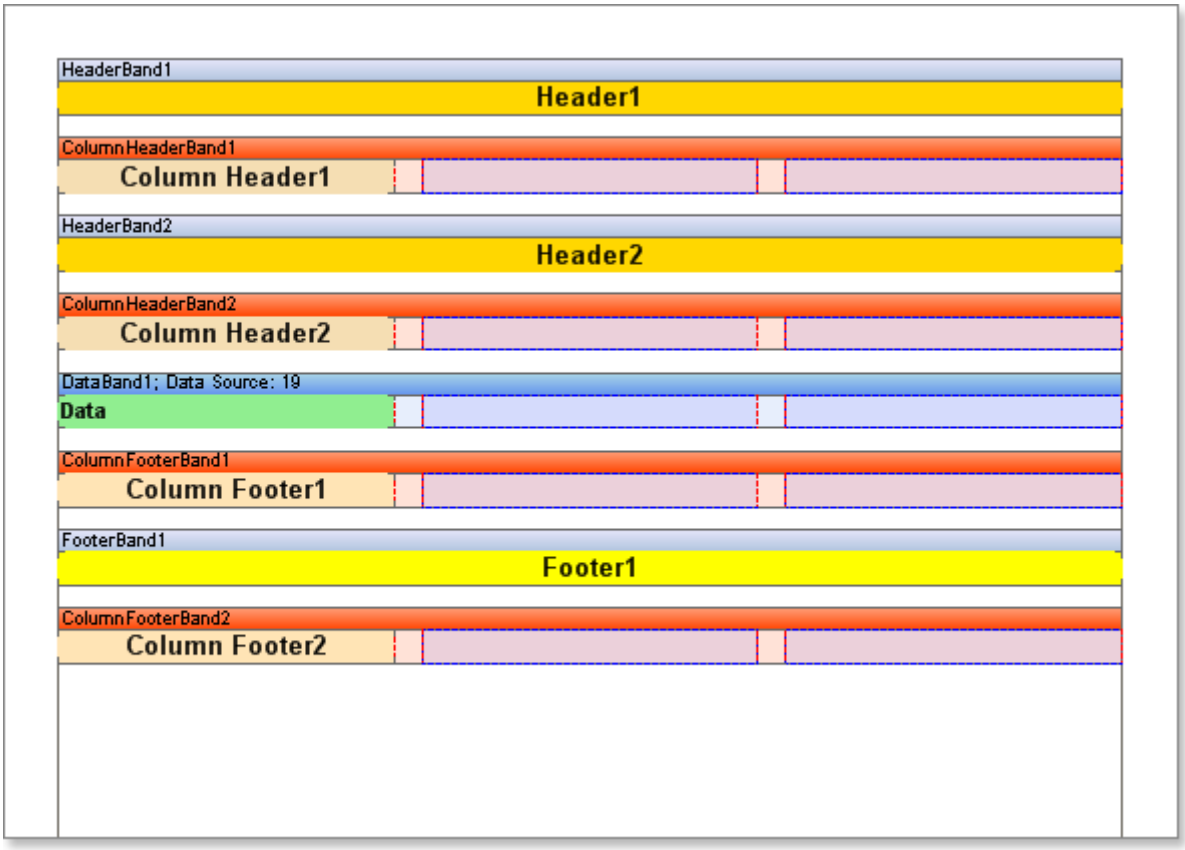
If you want to ensure that the same number of column footers are shown as the number of columns on a page without considering the number of strings available you can use the **PrintIfEmpty** property of the **ColumnFooter** band. If you set this property to true, then one footer will be output for each column regardless of the amount of available data.

Important: It is important to remember that when the **MinRowsInColumn** property of the **DownThenAcross** mode is used, the report generator is not able to indicate the exact number of rows. Therefore, when using the **MinRowsInColumn** property, set the **PrintIfEmpty** property to true.

Header and Footer Combinations

When outputting headers and footers for columns on a page it is very important to consider what the order in which the bands will be output on the page.

To see this in action create a report using multiple Header bands, Footer bands, **ColumnHeader** bands, **ColumnFooter** bands and just one **Data** band in a random order.



There are two modes used to output columns which will affect the output, and these will be reviewed in the following topics.

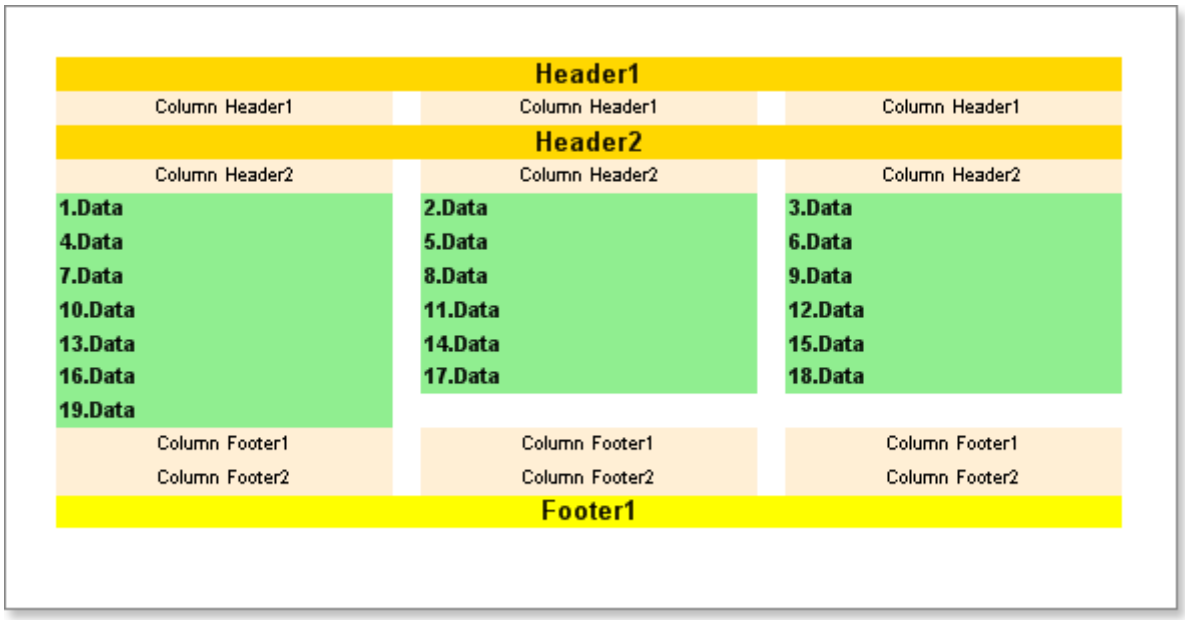
AcrossThenDown Column Mode

In **AcrossThenDown** mode all header bands are output in order of their position in the report template.

In our example as shown below the Header1 band will be output first, then the ColumnHeader1 band will be output three times over the every column. Next the Header2 band is output, and then ColumnHeader2 band over the every column.

Bands are output in order of their position on a page. This allows you to combine both types of header band to get the result you want.

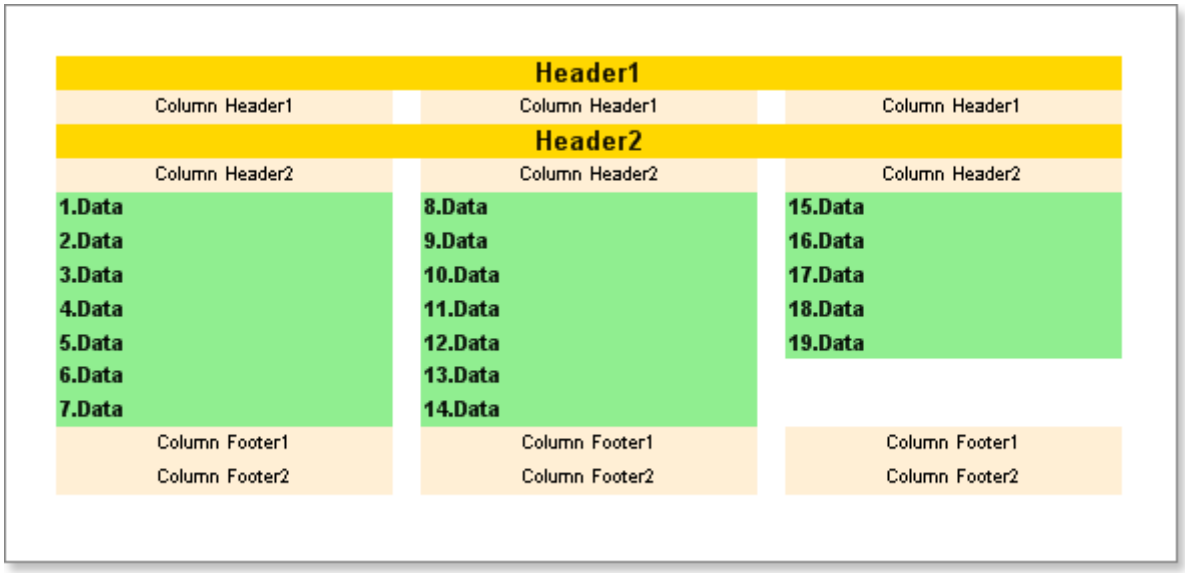
Footer bands are output differently. The ColumnFooters are output first. Then the Footer bands are output after all data rows. However, if the **PrintOnAllPages** property of the Footer bands is set to true, then the bands will be output in order of their position on a page. It is important to remember that if the **PrintOnAllPages** property of the Footer band is set to false, then this band will be output only after all data rows.



DownThenAcross Column Mode

This mode is similar to the Across Then Down mode.

All bands are output in the same order as they are placed on a page. However, if the **PrintOnAllPages** property of the Footer band is set to true, then all Footer bands are output in the same order as they are placed on page. If the **PrintOnAllPages** property of the Footer band is set to false, then only ColumnFooter bands are output and the Footer bands are ignored.



Pagination

Sometimes it is necessary to number pages. Page numbering is applied using system variables.
Page numbering is set by adding system variables into an expression. The code below shows how

```
{PageNumber}
```

```
{PageNofM}
```

```
{TotalPageCount}
```

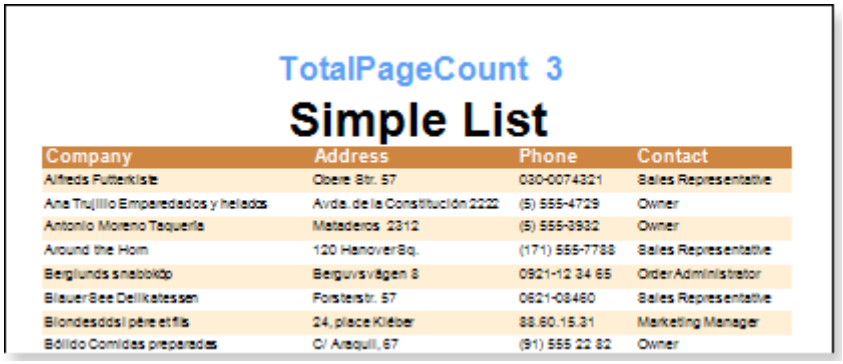
Page Number

Let see page numbering using the **PageNumber** system variable. When using this variable, the page number will be displayed on each page. Place where the page number is shown depends on which band is the text component, in expressions of what the system variable is used.

On the picture above the **PageNumber** system variable was used on the **PageHeaderBand** band. System variable can be used in any text component. The text component can be placed on any page band.

Total Page Count

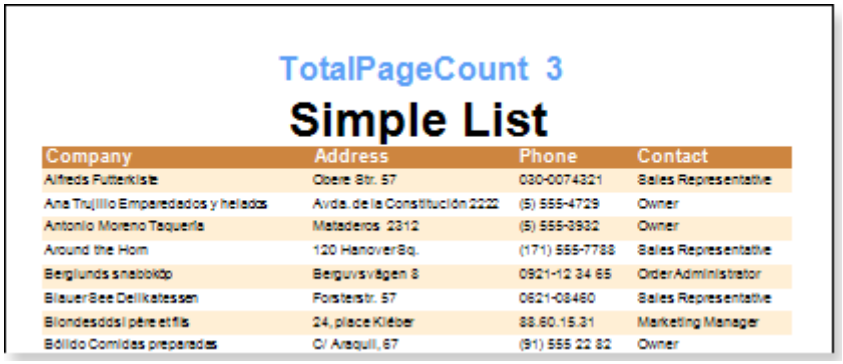
The **TotalPageCount** system variable is used to output the total number of pages.



On the picture above you can see how total number of pages is output. The **TotalPageCount** system variable is used with the **PageNumber** system variable. Usually it looks like this: **{PageNumber} Of {TotalPageCount}**. For example, **5 of 10**.

Page NofM

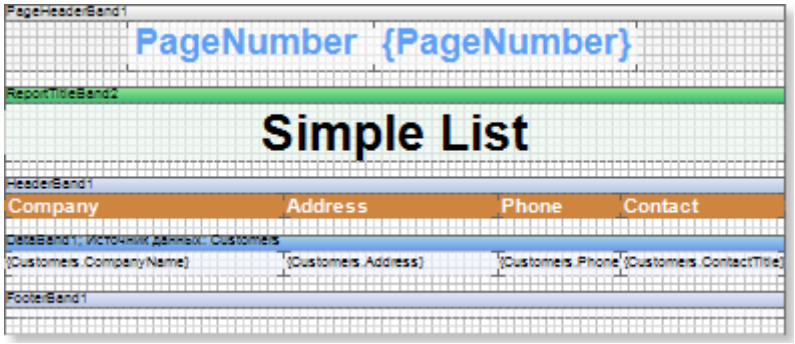
In order to show the page number of the total number of pages in the report generator the **PageNofM** system variable is used. This variable is a combination of system variables, such as the **PageNumber** and the **TotalPageCount**, i.e. it will print the page number on the total number of pages.



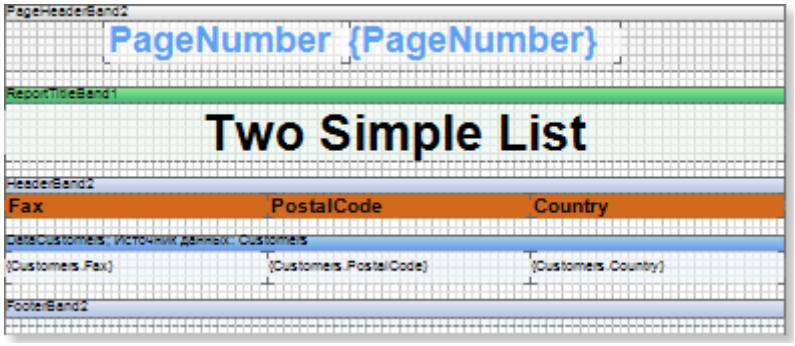
On the picture above the **"Page 1 of 3"** shows that the first page of three pages is available. The **PageNofM** depends on localization so it should be used very carefully.

ResetPageNumber Property

The numbering of the pages of the report begins with the number 1 and is defined consistently for each page built by the report.

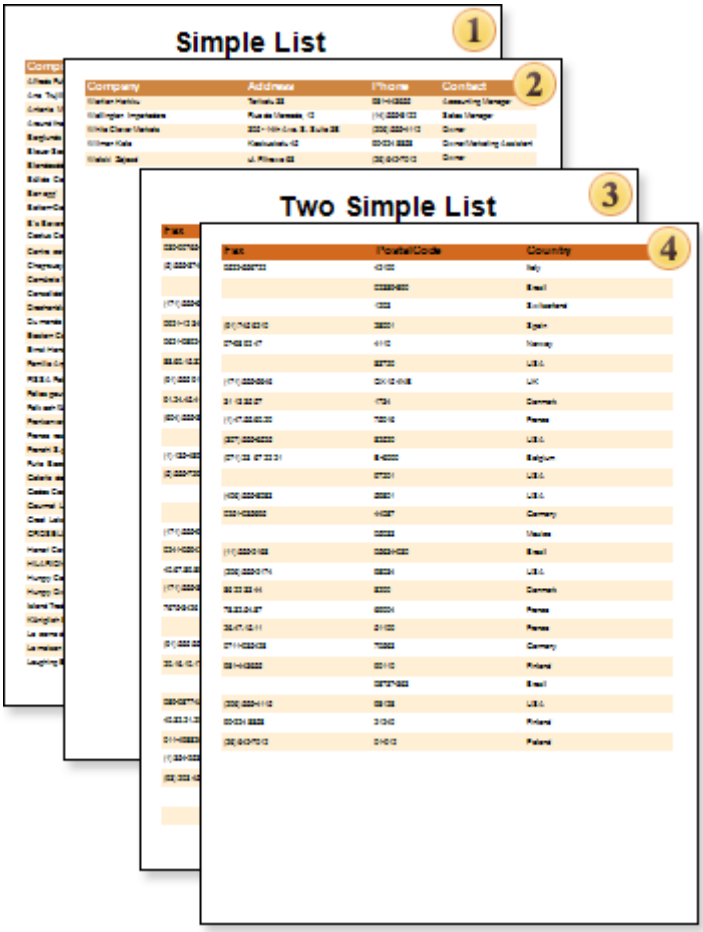


On the picture above the first page of a template is represented.

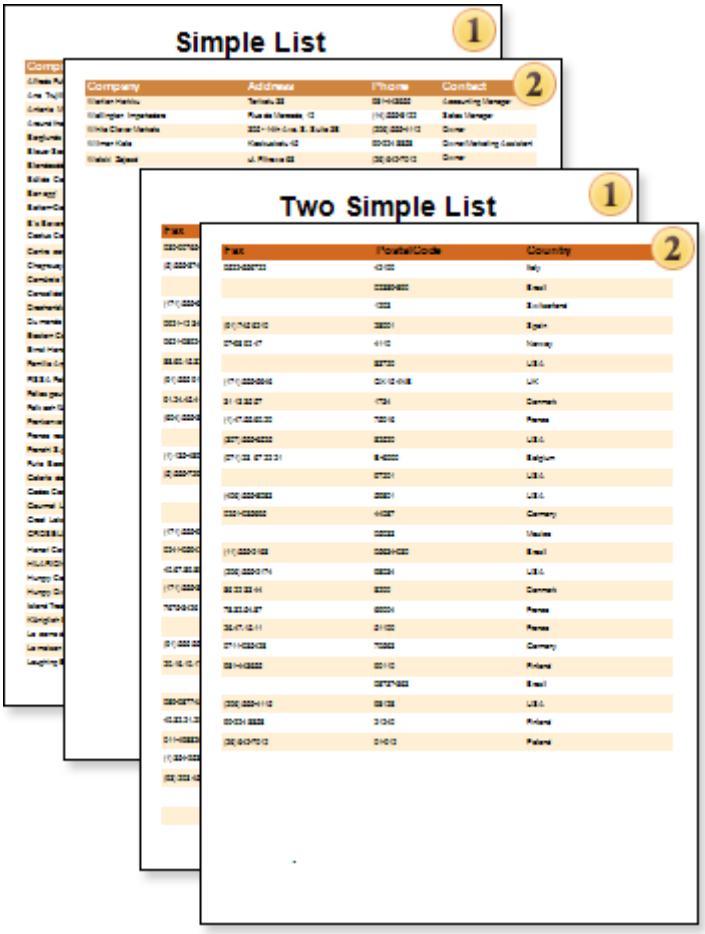


On the picture above the second page of a template is represented.

If, when report rendering, the **ResetPageNumber** is set to **false**, then numeration will look like on the picture below:



If the set the **ResetPageNumber** page property to **true**, then numeration for each page of a template will start from 1:



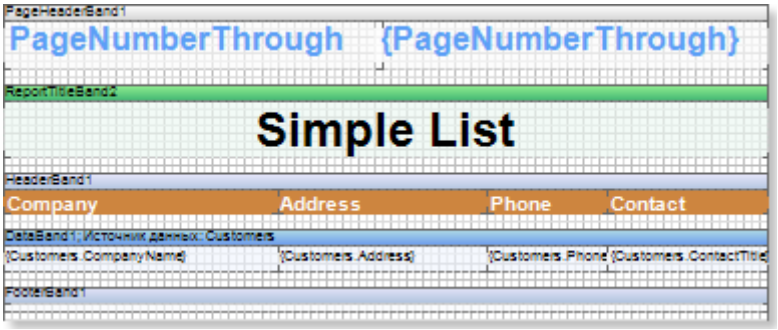
Information: The **ResetPageNumber** property works with the following variables: **PageNumber**, **PageNofM**, **TotalPageCount**. With system variables: **PageNumberThrough**, **PageNofMThrough**, **TotalPageCountThrough** - this property **does not work**.

By default the property is set to **false**.

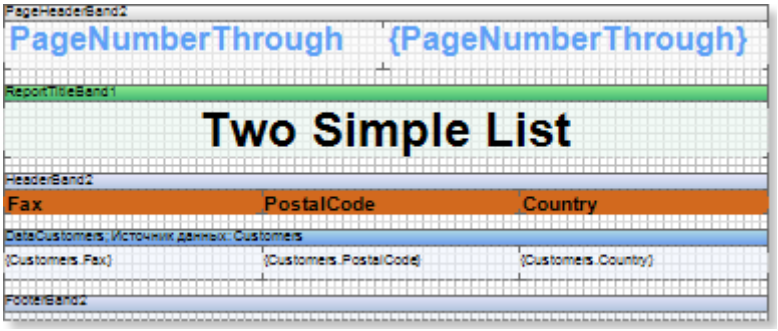
Sequentially Numbered Pages

Sequential numbering (numbering without taking into account the **ResetPageNumber** property) set the **SystemVariables**:

1. {PageNumberThrough} - **PageNumberThrough**, displays the page number;
2. {TotalPageCountThrough} - **TotalPageCountThrough**, displays the total number of pages of the rendered report;
3. {PageNofMThrough} - **PageNofM**, is a combination of **PageNumberThrough** and **TotalPageCountThrough**, and displays the page number in relation to the total number of pages in the rendered report..

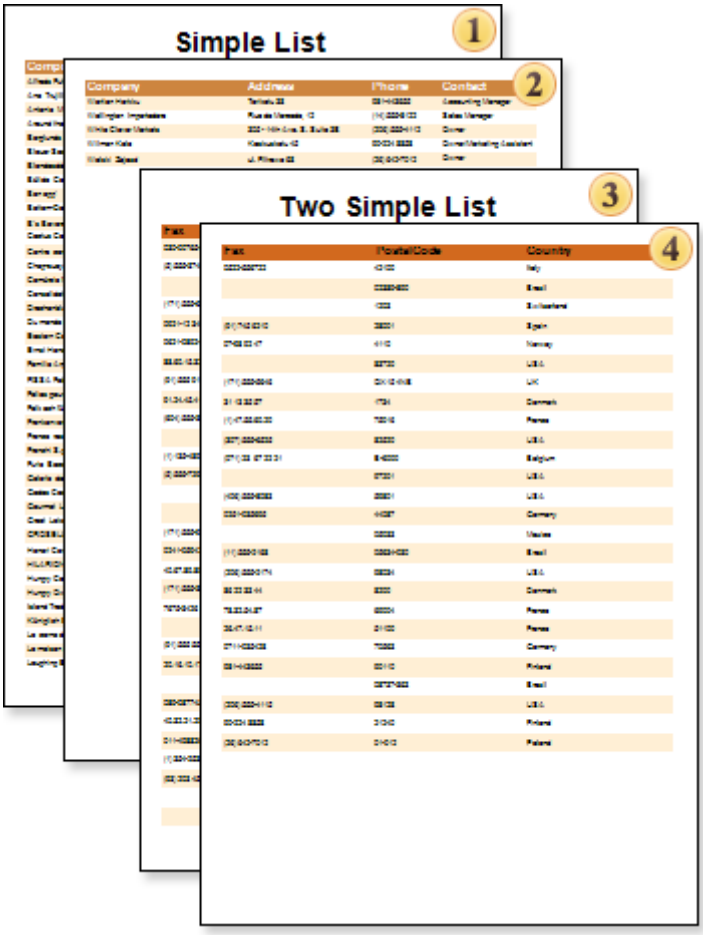


The picture above shows the first page of the report template.



The picture above shows the second page of the report template.

After rendering a report, even if the **ResetPageNumber** property of the page is set to **true**, the numbering of pages of the rendered report is to be consistent.



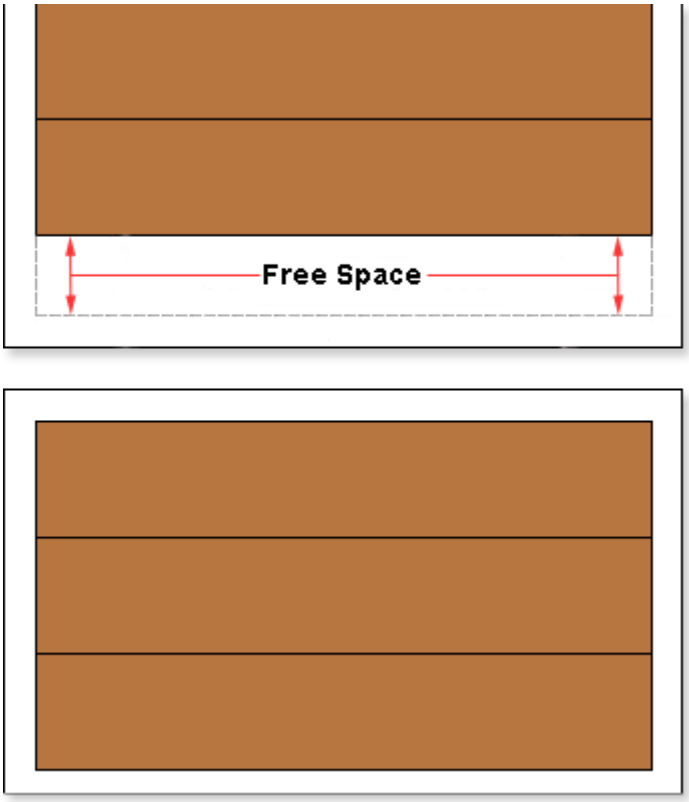
In other words, if the **ResetPageNumber** property is set to **true**, then, when using the system variables, mentioned above, the numeration will not be reset. So it will continue to be consistent for each page of the rendered report.

Breaking Component

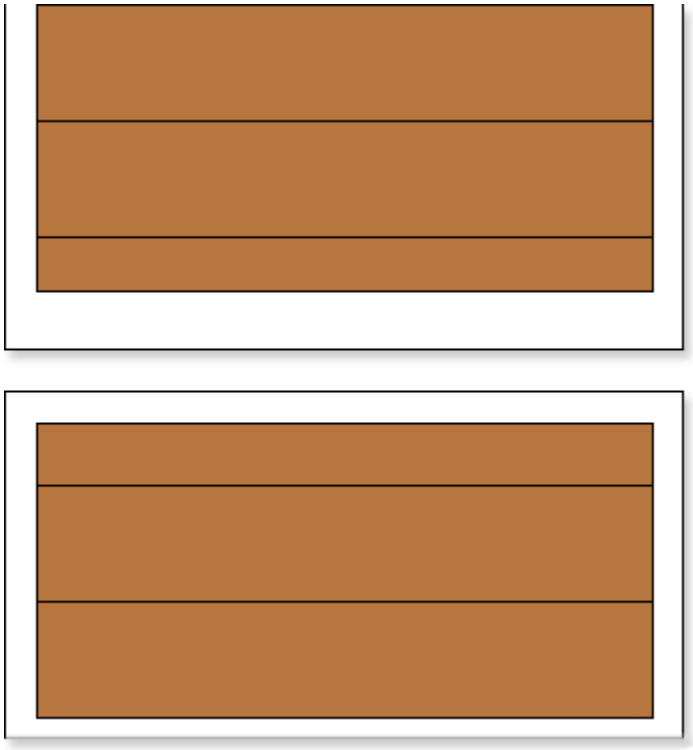
If, when rendering a report, the component will not fit the entire page, it will be carried to the next page. In addition, there are cases where the component has a size larger than the page size and cannot be output entirely on a page. In this case, you can use the **CanBreak** property. Components for which this property is set to **true**, can be "broken" with the Report Engine. I.e. the first part of a component will be printed on one page, and the second one on the next page. For example, a component of the **Text** has 10 lines, on the first page 7 lines will be output, and 3 lines on the next page.

Breaking Bands

How to use the **CanBreak** property of bands. The picture below shows two pages of a rendered report, which has 5 bands. The picture shows: the first and the second bands are output on the first page. The third band could not fit the bottom of the first page, so it was moved to the next page, along with the fourth and fifth bands.



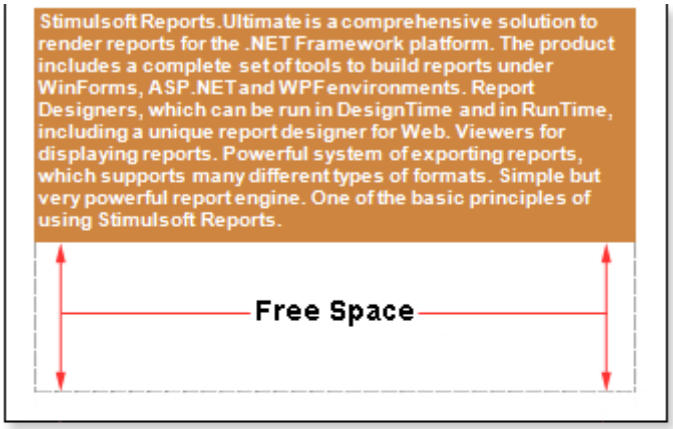
In this case, free space available remained on the first page of the report, because the band could not fit entirely and was moved to with the report engine to the next page. If to set the **CanBreak** property to **true**, then this will be "broken. The picture below shows how the of the third band is broken.

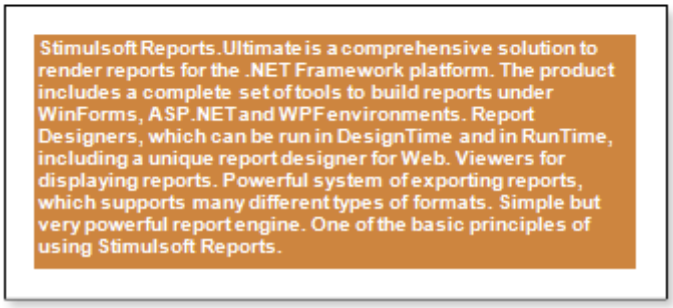


In this case we see that the third band could not fit, so it was broken: one part was left on the first page, and the second was moved to the next page, respectively. So all the space of the page was used. It should also take into account that the band may not fit within a single page. If the **CanBreak** is set to **false**, then it will be moved to the next page. If, on the next page, the band does not fit completely, it will be forcibly broken. You should know that special bands are displayed on the first page, and the remaining space of the page will be used to output the broken band. It is worth noting that the band may be output on more than one page. There are no limitations on the number of pages in which parts of the broken band can be output. By default, the **CanBreak** property is set to **false**.

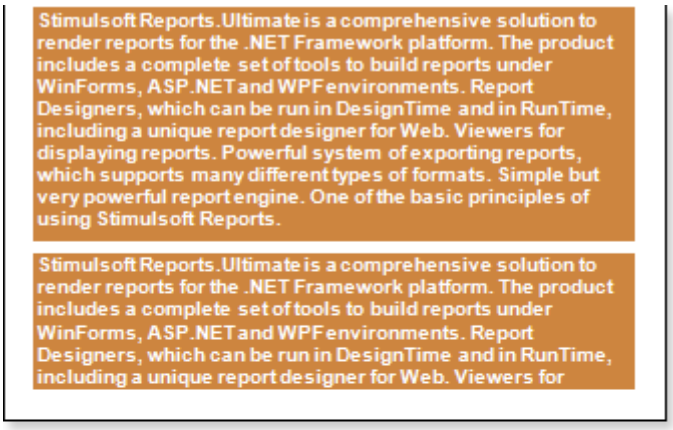
Breaking Text

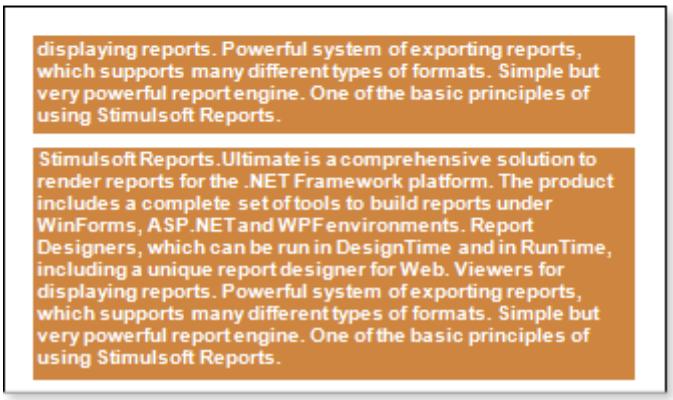
By default, the **CanBreak** property of the **Text** component is set to **false**. Such a Text component will not be broken if it is not enough space to print on one page, and would be moved to the next page.





As seen on the picture above, free space left at the bottom of the first page. To avoid this, set the **CanBreak** property to **true**. And then, a **Text** component is broken, for example, as shown on a picture below:



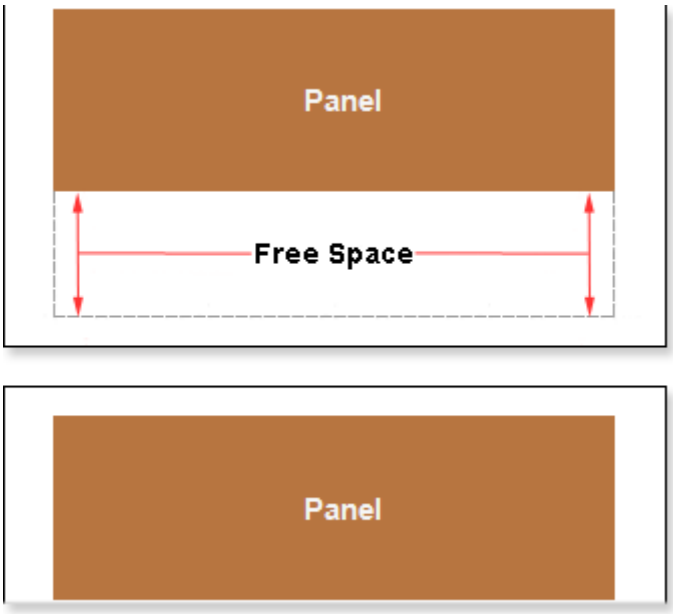


In this case, a **Text** component could not fit entirely on the bottom of a page, so it was broken. I.e. a part of the component remains on the same page, and another part was moved to the next one. Note that the text component is broken by row. Small amount of free space remains, as report generator must output the full height of a row and the text remains readable. Also note that the break of the text component will not work if the **CanBreak** property in a container, which has a text component, is set to **false**. Because the container would be moved to the next page completely. Accordingly, together with it, a text component will be transferred and the break will not work. So, if you

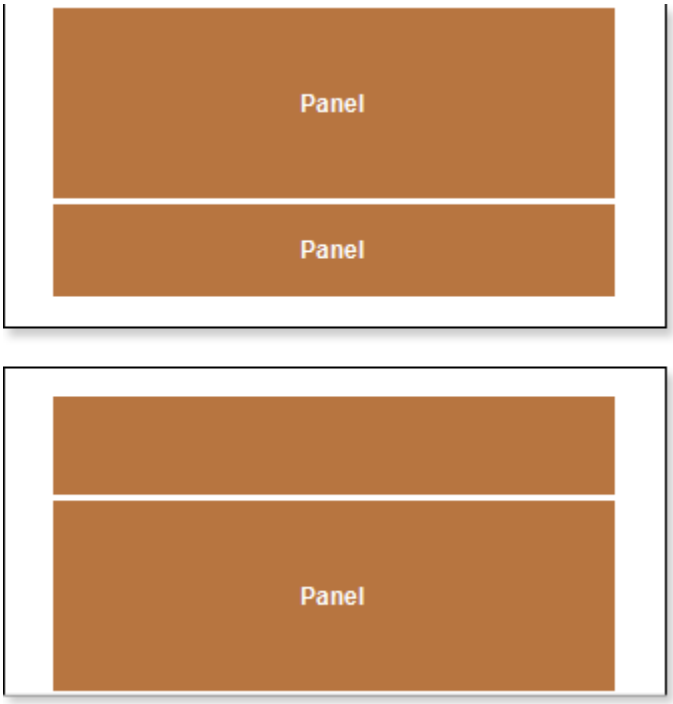
need a break, then set the **CanBreak** property to **true** for the Text component and container to what the text component is placed.

Breaking Panels

Sometimes, in a report template, where the **Panel** is used, all data cannot fit one page. If the **CanBreak** property is set to **false**, then a report, may look like on the picture below.



As shown in the picture above, the **Panel** was moved to another page, and free blank space remained on the previous page. If the **CanBreak** property is set to **true**, then the report may look like on the picture below:



As shown in the picture above, the **Panel** was broken, i.e. a part of it remained on the first page, and the other was moved to the next page. It should also take into account that the panel may not fit a single page. If to set the **CanBreak** property to **false**, then it will be moved to the next page. If on the next page the panel does not fit

completely, it will be forcibly broken. You should know that special bands are displayed on the first page, and the remaining space of the page will be used to output the broken panel. It is worth noting that the panel may be output on more than one page. There are no limitations on the number of pages in which parts of the broken panel can be output. By default, the **CanBreak** property is set to **false**.

Breaking RichText

By default, the **CanBreak** property of the **RichText** component is set to **false**. Such a text component will not be broken, if it is not enough space to print it on one page, and would be moved to the next page.

As you can see on the picture above, on the free space remained at the bottom of the first page. To avoid this, set the **CanBreak** property to **true**. And then, a component of the **RichText** will be broken (see the picture below):

As shown in the picture above, the **RichText** was broken, i.e. a part of it remained on the first page, and the other was moved to the next page. It should also take into account that the component may not fit a single page. You should know that the text component is broken rowwise. Also note that the breaking of the text component will not work if the **CanBreak** property of the band, in what the text component is placed, is set to **false**, because the band will be moved entirely to the next page. So the text component will be moved together with the band. So, if you need the text component to be broken, then values of **CanBreak** properties for the text component and the band should be set to **true**.

Breaking Images

In some cases the **Image** does not fit one page. So the image will be moved to the next page.

As you can see on the picture above, free space remained on the first page. To avoid this, set the **CanBreak** property to **true**. And then, the **Image** component will be broken, as seen on the picture below:

As shown in the picture above, the **Image** component was broken, i.e. a part of it remained on the first page, and the other was moved to the next page. Also note that the breaking of the **Image** component will not work if the **CanBreak** property of the band, in what the **Image** component is placed, is set to **false**, because the band will be moved entirely to the next page. So the **Image** component will be moved together with the band. So, if you need the Image to be broken, then values of**CanBreak** properties for the Image and the band should be set to **true**.

Auto-break

If a component of the report template is more than a page, then, when rendering a report, the component does not fit a page. If the **CanBreak** property is set to **true**, then the component will be broken into parts. If the **CanBreak** property is set to **false**, and the component is larger then the page of a report, the then report engine, tries to move it to the second page. If the data do not fit the second page, they will be forcibly broken, regardless of the value set for the **CanBreak** property and the availability of this property for the component of the report template. Moreover, when forced breaking, a blank page is output before the component. I.e. the first page of the report is empty, and each time data output begins with a new page. In this case, also all special bands are output on the page.

Breaking and Page Bands

There is no possibility for the **PageHeaderBand** and **PageFooterBand** to change the value of the **CanBreak** property, because it is always set to the one value. By default, the **CanBreak** property is set to **true**. This means that, when designing a report, if sizes of page bands is more than a page size, then bands will broken. You should also take into account the value of the property of the component, located on the band page. If the **CanBreak** property of a component placed on the band page is set to **false**, then in that case, there will be auto-break. If the **CanBreak** property of a component placed on the band page is set to **true**, then the break will be executed, depending on the type of a component (text, panel, picture, Rich Text).

Hierarchical Band

The **HierarchicalBand** is used to display report data as a tree. The picture below shows an example of a hierarchical report:

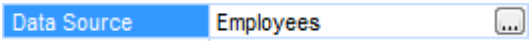
Hierarchical report

Employee	Phone	City	Region
Andrew Fuller	(206) 555-9482	Tacoma	WA
Steven Buchanan	(71) 555-4848	London	
Anne Dodsworth	(71) 555-4444	London	
Robert King	(71) 555-5598	London	
Michael Suyama	(71) 555-7773	London	
Laura Callahan	(206) 555-1189	Seattle	WA
Margaret Peacock	(206) 555-8122	Redmond	WA
Nancy Davolio	(206) 555-9857	Seattle	WA
Janet Leverling	(206) 555-3412	Kirkland	WA

Data Output

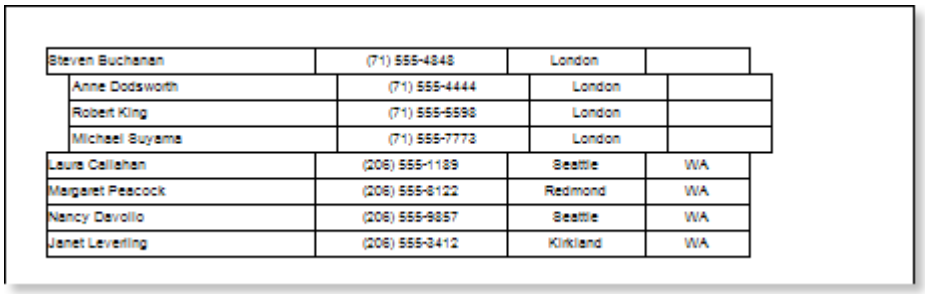
In order to obtain a structured list in a report as a tree, you must follow these steps:

1. Specify the **DataSource** for the **HierarchicalBand** using, for example, the **DataSource** property:



2. Set the **KeyDataColumn**, i.e. select the data column by what an identification number of data rows will be assigned. For example, a **EmployeeID** data column;
3. Set the **MasterKeyDataColumn**, i.e. select the data column by which a reference to the primary table key of the parent entry will be specified. For example, a **ReportsTo** data column;
4. Set the **Indent**, i.e. specify the indent distance of the child entry relative to the parent entry. For example, the **Indent** value will be equal to **20** units of a report (centimeters, inches, one hundredth inches, pixels);
5. Set the **ParentValue**, i.e. specify an entry that will be a parent for all rows. For example, set the **ParentValue** property to **2**.

The picture below shows an example of a rendered hierarchical report:



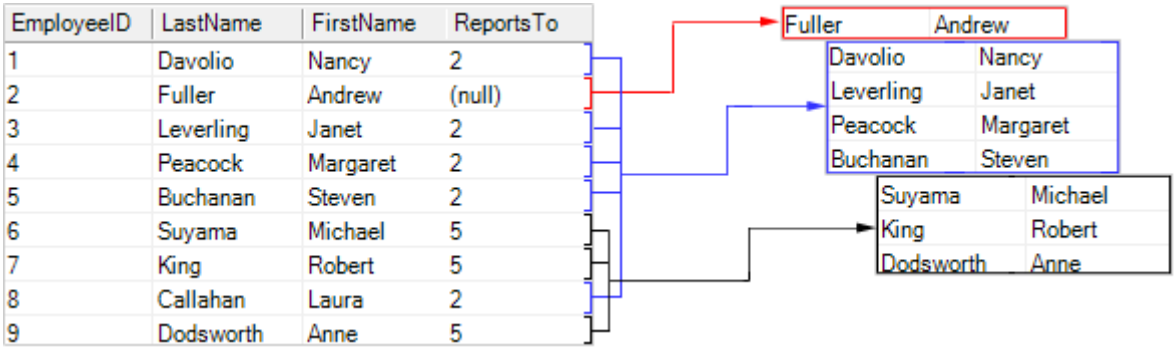
Steven Buchanan	(71) 555-4848	London	
Anne Dodsworth	(71) 555-4444	London	
Robert King	(71) 555-5598	London	
Michael Suyama	(71) 555-7773	London	
Laura Callahan	(206) 555-1189	Seattle	WA
Margaret Peacock	(206) 555-8122	Redmond	WA
Nancy Davolio	(206) 555-9857	Seattle	WA
Janet Leverling	(206) 555-3412	Kirkland	WA

KeyDataColumn Property

The **HierarchicalBand** has the **KeyDataColumn** property. This property is required for filling. If the **KeyDataColumn** is not specified, the report generator will not be able to render a report. The value of this property can be any data column from the selected **HierarchicalBand** of the data source, which entries will be keys for creating a report. For example, if the **Employees** data source is specified to the **HierarchicalBand**, then the value of the **KeyDataColumn** property is the **EmployeesID** data column, because the entry of this column is the key and contains unique codes of employees.

MasterKeyDataColumn Property

In order to represent an hierarchy in the report, you must specify the value of the **MasterKeyDataColumn** property. This property is required for filling. If the value of the **MasterKeyDataColumn** is not specified, the report generator cannot determine the hierarchy in the report. The value of this property will be a data column from the selected **HierarchicalBand** of the data source, which entries are the master key for creating an hierarchy in the report. For example, if the **Employees** data source is specified for the **HierarchicalBand**, then the **MasterKeyDataColumn** property is the **ReportsTo** column data. The values of this data column are used to specify to what this element in the table is subordinated. Usually, this column indicates the keys in the data column, which is a value of the **KeyDataColumn** property. The picture below shows the scheme of an hierarchy of the **ReportsTo** data column:



ParentValue Property

The **ParentValue** property is used to identify entries which will be the parent rows for the remaining rows in a report. Parent rows are rows which are placed on the top level of hierarchy and in which all other elements are included. The report must have at least one parent line, if the parent line is missing, the report cannot be rendered. The **ParentValue** property can take any value, which is an entry in the data column, which is listed as the **MasterKeyDataColumn**. For example, if the **MasterKeyDataColumn** property is the **ReportsTo** data column, then the value of the **ParentValue** property will be entries in this column. The picture below shows an example of the **EmployeeID**, **LastName**, **City**, **Region**, **ReportsTo** data columns of the **Employees** data source:

EmployeeID	LastName	City	Region	ReportsTo
1	Davolio	Seattle	WA	2
2	Fuller	Tacoma	WA	(null)
3	Leverling	Kirkland	WA	2
4	Peacock	Redmond	WA	2
5	Buchanan	London	(null)	2
6	Suyama	London	(null)	5
7	King	London	(null)	5
8	Callahan	Seattle	WA	2
9	Dodsworth	London	(null)	5

As can be seen in the **ReportsTo** data column the following entries are: **(null)**, **2** and **5**, i.e. any of these entries may be the value of the **Parent Value** property. If the value of this property is not specified, or is specified as a "space", then the default value is used. By default, the value of the **Parent Value** property is set to null, i.e. the parent row for all rows will be a line where there is a **(null)** entry in the **ReportsTo** data column. In this case, this is a row with the **ID 2**. The picture below shows an example of a rendered report:

Employee	City	Region
Fuller	Tacoma	WA
Buchanan	London	
Dodsworth	London	
King	London	
Suyama	London	
Callahan	Seattle	WA
Peacock	Redmond	WA
Davolio	Seattle	WA
Leverling	Kirkland	WA

If the value of the **Parent Value** property is set to **2**, then the parent row for all rows will be a row where there is a **2** entry in the **ReportsTo** column data. In this case, these are rows with **ID 1,3,4,5,8**. The picture below shows an example of a report, where the value of the **Parent Value** property is set to the **2** value:

Employee	City	Region
Buchanan	London	
Dodsworth	London	
King	London	
Suyama	London	
Callahan	Seattle	WA
Peacock	Redmond	WA
Davolio	Seattle	WA
Leverling	Kirkland	WA

Indent Property

In order to visualize the hierarchy of a report you need to change a value of the **Indent** property. The value of the **Indent** property is the distance at which an entry in the hierarchy, relative to the previous level of the tree, will be moved. If the **Indent** property is set to 0, then the indent will not be performing. The picture below shows an example of a rendered hierarchical report with the indent of 0:

Employee	City	Region
Fuller	Tacoma	WA
Buchanan	London	
Dodsworth	London	
King	London	
Suyama	London	
Callahan	Seattle	WA
Peacock	Redmond	WA
Davolio	Seattle	WA
Leverling	Kirkland	WA

If the **Indent** property is set to any value greater than 0, for example 10, the shifting will be on 10 units of a report (centimeters, inches, one hundredth of inch, pixels). The picture below shows an example of a rendered hierarchical report with the indent of 10 units in the report:

Employee	City	Region
Fuller	Tacoma	WA
Buchanan	London	
Dodsworth	London	
King	London	
Suyama	London	
Callahan	Seattle	WA
Peacock	Redmond	WA
Davolio	Seattle	WA
Leverling	Kirkland	WA

If you want a text component, which is located in the **HierarchicalBand**, do not move, you should change the value of the **Locked** property of this text component. If the **Locked** property is set to **true**, then the text component will not be shifted. If the **Locked** property is set to **false**, then the text component will be shifted. The picture below shows an example of a rendered hierarchical report:

Employee	City	Region
Fuller	Tacoma	WA
Buchanan	London	
Dodsworth	London	
King	London	
Suyama	London	
Callahan	Seattle	WA
Peacock	Redmond	WA
Davolio	Seattle	WA
Leverling	Kirkland	WA

As can be seen on the picture above, the **Locked** property of the **Employee** text component is set to **false**, so the entries were shifted. And for the **City** and **Region** text components, this property is set to **true**, so the entries were not shifted.

Important! The parent entry is not shifted. Only subordinate entries are shifted: the lower the priority is, the further is shifting, relative to the parent entry.

Child Band

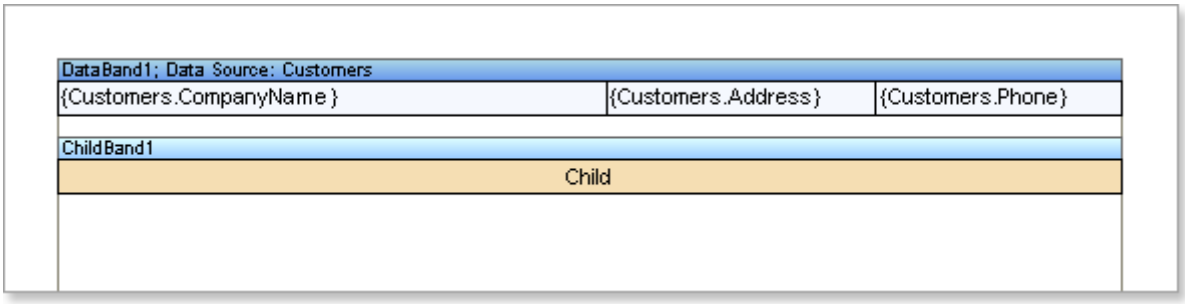
The Child Band can be used in tandem with other bands. It can be placed after any band on a page, including after the Header band or the GroupHeader band. It allows the parent band to be effectively extended whilst the child can behave differently, for example having a different background color.

Note: The Child band can be used in combination with any other bands placed on a page.

Using The Child Band With Data Bands

The Child band allows you to output two bands on one data row.

To use the child band in this way you would create a new report, put a Data band on the page, and then put a Child band after the Data band.

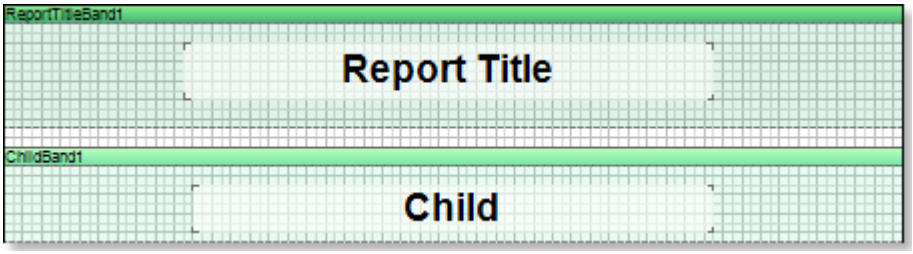


When you run the report the Child band will be printed as many times as the Data band. In other words the Child band acts as a continuation of the Data band but is still a band in its own right possessing all properties available with other bands.

Alfreds Futterkiste	Obere Str. 57	030-0074321
Child		
Ana Trujillo Emparedados y helados	Avda. de la Constitución 22	(5) 555-4729
Child		
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932
Child		
Around the Horn	120 Hanover Sq.	(171) 555-7788
Child		
Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65
Child		
Blauer See Delikatessen	Forsterstr. 57	0621-08460
Child		
Blondesddsl père et fils	24, place Kléber	88.60.15.31
Child		
Bólido Comidas preparadas	C/ Araquil, 67	(91) 555 22 82
Child		
Bon app'	12, rue des Bouchers	91.24.45.40
Child		
Bottom-Dollar Markets	23 Tsawassen Blvd.	(604) 555-4729
Child		
B's Beverages	Fauntleroy Circus	(171) 555-1212
Child		
Cactus Comidas para llevar	Cerrito 333	(1) 135-5555
Child		
Centro comercial Moctezuma	Sierras de Granada 9993	(5) 555-3392
Child		
Chop-suey Chinese	Hauptstr. 29	0452-076545
Child		
Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647

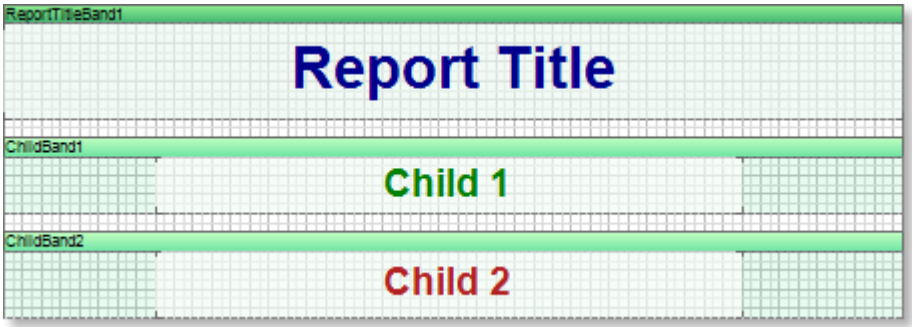
Multi Line Header

The **ChildBand** is a band that is a continuation of the band, after which it is placed.



In the picture above shows the **ChildBand** is placed after the **ReportTitleBand**, respectively, it is a continuation of this **ReportTitleBand**.

There are no limitations on the number of **ChildBands** placed on a page.



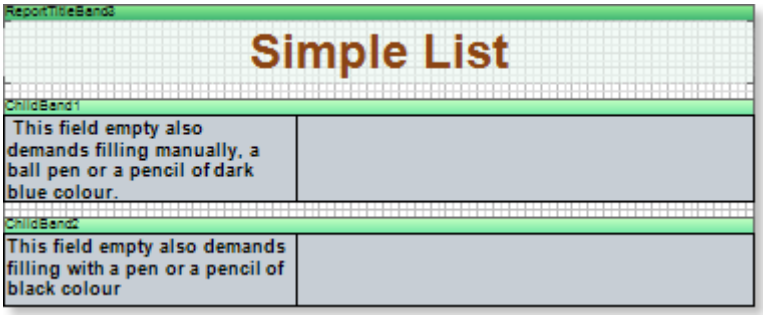
The picture above shows two **ChildBands**, which are a continuation of the **ReportTitleBand**.

Suppose there is a report with the report title that consists of a few lines. If the text is placed on the **ReportTitleBand**, then visually it may look not entirely correct:

Even when using the **GrowToHeight** property, then visually it cannot be convenient:

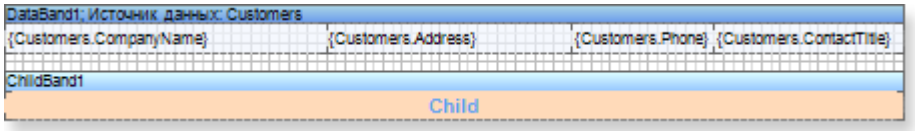
Therefore, in some cases, the title of the report is better represent with the **ChildBand**:

The picture below shows the report title located in the **ReportTitleBand** and two **ChildBands**.



Child Band and Data

How to output two bands on one data row? You can use the **ChildBand**. Create a new report. Put the **DataBand** on a page. Put the **ChildBand** under the **DataBand**.



Run a report for execution. As you can see, the **ChildBand** was printed as many times as the **DataBand**. I.e. the **ChildBand** is a continuation of the **DataBand**. But at the same time it remained to be a band, with all its properties.

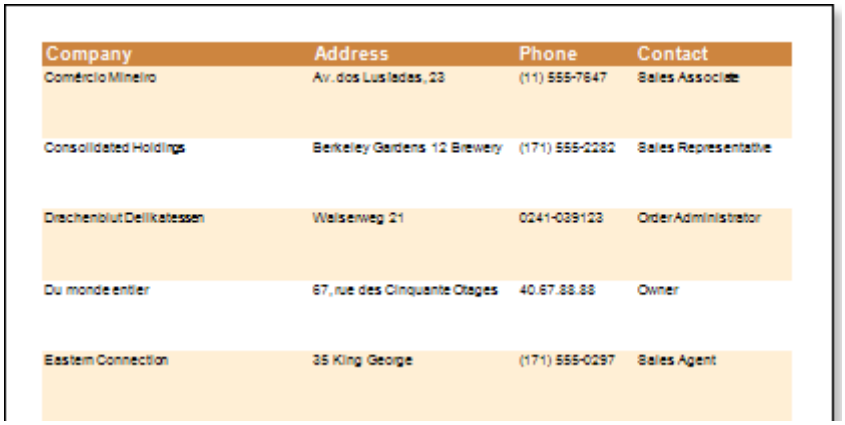
Company	Address	Phone	Contact
FISSA Fabrica Inter. Salchichas S.A.	C/ Moralzarzal, 86	(91) 555 94 44	Accounting Manager
	Child		
Folies gourmandes	184, chaussée de Tournai	20.16.10.16	Assistant Sales Agent
	Child		
Folk och fä HB	Åkergatan 24	0695-34 67 21	Owner
	Child		
Frankenversand	Berliner Platz 43	089-0877310	Marketing Manager
	Child		
France restauration	54, rue Royale	40.32.21.21	Marketing Manager
	Child		
Franchi S.p.A.	Via Monte Bianco 34	011-4988260	Sales Representative
	Child		

The **ChildBand** can be used not only with the **DataBand**. It can be placed after any band on a page. For example, after the **Header** band or after the **GroupHeader** band.

The **Child** band can be used in association with any band.

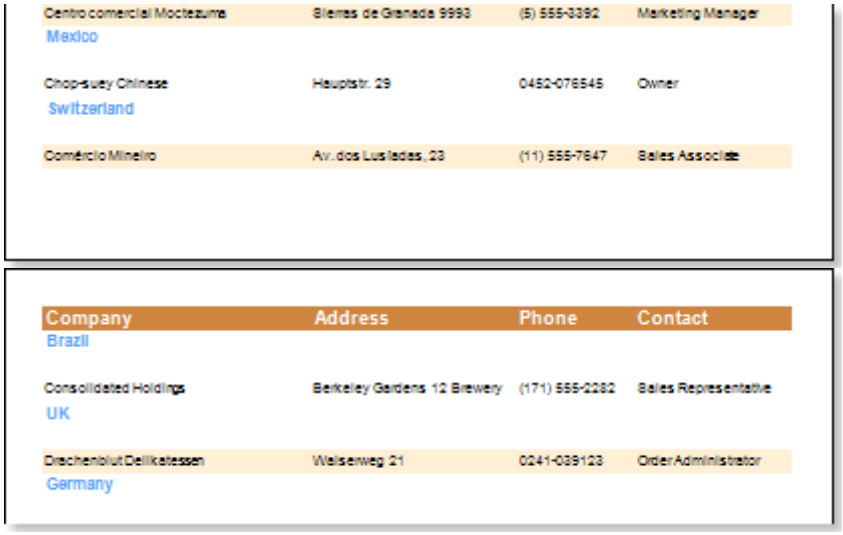
KeepChildTogether Property

For example, add the **ChildBand** to the **DataBand**, as the result a data row and an empty row (**ChildBand** row) is output, visually it looks like a high line.



Company	Address	Phone	Contact
Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647	Sales Associate
Consolidated Holdings	Berkeley Gardens 12 Brewery	(171) 555-2282	Sales Representative
Drachenblut Delikatessen	Walserweg 21	0241-039123	Order Administrator
Du monde entier	67, rue des Cinquante Otages	40.67.88.88	Owner
Eastern Connection	35 King George	(171) 555-0297	Sales Agent

Add data to the **ChildBand**, for example **Country.**
The picture below shows that instead of empty space, the country name will be output.



So as to avoid breaking data, meaning when **Company**, **Address**, **Phone**, **Contact** remained on one page, and the second part (in our case, **Country**) was moved to another page, the **ChildBand** has the **KeepChildTogether** property.

By default the property is set to **true**.

Empty Band

The **EmptyData** band is used to fill free space on the bottom of a page with additional empty data rows formatted to match the displayed data. This example shows a page without an **EmptyData** band:

Adding an **EmptyData** band to the same page changes the look of the empty part of the page to match the formatting of the rest of the data.

Example

Create a new report with borders around the text items on the data band. Then drop an Empty Data band after the Data band. If there is more than one **Data** band on the page then you should place the **EmptyData** band after the last **Data** band, but before any footer bands.

Note: To output Footer bands on the bottom of a page set the **PrintAtBottom** property of each Footer band to true.

Then add text objects to the empty band to match those on the Data band. The result should look something like this:

If you then run the report you will see that the empty space is replaced with formatted empty data rows:

Empty Band Modes

The **Empty** band has only one special property - **SizeMode**. This property indicates the behavior of the Empty Band on the bottom of a page. There are 4 values of the property: **IncreaseLastRow**, **DecreaseLastRow**, **AlignFooterToBottom**, **AlignFooterToTop**. The **IncreaseLastRow** indicates that if, when filling the page by an

Empty band, there is a free space to partially output an Empty Band, then it is possible to increase the last row. The picture below shows this.

DecreaseLastRow. The last row of the **Empty Band** will be decreased by height. The picture below shows this.

AlignFooterToBottom. If there is no free space for the **Empty Band** then this band is not output. The picture below shows this.

AlignFooterToTop. (this is the default value of the **SizeMode** property). The Footer Bands will be output on the bottom (the **PrintAtBottom** = true) and moved to top to fill the free space of the Empty Band. The picture below shows this.

Watermarks

Sometimes it is required to output watermark on a page. Watermark is an inscription or an image that is placed under or over elements of a page. Stimulsoft Reports has three modes to output watermarks: the **Watermark** of a page, the **Overlay** band and direct placing on a page.

Watermark Property

The **Watermark** property allows user to output one image and one inscription on the background or foreground. The **Watermark** property has sub-properties to output watermarks.

Watermark	(Watermark)
Aspect Ratio	False
Right to Left	False
Enabled	True
Text	
Angle	45
Font	**Arial; 100pt**
Text Brush	Solid
Show Behind	False
Show Image Behin	True
Image	[Not Assigned]
Image Alignment	Middle Center
Image Multiple Fact	1
Image Stretch	False
Image Tiling	False
Image Transparenc	0

On the table below Text properties for watermark are described.

Text	A text that is used to output a watermark
Text Brush	A brush to output a watermark
Font	A font that is used to output a watermark
Angle	An angle to rotate a watermark
ShowBehind	Show text of a watermark on the background or foreground

An example how properties can be used is shown on the picture below.

On the table below Image properties for watermark are described.

Image	An image to output
ImageAlignment	This property is used to align an image on a page
ImageMultipleFactor	A multiplier that is used to change image size
AspectRatio	Saves proportions of an image
ImageTiling	If to set this property to **true**, then it will be tiled throughout a page
ImageTransparency	This property is used to set image transparency
ImageStretch	Stretches an image on a page
ShowImageBehind	Shows an image of a watermark on the background or foreground

Also there is another **Enabled** property. This property enables or disables watermark output.

Overlay Band

The **OverlayBand** is used to output text, images, primitives and other data.

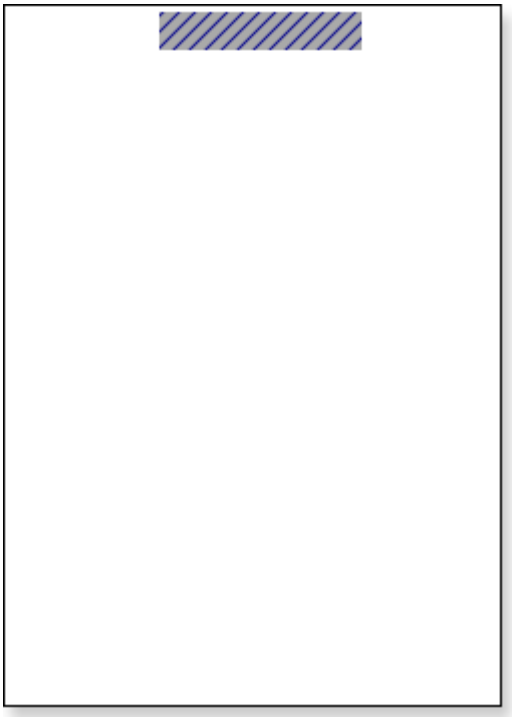
The **OverlayBand** is placed on the top of all other bands. The **Watermark**, for example, is placed in the foreground or in the background. The advantage of the **OverlayBand** over **Watermark** is that it is not a page element but a band which has properties of bands.
Watermark is either printed on all pages or not printed. The **OverlayBand** band allows selecting 7 ways of printing. In **Watermark,** for the same operation script should be printed.

The **PrintOn** property has 7 values:
- **All page;**
- **ExceptFirstPage;**
- **ExceptLastPage;**
- **ExceptFirstAndLastPage;**
- **OnlyFirstPage;**
- **OnlyLastPage;**
- **OnlyFirstAndLastPage.**

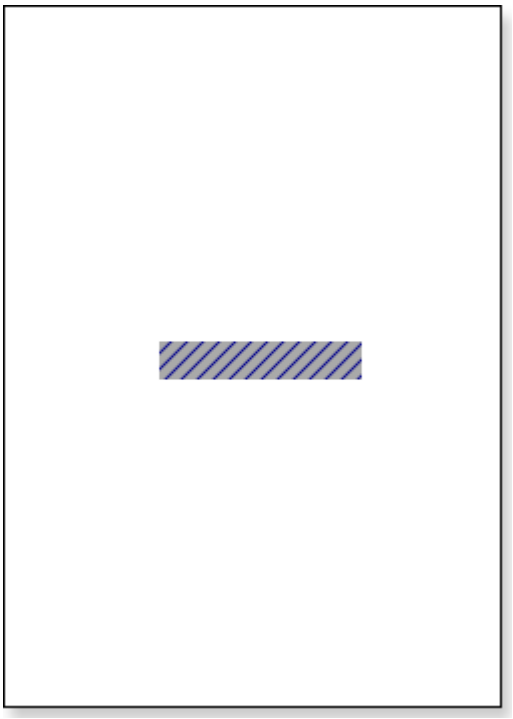
Vertical Alignment Property

The **VerticalAlignment** property is used to define the place of the "watermark" inscription which is output using the **OverlayBand** band. This property may have three values:

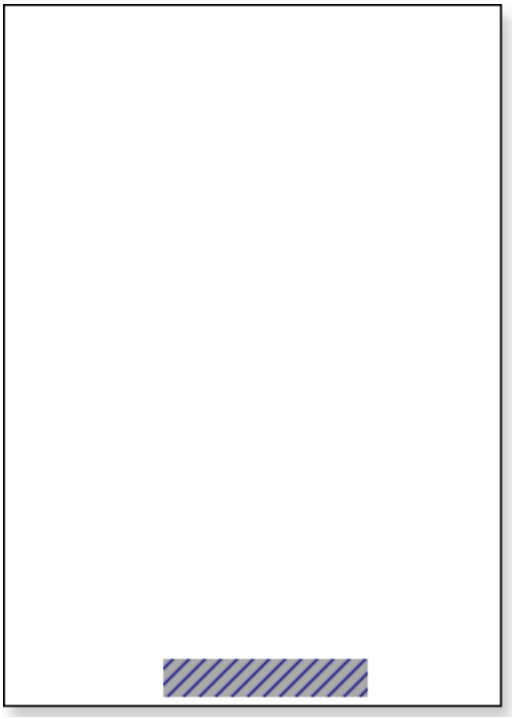
1. **Top**. The **OverlayBand** will be output on the top of a rendered report before the page header and the page header.



2. **Center**. The **OverlayBand** will be output on the center of a rendered report and in front of data placed on the page.

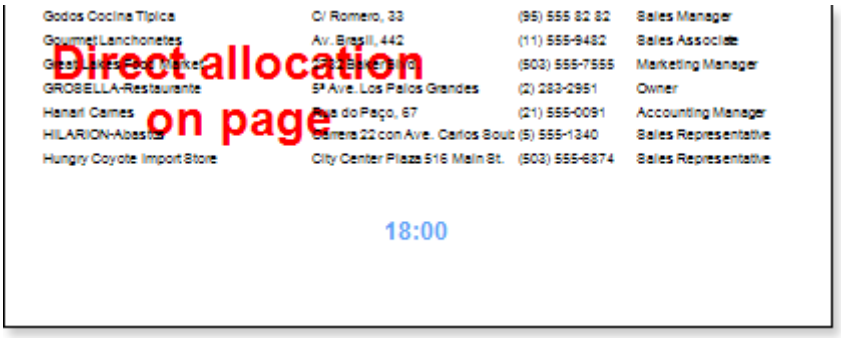


3. **Bottom**. The **OverlayBand** will be output on the bottom of a page of a report and after the page footer.



Direct Allocation on Page

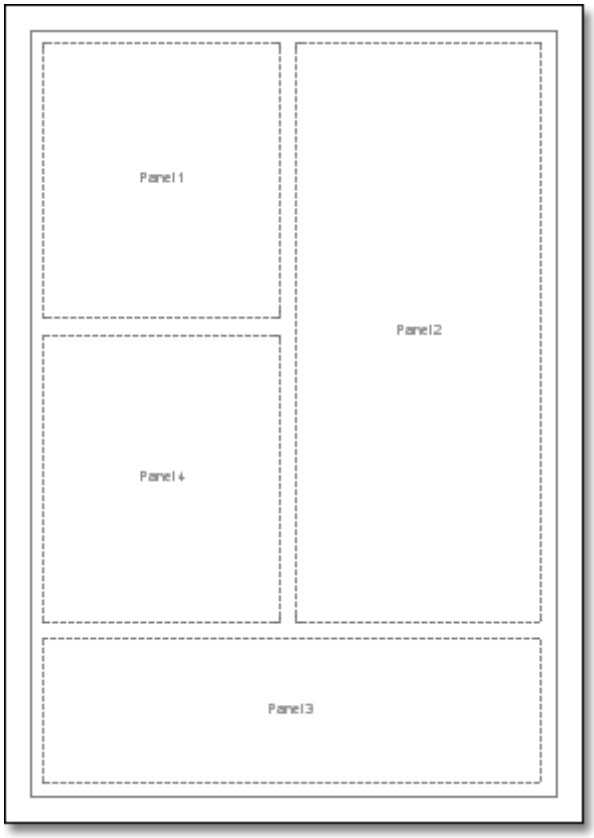
One of the options for placement of the "watermark" inscription is a direct placement on the page. This means that the direct placement of any component, which will be the "watermark" inscription on a page of a report template.



The picture above shows the "watermark" by means of the direct placement a text component on a template of a page.
Direct placement on a page allows showing an inscription on the background but at any of the working space.
There is the **Linked** property. This **Linked** property may have two values: **true** and **false**.
If the property is set to **false**, then the relation with "owner" is not fixed. In other words the "owner" is the report template item on which the **TextBox** component is placed.
If the property is set to **true**, then the relation with "owner" is fixed. In other words the **TextBox** component may change the position but it will be referred to the item on what it is fixed.

Panels

Panel is a rectangular region that may contain other components including bands. If to move a panel then all components in it are moved too. The panel can be placed both on a band and on a page. This gives unique abilities in report creation.



Placing Bands on Panel

A panel can be placed on a page, on a band, and on another panel. Almost all components of a report can be paced on a panel. But not all bands can be placed on a panel. A table below shows which bands can placed.

Band name	It is possible to place a band on a panel
ReportTitle	No
ReportSummary	No
PageHeader	No
PageFooter	No
GroupHeader	Yes
GroupFooter	Yes
Data	Yes
Hierarchical Data	Yes
Child	Yes
Header	Yes
Footer	Yes

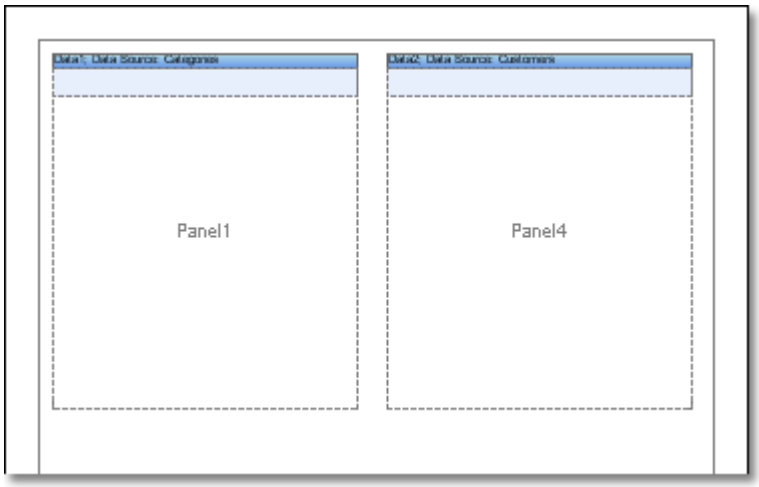
As seen, report bands and page bands cannot be placed on a report. All other bands can be placed on a panel.

Placing Panels

There are three ways of placing panels: on a page, on a band and in another panel. The below topics describes all these variants.

Placing Panels on Page

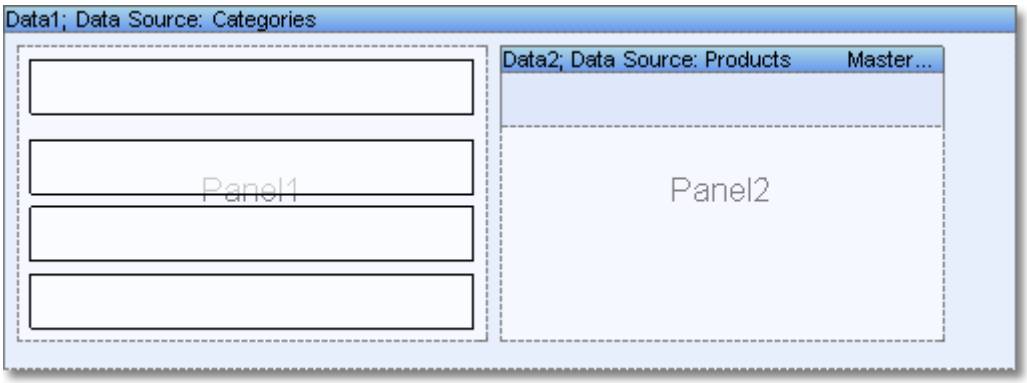
It is the first way. Basically it is used as organization some independent streams of printing. Panels can be places on any part of a page. Each panel is a small page. So it is allowed placing some small pages with bands and components on one page. So it is possible to render a lot of complex reports.



Notice. Number of panels on one page is unlimited.

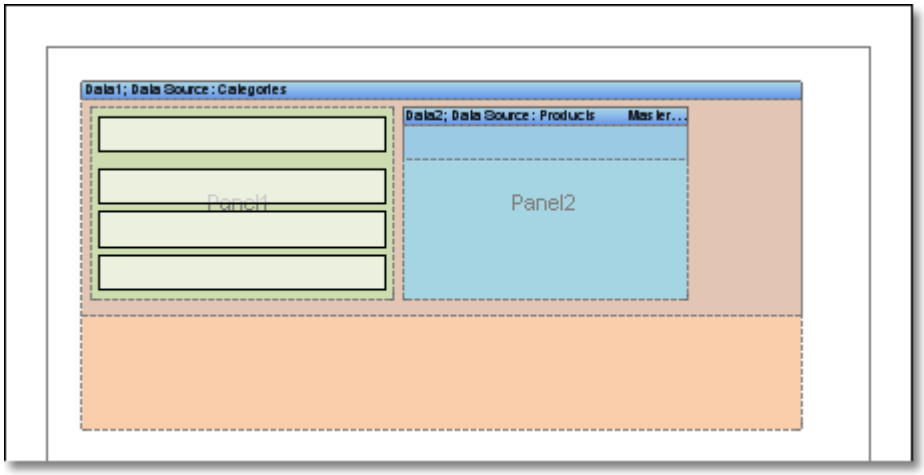
Placing Panels on Band

The second way is when the panel in placed on a band. This variant is used both for grouping simple components on a panel and to output bands on a band. This allows rendering very complex reports. But it is important to know that the report template can be difficult in "reading".



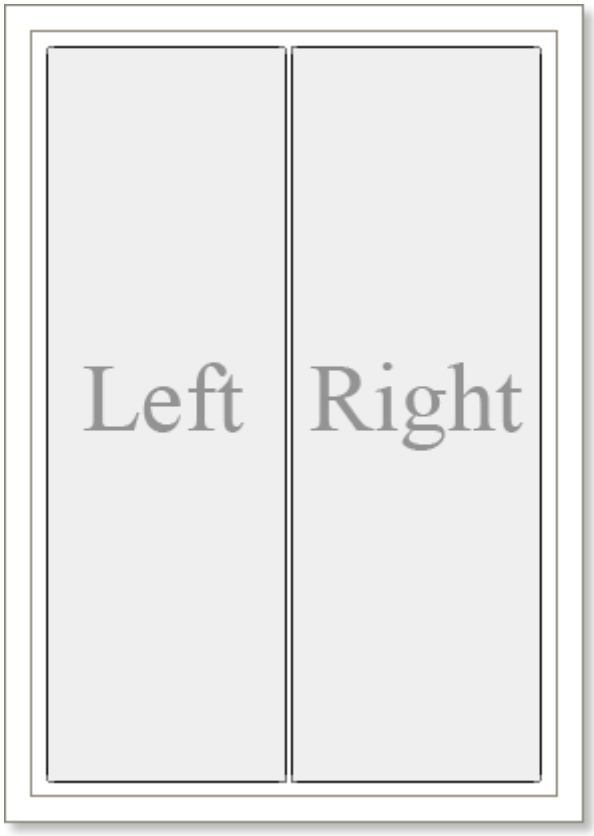
Placing Panels on Panel

The third way – when a panel is placed on another panel. This variant is combination of two previous ones. It is very important to know that panels insertion should be used very carefully. Number of insertions in unlimited but such report will not have good look.



Side-by-Side Reports

Side-by-side report is a report in what containers can help to speed up report creation. Two lists of rows are output simultaneously in this report. Both lists are independent from each other. Usually it is necessary to use the **Sub report** component to create such a report. But it is much easier to create a report with panels.



How to build a **Side-by-Side** report. Put two containers on a page. Set the **DockStyle** property of one component to **Left**. Set the **DockStyle** property of the second component to **Right**. Docking component is necessary to take all space on a page by the height. In cases it should not be done. Leave some space between lists to separate them. Put two bands on the first panel: the **Header** band and the **Data** band. The first list will output using these bands. Do the same in the second container. As a result two lists will be output on one page simultaneously.

Companies

Company	Phone
Hungry Owl All-Night Grocers	2967 542
Island Trading	(198) 555-8888
Königlich Essen	0555-09876
La corne d'abondance	30.59.84.10
La maison d'Asie	61.77.61.10
Laughing Bacchus Wine Cellars	(604) 555-3392
Lazy K Kountry Store	(509) 555-7809
Lehmanns Marktstand	069-0245984
Let's Stop N Shop	(415) 555-5859
LILA-Supermercado	(9) 331-6954
Lino-Delicatessen	(8) 34-56012
Lonesome Pine Restaurant	(503) 555-9573
Magazzini Alimentari Riuniti	035-640230
Maison Dewey	(02) 201 24 67
Mère Paillarde	(514) 555-8054
Morgenstern Gesundkost	0342-023176
NorthSouth	(171) 555-7555
Oceano Atlantico Ltda.	(1) 135-5555
Old World Delicatessen	(907) 555-7584
Ottilies Käseladen	0221-0644327
Paris specialités	(1) 42.34.22.66
Pericles Comidas clásicas	(5) 552-3745
Piccolo und mehr	6562-9722
Princesa Isabel Vinhos	(1) 356-5634
Que Delícia	(21) 555-4252
Queen Cozmo	(11) 555-1189
QUICK-Stop	0372-035188
Rancho grande	(1) 123-5555
Rattlesnake Canyon Grocery	(505) 555-5939
Ringgisti Caseificio	0522-556721
Ricardo Adocicados	(21) 555-3412
Richter Supermarkt	0897-034214
Romero y tomillo	(91) 745 6200
Santé Gourmet	07-98 92035
Save-a-lot Markets	(208) 555-8097
Seven Seas Imports	(171) 555-0117
Simons bistro	31 1234 56
Spécialités du monde	(1) 47.55.6010
Split Rail Beer & Ale	(307) 555-4680
Suprêmes délices	(071) 23 67 22 20
The Big Cheese	(503) 555-3092
The Cracker Box	(406) 555-5834

Products

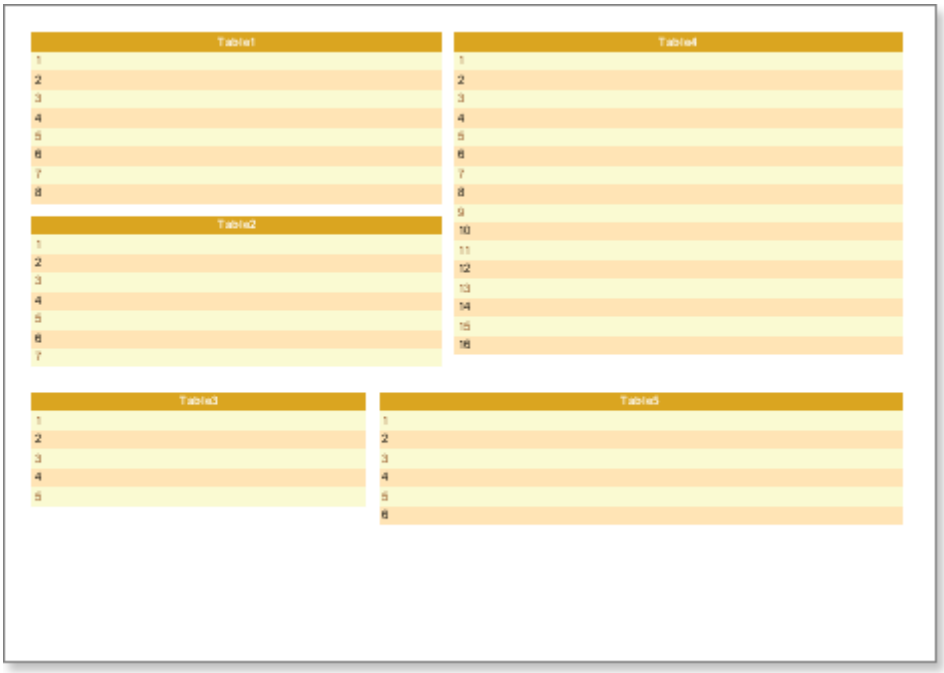
Product	Price
Ipoh Coffee	46,00p.
Jack's New England Clam Chowder	9,65p.
Konbu	6,00p.
Lakkalikööri	18,00p.
Laughing Lumberjack Lager	14,00p.
Longlife Tofu	10,00p.
Louisiana Fiery Hot Pepper Sauce	21,05p.
Louisiana Hot Spiced Okra	17,00p.
Manjimup Dried Apples	53,00p.
Mascarpone Fabioli	32,00p.
Mishi Kobe Niku	97,00p.
Mozzarella di Giovanni	34,80p.
Nord-Ostsee Matjesheringe	25,89p.
Northwoods Cranberry Sauce	40,00p.
Nuka Cola Kuli-Kuli	14,00p.
Original Frankfurter grüne Soße	13,00p.
Outback Lager	15,00p.
Pâte chinoise	24,00p.
Pavlova	17,45p.
Perth Pasties	32,80p.
Queso Cabrales	21,00p.
Queso Manchego La Pastora	38,00p.
Raclette Courdavault	55,00p.
Ravioli Angelo	19,50p.
Rhönbräu Klosterbier	7,75p.
Röd Kaviar	15,00p.
Rogede sild	9,50p.
Rössle Sauerkraut	45,60p.
Saarquasch Alt	14,00p.
Schoggi Schokolade	43,90p.

(c) 2003-2006 stimulsoft

Page 2 of 3

Multiple Tables on One Page

Sometimes it is required to output multiple tables on a page and, what is very important, to output them on different parts of a page. Such report can be rendered using the **Sub Report**. But it is much easier to do this using panels. All it is required to do is to place panels and put band on them. On the picture below a sample of such a report is shown.

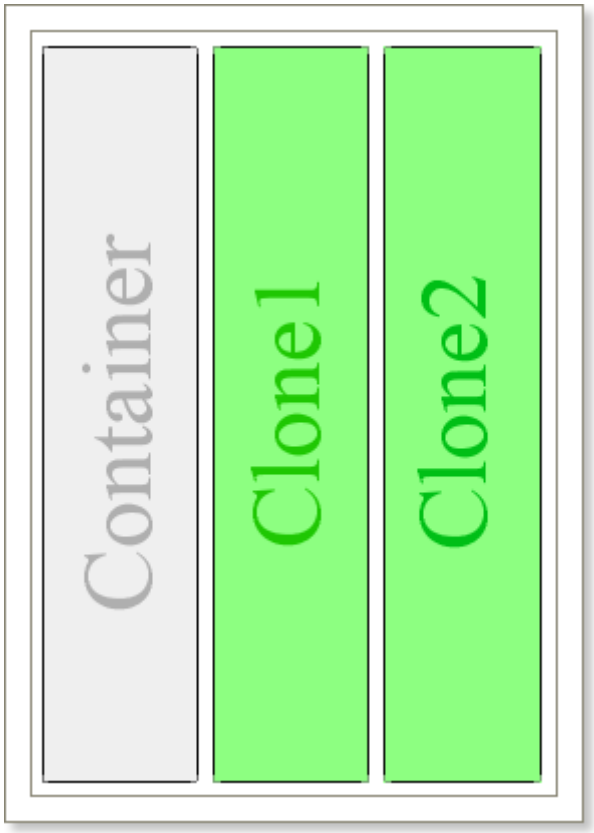


Cloning

The unique Clone component is included into Stimulsoft Reports. This component is used to clone parts of a report into a required part of a report. Cloning can be used only in panels.

Notice. The **Clone** component can work with the **Panel** component.

How it works? Put a panel on a page. Put bands to output lists. Place a panel on the left part of a page. Place a **Clone** component on the right side of a page. Then, in the **Clone** component designer, indicate the panel that should be cloned. In our case it is the panel that was created on a page.



Run a report. The panel will be rendered first. The list will be output on the left side of a page. Then the list will be continued to output on the place where the **Clone** component is placed. The **Clone** component clones all bands of the panel. Using the **Clone** component it is possible to render complex reports with columns. The first column is output using the panel and other columns - using the **Clone** component. It is important to consider the order of placing Clone components on a page.

Notice. **Panel** components and their clones will output in order of placing components on a page.

Cross Table

The **Cross table** is a special component that is used to process, group and summarize data from the data source. The result is represented as a table. The **Cross table** can be placed both directly on a page or on a **Data** band. If a table that is created as a result of a **cross table** rendering does not fit in the one page, then can be printed on some pages. The component has many properties and settings.

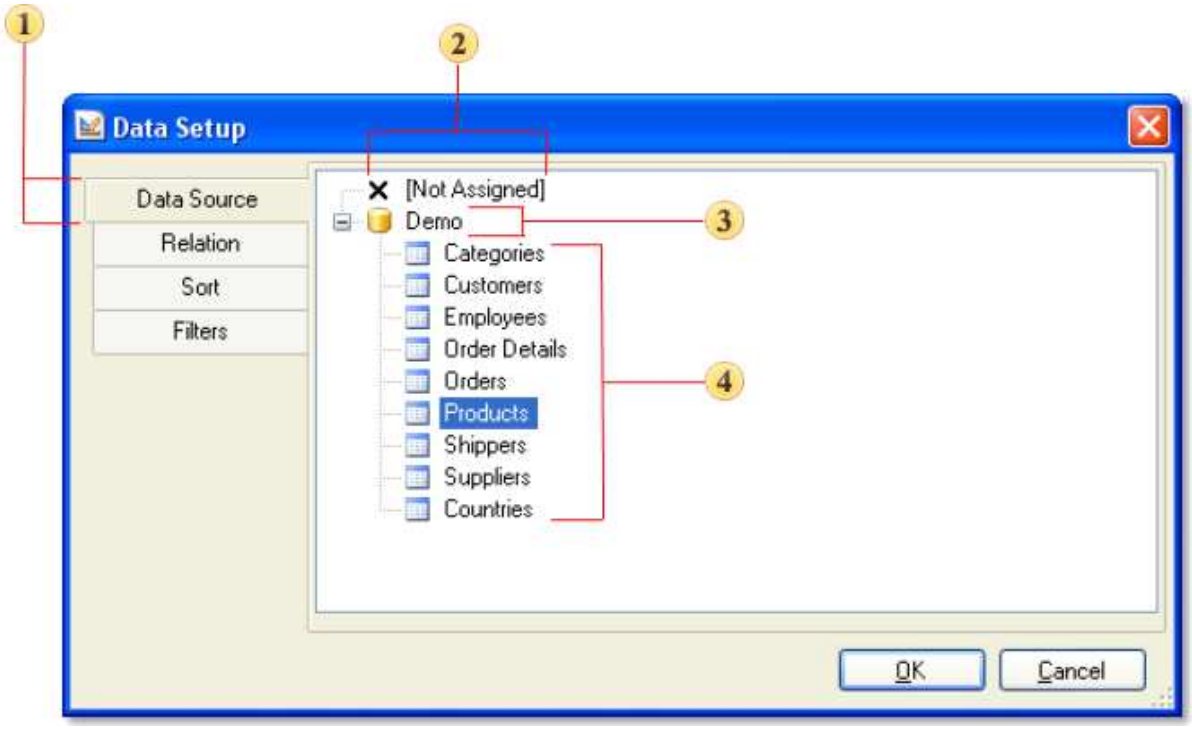
Data Source Property

Data are the base for cross table rendering. So the cross table rendering should be started from selecting the data source. The data source can be selected using the Data source.

It is necessary to specify the data source that will be used. There are several ways how to do this.
The first way. You may use either the **DataSource** property or the Table editor.

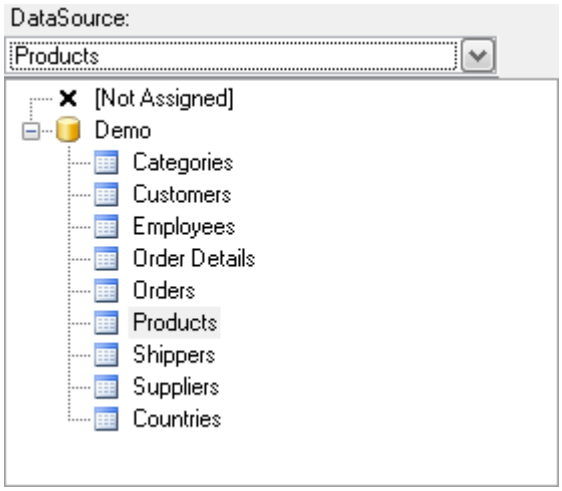
Data Source	Products

A data source can be selected by clicking the first tab of the Data band editor. All data sources are grouped in categories. Each category corresponds to one connection with data in the report data dictionary.



1 The tab to select the data source;
2 Select this node if you do not need to specify the data source;
3 The "Demo" data category;
4 The "Demo" data source category.

The second way. The data source can be selected using the cross table editor. It can be called by double click on the cross table.



Cross Table Items

After selecting the data source you need to specify the following items: columns, rows, and cells for summation.

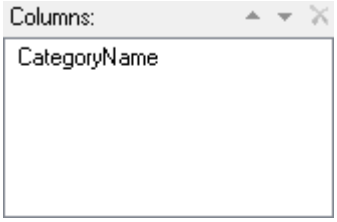
Columns

On a picture below you may see how the columns are positioned on a table.

Columns

Products	CategoryName								
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	24	29		38		20	42	168
Brazil	20								20
Canada		113	17			136			266
Denmark								100	100
Finland	57		75						132
France	86			98				62	246
Germany	125	32	140		22		26	10	355
Italy				23	57				80
Japan		39				29	39	55	162
Netherlands			51						51
Norway				164					164
Singapore	17	27			26				70
Spain				108					108
Sweden					165			224	389
UK	56	13	74						143
USA	183	259					15	208	665
Total	559	507	386	393	308	165	100	701	3119

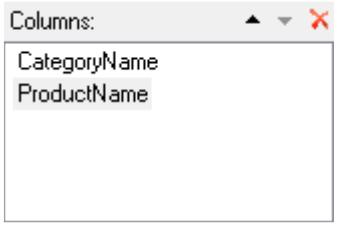
It is allowed to specify one or several columns at once. For example, in cross table only one column is specified:



As a result we get grouping by values of this column:

CategoryName								
Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total

If to specify more than one column:



Grouping is output by values of two columns. Values of the first column are output first. Then the value from the second column is output:

CategoryName, ProductName																									
Beverages														Condiments											
Chai	Chang	Chartreuse verte	Côte de Blaye	Guaraná Fantástica	Ipoh Coffee	Lakkalikööri	Lapland Lumberjack Lager	Outback Lager	Rhönbräu Klosterbier	Sasquatch Ale	Steeleye Stout	Total	Aniseed Syrup	Chef Anton's Cajun Seasoning	Chef Anton's Gumbo Mix	Genen Shouyu	Grandma's Boysenberry Spread	Gula Malacca	Louisiana Fiery Hot Pepper Sauce	Louisiana Hot Spiced Okra	Northwoods Cranberry Sauce	Original Frankfurter grüne Soße	Sirop d'érable	Vegie-spread	Total

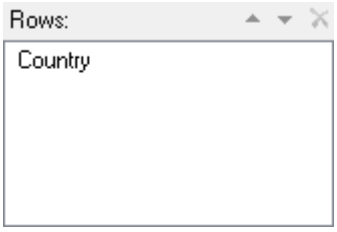
Rows

On a picture below you may see how the rows are positioned on a table.

Products	CategoryName								
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	24	29		38		20	42	168
Brazil	20								20
Canada		113	17			136			266
Denmark								100	100
Finland	57		75						132
France	86			98				62	246
Germany	125	32	140		22		26	10	355
Italy				23	57				80
Japan		39				29	39	55	162
Netherlands			51						51
Norway				164					164
Singapore	17	27			26				70
Spain				108					108
Sweden					165			224	389
UK	56	13	74						143
USA	183	259					15	208	665
Total	559	507	386	393	308	165	100	701	3119

Rows

Grouping is done only by its values for one row:



Get the result shown on a picture below. All values of the specified row are represented in one level.

Country
Australia
Brazil
Canada
Denmark
Finland
France
Germany
Italy
Japan
Netherlands
Norway
Singapore
Spain
Sweden
UK
USA
Total

Specify two rows:

A cross table is grouped in two levels vertically:

Country	City
Australia	Melbourne
	Sydney
	Total
Brazil	Sao Paulo
	Total
Canada	Montréal
	Ste-Hyacinthe
	Total

In a cross table you may not specify columns or rows. For example, if columns are not specified, then grouping will be done by rows. For some reports this property is very important for a cross table. The picture below shows one those reports:

CategoryName	CompanyName	UnitsInStock
Beverages	Aux joyeux ecclésiastiques	281,5
	Bigfoot Breweries	46
	Exotic Liquids	37
	Karkki Oy	18
	Leka Trading	46
	Pavlova, Ltd.	15
	Plutzer Lebensmittelgroßmärkte AG	7,75
	Refrescos Americanas LTDA	4,5
	Total	455,75
Condiments	Exotic Liquids	10
	Forêts d'érables	28,5
	Grandma Kelly's Homestead	65
	Leka Trading	19,45
	Mayumi's	15,5
	New Orleans Cajun Delights	81,40
	Pavlova, Ltd.	43,9
	Plutzer Lebensmittelgroßmärkte AG	13
	Total	276,75

Summary Cells

Summary cells are the elements of a cross table, which set rules for cells formatting on intersection of columns and rows of a summary cell. On a picture below the structure of a simplest cross table is represented.

Columns

Rows

Summary Cells

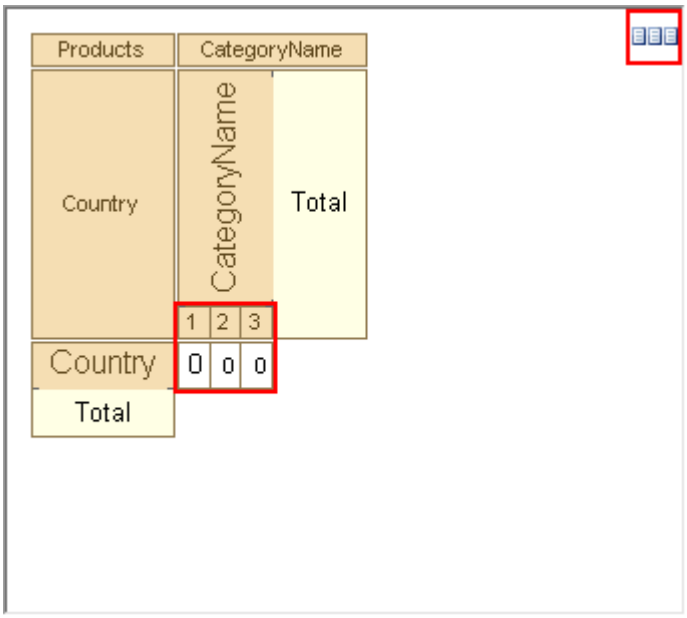
Products	CategoryName								
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	24	29		38		20	42	168
Brazil	20								20
Canada		113	17			136			266
Denmark								100	100
Finland	57		75						132
France	86			98				62	246
Germany	125	32	140		22		26	10	355
Italy				23	57				80
Japan		39				29	39	55	162
Netherlands			51						51
Norway				164					164
Singapore	17	27			26				70
Spain				108					108
Sweden					165			224	389
UK	56	13	74						143
USA	183	259					15	208	665
Total	559	507	386	393	308	165	100	701	3119

In a summary cell all values from the data source which are suitable for a particular condition are grouped. The condition is the coincidence of the value of the column and the row from a data source with the value of the column and row of a cross-table. The value of a cross table column and a row is indicated by intersection where the summary cell is placed. For example, see a simple cross table on a picture below:

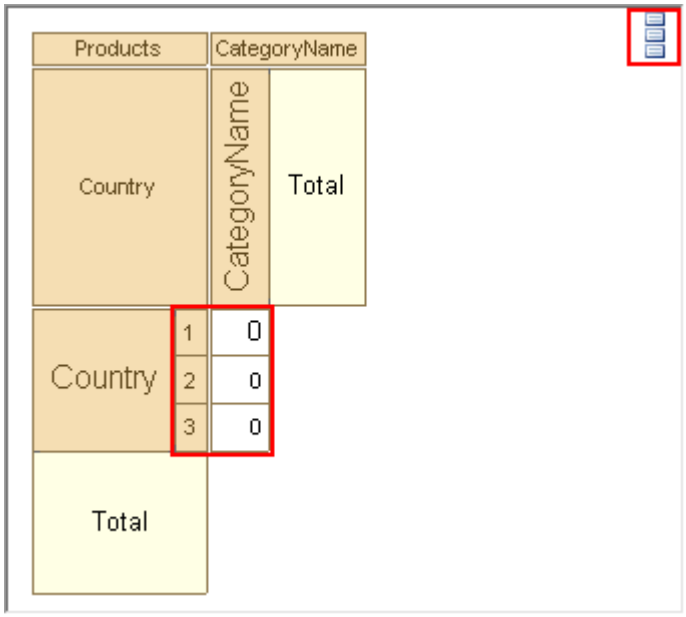
Products	CategoryName								
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	24	29		38		20	42	168
Brazil	20								20
Canada		113	17			136			266
Denmark								100	100
Finland	57		75						132
France	86			98				62	246
Germany	125	32	140		22		26	10	355
Italy				23	57				80
Japan		39				29	39	55	162
Netherlands			51						51
Norway				164					164
Singapore	17	27			26				70
Spain				108					108
Sweden					165			224	389
UK	56	13	74						143
USA	183	259					15	208	665
Total	559	507	386	393	308	165	100	701	3119

The red rectangle indicates the summary cell with the 140 values and also a column and a row of this cell. In this cell all values from the data source which CategoryName column is equal to **Confection** and Country row is equal to **Germany** were grouped. The rules of grouping are set using the **Summary** property of a summary cell.

If more than one summary cell is set in a Cross table then it is possible to define the direction of placing of these cells. The reporting tool can place them horizontally from left to right or vertically from top to bottom. On a picture below a table with horizontally placed summary cells is shown.

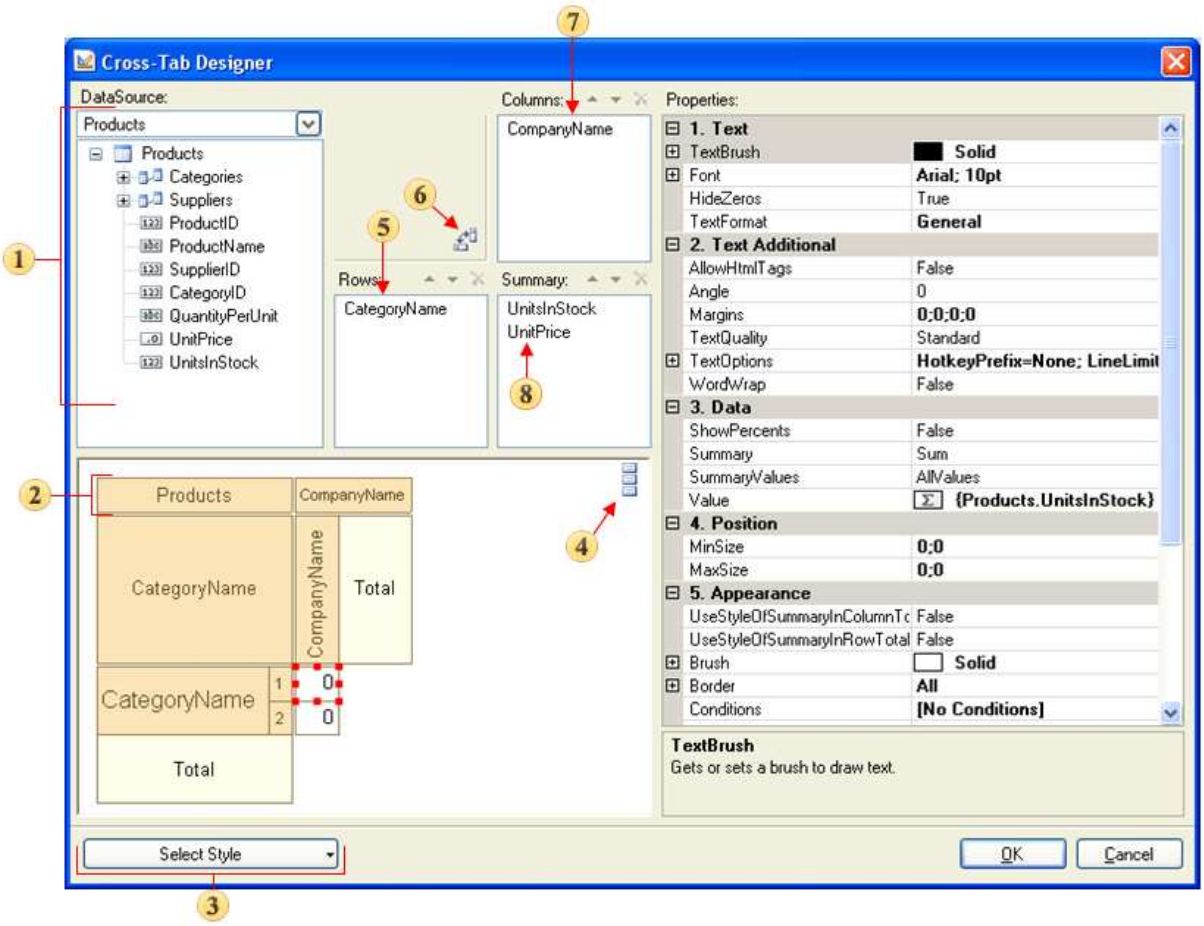


On a picture below a table with vertically placed summary cells is shown.



Cross Table Editor

For editing a cross table the Cross Table editor is used. The editor is divided in four zones: data source selection, columns, rows and cells selection, cross table preview and a property grid.



1 The Data source that is used for the table creation;
2 The Table Header;
3 Select a style of the table appearance;
4 Change the direction of summary in a table;
5 The list of rows;
6 This button is used to change columns and rows;
7 A list of columns;
8 A list of Summaries.

Data Summary Types

When rendering a cross-table, the report generator should know how the values in the summary cells will be summarize. Summation function is set using the Summary property of a summary cell. For each summary cell its own function can be specified. A Cross Table works with the following functions:

Function	Description
None	Do not summarize the cell values
Sum	Returns the sum of values that are contained in the cell
Average	Returns the average of values that are contained in the cell
Min	Returns the minimal of values that are contained in the cell
Max	Returns the maximal of values that are contained in the cell
Count	Returns the number of values that are contained in the cell
CountDistinct	Returns the number of distinct values that are contained in the cell

Image	A cross table will show the first value as an image

In addition to the Summary property, there is another property that affects on the summary. This is the Summary Values property. This property identifies and process the 0 and null values when calculating totals.

Sort Direction

The values of the source data that are used to group rows and columns are always re-sorted with the component of a cross-table. Resorting is necessary in order that, when showing a cross-table, rows and columns do not contain duplicates. But this behavior can be changed. The type sorting is specified using two properties: **SortDirection** and **SortType**. These properties are available for columns and rows of a cross-table.

SortDirection	Asc
SortType	ByDisplayValue

Using the **SortDirection** property it is possible to set the direction of sorting. Sorting can be in ascending order, descending, or no sorting. The **SortType** property sets the source of values for sorting: by value or by the displayed value. The picture below shows a table, sorted in two different directions.

Country	UnitsInStock
Australia	168
Brazil	20
Canada	266
Denmark	100
Finland	132
France	246
Germany	355
Italy	80
Japan	162
Netherlands	51
Norway	164
Singapore	70
Spain	108
Sweden	389
UK	143
USA	665
Total	3119

Country	UnitsInStock
USA	665
UK	143
Sweden	389
Spain	108
Singapore	70
Norway	164
Netherlands	51
Japan	162
Italy	80
Germany	355
France	246
Finland	132
Denmark	100
Canada	266
Brazil	20
Australia	168
Total	3119

Conditions

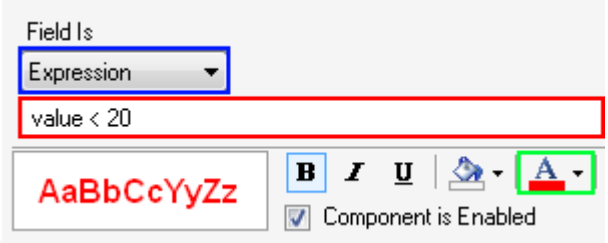
Often, when rendering a cross table, it is necessary that, according to certain conditions, the appearance of a cell will be changed. To achieve this, you can use the Conditions property of columns, rows and, summary cells.

Conditions	**[No Conditions]**

In order to specify the condition, it is necessary to select a component for what this condition will be executed and call the Conditions editor from the properties panel or from the toolbars.

For example, we need to mark summary cells which values are less than 20.

Add a new conditional formatting for the cell. Make three changes in the condition (see picture below).



Change the value of the Field Is field on the Expression (marked with blue). Specify the required expression (marked with red):

value <20

The value variable contains the total value of the summary cell. And change the text color of cells to red (marked with green). An example of report rendering is shown on the picture below.

Products	CategoryName								
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	24	29		38		20	42	168
Brazil	20								20
Canada		113	17			136			266
Denmark								100	100
Finland	57		75						132
France	86			98				62	246
Germany	125	32	140		22		26	10	355
Italy				23	57				80
Japan		39				29	39	55	162
Netherlands			51						51
Norway				164					164
Singapore	17	27			26				70
Spain				108					108
Sweden					165			224	389
UK	56	13	74						143
USA	183	259					15	208	665
Total	559	507	386	393	308	165	100	701	3119

Showing Totals

Rows and **Columns** of a cross-table have the **ShowTotal** property, which allows you to show or hide totals by rows and columns. If this property for **Rows** and **Columns** is set to **true**, then the totals by rows and columns are visually displayed. The picture below shows an example of a cross-table with a visually displayed results:

Products	Category/Name								
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	24	29		38		20	42	168
Brazil	20								20
Canada		113	17			136			266
Denmark								100	100
Finland	57		75						132
France	86			98				62	246
Germany	125	32	140		22		26	10	355
Italy				23	57				80
Japan		39				29	39	55	162
Netherlands			51						51
Norway				164					164
Singapore	17	27			26				70
Spain				108					108
Sweden					165			224	389
UK	56	13	74						143
USA	183	259					15	208	665
Total	559	507	386	393	308	165	100	701	3119

If, for example, the **ShowTotal** property is set to **false** for rows, then the total by rows will not be displayed. The picture below shows an example of a cross-table, where the **ShowTotal** property of rows is set to **false**:

Products	Category/Name								
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	24	29		38		20	42	168
Brazil	20								20
Canada		113	17			136			266
Denmark								100	100
Finland	57		75						132
France	86			98				62	246
Germany	125	32	140		22		26	10	355
Italy				23	57				80
Japan		39				29	39	55	162
Netherlands			51						51
Norway				164					164
Singapore	17	27			26				70
Spain				108					108
Sweden					165			224	389
UK	56	13	74						143
USA	183	259					15	208	665

If, for example, the **ShowTotal** property for columns is set to **false**, then total by columns will not be displayed. The picture below shows an example of a cross-table, where the **ShowTotal** property of columns is set to **false**:

Products	Category/Name							
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood
Australia	15	24	29		38		20	42
Brazil	20							
Canada		113	17			136		
Denmark								100
Finland	57		75					
France	86			98				62
Germany	125	32	140		22		26	10
Italy				23	57			
Japan		39				29	39	55
Netherlands			51					
Norway				164				
Singapore	17	27			26			
Spain				108				
Sweden					165			224
UK	56	13	74					
USA	183	259					15	208
Total	559	507	386	393	308	165	100	701

By default, the **ShowTotal** property for rows and columns is set to **true**, i.e. totals by rows and columns are displayed.

Processing Values for Summary

The **Cross Table** has the **SummaryValues** property, which allows you to display the total number of values of the cross-table, considering or not considering to 0 and/or null values. The **SummaryValues** property can take three values, depending on the value of the property, the number of values will be displayed as a result. Values of the **SummaryValues** property and their description are described in the table below:

Function	Description
AllValues	All values, contained in a cell
SkipZerosAndNulls	Skip **0** **null** values, contained in a cell
SkipNulls	Skip **null** values, contained in a cell

Word Wrap

Each component of the cross-table has the **WordWrap** property, which lets you wrap text from one line to another. If the **WordWrap** property is set to **false**, then the text is in one line, and if it does not fit in one line it will be cut. The picture below shows an example of a cross-table with the **WordWrap** property set to **false**:

CategoryName	Beverages
UnitsInStock, UnitPric	455,75
	37,98p.

If the **WordWrap** property is set to **true**, then text wrapping goes automatically. When wrapping a text on the new line the vertical and horizontal alignment are taken into the account. The picture below shows an example of a cross-table that has the **WordWrap** property set to **true**:

CategoryName	Beverages
UnitsInStock,	455,75
UnitPrice	37,98p.

By default, the **WordWrap** property of cross-table components is set to **false**.

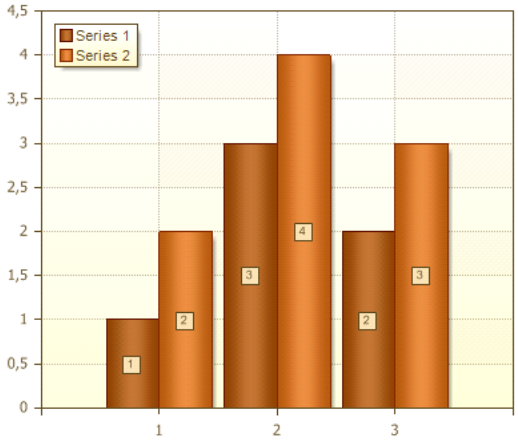
Charts

Column Area

Data that is arranged in columns or rows. Column charts are useful for showing data changes over a period of time or for illustrating comparisons among items:

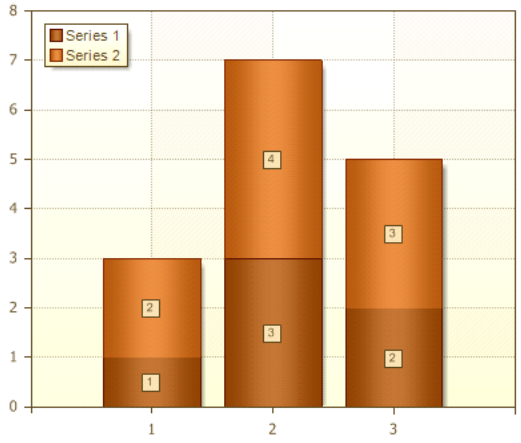
Clustered Column

Clustered column charts compare values across categories.

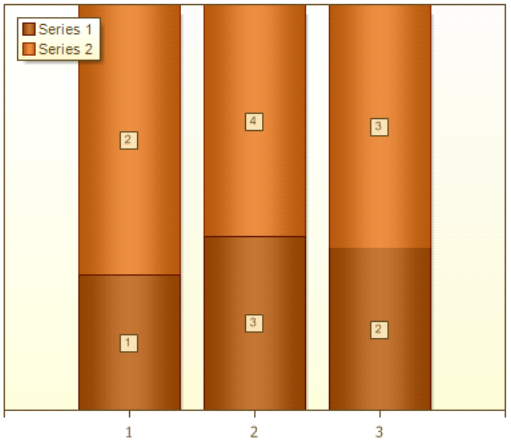


Stacked Column

Stacked column charts show the relationship of individual items to the whole, comparing the contribution of each value to a total across categories.



Full-Stacked Column
Full-Stacked column allows comparing the percentage of each value.

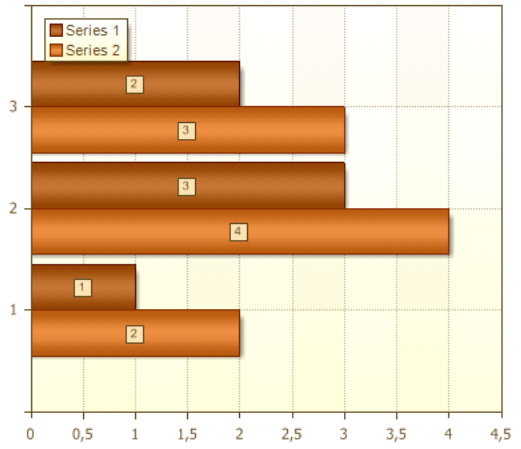


Bar Area

Data that is arranged in columns or rows on a worksheet can be plotted in a bar chart. Bar charts illustrate comparisons among individual items. Bar Area should be used if:

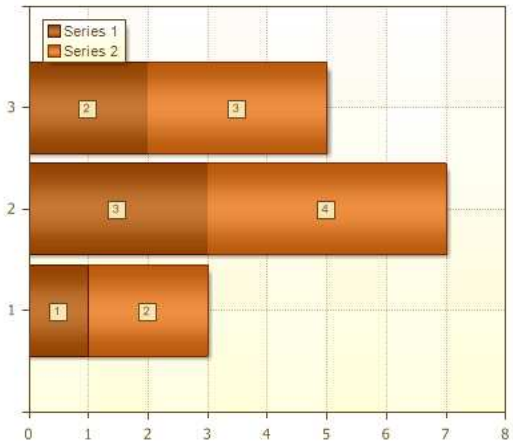
- Ticks are long.
- If a values show duration.

Clustered Bar
Clustered bar charts compares values across categories. In a clustered bar chart, the categories are typically organized along the vertical axis, and the values along the horizontal axis.

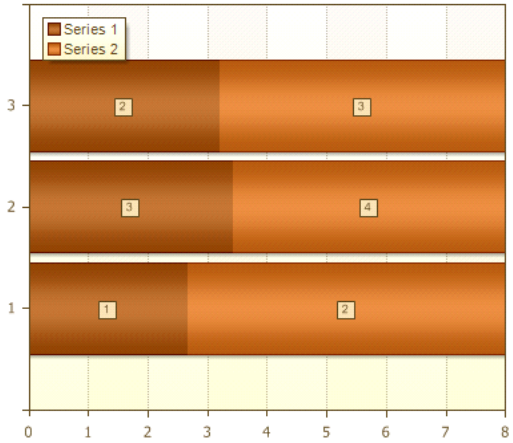


Stacked Bar

Stacked bar charts show the relationship of individual items to the whole.



Full-Stacked Bar
This type of charts allows comparing percentage of each value with the total inside the category.

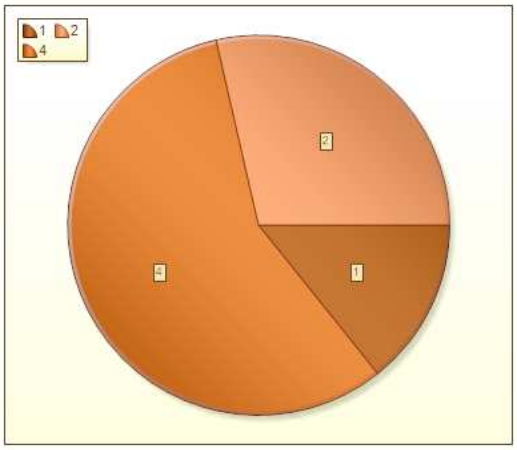


Pie Area

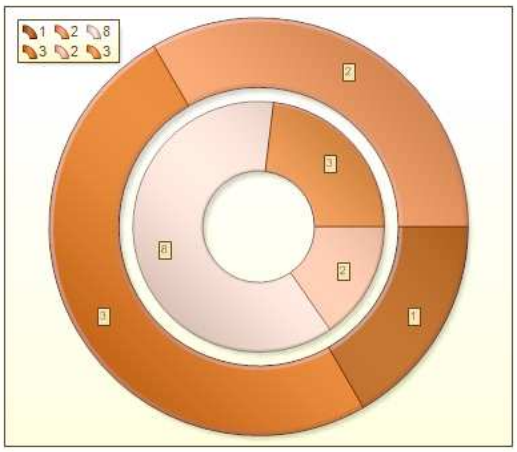
Data that is arranged in one column or row only on a worksheet can be plotted in a pie chart. In a pie chart, the arc length of each sector, is proportional to the quantity it represents. Together, the sectors create a full disk.
Pie charts should be used if:

- It is required to show one row of data;
- All values are positive and greater than 0;
- A values belongs to no more than 7 categories;
- Categories corresponds to some parts of the whole chart disk.

Pie
Pie charts display the contribution of each value to a total. It is possible to manually pull out the slices of a pie chart to emphasize them.



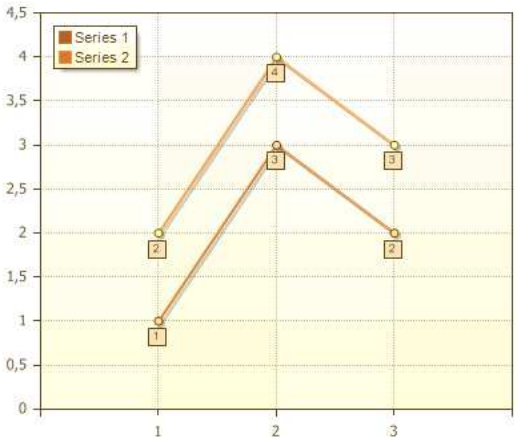
Doughnut
A doughnut chart is functionally similar to a pie chart, with the exception of a blank center and the ability to support multiple statistics as one.



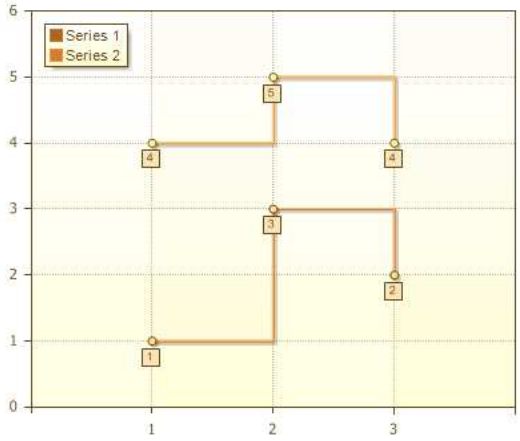
Line Area

Data that is arranged in columns or rows on a worksheet can be plotted in a line chart. Line charts can display continuous data over time, set against a common scale, and are therefore ideal for showing trends in data at equal intervals.

Line
Line and line with markers are used to indicate individual data values, line charts are useful to show trends over time or ordered categories, especially when there are many data points and the order in which they are presented is important.

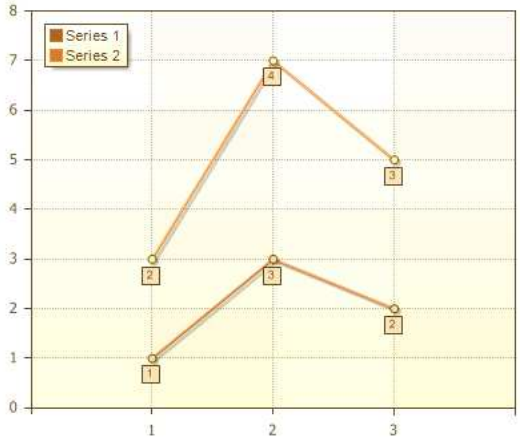


Stepped Line

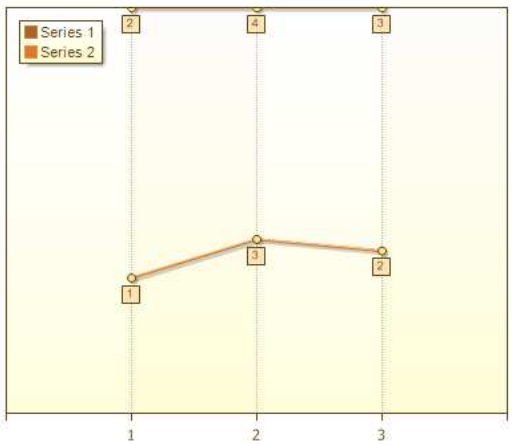


Stacked Line

Displayed with or without markers to indicate individual data values, stacked line charts are useful to show the trend of the contribution of each value over time or ordered categories. If there are many categories or the values are approximate, you should use a stacked line chart without markers.

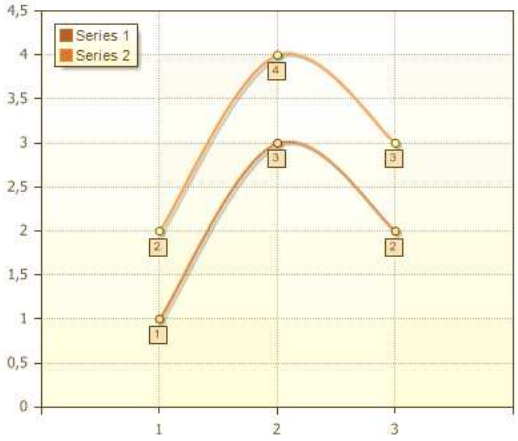


Full-Stacked Line

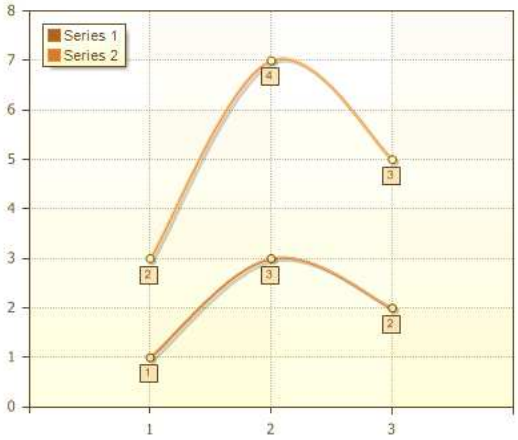


Spline Area

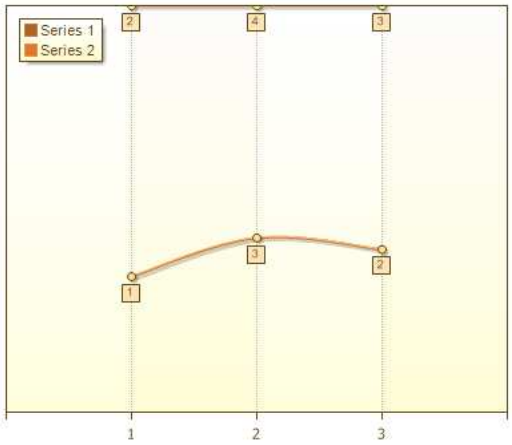
Spline



Stacked Spline



Full-Stacked Spline

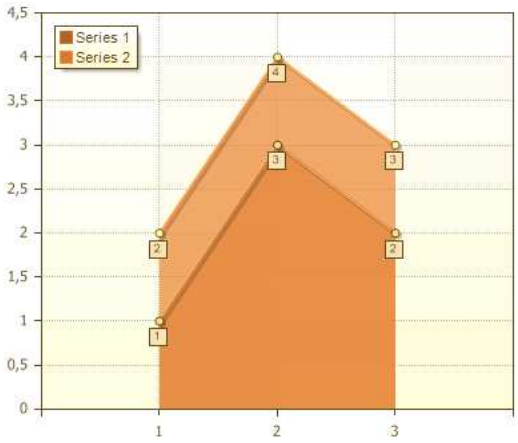


Area

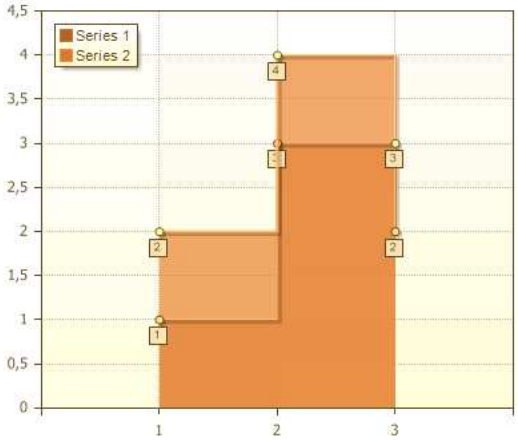
Data that is arranged in columns or rows on a worksheet can be plotted in an area chart. Area charts illustrate changes depending on time period and can be used to attract attention to summary value in compliance with trend. For example, data which shows profit depending on time can be created in Area charts to attract attention to total profit.

Area

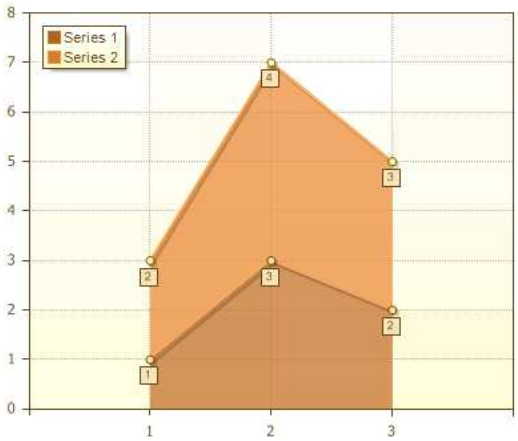
Area charts display the trend of values over time or categories.



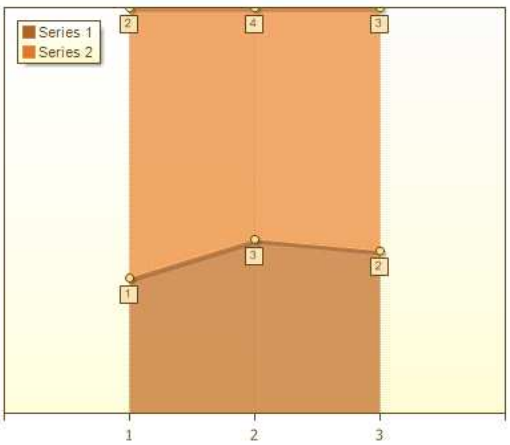
Stepped Area



Stacked Area
Stacked area charts display the trend of the contribution of each value over time or categories.

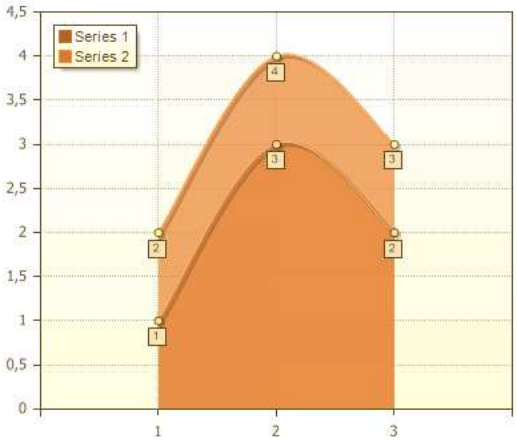


Full-Stacked Area
Full-Stacked Area charts display the trend of the percentage each value contributes over time or categories.

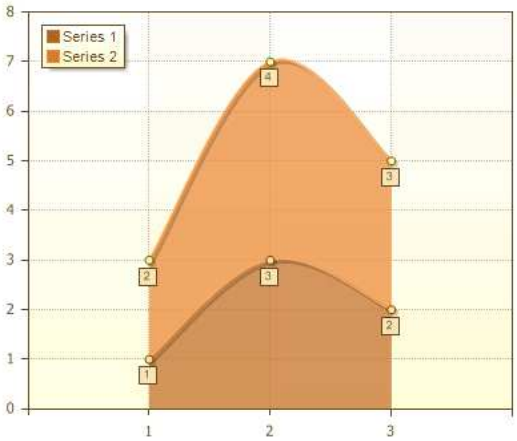


Spline Area

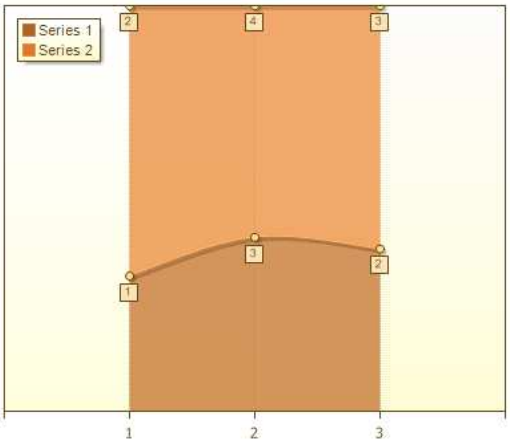
Spline Area



Stacked Spline Area



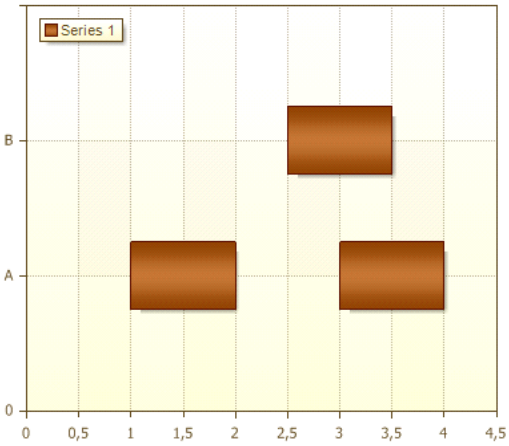
Full-Stacked Spline Area



Gantt Area

A Gantt chart is a type of bar chart that illustrates a project schedule.

Gantt
Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project.

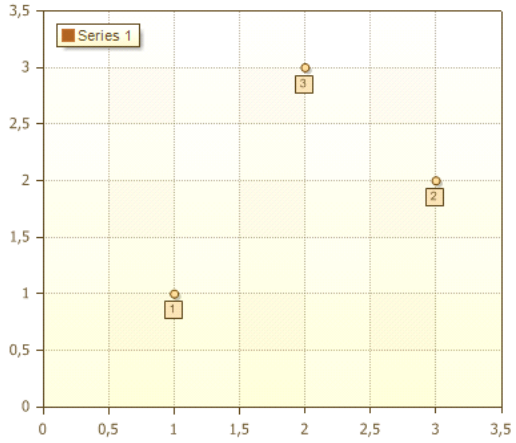


Scatter Area

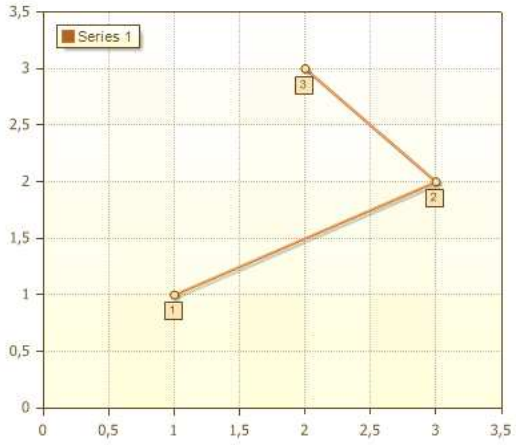
Data that is arranged in columns and rows on a worksheet can be plotted in an xy (scatter) chart. Scatter charts show the relationships among the numeric values in several data series, or plots two groups of numbers as one series of xy coordinates.
A scatter chart has two value axes, showing one set of numerical data along the horizontal axis (x-axis) and another along the vertical axis (y-axis). It combines these values into single data points and displays them in uneven intervals, or clusters. Scatter charts are commonly used for displaying and comparing numeric values, such as scientific, statistical, and engineering data.
Scatter charts should be used if:

- It is required to change the scale of the horizontal axis;
- Values for horizontal axis are not evenly spaced;
- There are many data points on the horizontal axis;
- It is required to show similarities between large sets of data instead of differences between data points;
- It is required to compare large numbers of data points without regard to time - the more data that you include in a scatter chart, the better the comparisons that you can make.

Scatter
This type of chart compares pairs of values. Use a scatter chart without lines when you have data in a specific order.



Scatter Line
This type of chart can be displayed with or without straight connecting lines between data points. These lines can be displayed with or without markers.



Scatter Spline
This type of chart can be displayed with or without a smooth curve connecting the data points. These lines can be displayed with or without markers. Use the scatter chart without markers if there are many data points.

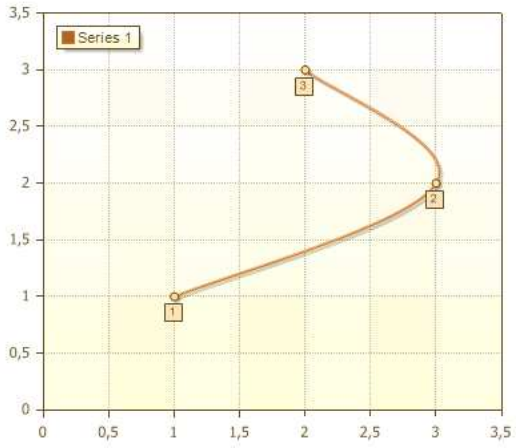


Chart Properties

Main Properties

1. Chart

Property name	Description
Chart Type	Chart Type
Area	Properties of the current area
Legend	Properties of the chart legend
Series	A collection of chart series
Series Labels	Properties of series labels
Style	Sets a Chart style

2. Chart Additional

Property name	Description
Constant Lines	Sets a collection of constant lines of chart
Process at End	Sets that a chart is processed at the end of the report execution
Rotation	Sets a rotation angle of a chart
Horizontal Spacing	Sets horizontal spacing between the chart area and axis area
Vertical Spacing	Sets vertical spacing between the chart area and axis area
Strips	Sets a collection of chart strips
Title	Sets chart title properties

3. Data

Property name	Description
Data Source	Get data source that is used for getting data
Data Relation	Get the link that is used for master-detail reports rendering
Master Component	Gets or sets the master component
Count Data	Gets or sets the count of rows for virtual data
Filter On	Gets or sets value indicates, that the filter is on
Filters	Gets or sets a collection of filters of chart data
Sort	Gets or sets the array of strings that describes rules of sorting

4. Position

Property name	Description
Left	Gets or sets the distance, between the left edge of the component and the left edge of its container's client area
Top	Gets or sets top position of the component

Width	Gets or sets width of the component
Height	Gets or sets height of the component
Min Size	Gets or sets minimal size
Max Size	Gets or sets maximal size

5. Appearance

Property name	Description
Brush	Gets or sets a brush to fill a component
Border	Gets or sets frame of the component
Conditions	Gets or sets a component condition
Use Parent Styles	Gets or sets a value which indicates that this component must use styles from parent component

6. Behavior

Property name	Description
Grow to Height	Gets or sets value which indicates that the height of this component increases/decreases to the bottom of a container
Dock Style	Gets or sets a type of the component docking
Enabled	Gets or sets a value which indicates will this component be available
Interaction	
Printable	Gets or sets value which indicates whether a component is printable
Print on	Gets or sets value which indicates on which pages component will be printed
Shift Mode	Gets or sets value which indicates the shift mode of a component

7. Design

Property name	Description
Name	Gets or sets a component name
Alias	Gets or sets a text that will be shown instead of a component name. If the text is not indicated then the name is shown
Restrictions	Gets or sets value which indicates the restrictions of a component
Locked	Gets or sets a value which indicates that moving is locked
Linked	Gets or sets value, indicates that the object snap to the container is turned on

Axis Area Properties

Property name	Description

Brush	Gets or sets a brush to fill area
Border Color	Gets or sets border color of area
Color Each	Gets or sets value which indicates that each series is drawn by its own color
Grid Lines Horizontal	Gets or sets horizontal grid lines on left axis
Grid Lines Horizontal Right	Gets or sets horizontal grid lines on right axis
Grid Lines Vertical	Gets or sets grid lines on vertical axis
Interlacing Horizontal	Gets or sets interlacing settings on horizontal axis
Interlacing Vertical	Gets or sets interlacing settings on vertical axis
Reverse Horizontal	Gets or sets value which indicate that all values on horizontal axis is reverse
Reverse Vertical	Gets or sets value which indicate that all values on vertical axis is reverse
Show Shadow	Gets or sets value which indicates whether it is necessary to draw shadow
X Axis	Gets or sets settings of XAxis
X Top Axis	Gets or sets settings of XTopAxis
Y Axis	Gets or sets settings of YAxis
Y Right Axis	Gets or sets settings of YRightAxis

Pie Area Properties

Property name	Description
Brush	Gets or sets a brush to fill area
Border Color	Gets or sets border color of area
Show Shadow	Gets or sets value which indicates whether it is necessary to draw shadow

Doughnut Area Properties

Property name	Description
Brush	Gets or sets a brush to fill area
Border Color	Gets or sets border color of area
Color Each	Gets or sets value which indicates that each series is drawn by its own color
Show Shadow	Gets or sets value which indicates whether it is necessary to draw shadow

Legend Properties

Property name	Description
Brush	Gets or sets a brush to fill a legend
Direction	Gets or sets direction of a legend
Horizontal Alignment	Gets or sets the text horizontal alignment of a legend
Vertical Alignment	Gets or sets the vertical alignment of a legend
Marker Alignment	Gets or sets the marker alignment

Border Color	Gets or sets a border color of a legend
Columns	Gets or sets a columns count of a legend
Font	Gets or sets a font of a legend
Horizontal Spacing	Gets or sets horizontal spacing from a legend border
Labels Color	Gets or sets a color of a legend text
Marker Size	Gets or sets marker size
Marker Visible	Gets or sets visibility of marker
Show Shadow	Gets or sets value which indicates whether it is necessary to draw shadow
Size	Gets or sets legend size
Title	Gets or sets legend title
Title Color	Gets or sets legend color
Title Font	Gets or sets legend font
Vertical Spacing	Gets or sets vertical spacing from a legend border
Visible	Gets or sets whether a legend should be visible

Title Properties

Property name	Description
Alignment	Gets or sets horizontal alignment of a title
Antialiasing	Gets or sets antialiasing of a title text
Brush	Gets or sets a brush to fill a title
Dock	Gets or sets a side to which a title will be docked
Font	Gets or sets a font of a title
Spacing	Gets or sets spacing from a title
Text	Gets or sets a title text
Visible	Gets or sets whether a title should be visible

Series Labels Properties

Property Name	Description
Brush	Gets or sets a brush to fill a series labels
Font	Gets or sets a font of an series labels
Marker Alignment	Gets or sets marker alignment
Angle	Gets or sets angle of a text rotation
Antialiasing	Gets or sets antialiasing of text titles
Border Color	Gets or sets a border color of an series labels
Draw Border	Gets or sets a value that indicates whether the border for Series Labels is drawn
Format	Gets or sets a text format
Label Color	Gets or sets label color

Legend Value Type	Gets or sets legend type value
Marker Size	Gets or sets marker size
Marker Visible	Gets or sets a value that indicates whether a marker is visible
Prevent Intersection	Gets or sets a value that includes algorithm of preventing intersection with the X axis
Show on Zero Values	Gets or sets forcibly showing zero values
Step	Gets or sets a step of showing series labels
Text After	Gets or sets a text that is shown after series
Text Before	Gets or sets a text that is shown before series
Use Series Color	Gets or sets a value that indicates whether colors are set for series are used
Value Type	Gets or sets a type of parameter that will be used in a series label
Visible	Gets or sets a value that indicates visibility of series labels

Series Labels (None) Properties

Property Name	Description
Marker Alignment	Gets or sets marker alignment
Angle	Gets or sets angle of a text rotation
Draw Border	Gets or sets a value that indicates whether border for series labels should be drawn
Format	Gets or sets text formatting
Legend Value Type	Gets or sets legend value type
Marker Size	Gets or sets marker size
Marker Visible	Gets or sets whether a marker should be visible
Show on Zero Values	Gets or sets force showing zero values
Step	Gets or sets a step of showing series labels
Text After	Gets or sets a text that is shown after series
Text Before	Gets or sets a text that is shown before series
Use Series Color	Gets or sets a value that indicates whether colors are set for series are used
Value Type	Gets or sets a type of parameter that will be used in a series label

Series Labels (Outside) Properties

Property Name	Description
Brush	Gets or sets a brush to fill a series labels
Font	Gets or sets a font of an series labels
Marker Alignment	Gets or sets marker alignment
Angle	Gets or sets angle of a text rotation
Antialiasing	Gets or sets antialiasing of Series Labels
Border Color	Gets or sets a border color of series labels

Draw Border	Gets or sets a value that indicates whether border for series labels should be drawn
Format	Gets or sets text formatting
Label Color	Gets or sets label color
Legend Value Type	Gets or sets legend value type
Line Length	Gets or sets length of a connecting line of a series label
Marker Size	Gets or sets marker size
Marker Visible	Gets or sets whether a marker should be visible
Show on Zero Values	Gets or sets force showing zero values
Step	Gets or sets a step of showing series labels
Text After	Gets or sets a text that is shown after series
Text Before	Gets or sets a text that is shown before series
Use Series Color	Gets or sets a value that indicates whether colors are set for series are used
Value Type	Gets or sets a type of parameter that will be used in a series label
Visible	Gets or sets a value that indicates visibility of series labels

Axis Properties

Property Name	Description
Arrow Style	Gets or sets arrow style
Labels	Gets or sets labels
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Range	Gets or sets range
Show Edge Values	Gets or sets show edge values
Start From Zero	Gets or sets a value that indicates how a chart should be shown on the chart area
Step	Gets or sets step
Ticks	Gets or sets ticks
Title	Gets or sets a title
Visible	Gets or sets a value that indicates visibility of axis

Grid Lines Properties

Property Name	Description
Color	Gets or sets color
Minor Color	Gets or sets minor ticks color
Minor Count	Gets or sets minor ticks count
Minor Style	Gets or sets minor ticks style

Minor Visible	Gets or sets minor ticks visibility
Style	Gets or sets style
Visible	Gets or sets visibility

Interlacing Properties

Property Name	Description
Interlaced Brush	Gets or sets Interlaced Brush
Visible	Gets or sets visibility

Clustered Column, Clustered Bar, Stacked Column, Full-Stacked Column, Stacked Bar Series Properties

1. Data

Property Name	Description
Conditions	Gets or sets a collection of conditions
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets section of sorting data by values, arguments, of without sorting
Sort Direction	Gets or sets sort direction
Auto Series Key Data Column	Gets or sets a data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Border Color	Gets or sets a border color
Brush	Gets or sets a series brush
Show Shadow	Gets or sets a shadow

3. Behavior

Property Name	Description
Show Zeros	Gets or sets a value that visualizes zero values of series
Width	Gets or sets a series column width
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Line, Scatter Line Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets a direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Labels Offset	Gets or sets vertical offset of labels in relation to its first position
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets series shadow

3. Behavior

Property Name	Description
Show Marker	Gets or sets marker showing

Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Stepped Line Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Labels Offset	Gets or sets vertical offset of labels in relation to its first position
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color

Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Point at Center	Gets or sets showing a value by the center of a line
Show Marker	Gets or sets marker showing
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Spline, Scatter Spline Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Labels Offset	Gets or sets vertical offset of labels in relation to its first position
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Show Marker	Gets or sets marker showing
Tension	Gets or sets tension of a line
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Area Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting

Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Brush	Gets or sets a brush
Labels Offset	Gets or sets vertical offset of labels in relation to its first position
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Show Marker	Gets or sets marker showing
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Stepped Area Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Brush	Gets or sets a brush
Labels Offset	Gets or sets vertical offset of labels in relation to its first position
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Point at Center	Gets or sets showing a value by the center of a line
Show Marker	Gets or sets marker showing
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}

List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Spline Area Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Brush	Gets or sets a brush
Labels Offset	Gets or sets vertical offset of labels in relation to its first position
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Show Marker	Gets or sets marker showing
Tension	Gets or sets tension of a line
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series

Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Stacked Line, Full-Stacked Line Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Show Marker	Gets or sets marker showing
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Stacked Spline, Full-Stacked Spline Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style

Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Show Marker	Gets or sets marker showing
Tension	Gets or sets tension of a line
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Stacked Area, Full-Stacked Area Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created

	series

2. Appearance

Property Name	Description
Brush	Gets or sets a brush
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Show Marker	Gets or sets marker showing
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Stacked Spline Area, Full-Stacked Spline Area Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters

Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Brush	Gets or sets a brush
Lighting	Gets or sets sets line lighting
Line Color	Gets or sets line color
Line Style	Gets or sets line style
Line Width	Gets or sets line width
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Show Marker	Gets or sets marker showing
Tension	Gets or sets tension of a line
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}

List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Pie, Doughnut Series Properties

1. Data

Property Name	Description
Conditions	Gets or sets a collection of conditions
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Border Color	Gets or sets series border color
Brush	Gets or sets a brush
Diameter	Gets or sets static diameter of a chart. If the value is zero, then the diameter will be calculated automatically
Lighting	Gets or sets sets line lighting
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Start Angle	Gets or sets the start angle of chart drawing
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title
Distance	Gets or sets a distance to pull out a chart slice
Cut Pie List	Gets or sets a list of pulled out slices

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}

List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Gantt Series Properties

1. Data

Property Name	Description
Conditions	Gets or sets a collection of conditions
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Border Color	Gets or sets a border color
Brush	Gets or sets a brush
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description
Show Zeros	Gets or sets a value that visualizes zero values of series
Width	Gets or sets a series column width
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

6. Value End

Property Name	Description
Value Data Column End	Gets or sets a data column name that indicates a value of data
Value End	Gets or sets a expression of the end value. For example: {Order.Value}
List of Values End	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Scatter Series Properties

1. Data

Property Name	Description
Filters	Gets or sets a collection of filters
Format	Gets or sets a data format in what series labels will be shown
Sort by	Gets or sets a selection of data sorting by values, arguments, or without sorting
Sort Direction	Gets or sets direction of sorting
Auto Series Key Data Column	Gets or sets data column name with the key-value that is used to create series automatically
Auto Series Color Data Column	Gets or sets a data column name that defines color of automatically created series. The color should be represented as a string
Auto Series Title Data Column	Gets or sets a data column name that defines a title of automatically created series

2. Appearance

Property Name	Description
Labels Offset	Gets or sets vertical offset of labels in relation to its first position
Marker Color	Gets or sets marker color
Marker Size	Gets or sets marker size
Marker Type	Gets or sets marker type
Show Shadow	Gets or sets whether a shadow must be shown

3. Behavior

Property Name	Description

Show Marker	Gets or sets marker showing
Y Axis	Gets or sets axis to what a series is assigned
Series Labels	Gets or sets series labels for this series
Show in Legend	Gets or sets value that allows showing series label in a legend
Show Series Labels	Gets or sets which type of series labels will be used: from chart settings or from settings from the series
Title	Gets or sets a series title

4. Argument

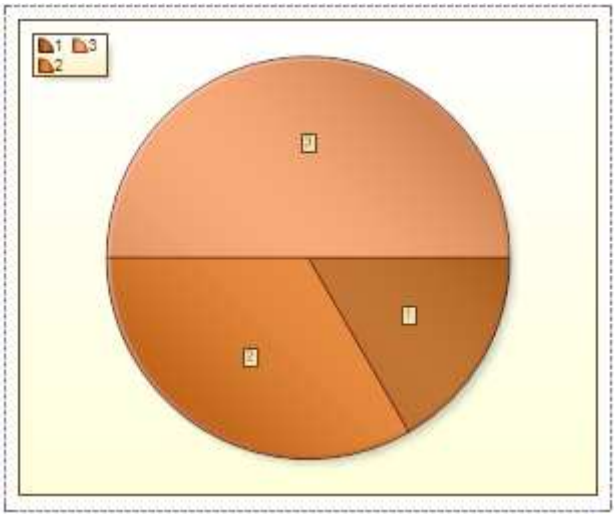
Property Name	Description
Argument Data Column	Gets or sets a data column name that indicates a value of an argument
Argument	Gets or sets an expression of an argument. For example: {Order.Argument}
List of Arguments	Gets or sets an expression that indicates a list of arguments. For example: 1;2;3

5. Value

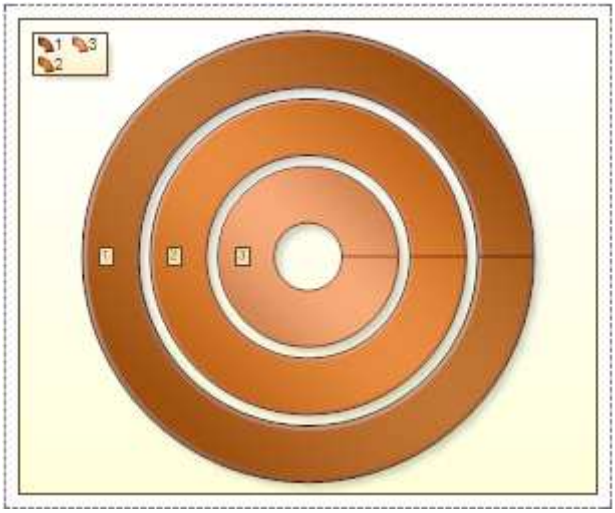
Property Name	Description
Value Data Column	Gets or sets a data column name that indicates a value of data
Value	Gets or sets an expression of a value. For example: {Order.Value}
List of Values	Gets or sets an expression that indicates a list of values. For example: 1;2;3

Area

Circular area or area without axes is a space where charts can be placed without axes. A circular area includes the main elements of the chart: series, chart title and a legend. In the area without axes the following chart types may be placed: **Pie** and **Doughnut**. The difference between these types of charts is that, for Pie type of a chart, rows are arranged in series. And for the Doughnut chart - rings. The picture below shows an example of a Pie chart, with three series:

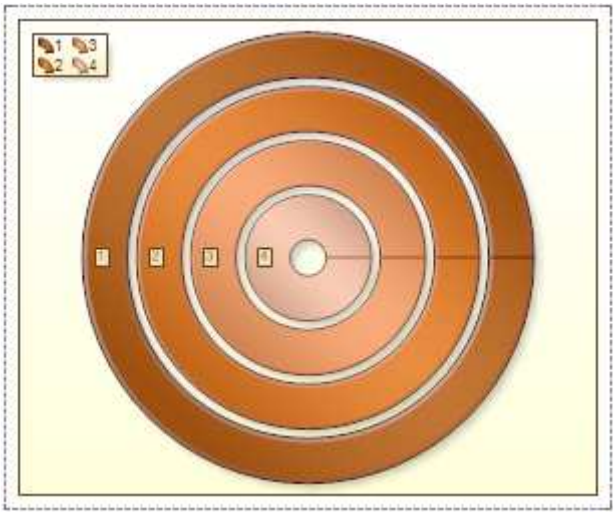


As can be seen from the picture, the series are arranged consecutively in a clockwise direction. In the Doughnut chart, the number of rows will match the number of rings. The picture below shows an example of a chart that has three rows:



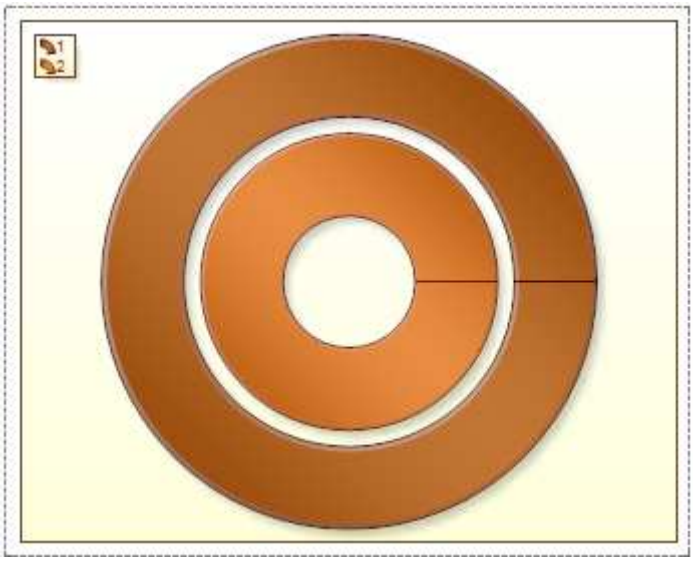
Doughnut

Doughnut chart is circular chart divided into sectors. It has a blank center and the ability to support multiple statistics as one. Doughnut illustrates proportion. On the picture below the doughnut chart sample is represented:

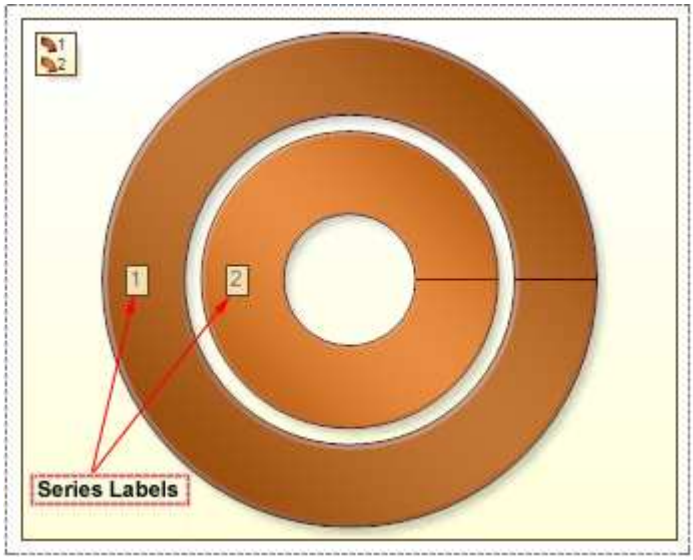


Series Labels

Series Labels can only be placed in the center on the doughnut chart. The **Series Labels** may have two values: **None** and **Center**. If the **Series Labels** property is set to **None**, then labels are not shown. The picture below shows the doughnut with no labels:

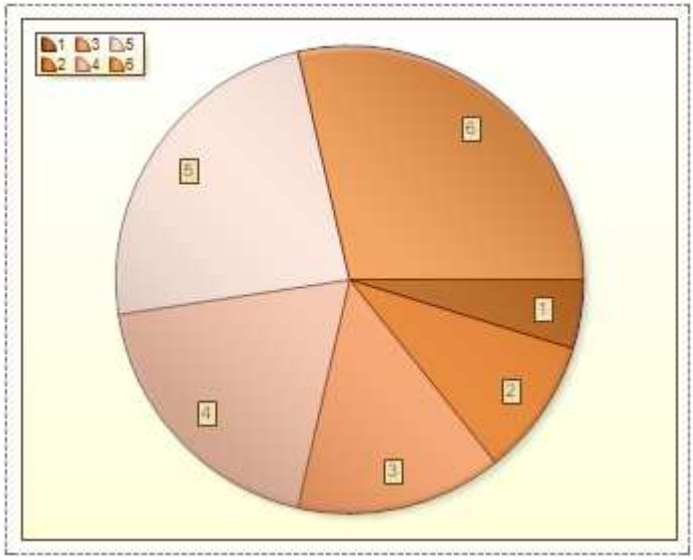


If the **Series Labels** property is set to **Center**, then labels are shown in the center of the chart ring. The picture below shows the doughnut with labels:



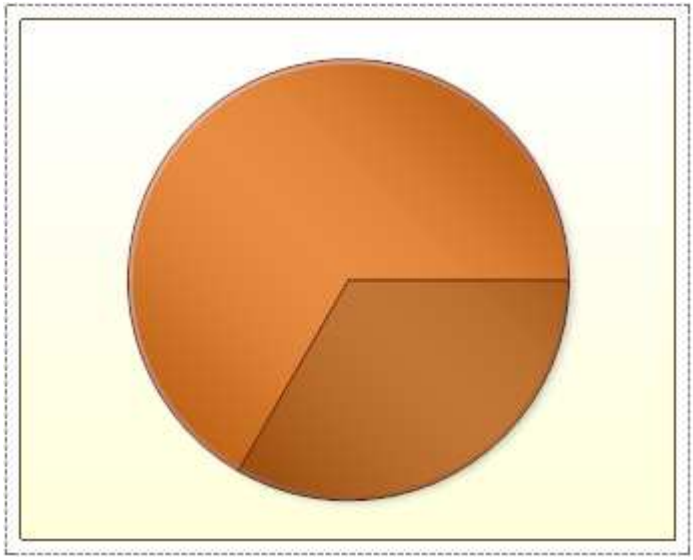
Pie

A **Pie** chart (or a circle graph) is circular chart divided into sectors, illustrating proportion. Each Series is a part of chart. In a pie chart, each sector, is proportional to the quantity it represents. Together, the sectors create a full disk.

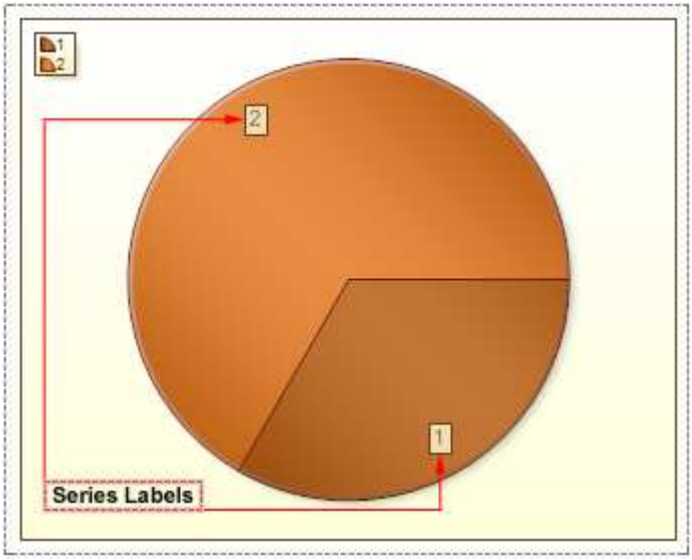


The location series labels, in the pie chart, depends on the value of the **SeriesLabels** property. This property may take the following values: None, Inside End, Center, Outside, Two Columns.

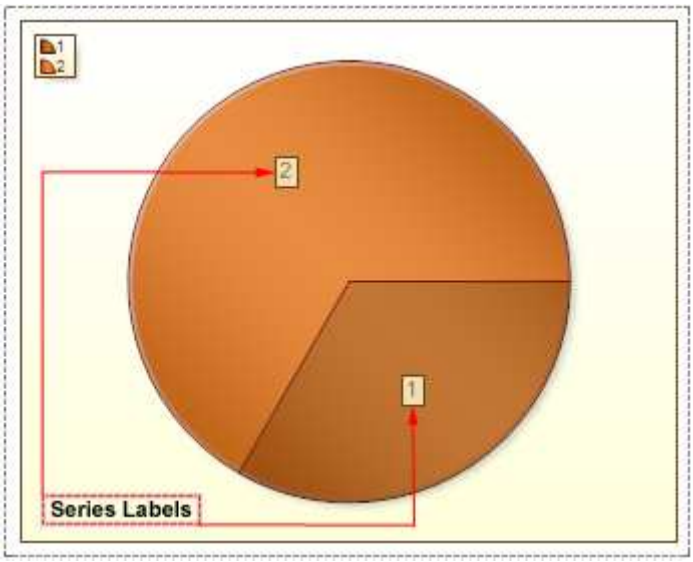
1. **None**. Series Labels are not shown. The picture below shows an example of a Pie chart with the **Series Labels** set to **None**:



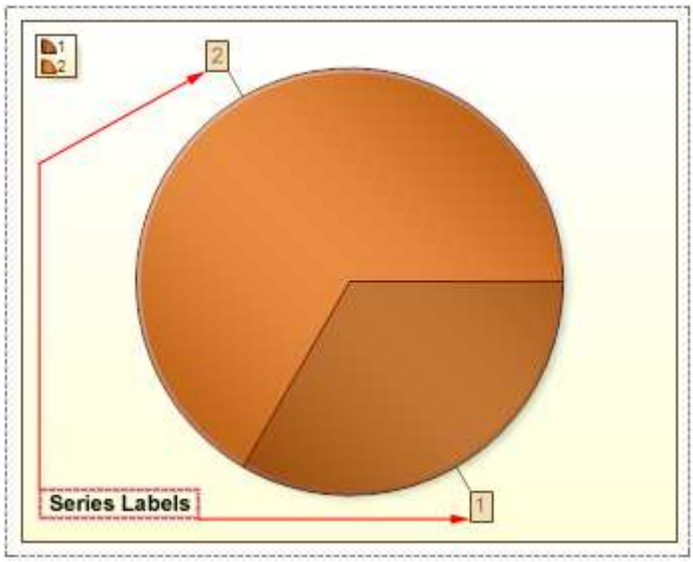
2. **Inside End**. Series Labels are displayed inside the slice and far from the center. The picture below shows an example of a Pie chart with the **Series Labels** set to **Inside End**:



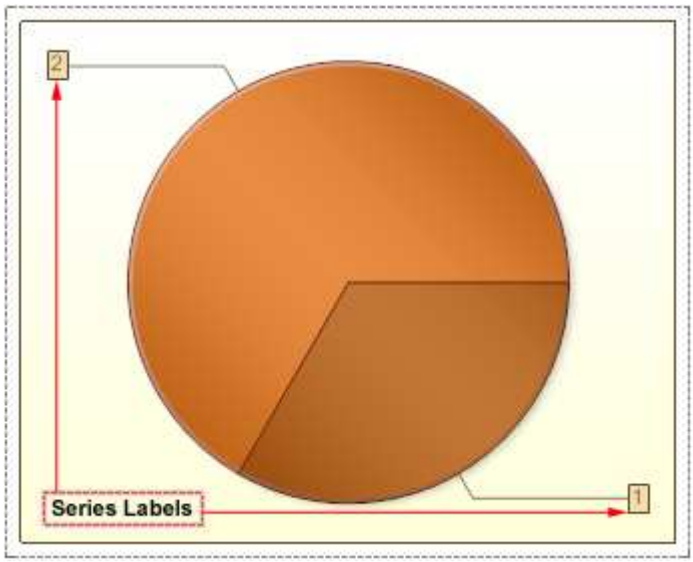
3. **Center**. Series Labels are displayed in the center of the slice. The picture below shows an example of a Pie chart with the **Series Labels** set to **Center**:



4. **Outside**. Series Labels are displayed outside the chart, but in a Pie area. The picture below shows an example of a Pie chart with the **Series Labels** set to **Outside**:

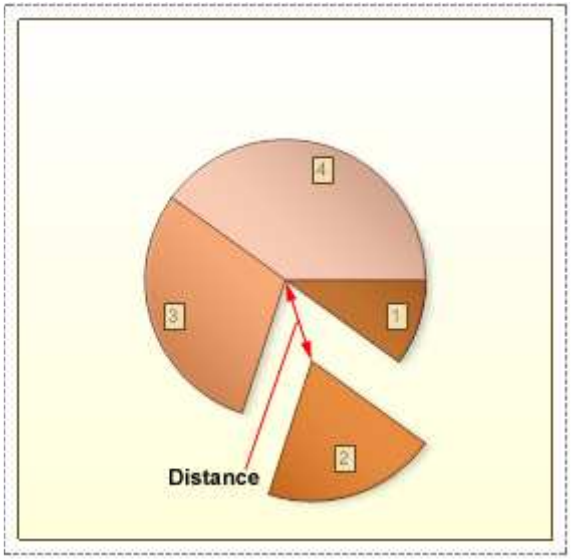


5. **Two Columns**. Series Labels are displayed outside the chart in two columns: on the left and right of the chart. The picture below shows an example of a Pie chart with the **Series Labels** set to **Two Columns**:

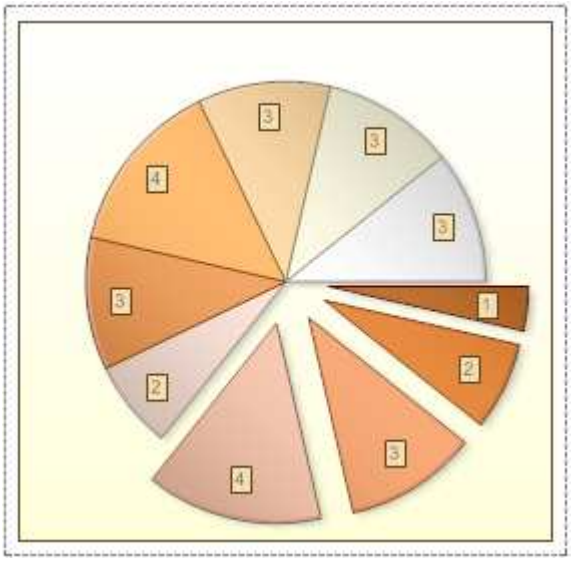


CutPieList Property

The Pie chart represents an opportunity to display the contribution of each value to a total while emphasizing individual values. To select a segment in a pie chart select and pull out, it is necessary, in the **Series Editor**, to specify values for the **Distance** and **CutPieList** properties of a series. The **Distance** property indicates is the distance from the center of the chart to the nearest point of the pull out segment. The **CutPieList** property has a list of series to be pulled out, separated with **';'**. The picture below shows an example of a pie chart, with the second slice of the first series pulled out. The distance is 60-hundredths of inches:



If the field of the **CutPieList** property is filled, and the field of the **Distance** property is not filled, then the segments will not be pulled out. If the field of the **Distance** property is filled, and the field **CutPieList** property is not filled, then all segments of this series will be pulled out to the distance, which corresponds to the value of the **Distance** property. The picture below an example of a chart with all segments of the series 1 being pulled out, because the field of the **CutPieList** property was not filled, and the **Distance** property set to 30-hundredths of an inch:

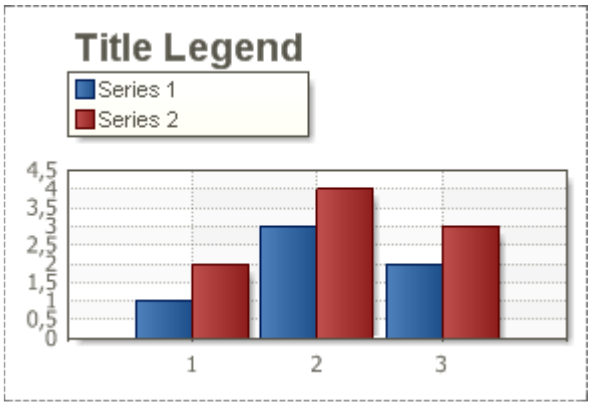


Legend

The chart may include a legend. A legend contains a list of the variables appearing in the chart and an example of their appearance. This information allows the data from each variable to be identified in the chart. The legend can be placed at any part of the chart.

Title Property

The **Title** property of the Legend allows setting the Legend title. The full path to this property is **Legend.Title**. If the the field of the **Title** property is not filled then the Legend title is not shown. The **Title** is shown over the Legend. The picture below shows a sample of the Chart with Legend where the "Title Legend" is the Legend title:

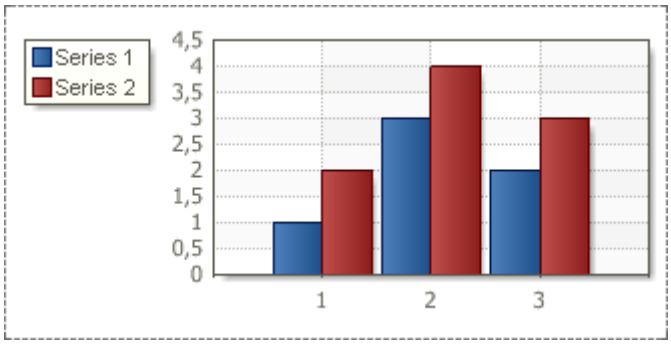


The **Title** property has the following properties:
- **TitleColor** - sets the Title color;
- **TitleFont** - sets the Title font size and font style.

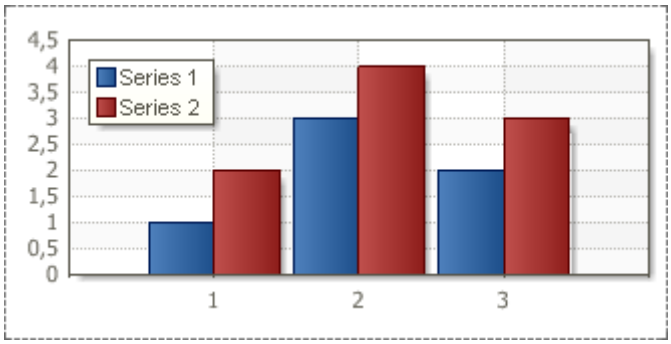
HorizontalAlignment Property

The **HorizontalAlignment** property of the Legend allows aligning the Legend position horizontally. The full path to this property is **Legend.HorizontalAlignment.** The property has the following values: **Left Out Side**, **Left**, **Center**, **Right**, **Right Out Side**.

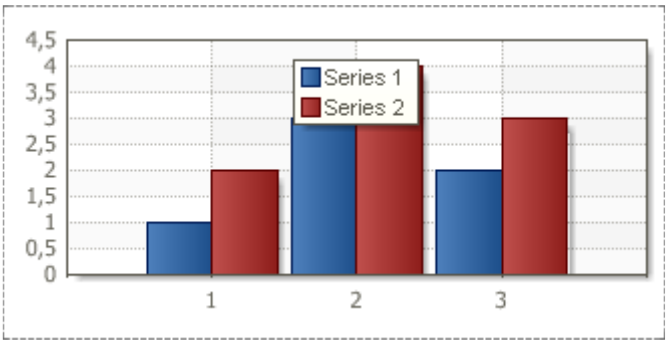
Description of values:
1. **Left Out Side**. The legend will be placed outside the Chart area on the left. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Left Out Side**:



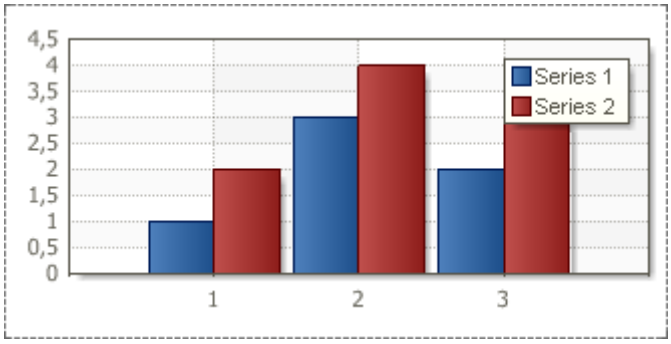
2. **Left**. The legend will be placed inside the Chart area on the left. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Left**:



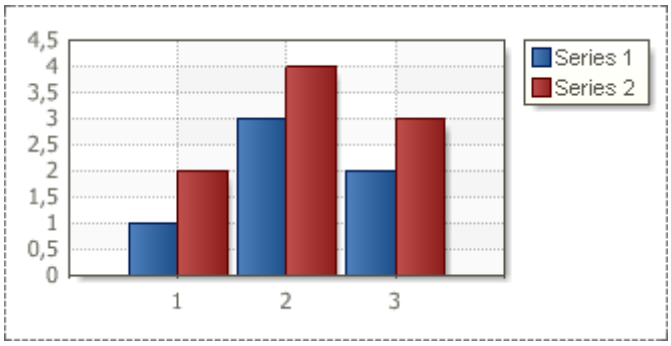
3. **Center.** The legend will be placed inside the Chart area in the center. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Center**:



4. **Right**. The legend will be placed inside the Chart area on the right. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Right**:



5. **Right Out Side**. The legend will be placed out side the Chart area on the right. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Right Out Side**:

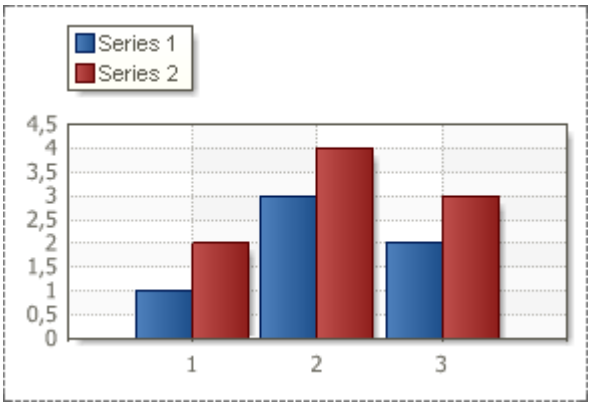


By default the **HorizontalAlignment** property is set to **Left**.

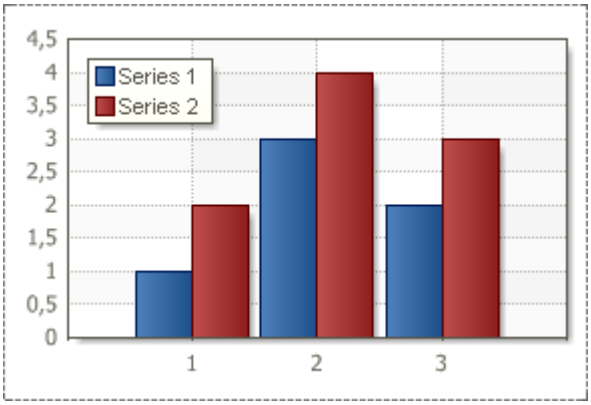
VerticalAlignment Property

The **Vertical Alignment** property of the Legend allows aligning the Legend position vertically. The full path to this property is **Legend.VerticalAlignment.** The property has the following values: **Top Out Side**, **Top**, **Center**, **Bottom**, **Bottom Out Side**.

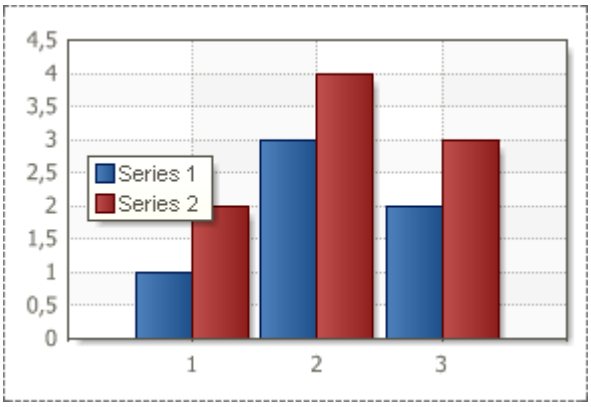
Description of values:
1. **Top Out Side**. The legend will be placed above and outside the Chart area. The picture below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Top Out Side**:



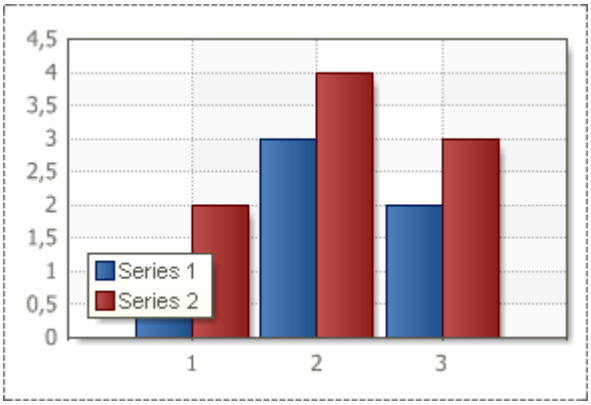
2. **Top**. The legend will be placed inside the Chart area on the top. The picture below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Top**:



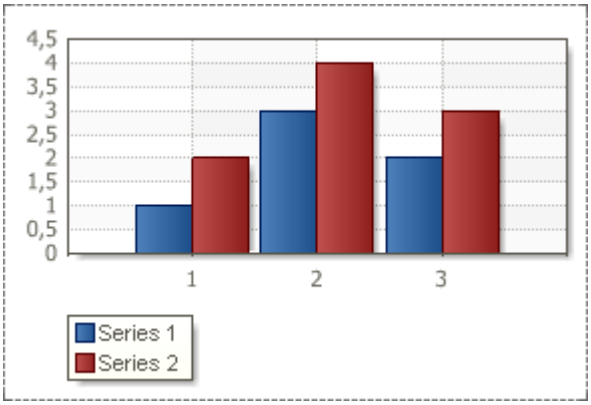
3. **Center**. The legend will be placed inside the Chart area and vertically in the center. The picture below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Center**:



4. **Bottom**. The legend will be placed inside the Chart area on the bottom. The picture below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Bottom**:



5. **Bottom Out Side**. The legend will be placed under and outside the Chart area. The picture below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Bottom Out Side**:



By default the **Vertical Alignment** property is set to **Top**.

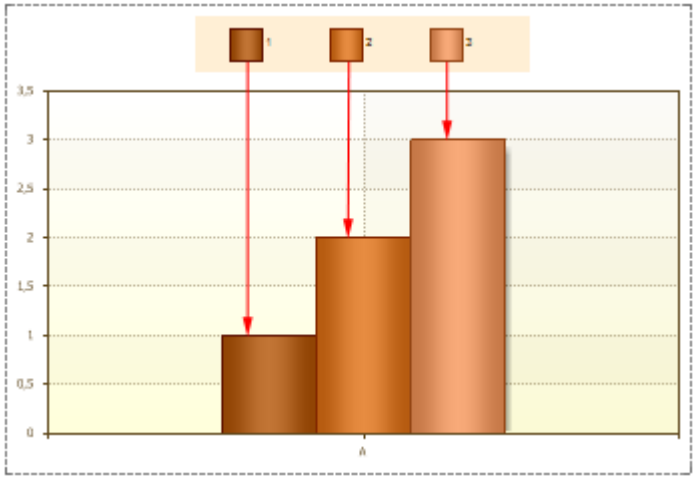
HorizontalSpacing and VerticalSpacing Properties

The **Horizontal Spacing** and **Vertical Spacing** properties allow setting the spacing (horizontal and vertical, respectively) between the Legend edge and the information on series. The full paths to these properties is **Legend.HorizontalSpacing** and **Legend.VerticalSpacing**. The picture below shows in arrows the horizontal and vertical spacing between the Legend edge and the Series 1:

These properties can take numeric values, and are required for filling. If values of the **Horizontal Spacing** and **Vertical Spacing** properties are negative, then the legend can be unreadable. The minimum value of these properties is 0.

Marker

Marker is an icon that indicates the chert row. The number of markers correspond to the number of rows. On the picture below a sample of chart with three rows and markers for them is shown:



Direction Property

The **Direction** allows selecting the order of showing markers. The full path to this property is **Legend.Direction.** The property has the following values: **Top to Bottom**, **Bottom to Top**, **Left to Right**, **Right to Left**.

Description of values:
1. **Top to Bottom**. Markers are shown in the "from top to bottom" order. The picture below shows a sample of the Legend which the **Direction** property is set to **Top to Bottom**:

2. **Bottom to Top**. Markers are shown in the "from bottom to top" order. The picture below shows a sample of the Legend which the **Direction** property is set to **Bottom to Top**:

3. **Left to Right**. Markers are shown in the "from left to right" order. The picture below shows a sample of the Legend which the **Direction** property is set to **Left to Right**:

4. **Right to Left**. Markers are shown in the "from right to left" order. The picture below shows a sample of the Legend which the **Direction** property is set to **Right to Left**:

By default the **Direction** property is set to **Top to Bottom**.

Columns Property

The **Columns** property allows changing the number of columns vertically or horizontally depending on the value of the **Direction** property. The full path to this property is **Legend.Columns**. The picture below shows a sample of the Legend which markers are split into two horizontal columns (the **Direction** property is set to **Top to Bottom**):

If to set the **Columns** property to **2**, and set the **Direction** property to **Left to Right**, then markers will be split into two vertical columns. The picture below shows a sample of the Legend which markers are split into two vertical columns (the **Direction** property is set to **Left to Right**):

The **Columns** property may have any values more than **0**. This property must be set. It cannot be left empty.

MarkerAlignment Property

The **Marker Alignment** property allows aligning markers either left or right from the "**Series**" name. The full path to this property is **Legend.Marker Alignment.** If the **Marker Alignment** property is set to **Left**, then the marker will be placed on the left from the "**series**" name. The picture below shows a sample of the Legend which the **Marker Alignment** property is set to **Left**:

If the **Marker Alignment** property is set to **Right**, then the marker will be placed on the right from the "**series**" name. The picture below shows a sample of the Legend which the **Marker Alignment** property is set to**Right**:

By default the **Marker Alignment** property is set to **Left**.

MarkerVisible Property

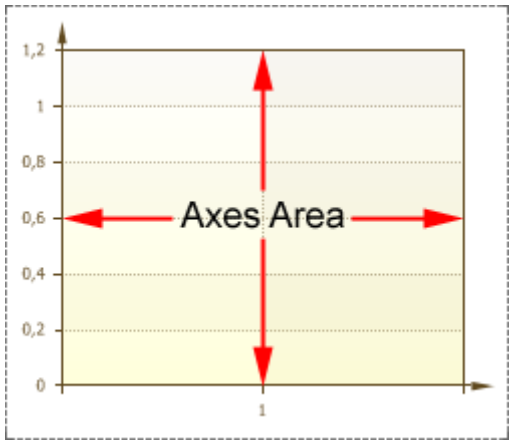
The **Marker Visible** property allows showing/hiding the legend markers. The full path to this property is **Legend.Marker Visible.** If the **Marker Visible** property is set to **true**, then markers are shown. The picture below shows a sample of the Legend which the **Marker Visible** property is set to **true**:

If the **Marker Visible** property is set to **false**, then the Legend markers are hidden. The picture below shows a sample of the Legend which the **Marker Visible** property is set to **false**:

By default the **Marker Visible** is set to **true**.

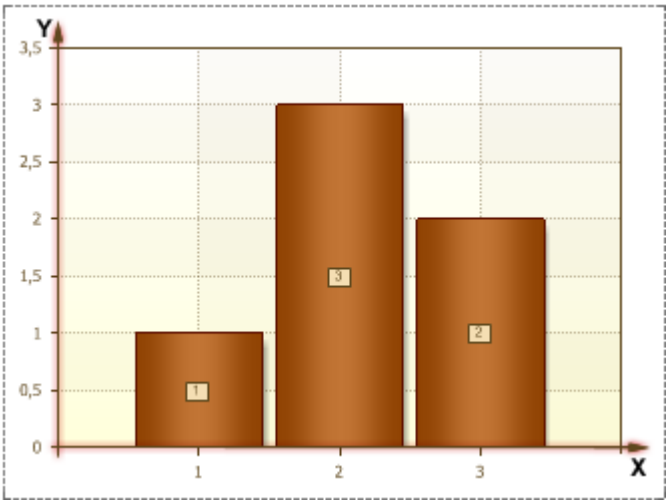
Axes Area

Axes Area is a space which includes all chart items such as data rows, axes, chart title, and legend. On the picture below the **Axes Area** is shown:

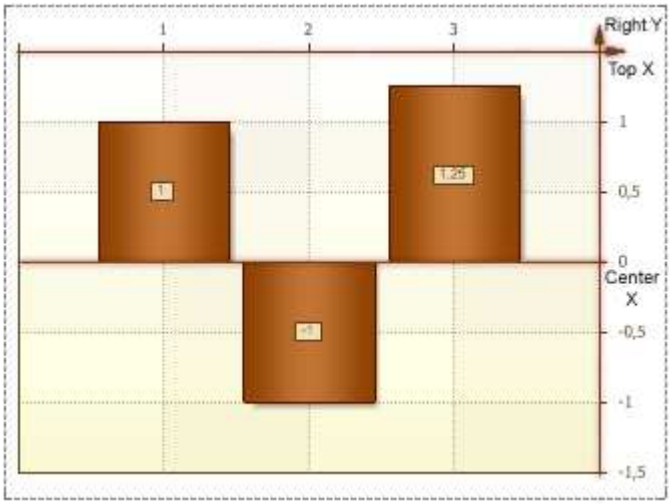


Axes

Axes Area has **X** and **Y** axes. The X axis, as a rule, is the axis of arguments, and the Y axis, is the axis of values.

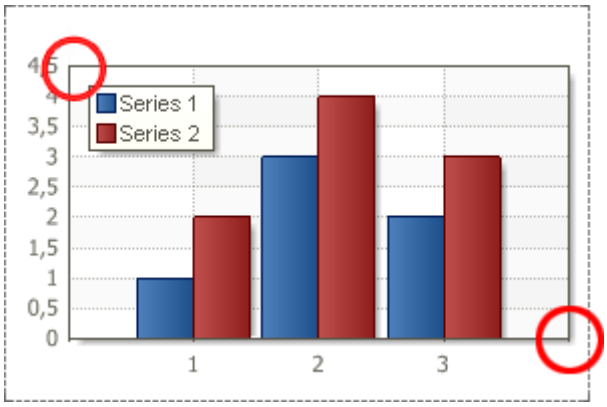


Besides, the **Axes Area** can contain top and central **X** axis, and right **Y** axis.

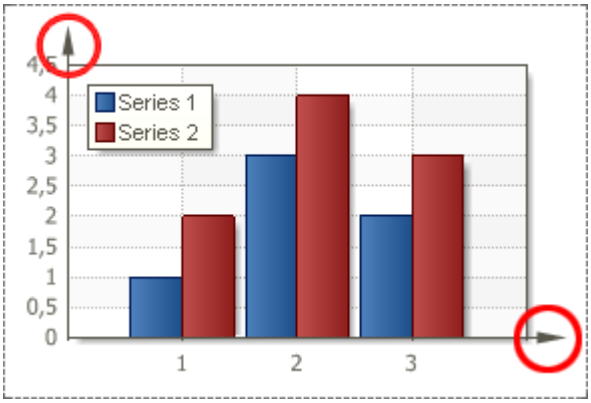


ArrowStyle Property

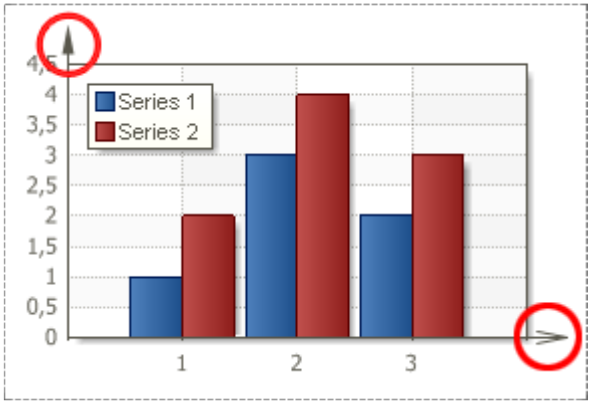
Each axis has its own direction. The direction is identified with marker (usually it is an arrow). To change the arrow style, use the **Arrow Style** property of an axis**.** The path to this property is **Area.Axes.ArrowStyle**. On the picture below the sample of a rendered chart with the **ArrowStyle** property set to the **None** default value:



As you can see, if the **ArrowStyle** property is set to **None**, then **X Y** axes do not have style. The **ArrowStyle** property can be set to **Triangle.** In this case the arrow style will look like on the picture below:



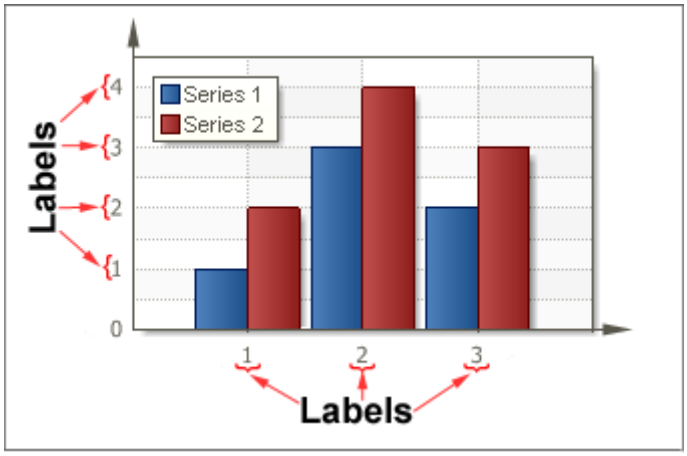
The **ArrowStyle** property can be set for each axis. Each axis may have its own values of the **Arrow Style** property. On the picture below different values of the **ArrowStyle** property of **X** and **Y** axes:



As seen from the picture above, the **ArrowStyle** property, of the **Y** axis is set to **Triangle.** And the the **ArrowStyle** property, of the **X** axis is set to **Lines.**

Labels

Labels are titles of X axis (the axis of the arguments) and Y (the axis values). Labels can take any string value. Any string value is transformed according to the selected format. If the report generator failed to convert a value to the selected format, then a direct string value is output. The picture below shows an example of a chart with arguments of Labels. The Format property is set to N:

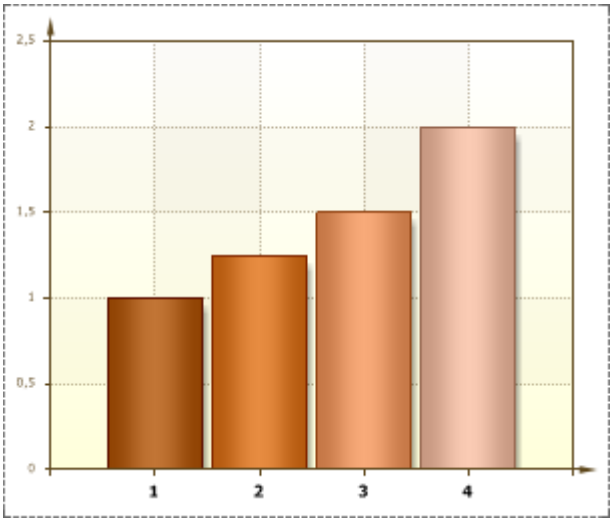


Also, Labels have a number of properties such as:

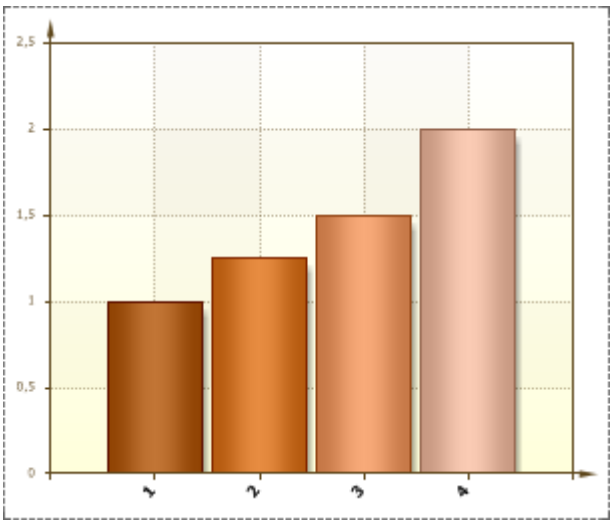
- **Angle** - sets an angle of inclination of labels;
- **Antialiasing** - sets smooth-edged type of labels;
- **Color** - sets the labels color;
- **Font** - sets the font type of labels;
- **Format** - changes the label format (numeric, percentage etc);
- **Placement** - changes the position of showing Labels;
- **Text before/Text after** - shows a text before/after Labels;
- **Text Alignment** - used for **Y** axis, aligns Labels;
- **Width** - changes the width of Label.

Angle Property

The **Angle** property is used to change the inclination of **Labels**. Specifies the angle, in degrees. The **Angle** property is set separately for each axis. The full path to this property is **Area.Axis.Labels.Angle**. By default, the value of the **Angle** property is set to **0.** So **Labels** are placed as it is shown on the picture below:

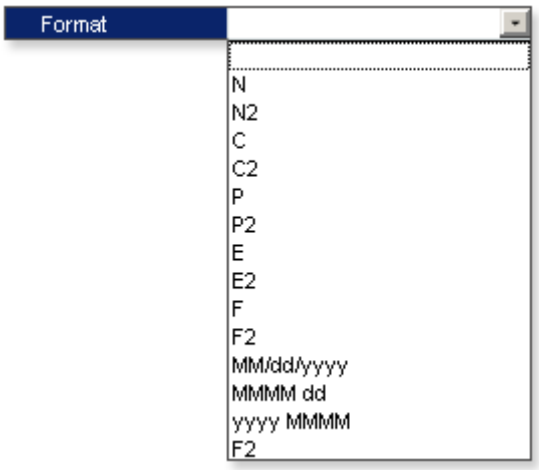


The value of this property can be negative and positive. If the value of the property is negative then Label is inclined clockwise. If the value of the property is positive then Label in inclined anticlockwise. The picture below shows the chart sample, which Angle property by the **X** axis is set to **50**:

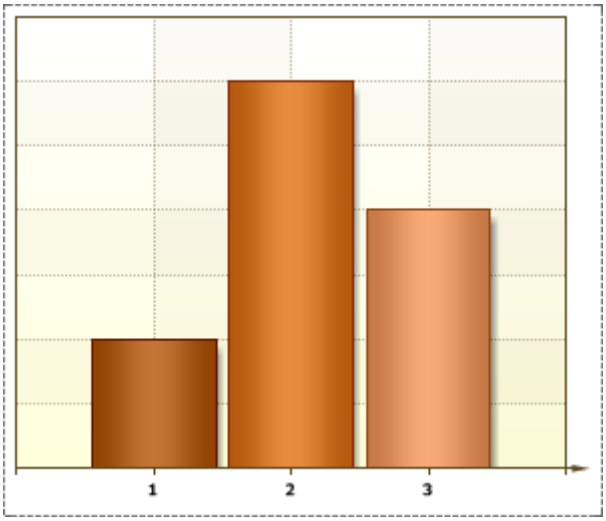


Format Property

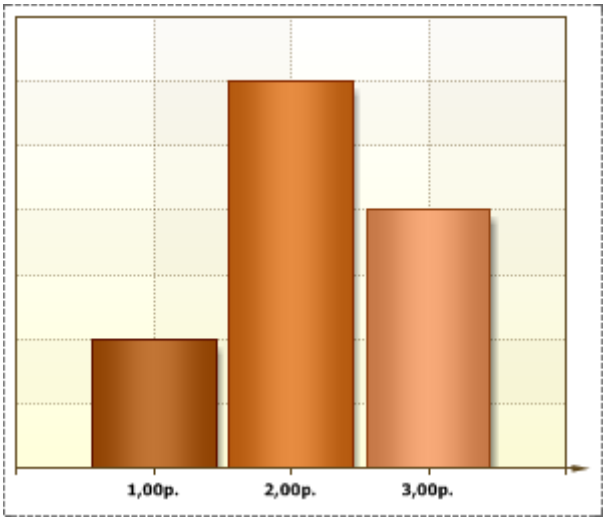
The **Format** property is used to to format the contents of Labels. The full path to this property is **Area.Axis.Labels.Format.** This property has multiple values.



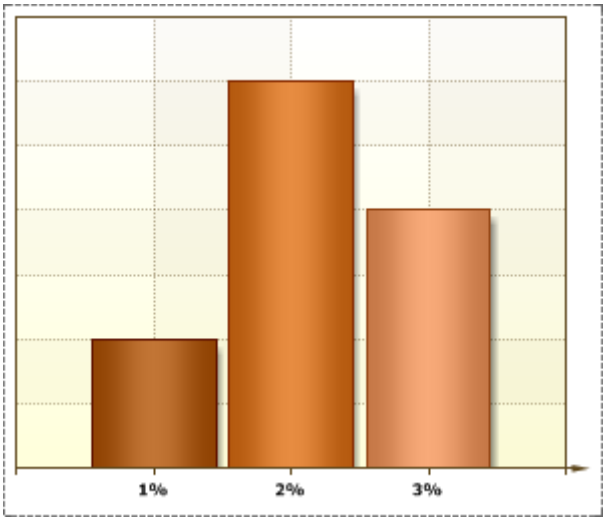
1. **Number.** The **N** value of the **Format** property is used for the general display of numbers. When filling the **Format**, after the **N** value, it is possible to specify the number of decimal places that you want to use. If no numbers are specified after **N** then decimal places will be shown only if they are present as a result of calculation. The picture below shows a chart with the **Format** property of Series Labels set to **N**:



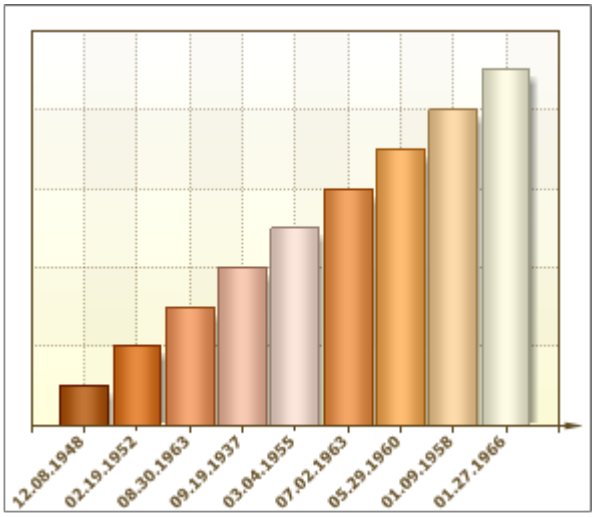
2. **Currency.** The **C** value of the **Format** property is used to display Labels with a currency symbol. After the **C** value, it is possible to specify the number of decimal places that you want to use. The picture below shows a chart with the **Format** property of Series Labels set to **C**:



3. **Percentage**. The **P** value of the **Format** property is used to display Labels with percent symbol. After the **P** value, it is possible to specify the number of decimal places that you want to use. The picture below shows a chart with the **Format** property of Series Labels set to **P**:



4 Date formatting. The **MM/dd/yyyy**, **MMMM dd**, **yyyy MMMM** values of the **Format** property convert values of arguments to date. **MM/dd/yyyy** - the date is shown like "01.20.2010", **MMMM dd** - the date is shown like "September 29", **yyyy MMMM** - the date is shown like "2010 March". The picture below shows a chart and its **Format** property is set to **MM/dd/yyyy**:

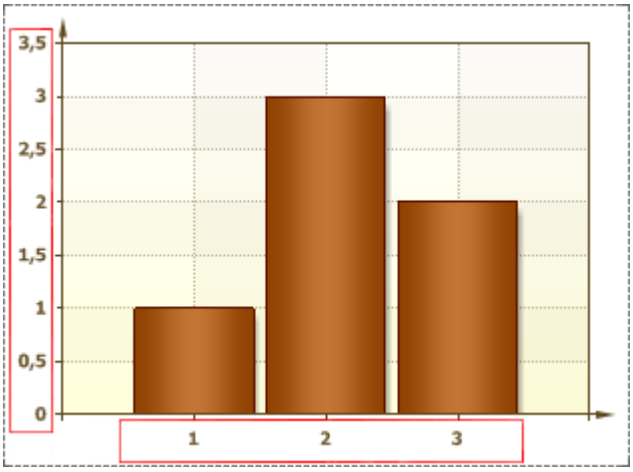


To reset the **Format** property of selected cells, and return to the default format, clear the Format by selecting empty field.

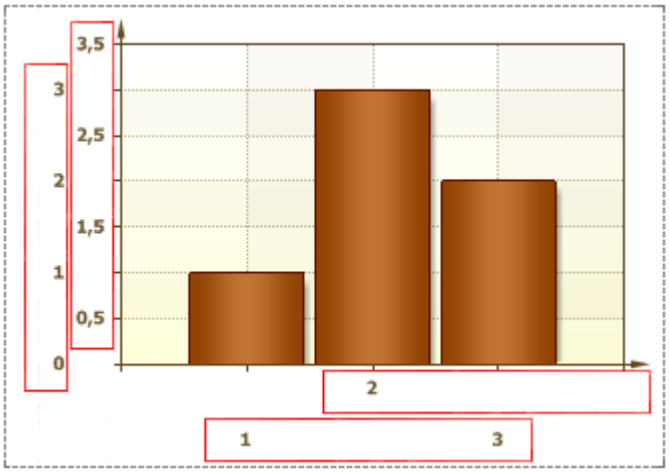
Placement Property

The **Placement** property is used to change position of labels. The full path to this property is **Area.Axis.Labels.Placement**. This property has three values: **One Line**, **Two Lines**, **None**.

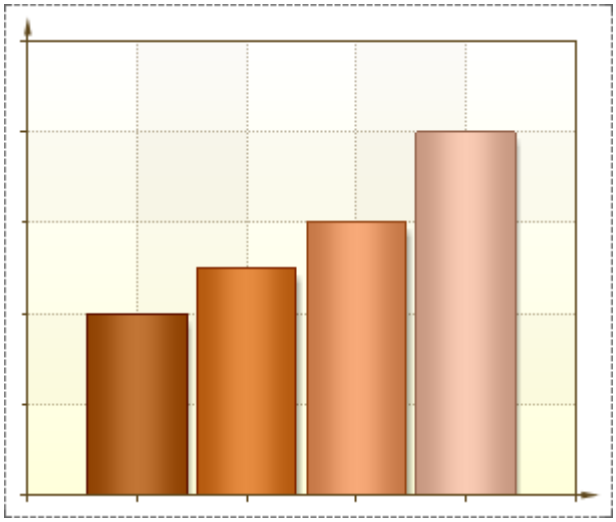
1. **One Line**. In this case, labels are placed in a line horizontally or vertically, depending on the X or Y axis, respectively. The picture below shows an example of a chart, with the **Placement** property set to **One Line** for of X and Y axes:



2. **Two Lines**. In this case, labels are placed in two lines horizontally or vertically, depending on the X or Y axis, respectively. The picture below shows an example of a chart, with the **Placement** property set to **Two Lines** for of X and Y axes:



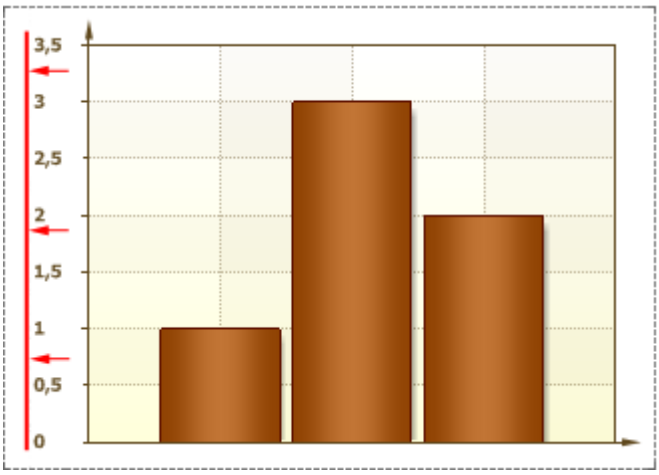
3. **None**. In the case labels are not shown. The picture below shows an example of a chart, with the **Placement** property set to **None** for of X and Y axes:



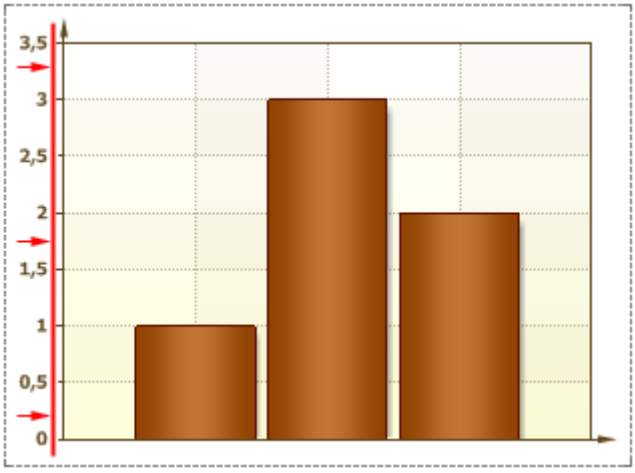
By default, the **Placement** property is set to **One Line**.

TextAlignment Property

The **TextAlignment** property is used to align labels on the chart or by Y axis. The full path to this property is **Area.Axis.Labels.TextAlignment**. If the **TextAlignment** property set to **Left**, then labels are aligned by the chart edge. The picture below shows an example of chart with the of **TextAlignment** property set to **Left**:



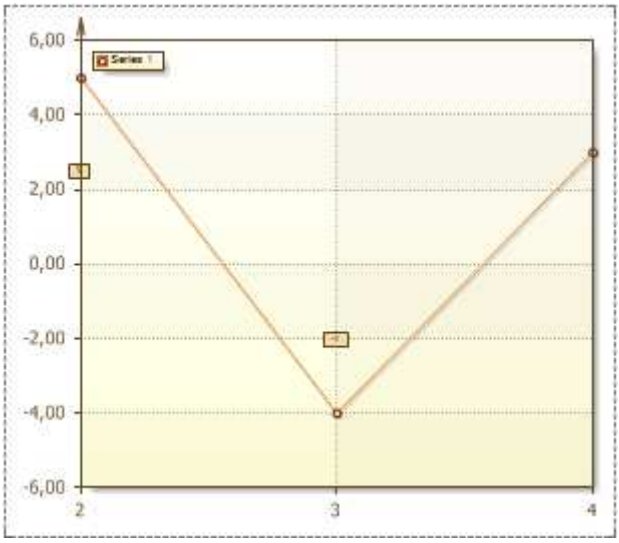
If the **TextAlignment** property set to **Right**, then the labels are aligned by the Y axis. The picture below shows an example of chart with the of **TextAlignment** property set to **Right**:



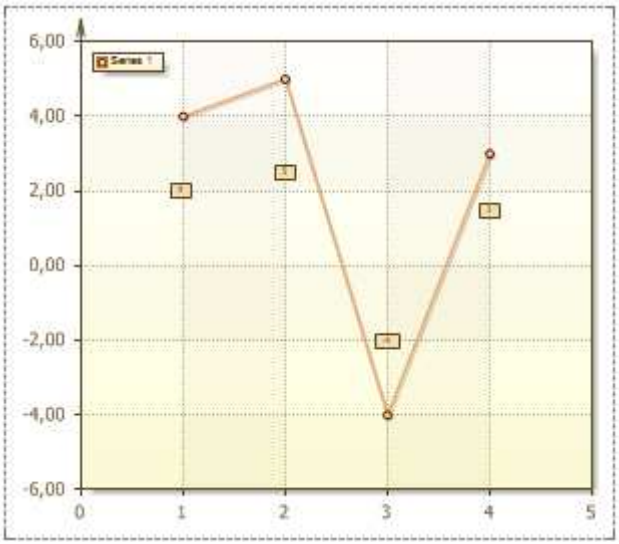
By default, the **TextAlignment** property is set to **Right**.

Range Property

The **Range** property is used to display the specified section of a chart. So a part of the chart within the specified values will be shown. The picture below shows a chart with the Range property set to the X-axis from 2 to 4:



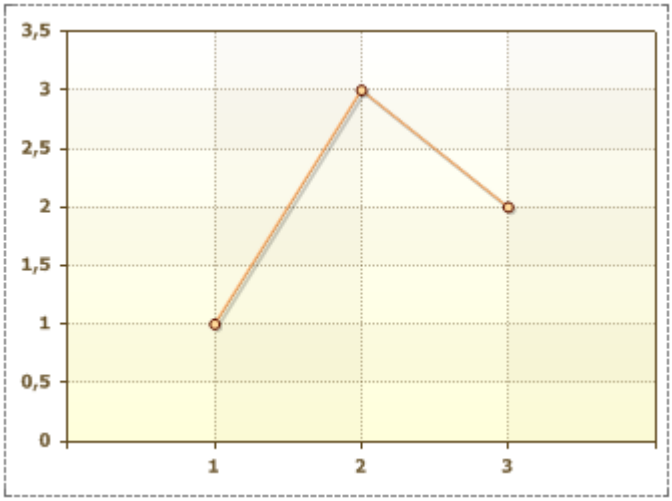
The Range consists of the values of three fields:
1. **Auto**. If the Auto field is set to true, then a chart is shown entirely, i.e. the range of values will be calculated automatically. The picture below shows an example of it:



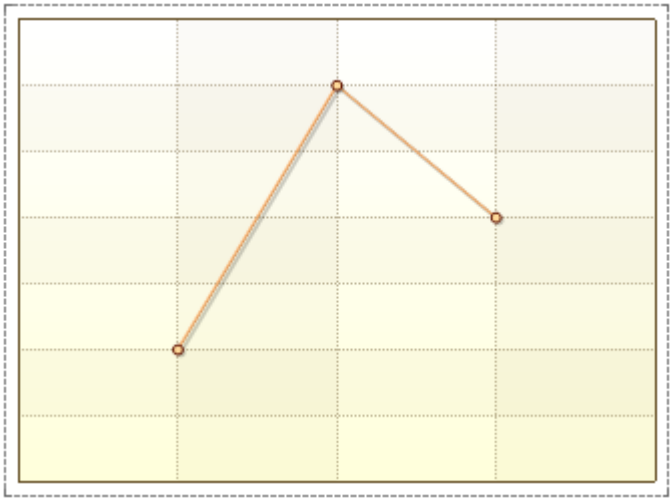
If the **Auto** field is set to **false**, then all values of the range which are specified in the **Minimum** and **Maximum** fields are considered. If the **Auto** field is set to **false**, and values the **Minimum** and **Maximum** fields are set to 0, then the chart will be shown entirely.
2. **Minimum** - sets the beginning of the range.
3. **Maximum** - sets the end of the range.
If the **Maximum** value is less then the **Minimum** value, then the chart will be displayed entirely.

Visible Property

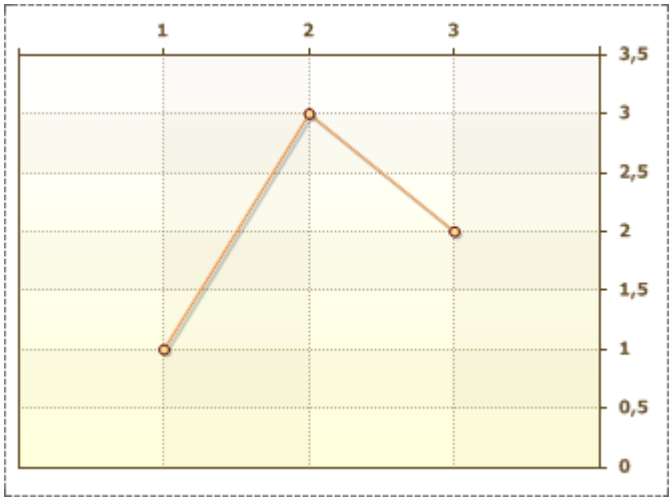
The **Visible** property is used to show X and Y axes. The picture below shows a chart with the **Visibility** property set to **true** (axes are visible):



If the **Visible** property is to set the **false**, then X and Y axes will not be shown. The picture below shows this:



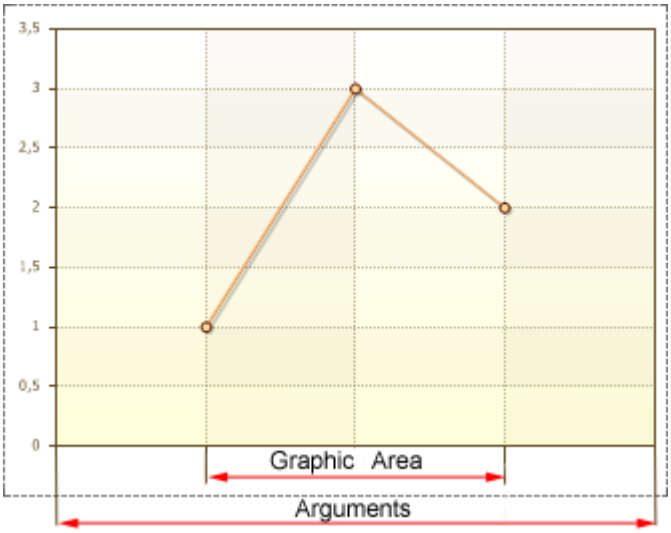
The **Visible** property has the X axis and the Y axis. It is possible to hide/show axes separately. Also, this property is used to display the top X axis and right Y axis. By default, for the axes, the property is set to **false**. The picture below shows an example of a chart, to display the top X axis and the right Y axis:



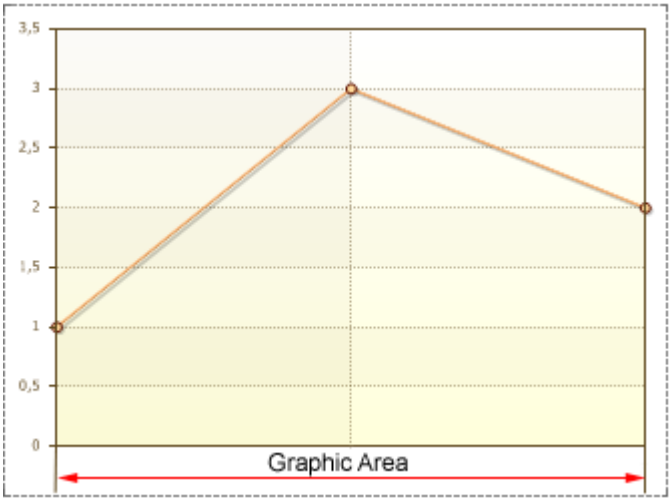
The **Visible** property has the top X axis and the right Y axis. It is possible a combination, for example, the top X axis and the left Y axis or the X axis and right Y axis or any other combinations.
By default the **Visible** property is set to **true**.

StartFromZero Property

By default, the **Start from Zero** property is set to **true**. I.e. arguments are shown from the start to the end, regardless of the location of the chart. The picture below shows an example of a chart with the **Start from Zero** property set to **true** for the X and Y axes:

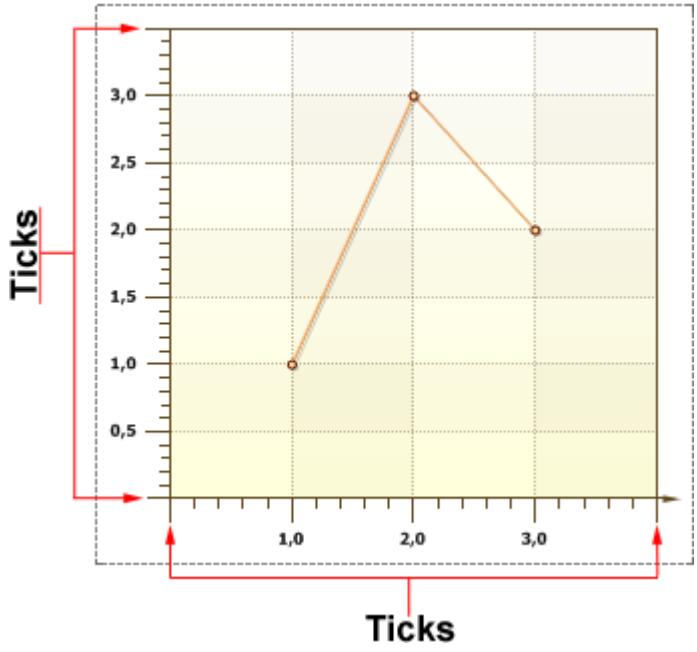


If the **Start from Zero** property to set **false**, then the Range of the chart area will be shown. The picture below shows an example of a chart with the **Start from Zero** property set to **false** for the X axis:



Ticks

Ticks are horizontal (for the Y axis) and vertical (for the X axis) lines, which visually show the unit interval and the proportion of segments. Under the **Ticks** labels are displayed. The picture below below shows a chart with ticks:



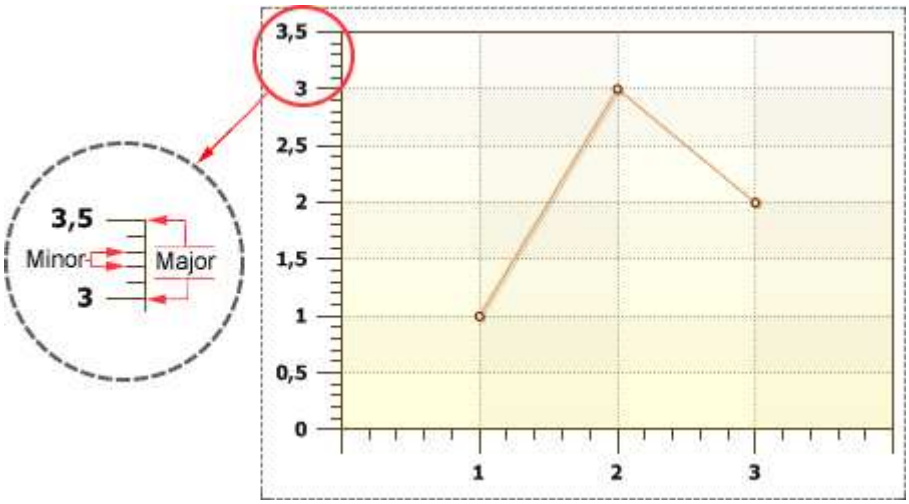
Ticks have the following properties:

- **Length** is the length of ticks, under which Labels are placed;
- **Minor Count** allows changing the number of intermediate lines (Minor ticks);
- **Minor Length** is the length of the intermediate lines (Minor ticks);
- **Minor Visible** is used to show/hide the intermediate lines (Minor ticks);
- **Step** controls the step of the unit interval, i.e. distance between ticks;
- **Visible** is used to show/hide **Ticks**, both basic and intermediate.

Minor

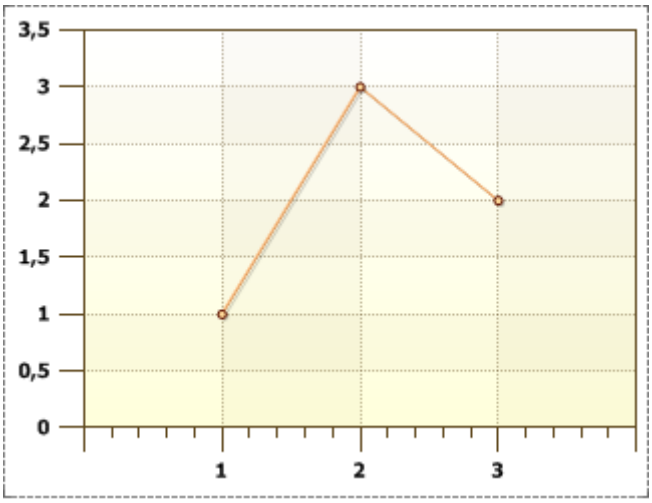
Minor ticks show the proportion of a single axis segment. **Minors ticks** have the following properties: **MinorCount**, **MinorLength**, **MinorVisible**.

1. **Minor Count** is used to change the number of Minor ticks. The value of this property can be any positive number or 0. The distance between two nearest Major ticks is divided into the number of Minor ticks into equal parts. The picture below shows an example of a chart, with the **Minor Count** property set to 4 for X and Y axes:



2. **Minor Length** is used to change the length of Minor ticks. The value of this property can be any positive number greater than 0, the field of this property can not be left blank. The length of Minor ticks can be longer than the length of Minor ticks.

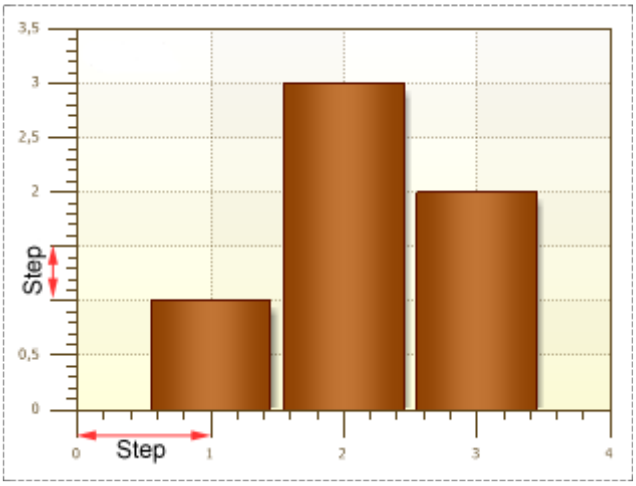
3. **Minor Visible** is used to show/hide Minor ticks on axes. If the **Minor Visible** property is set to **false**, then the Minor ticks are hidden. If the value of this property is set to **true**, then the Minor ticks are shown. The picture below shows an example of a chart, with the **Minor Visible** property set to **true** for X axis, and set to **false** for Y axis:



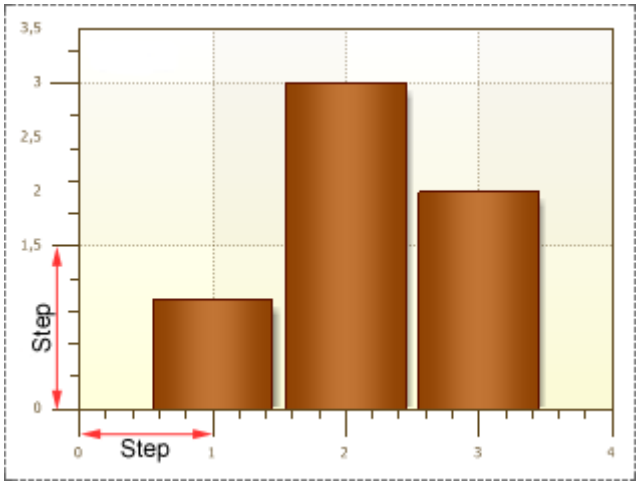
By default, the **Minor Visible** property is set to **false**.

Step Property

The **Step** property is used to change the step between Ticks, i.e. the distance between neighbor Major ticks. By default, the value of the **Step** property is set to 0. The picture below shows an example of a chart with the Step is installed to the 0 default value.

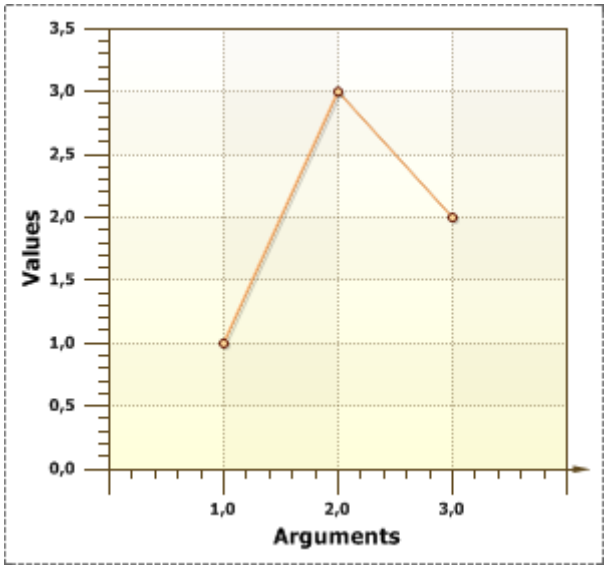


As one can see, if the value is 0, then the distance between two between neighbor Major ticks by the Y axis is **0.5**, and **1** by the X-axis. If to set the Step property to **Z** value, then the report generator will multiply **Z** value by the value of the unit interval. The result obtained is the distance between two neighbor Major ticks. The picture below shows an example of a chart, with the step on the Y axis set to **1,5**, and the X axis value set to **1**:



Title Property

The **Title** property is a title of axis. This property is used to display an axis title. Moreover, the **Title** property for each axis is given separately. The picture below shows a chart where the **X** axis is called the "**Arguments**", and the axis **Y** is called "**Values**":



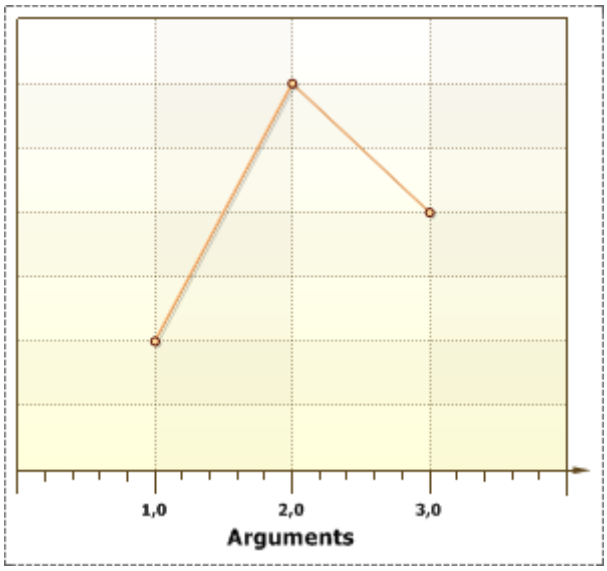
Also, the **Title** property has the following properties:

- **Alignment** is used to align the **Title**. It has the following values **Center** (align center), **Far** (align from the beginning of an axis), **Near** (align to the beginning of an axis);
- **Antialiasing** is used to produce smooth-edged **Titles**;
- **Color** is used to change a title text of an axis;
- **Font** is used to change the size, font style of a title text of an axis;
- **Text** is a field to type a title text of an axis. If the field is empty then the title of an axis is not displayed.

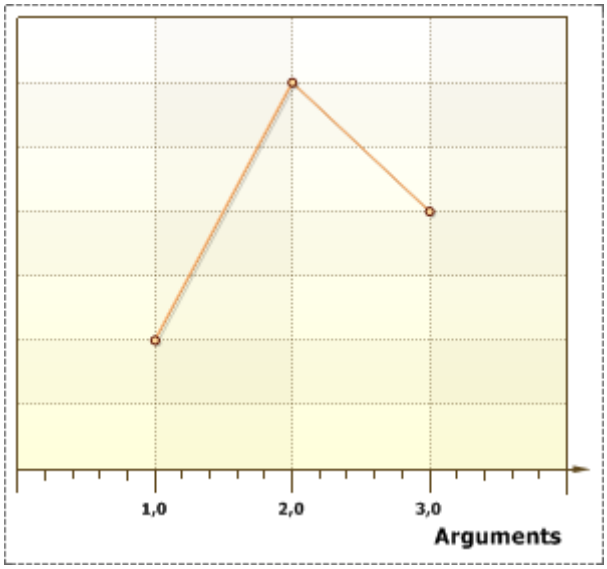
Alignment Property

The **Alignment** property is used to align a title of an axis. The full path to this property is **Area.Axes.Title.Alignment**. This property has the following values: **Center**, **Far**, **Near**.

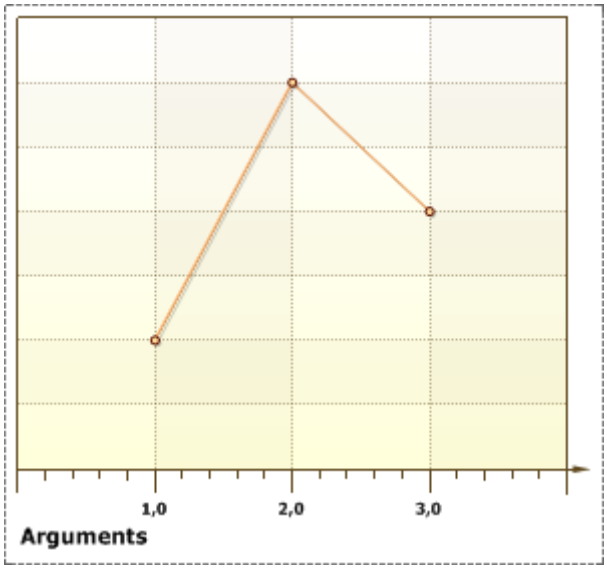
1. **Center**. Aligns the title of the axis by center by the axis. The picture below shows an example of a chart, with the **Alignment** property of a title of the X axis set to **Center**:



2. **Far**. Aligns the title of the axis on the opposite side from origin of coordinates. The picture below shows an example of a chart, with the **Alignment** property of a title of the X axis set to **Far**:



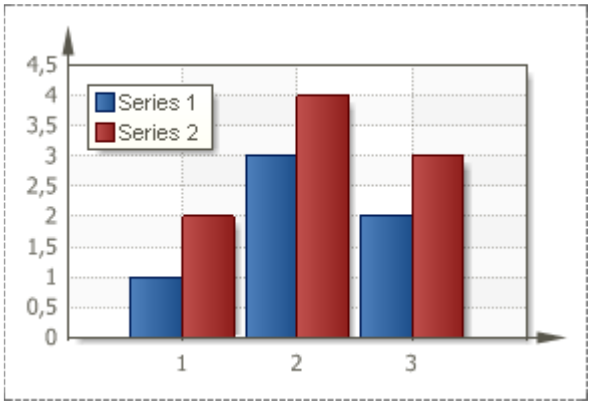
3. **Near**. Aligns the title of the axis on the near the origin of coordinates. The picture below shows an example of a chart, with the **Alignment** property of a title of the X axis set to **Near**:



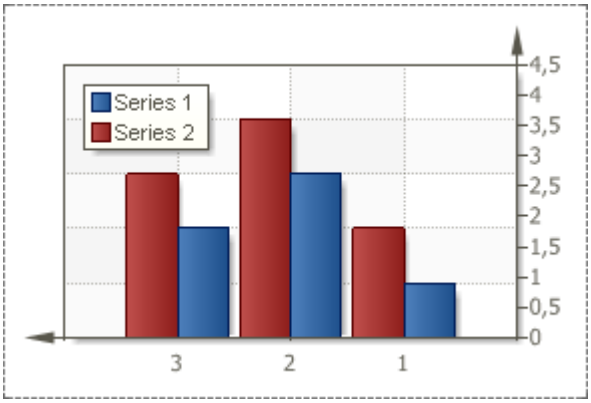
By default, the **Alignment** property of series is set to **Center**.

ReverseHorizontal Property

The **Reverse Horizontal** property is used to flip a chart horizontally. The picture below shows an example of a chart, with the **Reverse Horizontal** property set to **false** (As one can see, the values of the x-axis have left to right direction.):



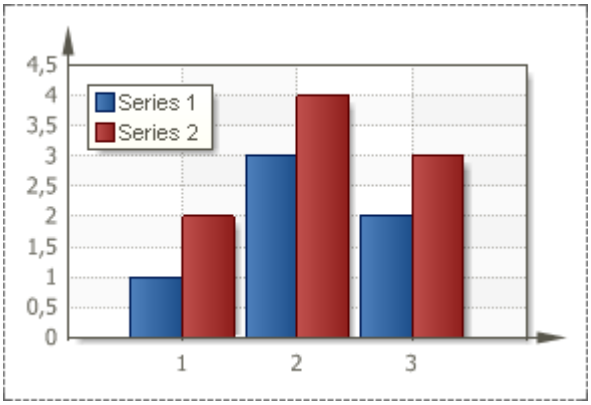
If the **Reverse Horizontal** property is set to **true**, then the chart will appear in the opposite direction horizontally. The picture below shows an example of a chart, with the **Reverse Horizontal** property is set to **true** (As one can see, the values of the x-axis have right to left direction.):



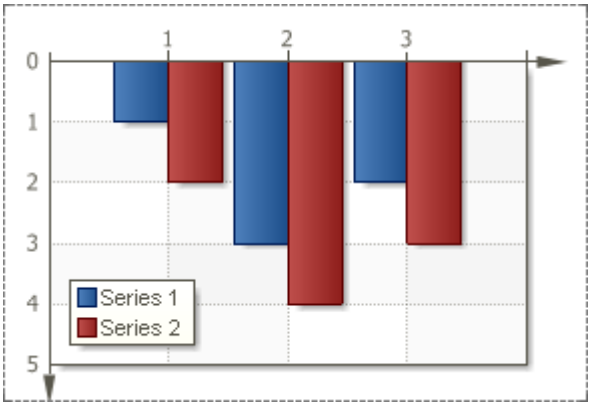
By default, the **Reverse Horizontal** property is set to **false**.

ReverseVertical Property

The **Reverse Vertical** property is used to flip a chart vertically. The picture below shows an example of a chart, with the **Reverse Vertical** property set to **false** (As one can see, the values of the x-axis have normal direction.):



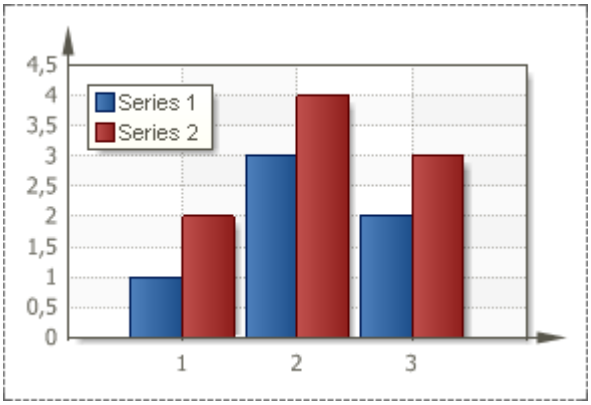
If the **Reverse Vertical** property is set to **true**, then the chart will appear in the opposite direction vertically. The picture below shows an example of a chart, with the **Reverse Vertical** property is set to **true** (As one can see, the values of the x-axis have downright direction.):



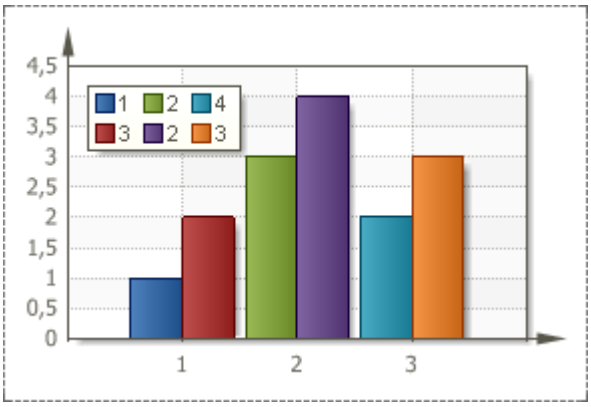
By default, the **Reverse Vertical** property is set to **false**.

ColorEach Property

The **Color Each** property is used (depends on the selected style) to set color for each value of a series. By default, the **Color Each** property is set to **false**, i.e. columns of one row have the same color. The picture below shows an example of a chart with the **Color Each** property set to **false** for two series:



If the **Color Each** property is set to true, then each value of X axis has its own color. The picture below shows an example of a chart with the **Color Each** property set to **true** for two series:



Series

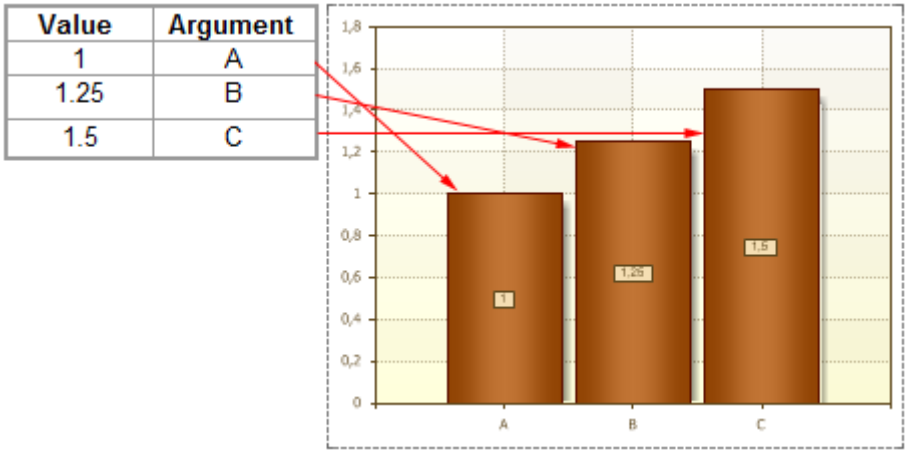
Series type depends on the chart type. They are divided into series, placed on doughnut charts, and placed in the axis area.

Data Connection

One of the main settings of the series is specifying the way of obtaining data. There are three ways to obtain data for the series:
1. To set the column data from the dictionary;
2. To specify an expression;
3. Manually specify values for the series as a list, through the '**;**' separator.

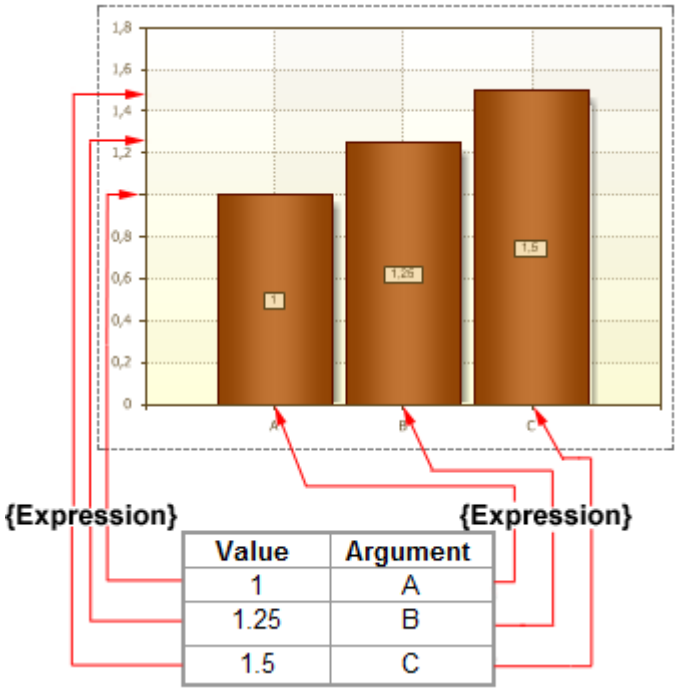
Data Column

The **Value Data Column** and **Argument Data Column** properties are used to connect a series by specifying a data column from the dictionary. The reporting tool renders series of charts by values and arguments of the column selected in the fields of the **Value Data Column** and **Argument Data Column** properties. For example, if the selected column of data from the data source contains the 1000 values, then all the 1000 values will be used in constructing the chart. The picture below shows an example of the chart, so the values from the selected data source column:



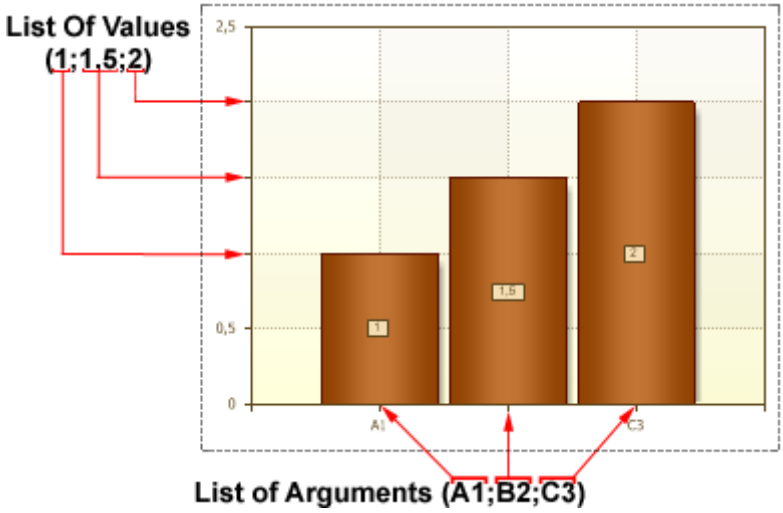
Expressions

To connect a series of data using the expression, you should use the **Value** and **Argument** properties. The values of these properties are expressions, the result of their calculation is used to obtain a single value of data and argument of data. If you use the Value and Argument properties, then, for this chart, it is necessary to select a data source (the Data Source property), because expressions specified in the fields of these properties are not lists of data and return only one value when calculating. Moreover, the **Value** property returns the value in Number format, but the **Argument** property allows any type of data. To make the report generator know which list should be used for the report, it is necessary to indicate the data source. Once the data source is specified, the report generator runs through all the records of the data source and calculates all the values and arguments according to expressions given in the fields of the **Value** and **Argument** properties. The result of the calculation is used to create a chart. Also, for the data in the data source, you can specify sorting and filtering. The picture below shows an example of a chart, rendered on the basis of results of values and arguments calculations of the selected column of the data source:



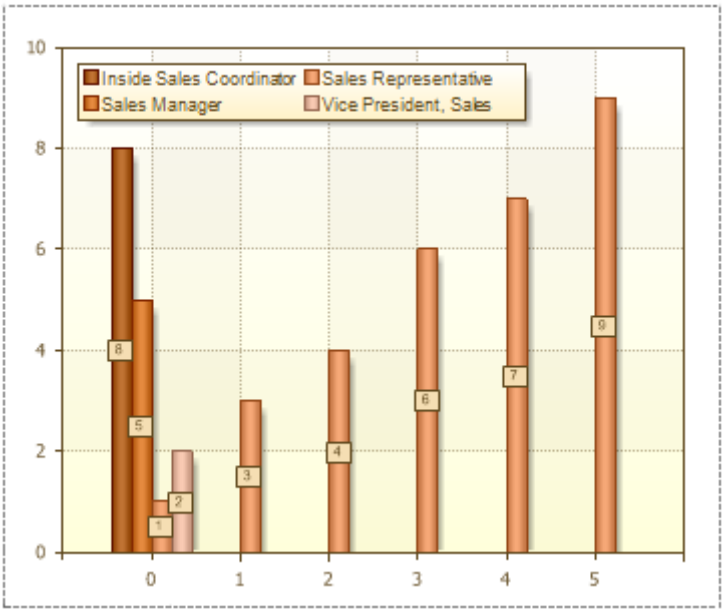
List of Values Property

If it is necessary to build a chart by the given values and arguments, then one should use the **List of Values** and the **List of Arguments** properties. The **List of Values** indicates values for creating series (values must be entered through the **';'** separator). The **List of Arguments** property indicates arguments for creating series (values must be entered through the **';'** separator). The order number of the **List of Values** property values corresponds to order number of the **List of Arguments** property values. The picture below shows an example a chart, designed by the list of values and arguments:



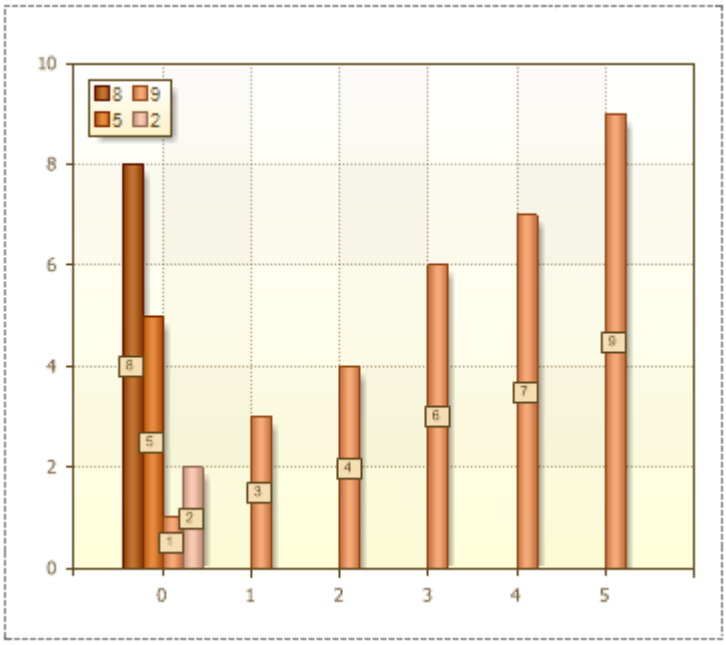
AutoSeries

Stimulsoft Reports can automatically create a series. Use the **Auto Series Key Data Column**, **Auto Series Color Data Column**, and **Auto Series Title Data Column** properties. A column from which values are taken to build the series is selected in the **Auto Series Key Data Column** property. A series is created for each unique value. The picture below shows an example of a chart with the **Auto Series Key Data Column** property set to **Employees.Title**:



There are 4 rows on the picture above. The 1st, 2nd, 4th series have one value, and the 3rd series has 6 values. This means that the **Employees** data source in the **Title** column contains **9** lines, and 6 lines have identical values (records), and the remaining three are different. Values (records) of rows in the data source are shown in a rendered chart in the legend, as well as the name of the series, if the field of the **Auto Series Title Data Column** property is empty. The **Auto Series Color Data Column** property is used to specify the color range, i.e. each series will have its own color. This property is subsidiary, and is not required to fill in the automatic creation of the series. Also, the subsidiary property and the **Auto Series Title Data Column** property, using what it is possible to change the title of

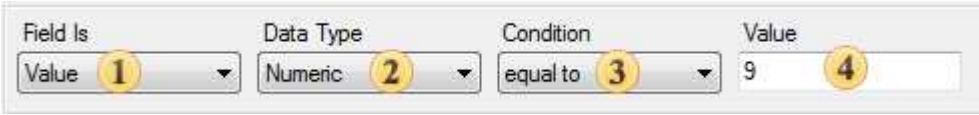
the series. The picture below shows an example of a chart, with the **Auto Series Key Data Column** property set to **Employees.Title**, and the **Auto Series Title Data Column** property set to **Employees.EmployeeID**:



As seen from the picture above, the series labels are changed. As the series labels, string values are taken from the columns of the data source that is listed in the **Auto Series Title Data Column** property, in this case, this is the **EmployeeID** column.

Filters

Sometimes, in creating reports, it is necessary to print, not all values from the data source, but only those that meet specific criteria. In order to select the required settings, data filtering is used. Filtering is set using the **Filters** property in the **Series Editor**. A condition is specified is each filter. If the condition is **true**, i.e. the result of its calculation is **true**. This means that this value will be used when chart rendering. If the result of calculation of the filter condition is **false**, then this value will be ignored. Each filter represents a condition for processing the data values. The picture below shows an example the filter panel:



1 The method of choosing the conditions by what filtering (Value or Argument) is done.
2 This field specifies the type of data with what condition will be working. Five types of data are available: **String**, **Numeric**, **DateTime**, **Boolean**, **Expression**. The data type affects how the report generator processes the condition. For example, if the data type is a string, then the method of work with strings is used. In addition, depending on the type of data the list of available condition operations is changed. For example, only for the **String** data type the **Containing** operation is available. The **Expression** data type is used to set the expression instead of the second value.
3 The type of operation with what it is possible to calculate a value of a condition. All available types of operations are available in the table below.
4 Values of the filter condition.

A list of available operations depends on the type of data. Below is a table of operations for each type of data with their descriptions.

Operation	Types of data				Description
	String	Numerical	DateTime	Boolean	
equal to	✓	✓	✓	✓	If the first value is equal to the second, then the condition is true.
not equal to	✓	✓	✓	✓	If the first value is not equal to the second, then the condition is true.
between		✓	✓		If the first value is in the range, then the condition is true.
not between		✓	✓		If the first value is not in the range, then the condition is true.
greater than		✓	✓		If the first value is greater then the second value, then the condition is true.
greater than or equal to		✓	✓		If the first value is greater then the second value of equal to the second value, then the condition is true.
less than		✓	✓		If the first value is less then the second value, then the condition is true.
less then or equal to		✓	✓		If the first value is less then the second value or equal to the second value, then the condition is true.
containing	✓				If the first value contains the second value, then the condition is true. This operation is used only for strings.
not containing	✓				If the first value does not contain the second value, then the condition is true. This operation is used only for strings.
beginning with	✓				If the first value starts with the second value, then the condition is true. This operation is used only for strings.
ending with	✓				If the first value ends with the second value, then the condition is true. This operation is used only for strings.

Conditions

If it is necessary to set the color of values in a chart, one can specify the condition. The **Conditions** property in the **Series Editor** is used to set up conditional formatting. The editor of conditions is called using this property. The picture below shows the main elements of the editor of conditions:

1 **Field Is**
This is used to select the type of conditions.
2 **Data Type**
This field specifies the type of data with what a condition will work. There are five types of data: **String**, **Numeric**, **DateTime**, **Boolean**, **Expression**. Data type affects on how the reporting tool processes a condition. For example, if the data type is a string, then the methods of work with strings are used. In addition, depending on the type of data the list of available operations of conditions is changed. For example, only for the **String** data type the **Containing** operation is available**.** The **Expression** data type provides the ability to specify an expression instead of the second value. In this case the reporting tool will not check the compatibility of the first and the second values of the condition. Therefore, the user should care about the correctness of the expression.
3 **Condition**
A type of operation using what the calculation of values will be done.
4 **Value**
The first value of a condition.
5 **Color**
Select a color to mark values which corresponds to condition.

Series Editor

Setting the series includes a number of properties in the **Series Editor**, which is used to visually arrange the rows and change the advanced settings.

1. The **Border Color** property is used to change the border color of each series of a chart, i.e. each border of series has its own color.

2. The **Brush** property is used to change the type of filling and series color.

3. Depending on the value of the **Show Shadow** property, the shadow for series may be shown/hidden. If the **Show Shadow** property is set to **true**, then shadows are shown. If the **Show Shadow** property is set to **false**, then shadows are not shown.

4. . The **Show Zeros** property can take two values, depending on what zero values in a chart will be shown/hidden. If the **Show Zeros** property is set to **true**, then zero values are displayed on a chart. If the **Show Zeros** property is set to **false**, then zero values will not be displayed on a chart.

5. Using the **Width** property it is possible to change the width of the created values. A value of this property will change a value from 0 (a value greater than 0) to 1 (a value must be less than or equal to 1). The lowest value corresponds to the minimum width and maximum value corresponds to the maximum width.

6. The **Axis Y** property affects the location of the Y axis. If the **Axis Y** property is set to **Left Y Axis**, then the Y axis will be located on the left. If the **Axis Y** property is set to **Right Y Axis**, then the Y axis will be located on the right.

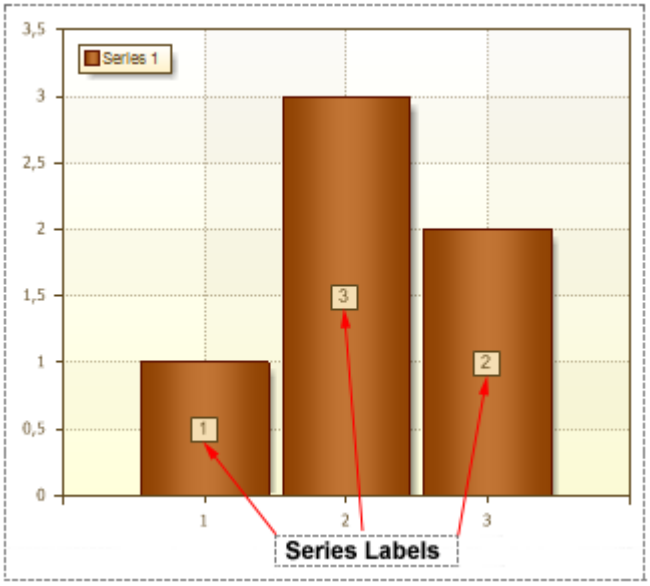
7. Using the **Show in Legend** property will change the display mode in a legend. If the **Show in Legend** property is set to **true**, then series are shown in a legend. If the **Show in Legend** property is set to **false**, then series are not shown in a legend.

8. The **Show Series Labels** property can take three values, according to which titles series will be shown/hidden. If the **Show Series Labels** property is set to **None**, then series labels not displayed. If the **Show Series Labels** property is set to **fromCharts**, then series labels are displayed according to parameters set in the **Series Labels** property of a chart. If the **Show Series Labels** property is set to **fromSeries**, then in the **Series Editor** the **Series Labels** property will appear. This property can be configured by setting the parameters, and Series Labels in a chart will be displayed in accordance with these parameters.

9. With help of the **Title** property it is possible to change the series labels. Any characters entered in the field of this property will be labels.

Series Labels

Series Labels is an information block which displays the value of each series. The picture below shows an example of a chart, with Series Labels:



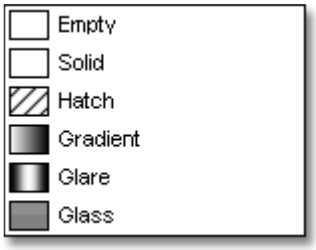
The **Series Labels** property is used to indicate position of series labels. The list of available options for this property depends on the type of chart. Also, the **Series Labels** property have some options that are used to change settings of Series Labels.

Series Labels Appearance

The following group of properties allows visually change the appearance of Series Labels: change the background color, titles, borders, font type, antialiasing.

Brush Property

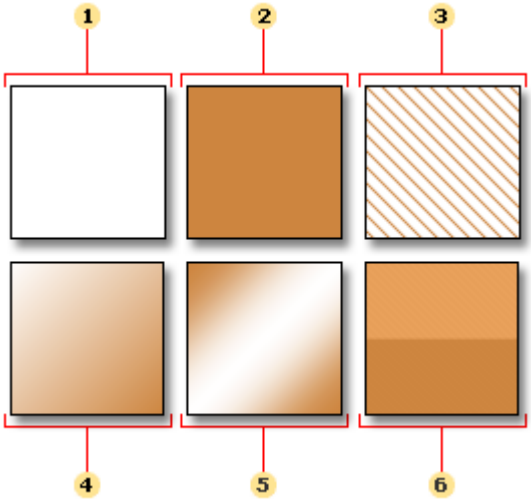
The **Brush** property is used to fill a background type and color in Series Labels. To change the background color and appearance of a Series Label use the **Brush** property within the Object Inspector.



Six types of Brushes are available within Stimulsoft Reports:

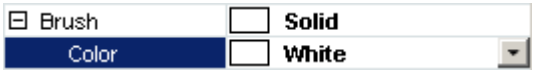
- **Empty**
- **Solid**
- **Hatch**
- **Gradient**
- **Glare**
- **Glass**

Below are representations of the results all six Brush types:

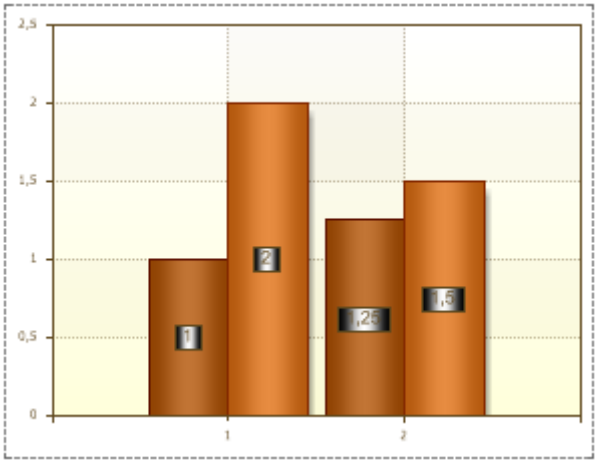


1 **Empty**	The background of a Series Label is transparent.
2 **Solid**	The background of a Series Label is filled with the color you specify.
3 **Hatch**	The background of a Series Label is filled with a texture. The background and foreground colors of the select can be specified individually..
4 **Gradient**	The background of a Series Label is filled with gradient. A Start color, an End color, and a Gradient ang specified.
5 **Glare**	The background of a Series Label is filled using the Glare effect.
6 **Glass**	The background of a Series Label is filled using the Glass effect.

The **Brush.Color** property is used to change the Series Labels color.



The picture below shows a sample of a chart with the Brush property set to **Glare**:



Font Property

The font for Series Labels can be set using the **Font** property within the Object Inspector.

Font	**Arial; 7pt**
Font Name	ab **Arial**
Font Size	**7**
Font Bold	**False**
Font Italic	**False**
Font Underline	**False**
Font Strikeout	**False**

Selecting font

Series Labels within a report can be output using different fonts. Three examples fonts are shown below:

AaBbCcDd

AaBbCcDd

AaBbCcDd

Any font that is installed on your machine can be used in Series Labels. However, when choosing a font try to select one that will also be present on a user machine or a report may not render as you would wish at runtime.

Font Size

The font size can be changed using the **Font.Size** property. For example:

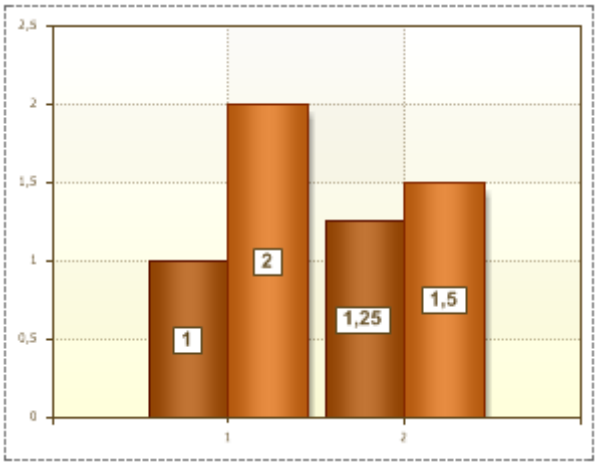
AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd

Font Styles

Different styles can be applied to the font. A font may include one or more styles such as regular, bold, semibold, italic, underlined, and strikeout. Examples of font styles are shown below:

AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
~~AaBbCcDd~~

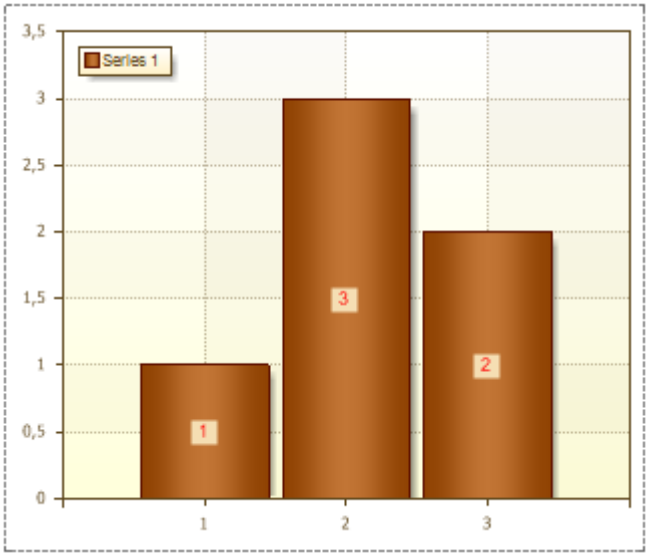
The picture below shows a chart with text set to **Arial, Bold** style**,** font size - **12:**



LabelColor Property

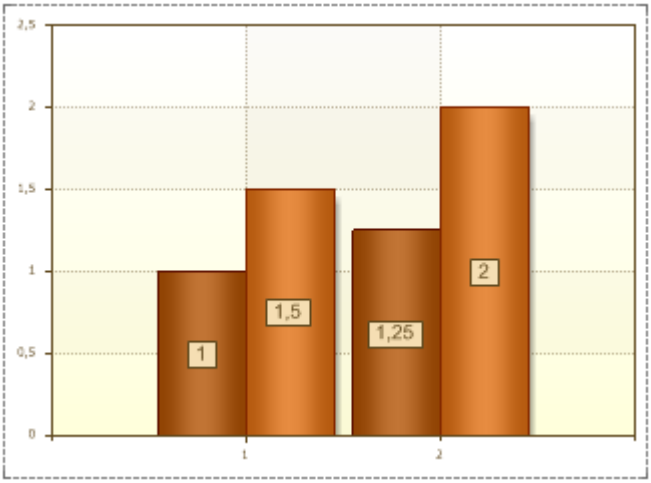
The **Label Color** property within the Object Inspector is used to change the color of Series Labels.

The picture below shows a chart with the **Label Color** property set to **red**:

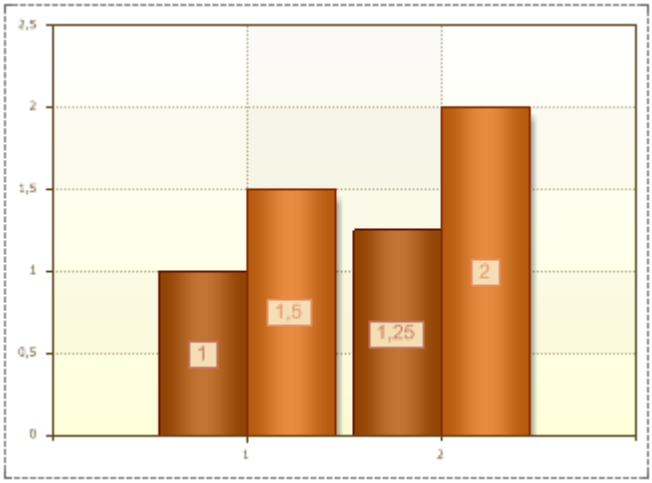


UseSeriesColor Property

The **Use Series Color** property is used to make the border color and the series label color match to the color of the series. If the **Use Series Color** property is set to **false**, then the border color and the color of series labels will correspond to the selected values of the **Border Color** and **Label Color** properties. The picture below shows an example of a chart, with the **Use Series Color** property set to **false**:

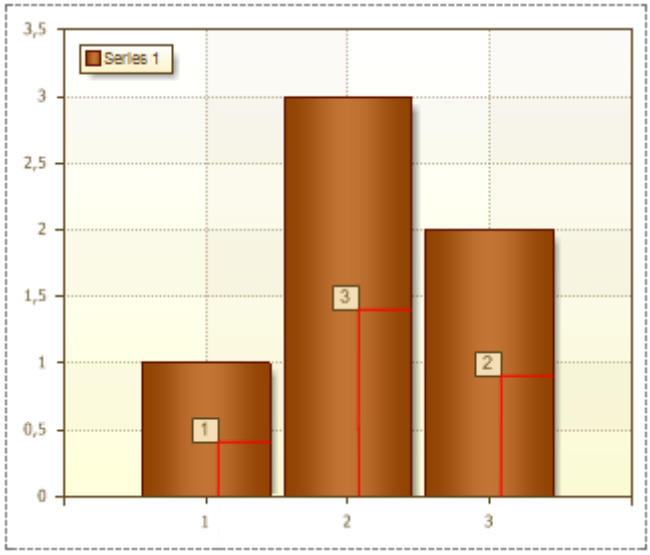


If the **Use Series Color** property is set to **true**, then the border color and series labels color will match to the color of series. The picture below shows an example of a chart, with the **Use Series Color** property set to **true**:

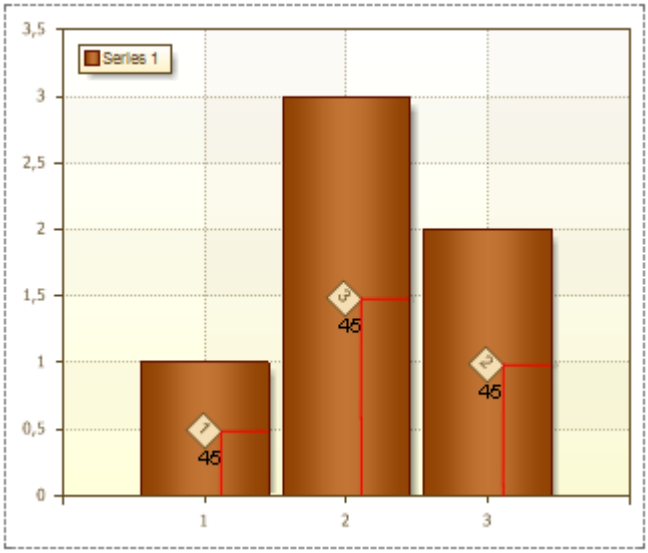


Angle Property

The **Angle** property allows changing the inclination angle of Series Labels. By default, this property is set to **0** (Series Labels is not inclined). The picture below shows the situation when the **Angle** property is set to **0**:

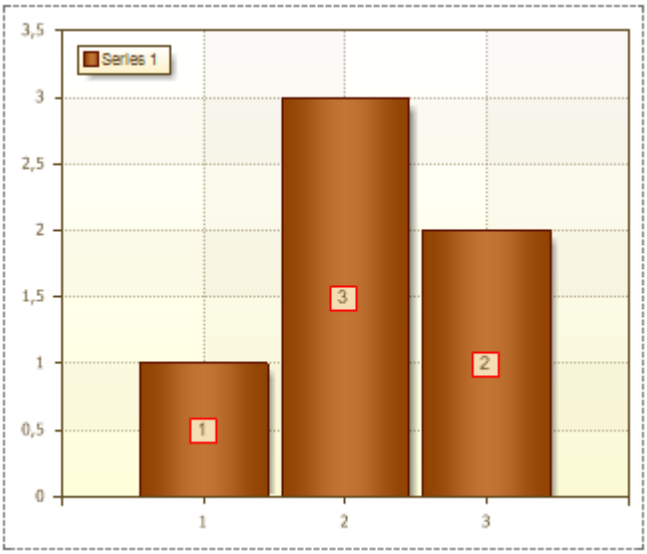


The value of the property can be negative and positive. If a value of the property is negative then Series Label is inclined anticlockwise. If the value of the property is positive then Label in inclined clockwise. The picture below shows a chart sample, which the **Angle** property for Series Labels is set to **45**:

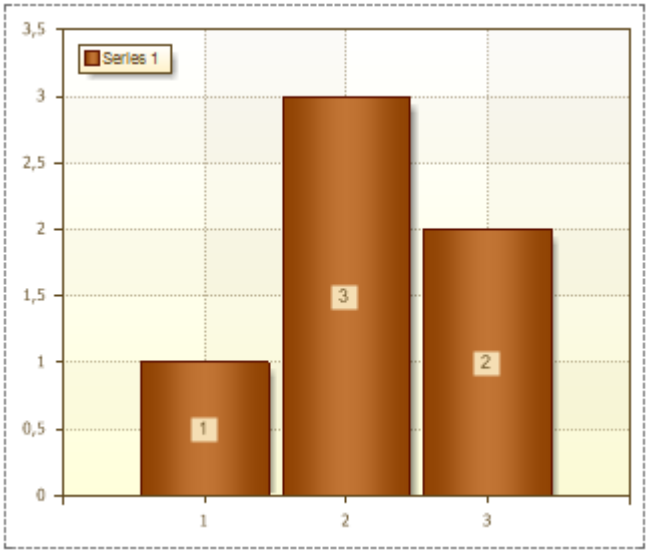


DrawBorder Property

The **Draw Border** property allows showing/hiding a border of Series Labels. It has two values: **true** and **false**. If the **Draw Border** is set to **true**, then the border is shown. The picture below shows a chart with borders around Series Labels (the borders are red):



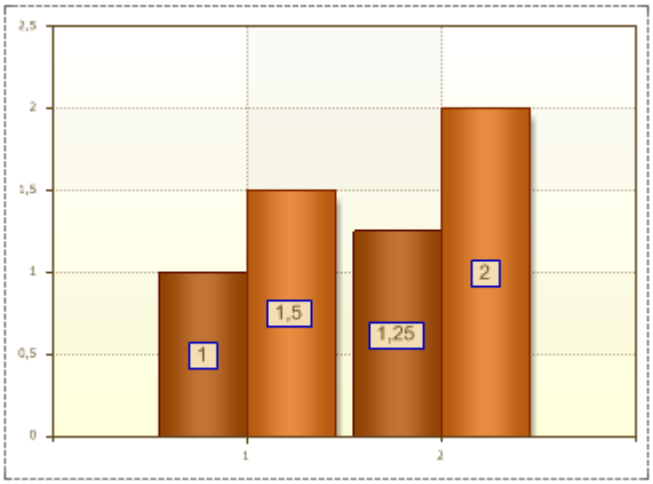
If the **Draw Border** is set to **false**, then the border is hidden. The picture below shows a chart without borders around Series Labels:



BorderColor Property

The **Border Color** property is used to change the border color of Series Labels.

The picture below shows a chart which Series Labels borders are blue:



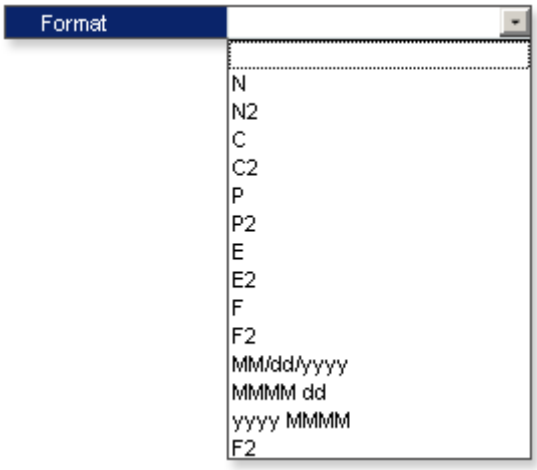
Antialiasing Property

Antialiasing property allows you producing smooth-edged Series Labels by partially filling the edge pixels. As a result, the edges of Series Labels blend into the background. The picture below shows a chart with the **Antialiasing** property set to **true**:

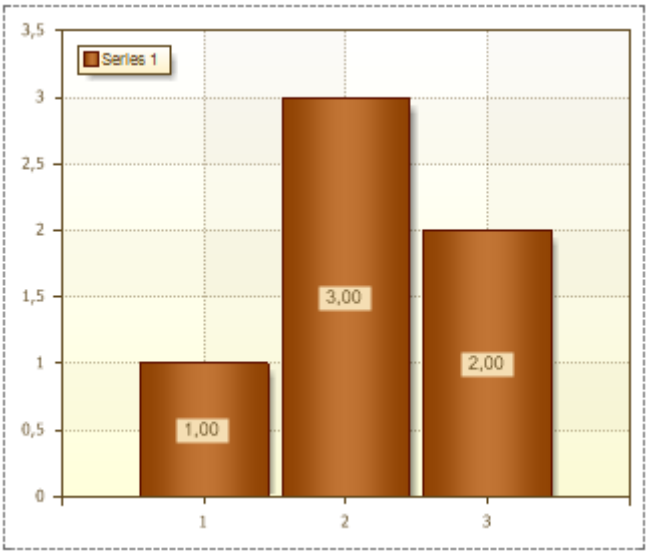
The picture below shows a chart with the **Antialiasing** property set to **false**:

Format Property

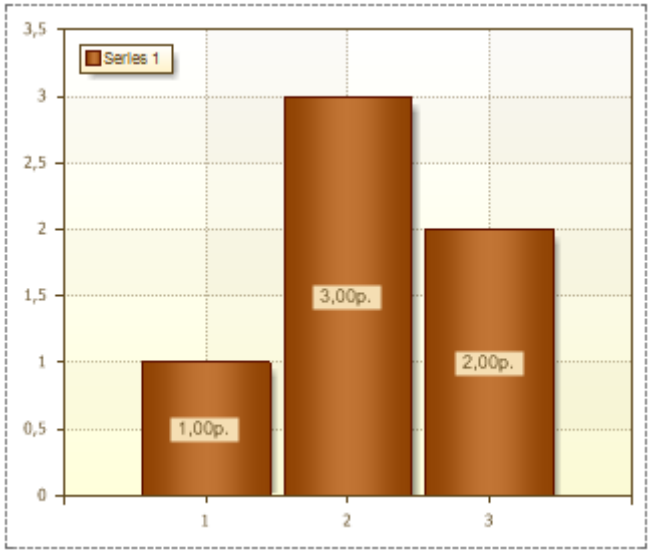
The **Format** property is used to format the contents of Series Labels. This property has multiple values.



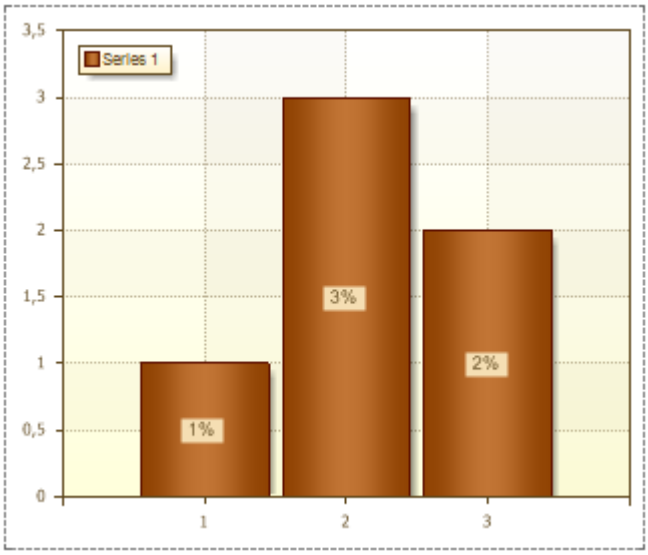
1. **Number.** The **N** value of the **Format** property is used for the general display of numbers. When filling the **Format**, after the **N** value, it is possible to specify the number of decimal places that you want to use. If no numbers are specified after **N** then decimal places will be shown only if they are present as a result of calculation. The picture below shows a chart with the **Format** property of Series Labels set to **N**:



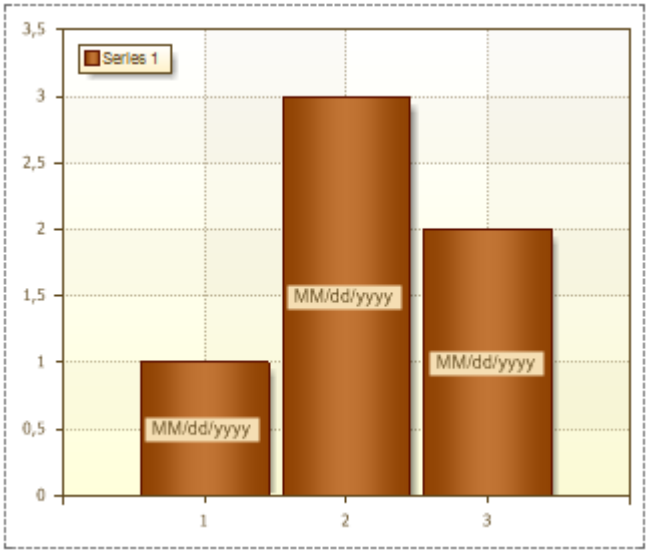
2. **Currency.** The **C** value of the **Format** property is used to display Series Labels with a currency symbol. After the **C** value, it is possible to specify the number of decimal places that you want to use. The picture below shows a chart with the **Format** property of Series Labels set to **C**:



3. **Percentage**. The **P** value of the **Format** property is used to display Series Labels with percent symbol. After the **P** value, it is possible to specify the number of decimal places that you want to use. The picture below shows a chart with the **Format** property of Series Labels set to **P**:



4 **Date**. The **MM/dd/yyyy**, **MMMM dd**, **yyyy MMMM** values of the **Format** property convert values of arguments to date. **MM/dd/yyyy** - the date is shown like "01.20.2010", **MMMM dd** - the date is shown like "September 29", **yyyy MMMM** - the date is shown like "2010 March". The picture below shows a chart and with the **Format** property set to **MM/dd/yyyy**

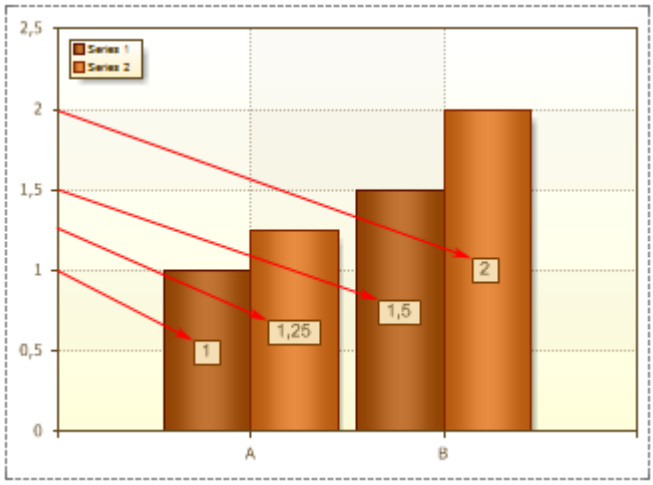


To reset the **Format** property of selected cells, and return to the default format, clear the Format by selecting empty field.

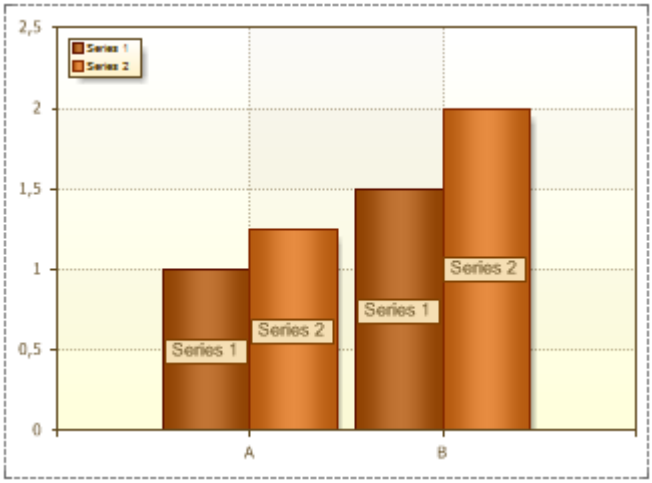
ValueType Property

The **Value Type** property is used to specify the type of a value that appears in the series labels. This property may take the following values: **Value**, **Series Title**, **Argument**, **Value - Argument**, **Argument - Value**, **Series Title - Value**, **Series Title - Argument**.

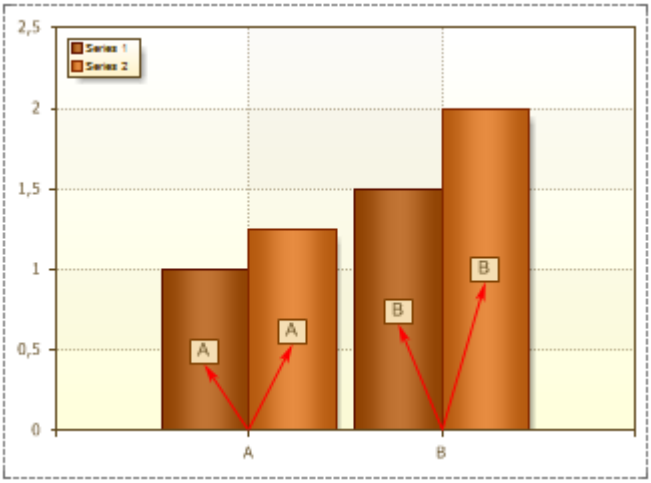
1. **Value**. The Series Labels are series values. The picture below shows an example of a chart with the **Value Type** property set to **Value**:



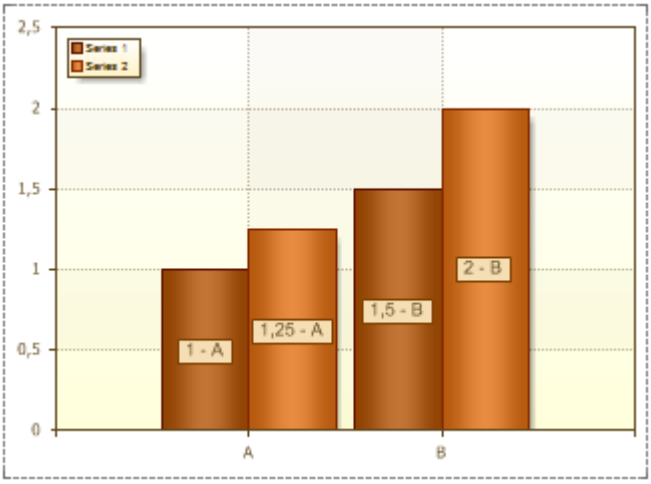
2. **Series Title**. The Series Labels are records in the **Title** field in the **Series Editor**. The picture below shows an example of a chart with the **Value Type** property set to **Series Title**:



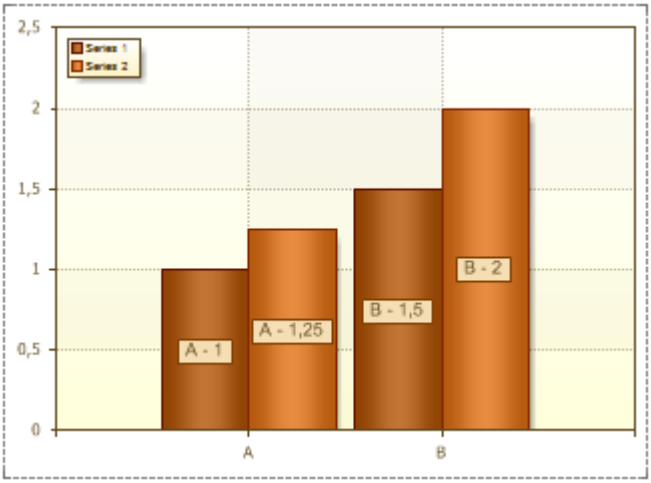
3. **Argument**. The Series Labels are the arguments. The picture below shows an example of a chart with the **Value Type** property set to **Argument**:



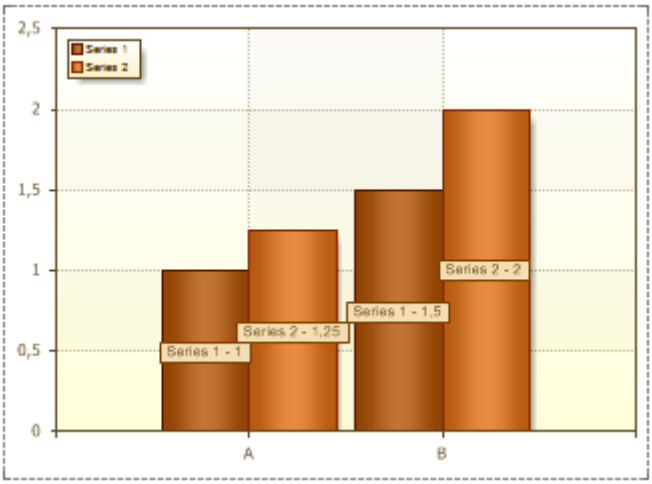
4. **Value - Argument**. The Series Labels are **Values** and **Arguments** of series. The picture below shows an example of a chart with the **Value Type** property set to **Value - Argument**:



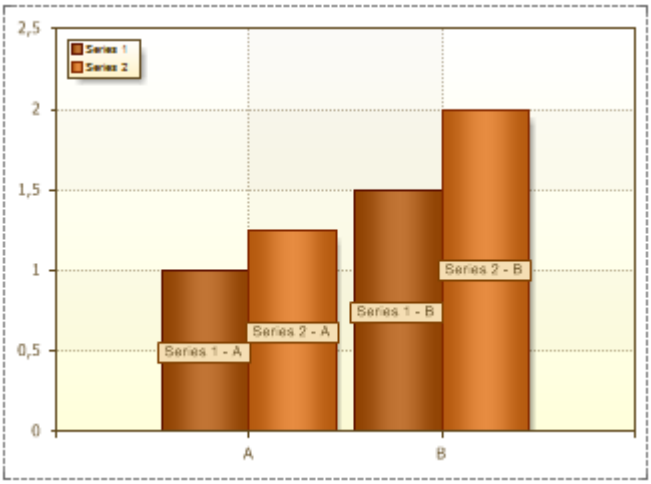
5. **Argument - Value**. The Series Labels are **Arguments** and **Values** of series. The picture below shows an example of a chart with the **Value Type** property set to **Argument - Value**:



6. **Series Title - Value**. The Series Labels are **Series Titles** and **Values**. The picture below shows an example of a chart with the **Value Type** property set to **Series Title - Value**:



7. **Series Title - Argument**. The Series Labels are **Series Titles** and **Arguments**. The picture below shows an example of a chart with the **Value Type** property set to **Series Title - Argument**:

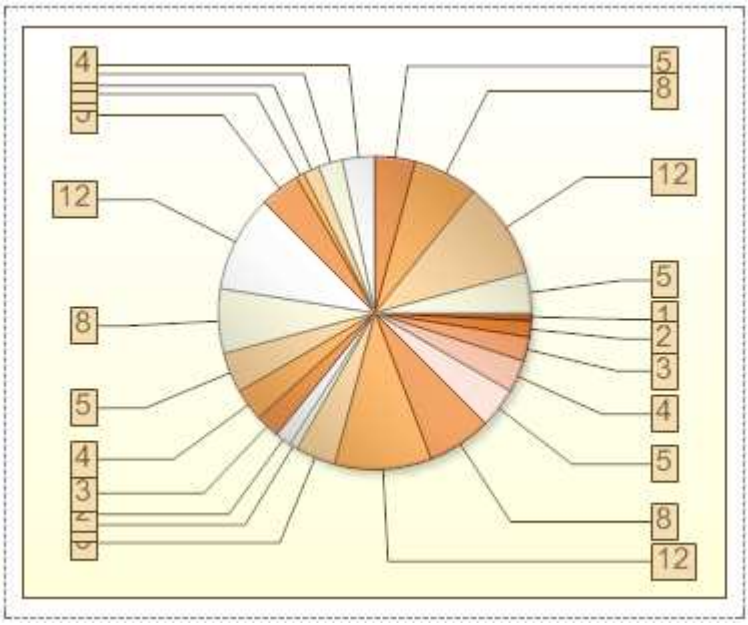


ValueType Separator

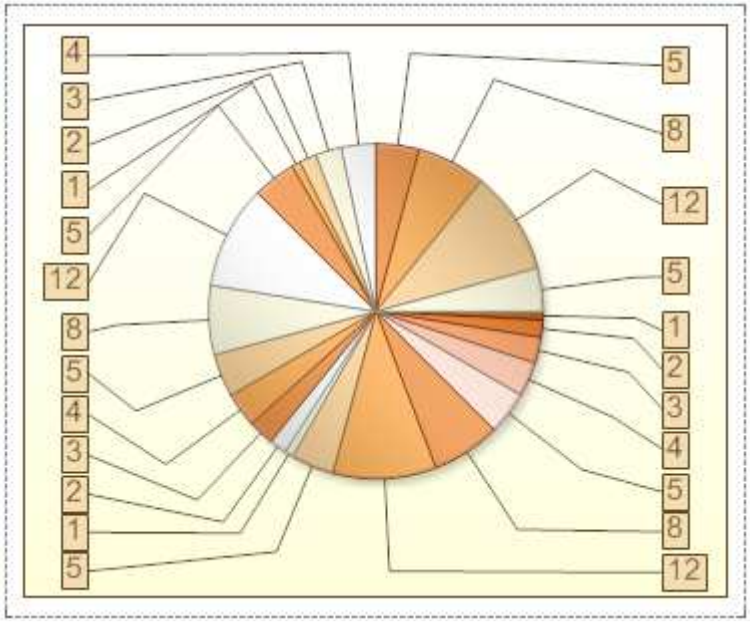
The **Value Type Separator** property is used to change the type of values separator in the series labels. By default, the **Value Type Separator** property is set to **'-'**. Any character or group of characters typed in the field of the **Value Type Separator** property, will be the delimiter (including the **'space'**). If the field is unfilled, then the separator is a **'space'**.

PreventIntersection Property

The **Prevent Intersection** property is used to avoid overlapping between series labels and with the borders of rendered values and axes. By default, the **Prevent Intersection** property is set to **false** and series labels may overlap, what makes them look bad or unreadable. The picture below shows an example of a chart, with the **Prevent Intersection** property set to **false**:

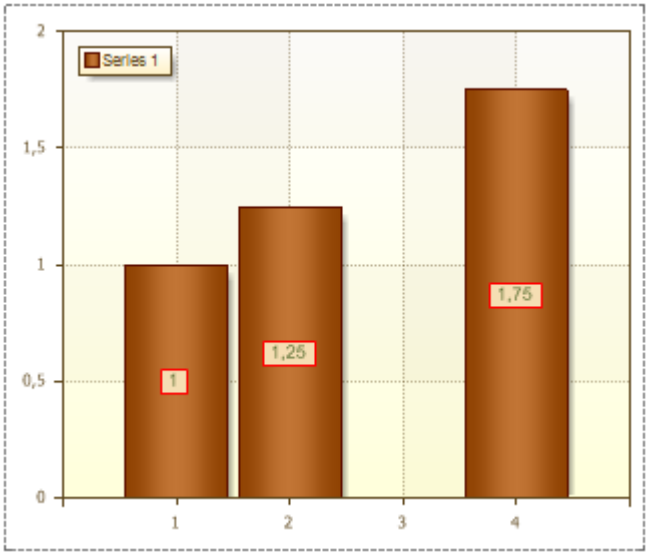


If the **Prevent Intersection** property is set to **true**, then the series labels will not overlap. The picture below shows an example of a chart, with the **Prevent Intersection** property set to **true**:

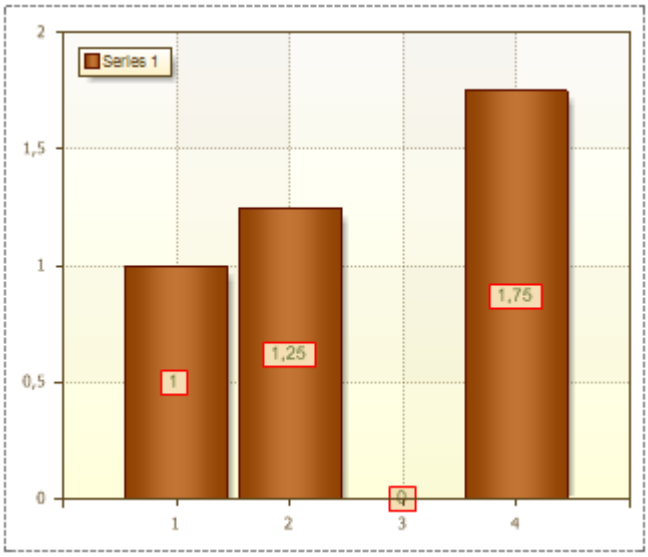


ShowOnZeroValues Property

Sometimes, when designing charts, 0 values of series can be met. Series labels of zero values can be displayed. The **Show on Zero Values** property is used to show/hide these series labels. If the **Show on Zero Values** property is set to **false**, then series labels of zero values will be hidden. The picture below shows an example of a chart with a zero value and the the **Show on Zero Values** property is set to **false**:



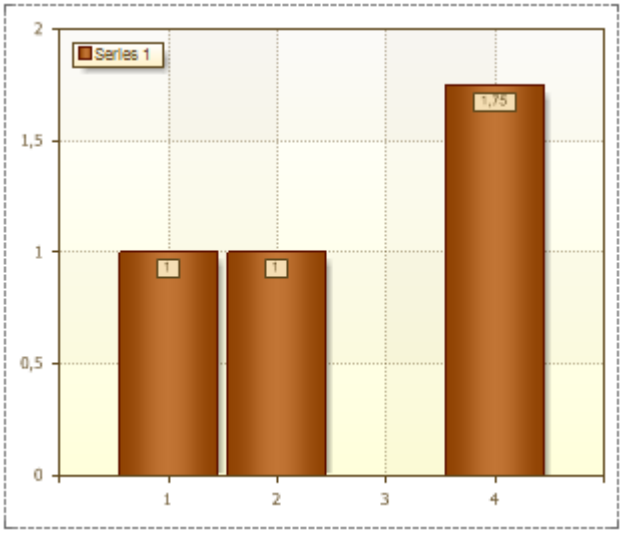
In this chart the 3rd argument is 0, and the series labels is not displayed. If the **Show on Zero Values** property is set to **true**, then series labels of zero values will be shown.The picture below shows an example of a chart with a zero value and the the **Show on Zero Values** property is set to **true**:



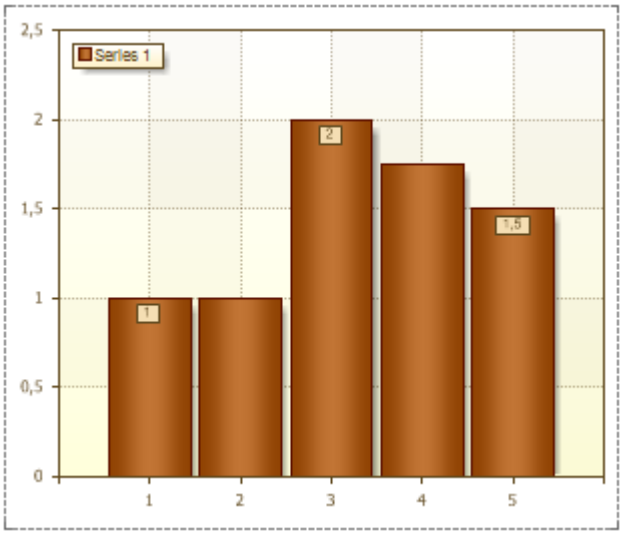
As can be seen from this picture, the 3rd argument is 0, and its title was shown.

Step Property

The **Step** property allows changing the step through what the Series Labels will be shown. By default, the **Step** property is set to **0**, so Series Labels will be shown on each Series. The picture below shows a chart with the **Step** property of Series Labels set to **0**:



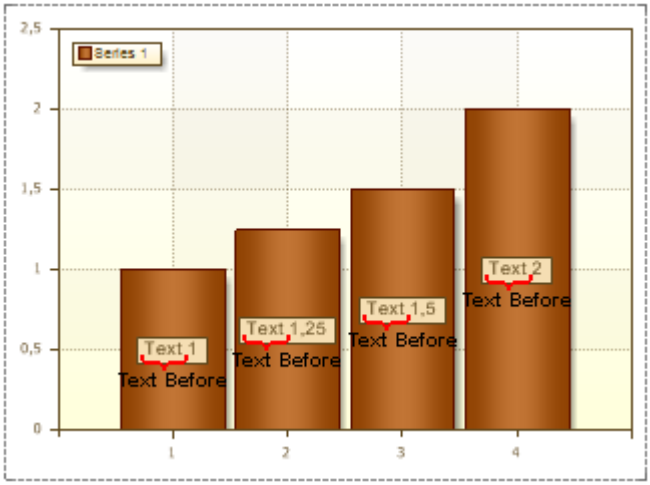
If the **Step** property is set to **2**, then Series Labels will be shown as it is shown on picture below:

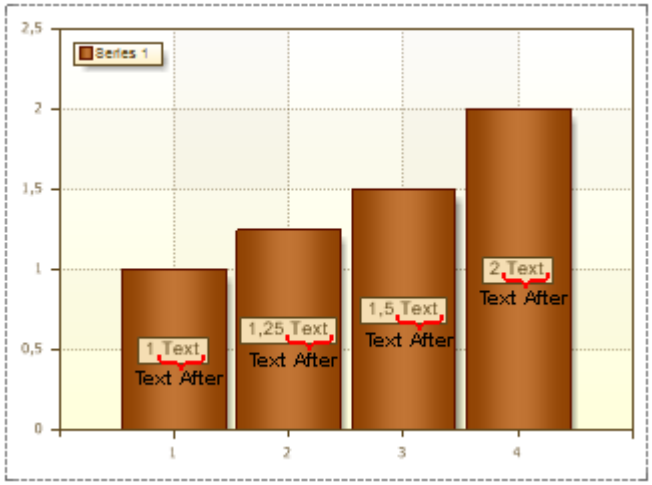


The value **1** of the **Step** property indicates that Series Labels will be shown for each value of Series.

TextBefore and TextAfter Properties

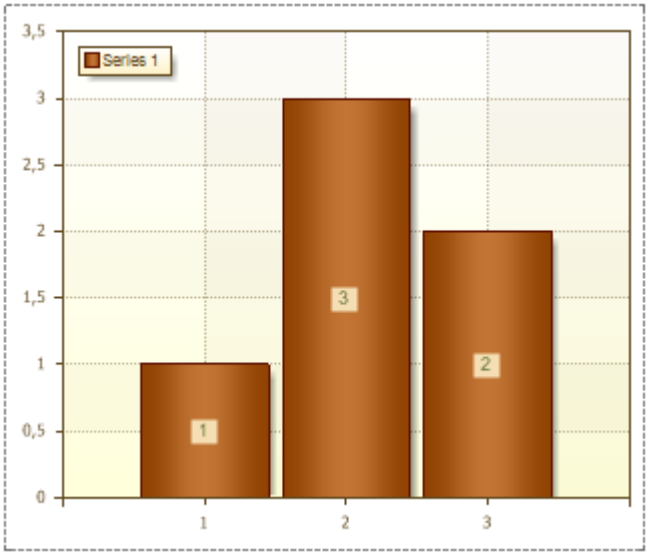
The **Text before** and **Text after** properties allows showing text before and after Series Labels. It is not necessary to use these properties. The pictures below show chart samples with a text before Series Labels (left) and a text after Series Labels (right):



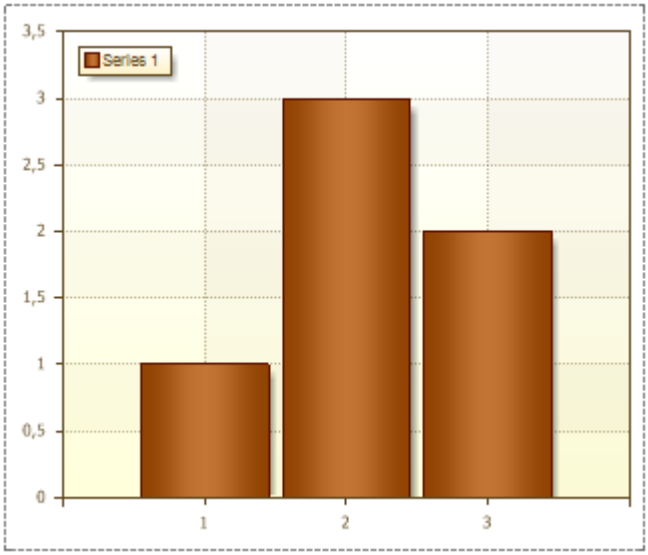


Visible Property

The **Visible** property is used to show/hide Series Labels, depending on the selected value. If the **Visible** property is set to **true**, then Series Labels are shown. The picture below shows a chart with Series Labels:



If the **Visible** property is set to **false**, then Series Labels are not displayed. The picture below shows a chart with hidden Series Labels:



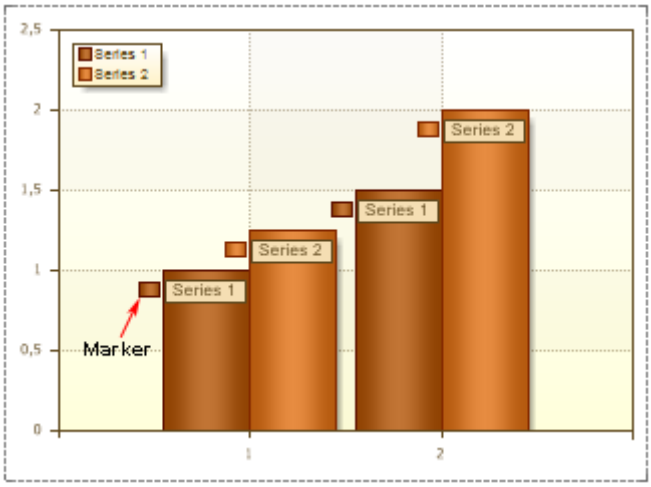
By default, the **Visible** property is set to **true**.

Marker

Marker is an icon that is shown near the Series Labels.

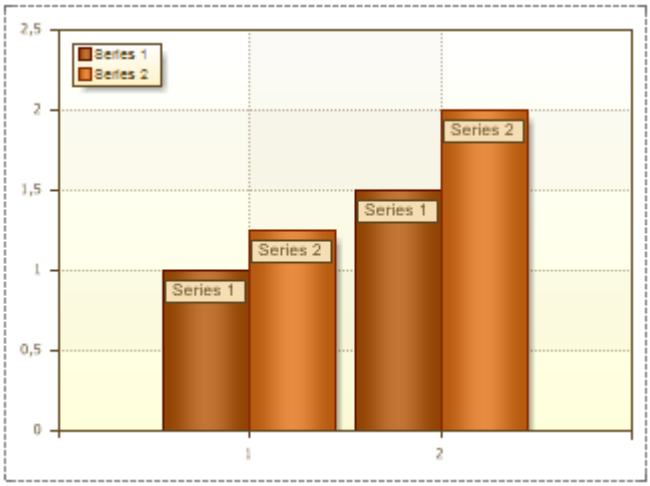
Marker Size	**8; 6**
Height	**6**
Width	**8**
Marker Visible	**True**

It is possible to change height and width of a **Marker**. The **Marker** takes the color of Series. The picture below shows a chart with **Markers**:

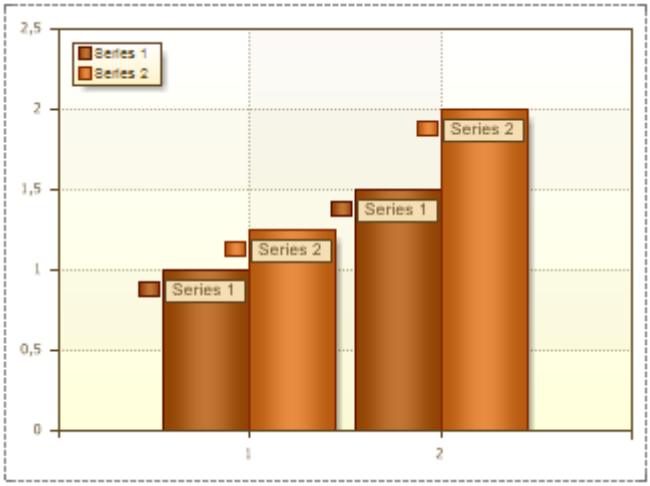


MarkerVisible Property

If to set the **Marker Visible** property to true then the **Marker** is shown. By default, the **Marker Visible** property is set to **false** and Markers are not visible. The picture below shows a chart with the **Marker Visible** property set to **false**:



The picture below shows a chart with the **Marker Visible** property set to **true**:



MarkerSize Property

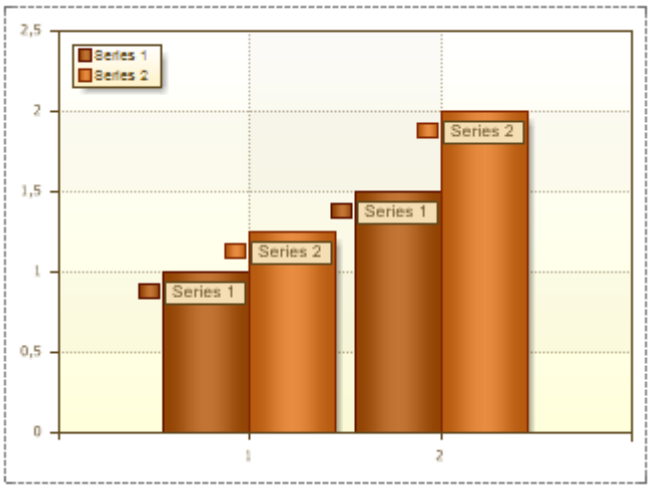
It is possible to change height and width of a **Marker**. The **Marker Size** property is used for this. It is possible to change **Height** and **Width** of a Marker.

Marker Size	**8; 6**
Height	**6**
Width	**8**
Marker Visible	**True**

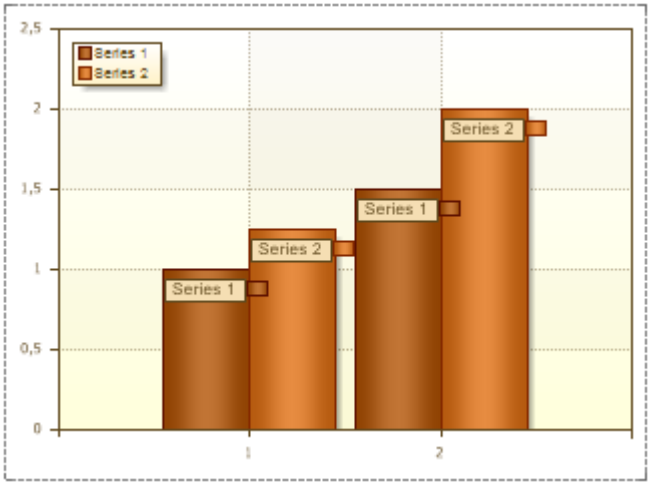
Marker Height and Width are set in pixels. If both values are more than **0**, then the Marker is shown.

MarkerAlignment Property

The **Marker Alignment** property allows aligning a marker on the left or right of Series Labels. If the **Marker Alignment** property is set to **Right**, then the marker is aligned to the left of Series Labels. The picture below shows the Markers aligned left:



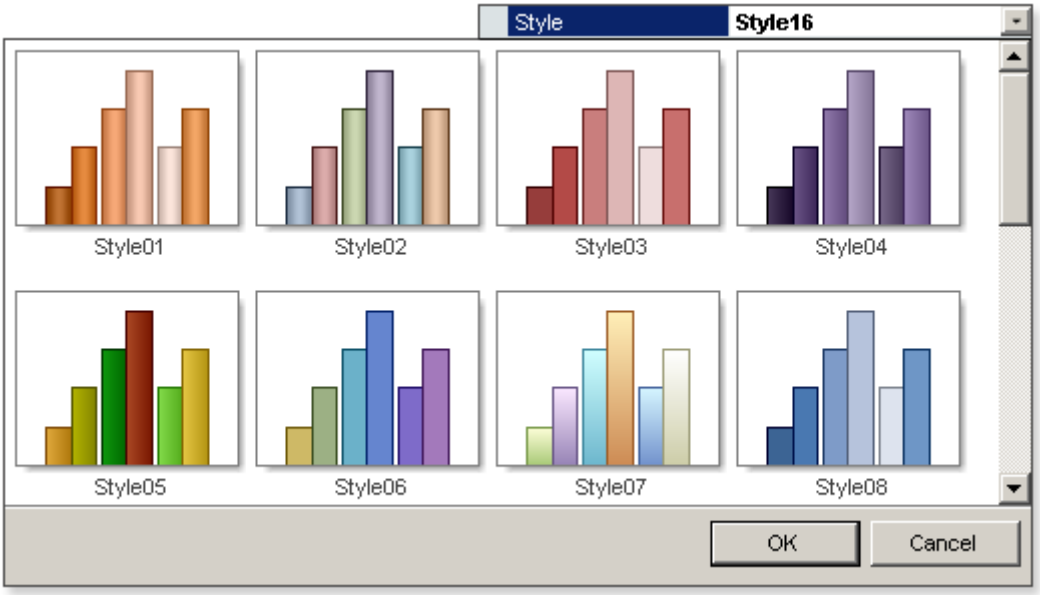
If the **Marker Alignment** property is set to **Right**, then the marker is aligned to the right of Series Labels. The picture below shows the Markers aligned right:



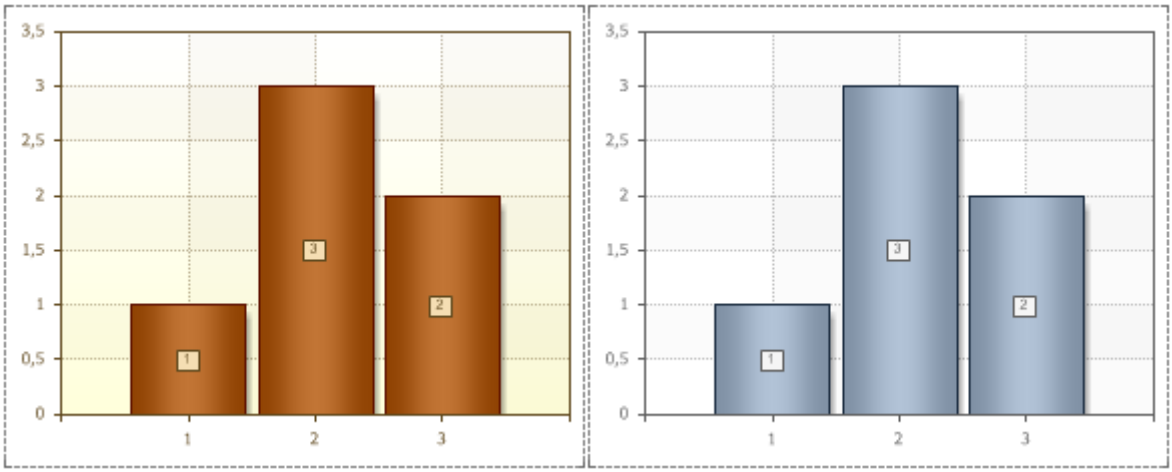
By default, the **Marker Alignment** property is set to **Left**.

Style

A style is a combination of various design attributes which can be applied to charts. The **Style** property is used to change the appearance of charts. The value of this property will be one of the chosen style diagrams.



Adding custom styles to the list of the chart styles can be done using the **Style Designer**. Also, it is possible to apply a style to each series. When working with chart styles, it is necessary to take into account the value of the **AllowApplyStyle** property. The picture below shows an example of two charts with different styles:



AllowApplyStyle Property

The **AllowApplyStyle** property is used for whether to apply a selected style in the field of the **Style** property. If the **AllowApplyStyle** property is set to **true**, then the report generator, when rendering, will take into account the value of the **Style** property. If the **AllowApplyStyle** property is set to **false**, then the report generator, when rendering, will take into account the values of appearance of series.

Report Inheritance

There are two ways of report inheritance:
1. Creation of the basic class of a report;
2. Creation of the master-report.

In both ways you should create a basic report in the designer that includes all necessary elements. You may add the following components to the basic reports:

- Pages;
- Components;
- Data sources;
- Variables;
- Connections.

After the report has been created you may either save the report as a special basic class (for this you should use the Save as command) or save the report as a regular report and then use it as a master report.
In the first case, you will get the C# or VB.NET class, and will be able to create new reports. For example:

```
Reports.Report master = new Reports.Report();
master.RegData(dataSet);
master.Design();
```

In order to use the basic report when creating a new report in the designer, you need to add the following string of a code:

```
StiReport.ReportType = typeof(Reports.Report);
```

Then all new reports will be automatically inherited from the basic class.
In the second way you need to use the following code:

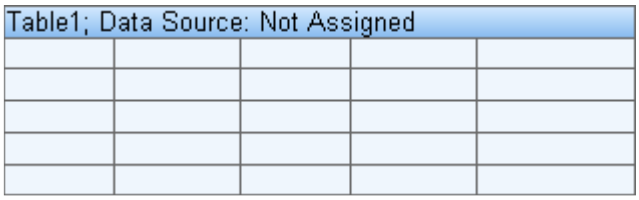
```
StiReport masterReport = new StiReport();
masterReport.Load("d:\\master-detail.mrt");

StiReport report = new StiReport();
report.RegData(dataSet);

report.MasterReport = masterReport.SaveToString();
report.Design();
```

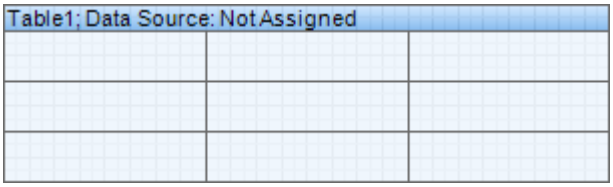
Table

The **Table** component is used to output data in a report. This component is similar to spreadsheets. The table consist of rows and columns in what data can be placed. See on a picture below a Table component with 5 columns and 5 rows.

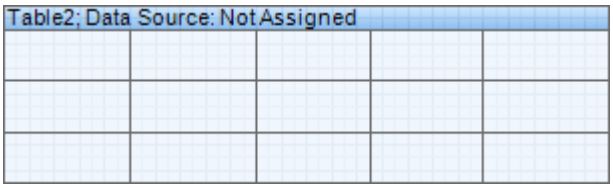


Columns

The **ColumnCount** property of the Table component is used to define the number of columns in a table. On the picture below the table with 3 columns is shown.

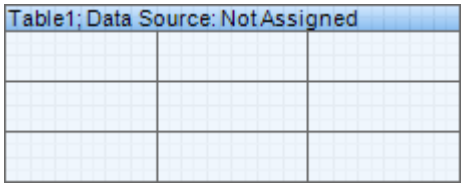


On the picture below the table with 5 columns is shown.

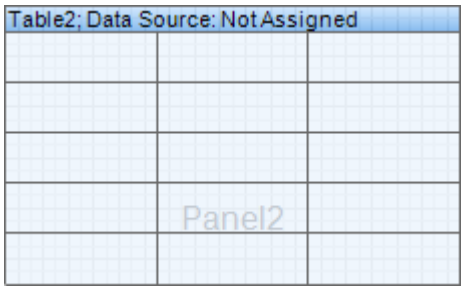


Rows

The **RowCount** property of the Table component is used to define the number of rows in a table. On the picture below the table with 3 rows is shown.

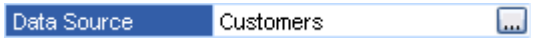


On the picture below the table with 5 rows is shown.

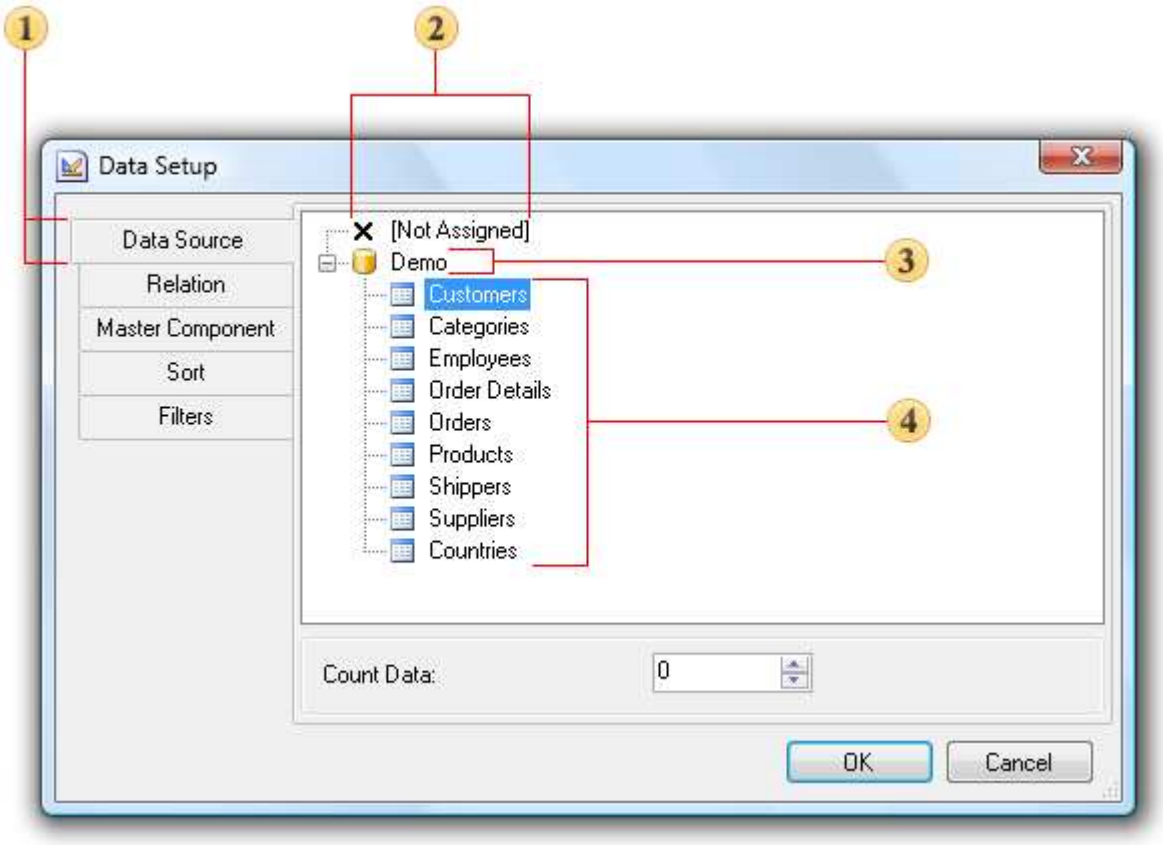


Data Source

It is necessary to define the data source to output data in the **Table** component. The reporting tool should know how many times do cells must be printed in a table. Therefore, the **Table** component should have the reference to the data source. There are several ways how to do this. You may use the Table editor. Double click on the Table header to call the editor. Also the Table editor can be called using the **DataSource** property of a Table.

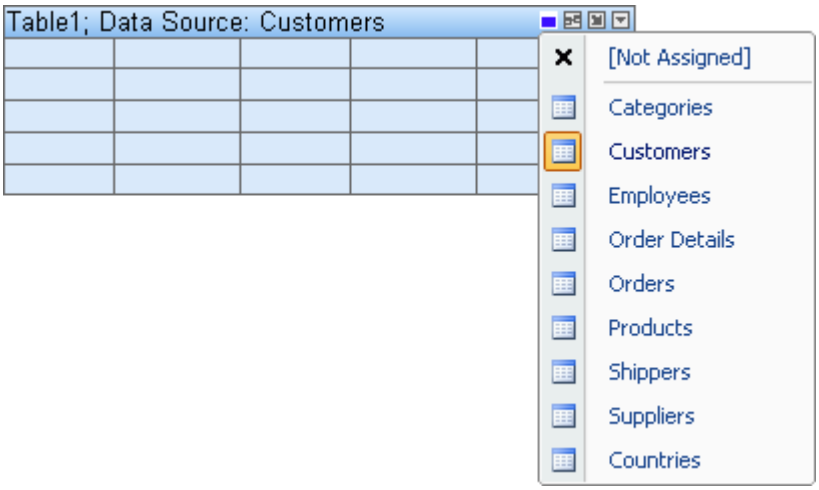


The Table editor allows selecting data source.
A data source can be selected by clicking the first tab of the editor. All data sources are grouped in categories. Each category corresponds to one connection with data in the report data dictionary. The picture below shows the Table editor.



1 The tab to select the data source;
2 Select this node if you do not need to specify the data source;
3 The "Demo" data category;
4 The "Demo" data source category.

The data source can be also selected using the quick access buttons.

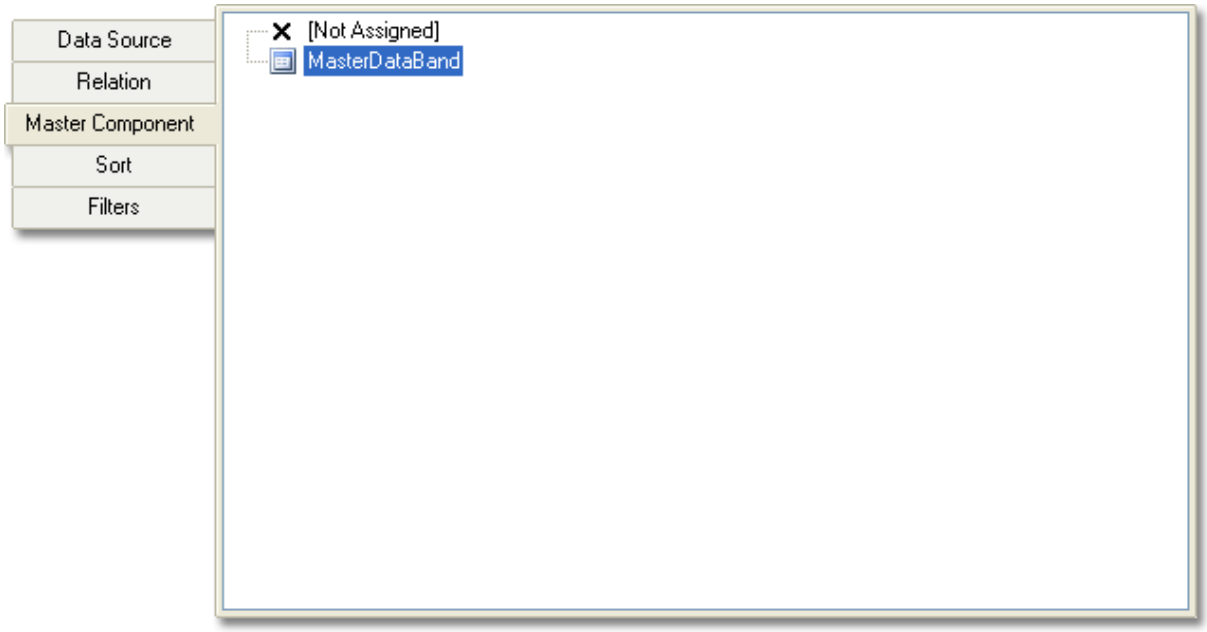


MasterComponent Property

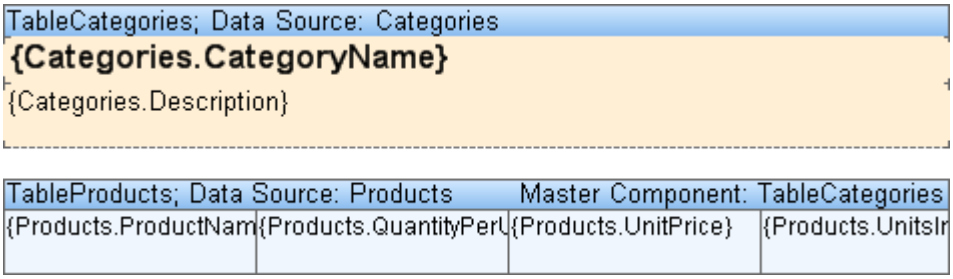
It is necessary to put two tables on a page for creating the Master-Detail using the Table component. Specify Master data source for the first table (this table is the Master table). Specify Detail data source to the second table (this table is the Detail table). Then you should bind these two tables using the **MasterComponent** property of a second table. There are several ways to set the Master table. The first way - you may set the Master table in the property grid.

MasterComponent	**MasterDataBand**

The second way is to set the Master table in the Table designer.



After filling the **MasterComponent** component two tables will be related to each other. When printing one data row from the Master data source (and, correspondingly, printing the Master table), the printing of appropriate rows from the Detail data source occurs (and, correspondingly, printing the Detail table). The Detail band will not be printed separately, only in relation to the Master band. On a picture below two related tables are represented.



TableProducts; Data Source: Products		Master Component: TableCategories	
{Products.ProductNam	{Products.QuantityPerU	{Products.UnitPrice}	{Products.UnitsIn

The picture below shows the result of two tables rendering.

Beverages

Soft drinks, coffees, teas, beers, and ales

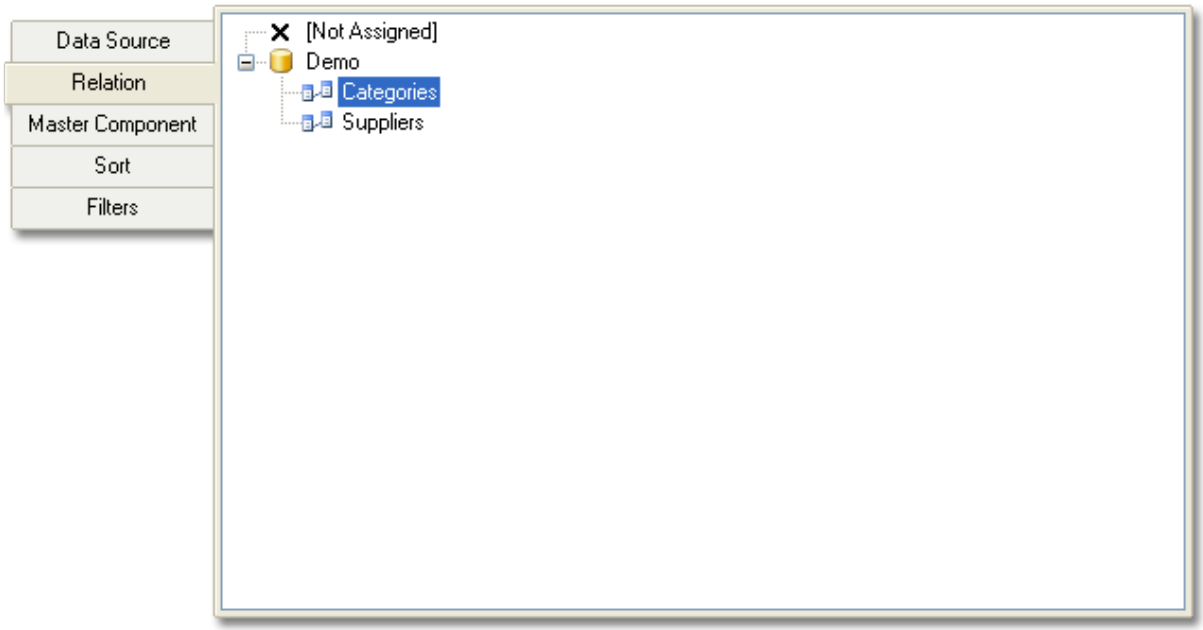
Chai	10 boxes x 20 bags	18	39
Chang	24 - 12 oz bottles	19	17
Guaraná Fantástica	12 - 355 ml cans	4,5	20
Sasquatch Ale	24 - 12 oz bottles	14	111
Steeleye Stout	24 - 12 oz bottles	18	20
Côte de Blaye	12 - 75 cl bottles	263,5	17
Chartreuse verte	750 cc per bottle	18	69
Ipoh Coffee	16 - 500 g tins	46	17
Laughing Lumberjack	24 - 12 oz bottles	14	52

Relation Property

Besides filling the **MasterComponent** property it is necessary to fill the **DataRelation** property of the Detail table. The relation is used for selecting the detailed data only for the specific row of the Master table. If the relation will not be specified then all records of the Detail data source of the Detail table will be output for each row of the Master data source of the Master table.

Data Relation	Categories

The relation can be selected using the **Data** table editor.



The selection is done between relations which are created between Master and Detail data sources and in what the Detail data source is the child data source.

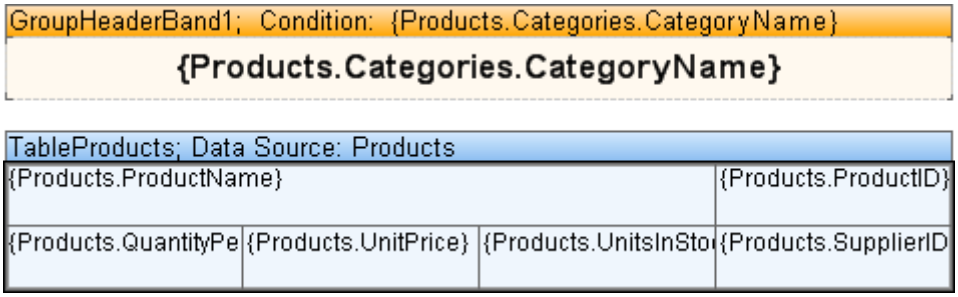
Tables and Bands in Master-Detail Lists

It is allowed binding bands and tables when rendering the Master-Detail reports. For example, the master component can be a band and the Detail component can be a table. The template of such a report is shown on a picture below.

The number of **Data** bands and **Tables** which interacts between each other is unlimited.

Tables and Grouping

It is easy to add grouping to a report with a table. For this you should put the **GroupHeader** band before the **Table** component and the **GroupFooter** band after the Table. The condition of grouping is specified for the **GroupHeader** component. The text component that outputs the condition of grouping is placed in the **GroupHeader** band. It is enough to group a table by the specified condition. On a picture below the table of grouping is shown.



See the picture below that demonstrates the report with grouping and a table.

Beverages

Côte de Blaye			38
12 - 75 cl bottles	263,5	17	18
Chartreuse verte			39
750 cc per bottle	18	69	18
Steeleye Stout			35
24 - 12 oz bottles	18	20	16
Guaraná Fantástica			24
12 - 355 ml cans	4,5	20	10

Table Header

Rows in a Table component can be specified as a header. In other words these rows will always be output in the beginning of a table. The **HeaderRowsCount** property is used to indicate how many rows will shown as headers. By default this property is set to 0. The number of header rows cannot be more than the number of rows in a table.

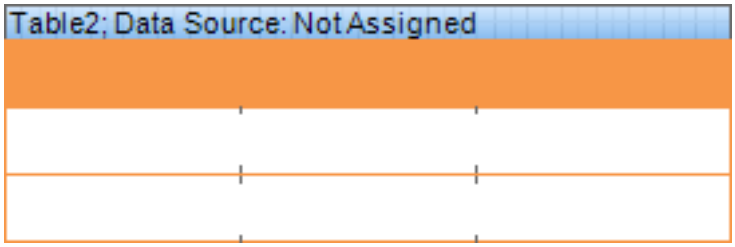
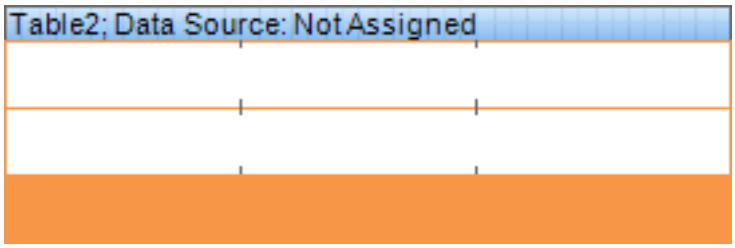


Table Footer

A table may include footer rows. These rows are output on the bottom of a table. The **FooterRowsCount** property is used to indicate how many rows will be used as footers. By default this property is set to 0. The number of footer rows cannot be more than the number of rows in a table.



Cells Width Autochange

When report rendering using the **Table** component, width of some cells can be changed. As a result this may lead to the change of a table size. There are two properties of Table component which are used to adjust cells size: the **AutoWidthType** property and the **AutoWidth** property.

AutoWidth Property

The **AutoWidth** property of a **Table** component indicates whether the reporting tool will fix the cells size after the report rendering.

1. The **AutoWidth** property is set to **None**. Column size is not changed. In this case setting the **AutoWidthType** property of a table and the **FixedWidth** property of cells will not affect on a table.
2. The **AutoWidth** property is set to **Page**. If a rendered table is placed on several pages then columns will have different width on different pages. It depends on data.

3. The **AutoWidth** property is set to **Report**. If a rendered table is placed on several pages then columns will have the same width in a report.

AutoWidthType Property

The **AutoWidthType** property of a table indicates how the reporting tool will fix cells width after report rendering.

1. None
Columns width is set depending on the cells contents of all table (the longest line by column is taken). If the **FixedWidth** property is set to true, then the column size is not changed.

Franchi S.p.A.	Via Monte Bianco 34	011-4988260	Sales Representative
Furia Bacalhau e Frutos do Mar	Jardim das rosas n. 32	(1) 354-2534	Sales Manager
Galería del gastrónomo	Rambla de Cataluña, 23	(93) 203 4560	Marketing Manager
Godos Cocina Típica	C/ Romero, 33	(95) 555 82 82	Sales Manager

(c) 2003-2009 Stimulsoft Page 1 of 3

CompanyName	Address	Phone	ContactTitle
Gourmet Lanchonetes	Av. Brasil, 442	(11) 555-9482	Sales Associate
Great Lakes Food Market	2732 Baker Blvd.	(503) 555-7555	Marketing Manager
GROSELLA-Restaurante	5ª Ave. Los Palos Grandes	(2) 283-2951	Owner

2. FullTable
Column width is set depending on the table width. In other words the width of all column cells is checked first (the column width is set by the longest line). If there is free space then it is equally distributed between all columns. If there is no enough space to output the longest lines, then the width of columns is decreased in equal parts between all columns.

Franchi S.p.A.	Via Monte Bianco 34	011-4988260	Sales Representative
Furia Bacalhau e Frutos do Mar	Jardim das rosas n. 32	(1) 354-2534	Sales Manager
Galería del gastrónomo	Rambla de Cataluña, 23	(93) 203 4560	Marketing Manager
Godos Cocina Típica	C/ Romero, 33	(95) 555 82 82	Sales Manager

(c) 2003-2009 Stimulsoft Page 1 of 3

CompanyName	Address	Phone	ContactTitle
Gourmet Lanchonetes	Av. Brasil, 442	(11) 555-9482	Sales Associate
Great Lakes Food Market	2732 Baker Blvd.	(503) 555-7555	Marketing Manager
GROSELLA-Restaurante	5ª Ave. Los Palos Grandes	(2) 283-2951	Owner

3. LastColumns

Column width is set depending on the table width. In other words the width of all column cells is checked first (the column width is set by the longest line). If there is free space then it is distributed to the last column which **FixedWidth** property is set to **false**. If there is no enough space to output the longest lines, then the width of the last columns is decreased and distributed between all columns which **FixedWidth** properties are set to **false**.

La corne d'abondance	67, avenue de l'Europe	30.59.84.10	Sales Representative
La maison d'Asie	1 rue Alsace-Lorraine	61.77.61.10	Sales Manager
Laughing Bacchus Wine Cellars	1900 Oak St.	(604) 555-3392	Marketing Assistant
Lazy K Kountry Store	12 Orchestra Terrace	(509) 555-7969	Marketing Manager

(c) 2003-2009 Stimulsoft

CompanyName	Address	Phone	ContactTitle
Lehmanns Marktstand	Magazinweg 7	069-0245984	Sales Representative
Let's Stop N Shop	87 Polk St. Suite 5	(415) 555-5938	Owner
LILA-Supermercado	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	(9) 331-6954	Accounting Manager

FixedWidth Property

The **FixedWidth** property is used together with the **AutoWidth** property of a **Table** component. If a table changes the column size (depending on the **AutoWidth** property) then the **FixedWidth** property that is set to **true** does not allow these changes. On a pictures below samples of using these property is shown. On the first picture the **FixedWidth** property is not used.

Company	Address	Phone	Contact
Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative
Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative
Blondesddsl père et fils	24, place Kléber	88.60.15.31	Marketing Manager

On the second picture the **FixedWidth** property of the Phone column is set to **true**.

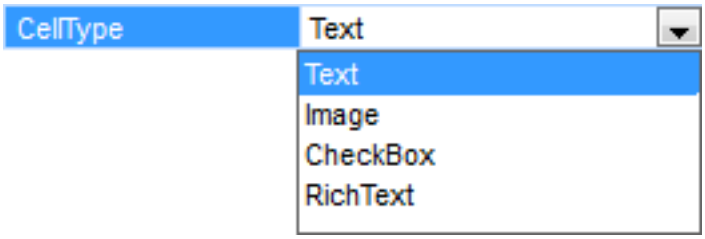
Company	Address	Phone	Contact
Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
Around the Horn	120 Hanover Sq.	(171) 555-77	Sales Representative
Berglunds snabbköp	Berguvsvägen 8	0921-12 34 6	Order Administrator
Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative
Blondesddsl père et fils	24, place Kléber	88.60.15.31	Marketing Manager

CellType Property

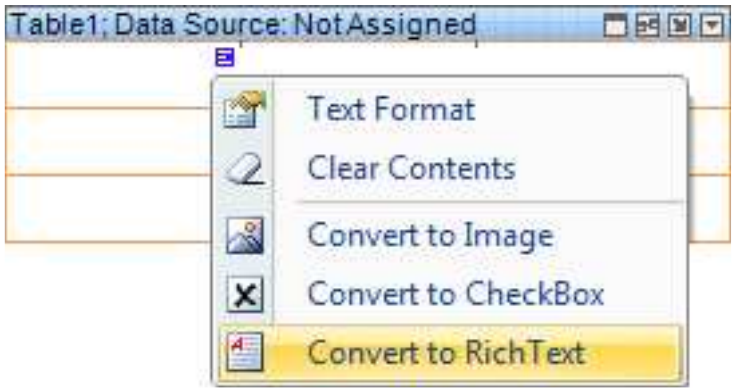
There are different types of cells can be placed In the Table component. They are a text, an image, a check, and a rich text.

1. Text is a cell will be output as a text. Cell settings are the same as the settings of a Text component;
2. Image is a cell will be output as a text.Cell settings are the same as the settings of an Image component;
3. Check is a cell will be output as a check for Boolean types of data. Cell settings are the same as the settings of a Check component;
4. Rich text is a cell will be output as a rich text. Cell settings are the same as the settings of a Rich Text component.

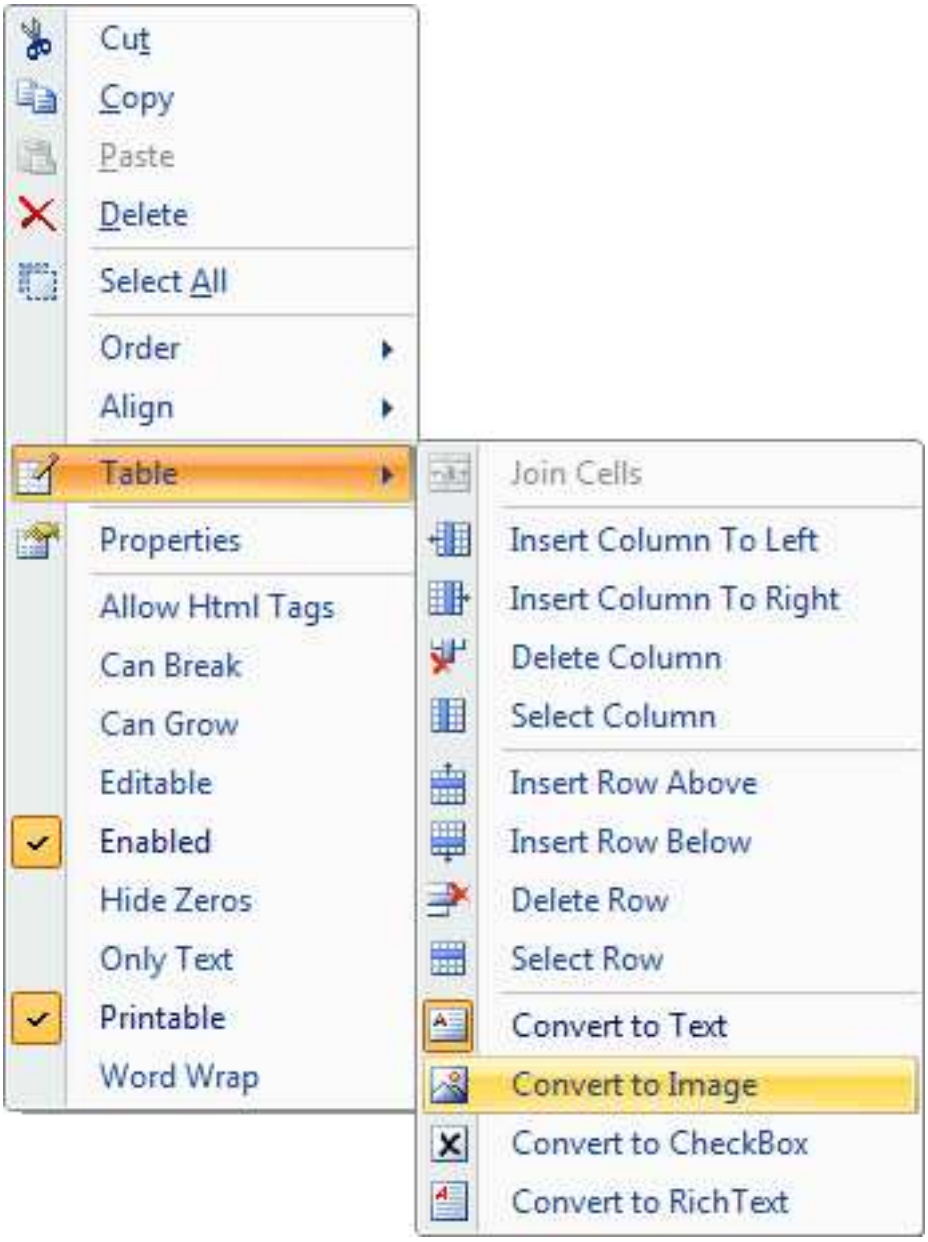
The **CellType** property is used to indicate a cell type.



Also it is possible to indicate a cell style by clicking the quick access button of a cell.

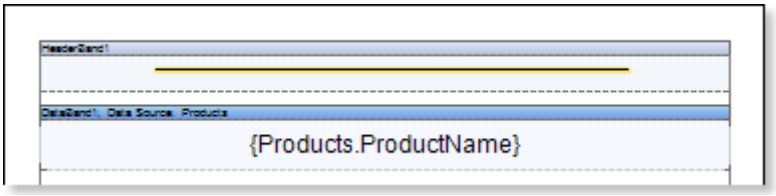


Or the context menu of a cell.

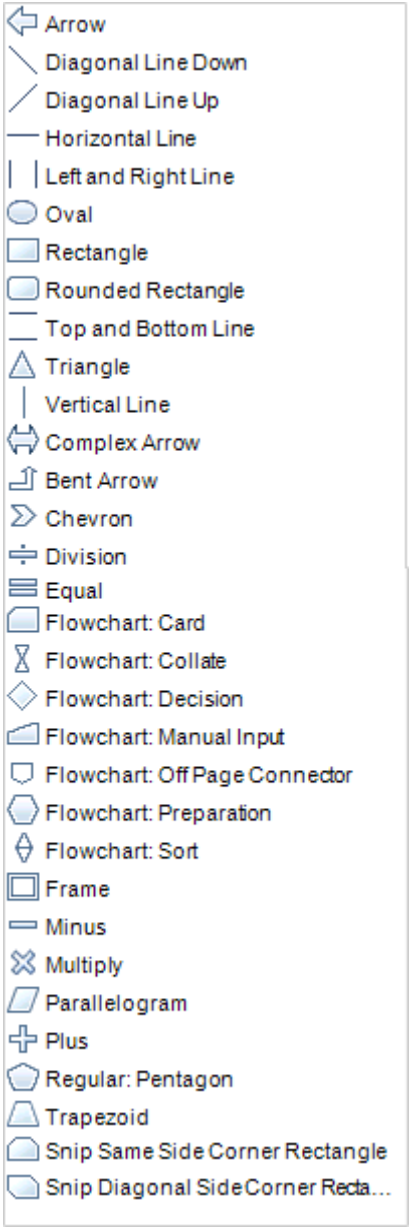


Primitives

Primitives include: **Horizontal Line** and **Shape**. Cross-primitives include: **Vertical Line**, **Rectangle** and **Rounded Rectangle**. **Horizontal line** is a line in the horizontal plane, which start and end points are located on the same component in a report. The picture below shows a report template with a list in which a **Horizontal Line** is located in the **HeaderBand**:

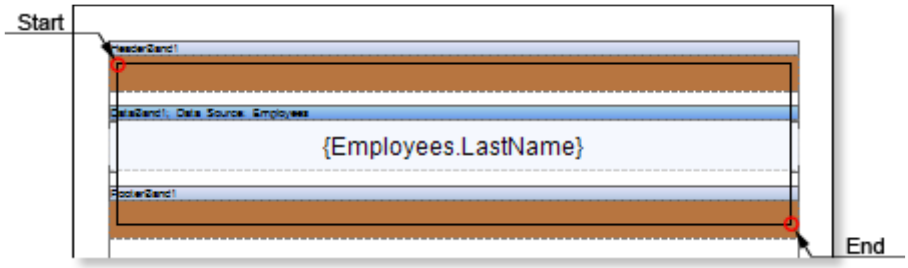


The **Shape** is a report component, which, depending on the type, shows this or that primitive. The **Shape Type** property is used to specify a primitive type. The picture below shows a list of values of the **Shape Type** property:

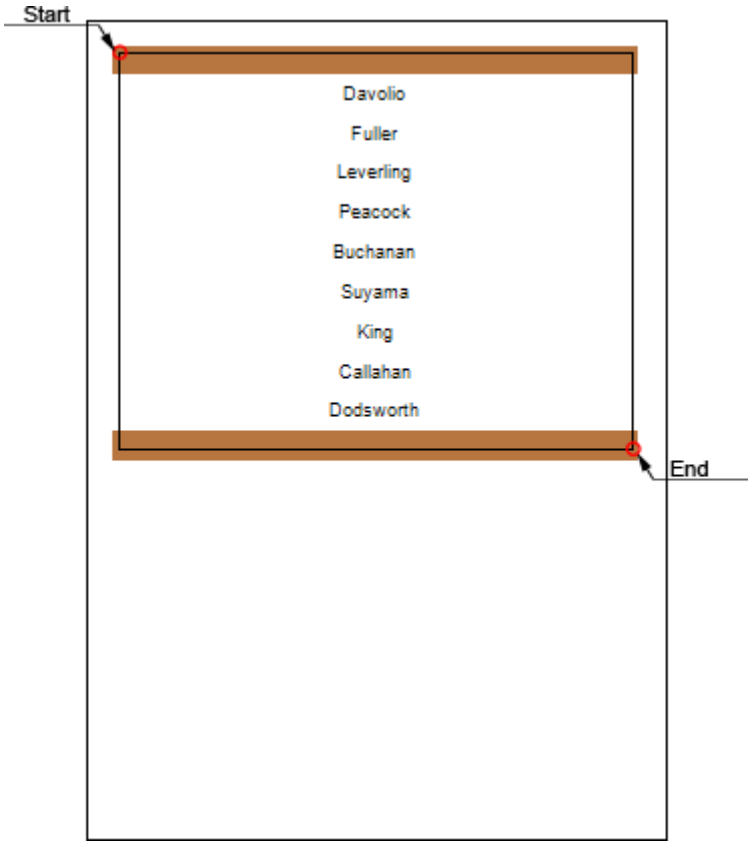


Cross-Primitives

Cross-primitives include: **Vertical Line**, **Rectangle** and **Rounded Rectangle**. The start and end points of cross-primitives can be placed on different components of a report. When designing a report with cross-primitives the report generator renders start and end points of a vertical line, and then, between two points, it renders a vertical line. The picture below shows an example of a report template with a rectangle:

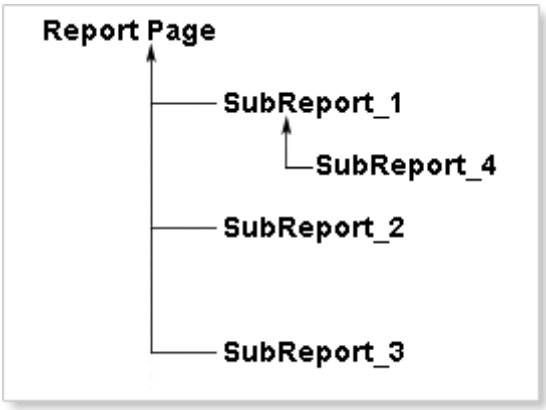


As can be seen in the picture, the start and end points of the **Rectangle** component are located on different bands: the start point is located in the **HeaderBand**, and the end point is in the **FooterBand**. When rendering the report, the report generator will render start and end points of the rectangle, and then it will render rectangle sides. The picture below shows an example of the rendered report pages with the **Rectangle** cross-primitive:



Sub-Reports

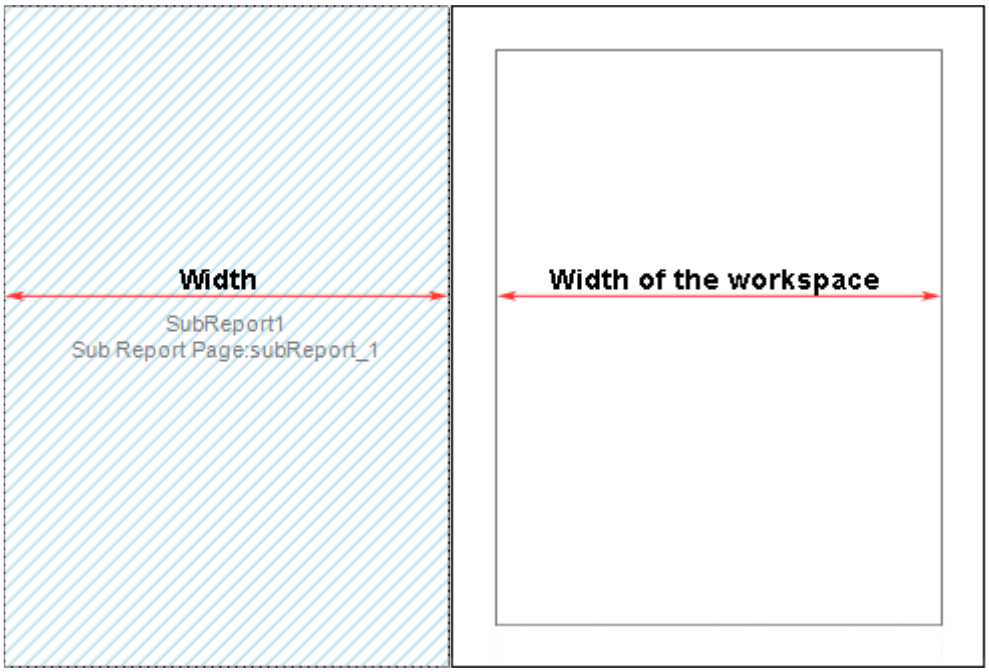
The **Sub-Report** is a report component that can be placed on a band, page, panel or any other component that can be a container for the sub-report. When placing this component, the reporting tool will add nested page into the report and bind it with the **Sub-Report**. When rendering a report, the reporting tool will build all sub-reports and place them in this container. On the nested page a report that has any structure can be created. Also the **Sub-Report** component can be placed on the nested page, so the nested page of the second level will be created. In other words it is possible to create complex hierarchy in a report. The picture below shows the hierarchy of a sub-report:



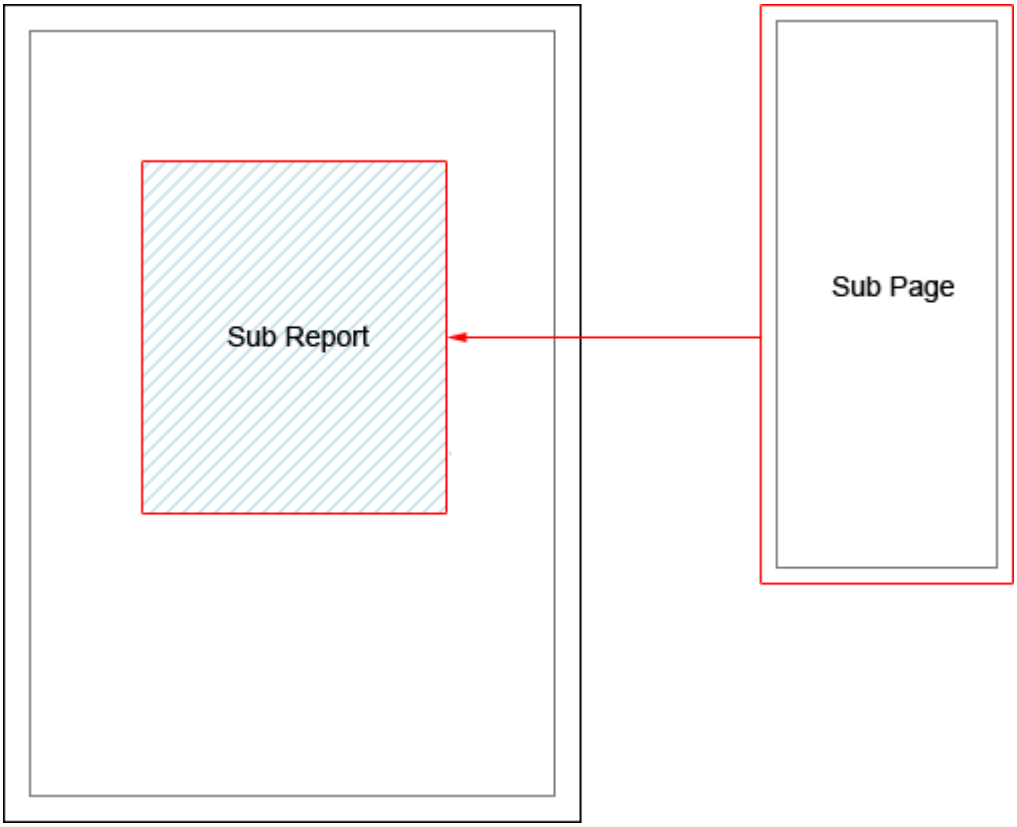
As seen on the picture above, **SubReport_1**, **SubReport_2**, **SubReport_3** components are placed on the report page and the **SubReport_4** component is placed on the sub-report page of the **SubReport_1** component. So the page of the **SubReport_4** component is the nested page of the second level.

Sub-Reports on Page

The **Sub-Report** component can be placed on any part of a page. The width of the nested page depends on the width of the **Sub-Report** component. The picture below shows a sample of the **Sub-Report** component and nested page:

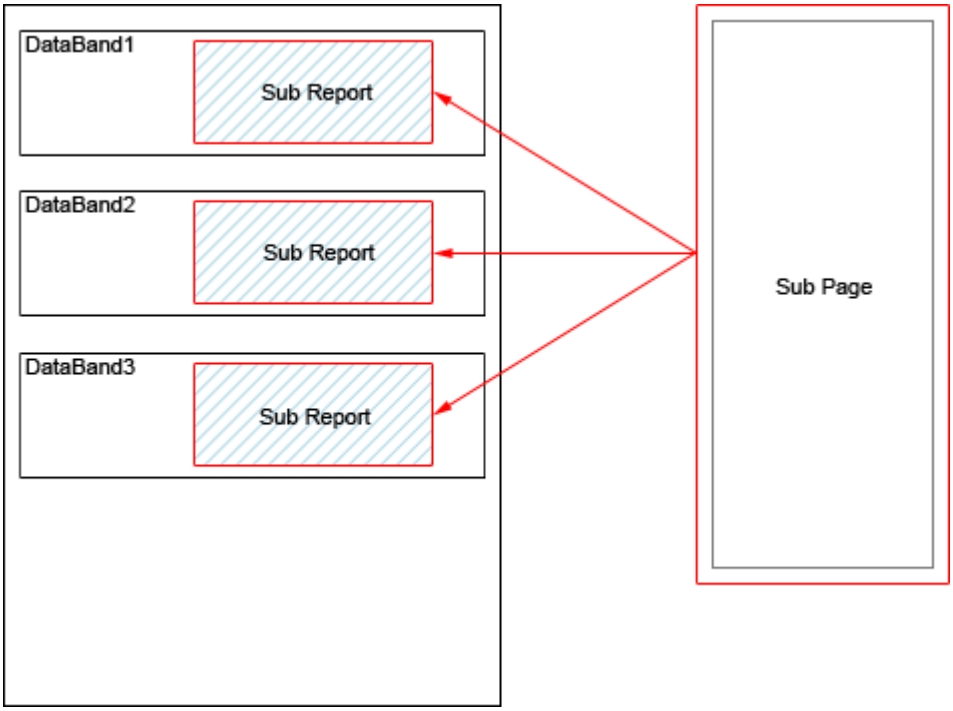


The **CanGrow** property of the **Sub-Report** component is always set to **true** but, when placing this component, it cannot be grown by height. So you should take into the account the height of the component on the nested page: it should not be higher than the **Sub-Report** component. When rendering a report, the **Sub-Report** component, placed on the report template, will be rendered as the report page item. When rendering a report, the reporting tool will render all sub-reports and place them in the container of the **Sub-Report** component. The picture below shows a sample of placing the nested page in a report:



Sub-Reports on Data Band

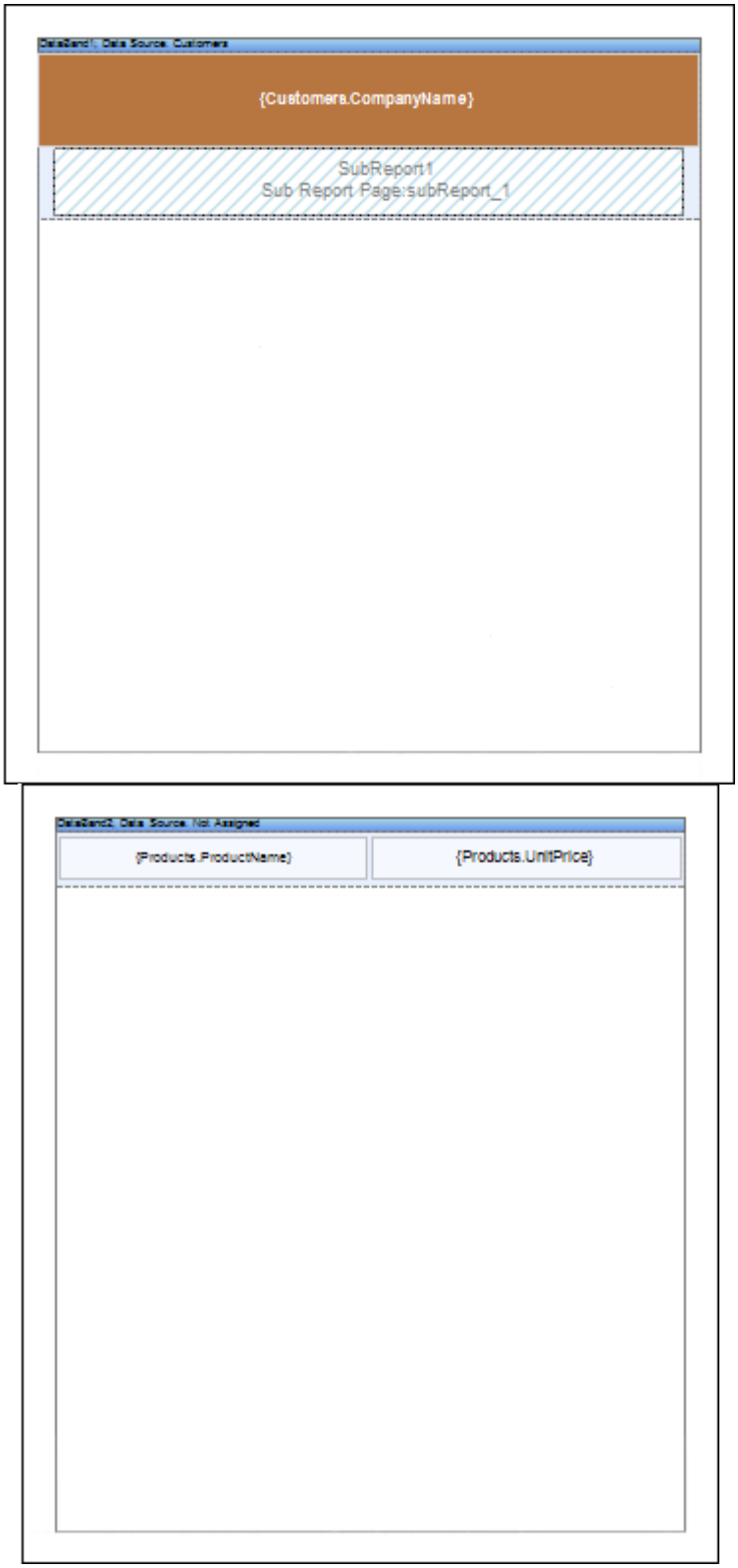
The **Sub-Report** component can be placed on the **DataBand**. When rendering a report, the **Sub-Report** will be rendered as the item of the **DataBand**, so this component will be printed in each **DataBand**. The picture below shows the scheme of rendering of the sub-report when placing the **Sub-Report** component in the **DataBand**:



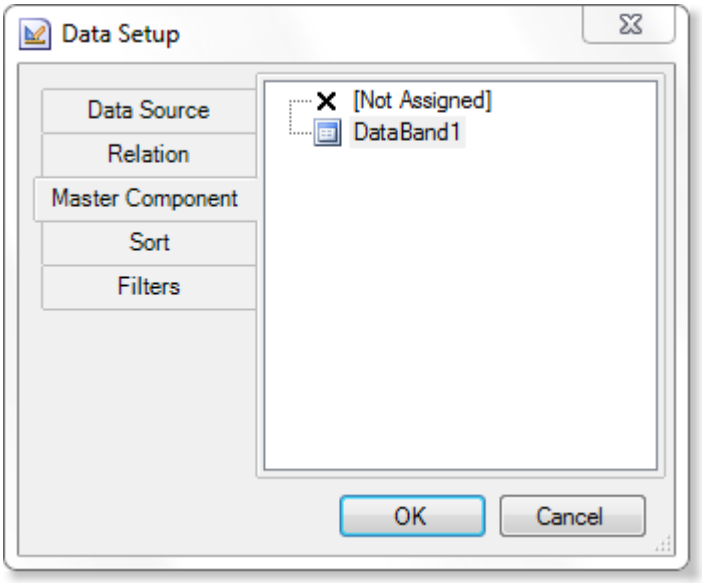
In this case the height of the component on the sub-report page of a report will be higher than the height of the **Sub-Report** component. So the **Sub Report** component is placed in the **DataBand** and rendered as the item of the **DataBand**, and, in this case, the **CanGrow** property works and the component can grow by height.

Master-Detail Reports and Sub-Reports

It is possible to design the **Master-Detail** report using the **Sub-Report** component. Put **DataBand1** on a page of a report template. Insert **Sub-Report** component into this band. Put **DataBand2** on the sub-report page. The picture below shows the report template:



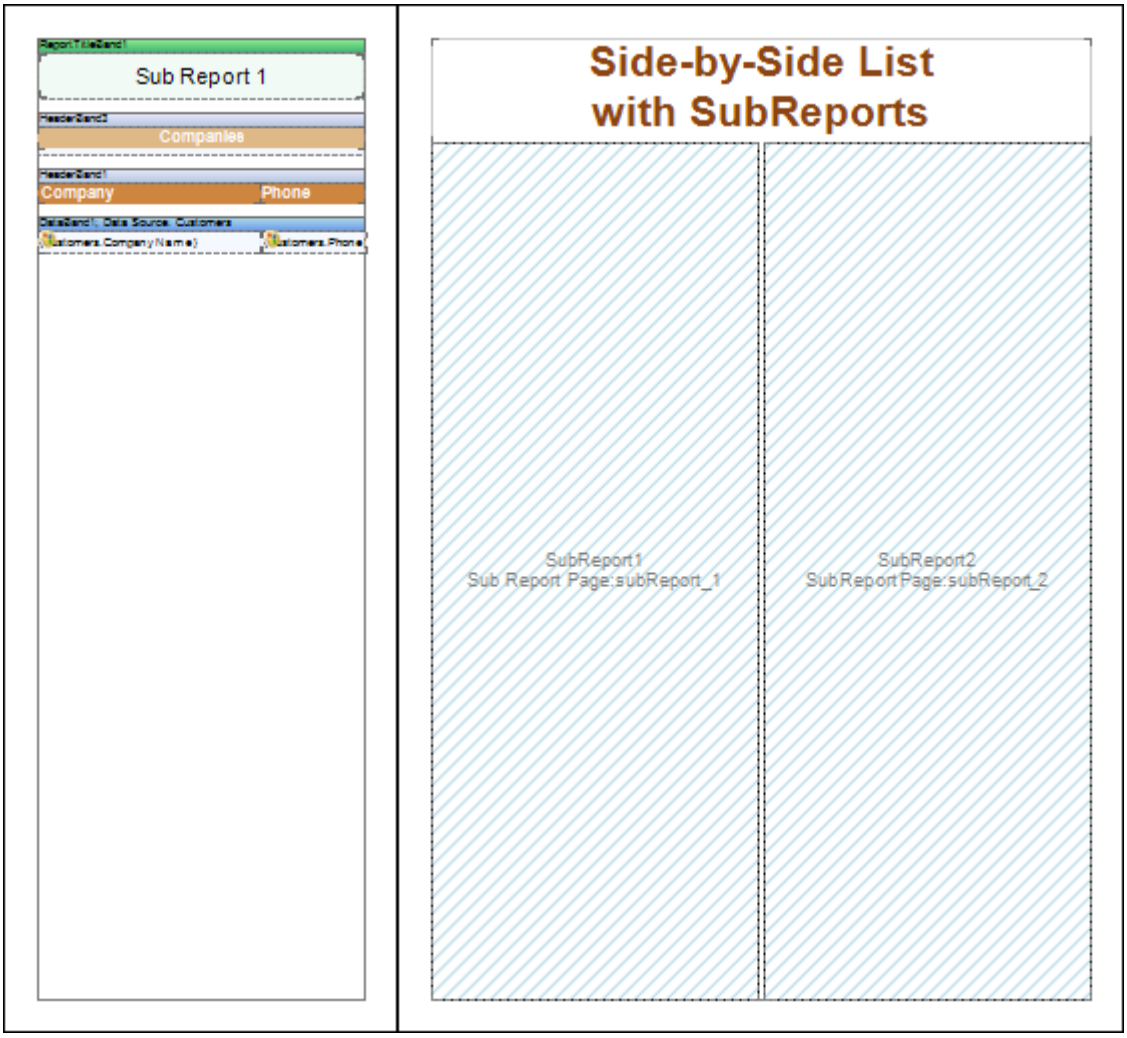
In this example the **DataBand1** can be defined as the **Master** for the **DataBand2** that is placed in the sub-report page of a report. For this you need to choose the **Master** component in the data settings. The picture below shows the sample of the **Data Setup** window:

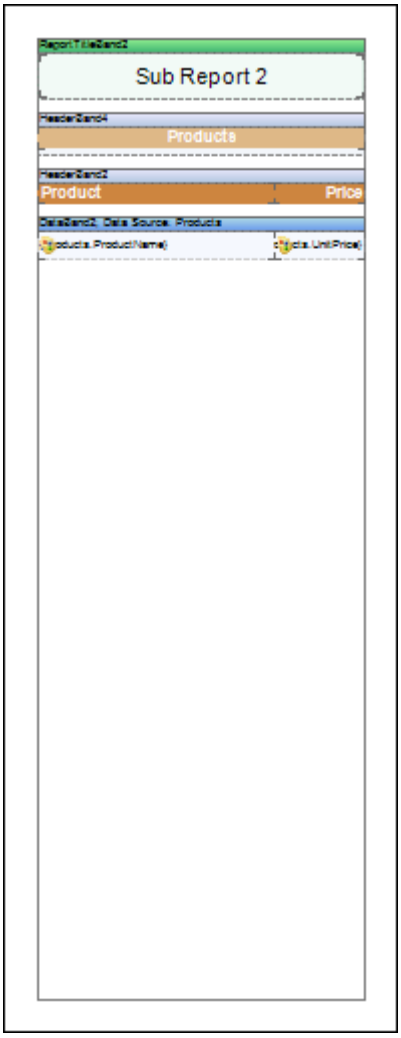


As you can see, the **DataBand1**, that is placed on the report page, is the **Master** in the **Master-Detail** report. If several **DataBands** are placed on the sub-report page then, when creating the **Master-Detail** report, the **Master** is either the **DataBand** in what the **Sub-Report** is placed or any other **DataBand**, placed in the sub-report page.

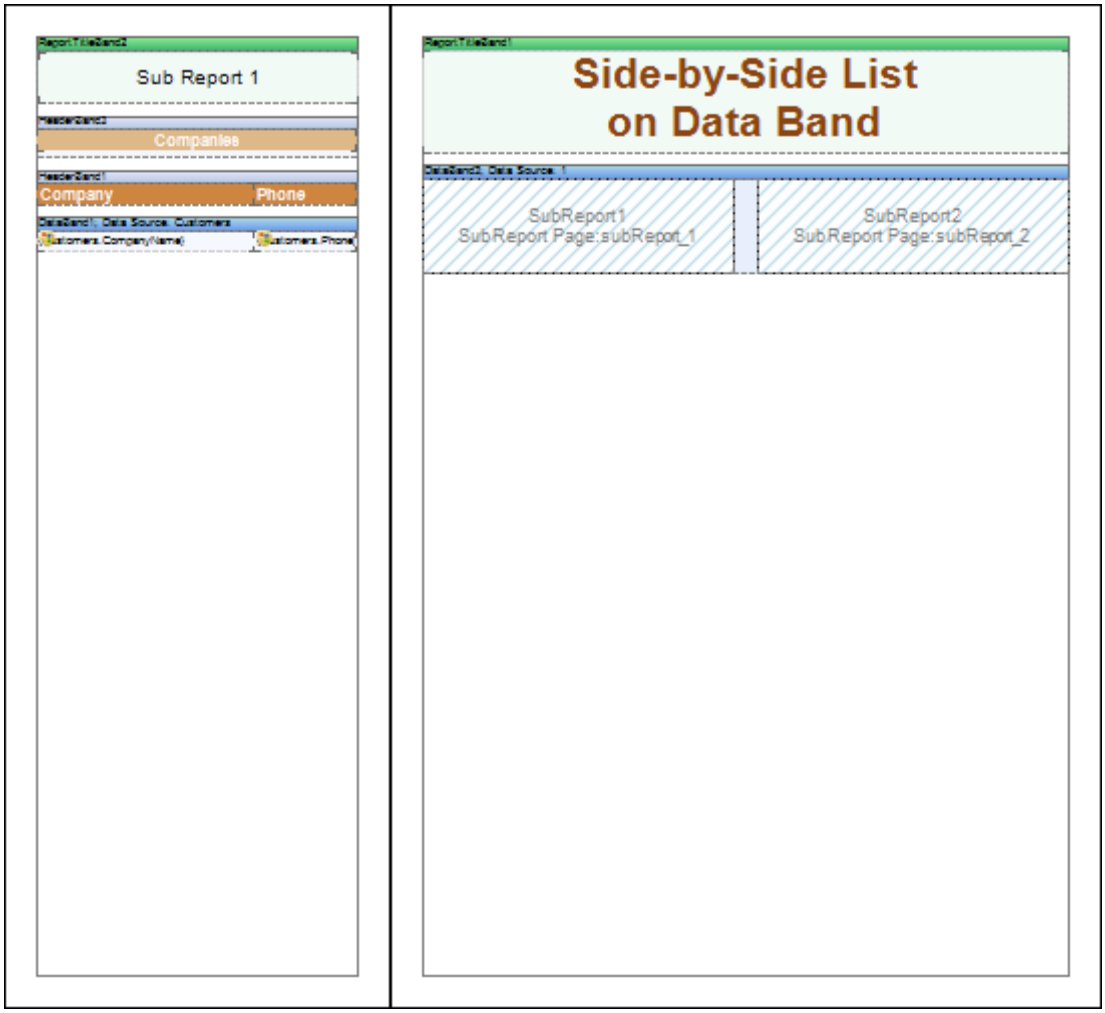
Side-by-Side Reports and Sub-Reports

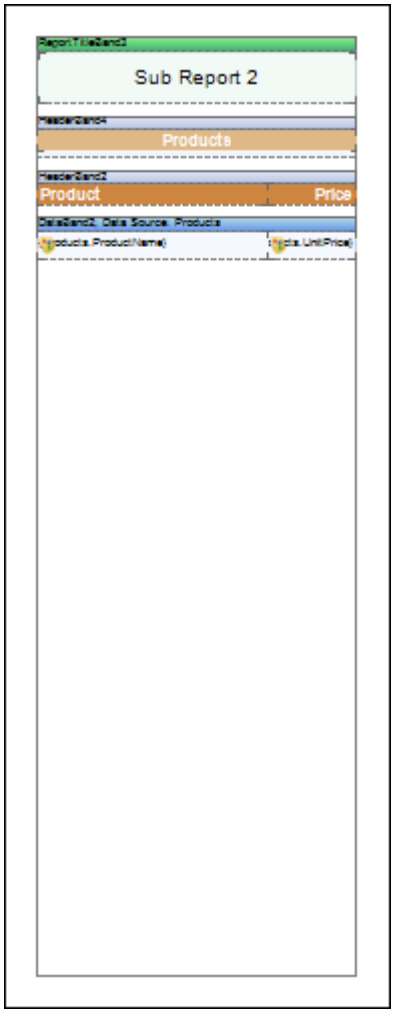
You can use the **Sub-Report** component to create the **Side-by-side** report. The **Side-by-side** report consists of independent lists of data, located side by side. The picture below shows an example of a **Side-by-side** report template with the location of the **Sub-Report** component on on a page of the report template:


Sub Report 1
Companies
Company
Phone
Side-by-Side List
with SubReports
SubReport1
Sub Report Page:subReport_1
SubReport2
SubReport Page:subReport_2



As you can see on the picture above, when rendering a report, independent data lists will be displayed, i.e. two **Side-by-side** sub-reports will be built. Thus it is possible to build more complex reports: for example, put three **Sub-Report** components together side by side, and then, when rendering a report, three independent data lists, i.e. three **Side-by-side** sub-reports will be output. You should also remember that the **Sub-Report** can be placed in the **DataBand**. Accordingly, put two or more **Sub-Report** components to build **Side-by-side** reports in one **DataBand**. The picture below shows an example of the **Side-by-side** report templates with the location of the **Sub-Report** component in the **DataBand**:


Sub Report 1
Companies
Company
Phone
Side-by-Side List
on Data Band
SubReport1
SubReport Page:subReport_1
SubReport2
SubReport Page:subReport_2



Scripts

Stimulsoft Reports supports a choice of languages for report generation.

Programming Language of Report

The report generator uses a single specified programming language to generate the report code and handle report events. If the current programming language of a report does not suit your requirements you can change it. The options are currently C# or VB.

Changing The Language Of The Current Report

To do this select **File | Report Setup**. A new dialog will be displayed

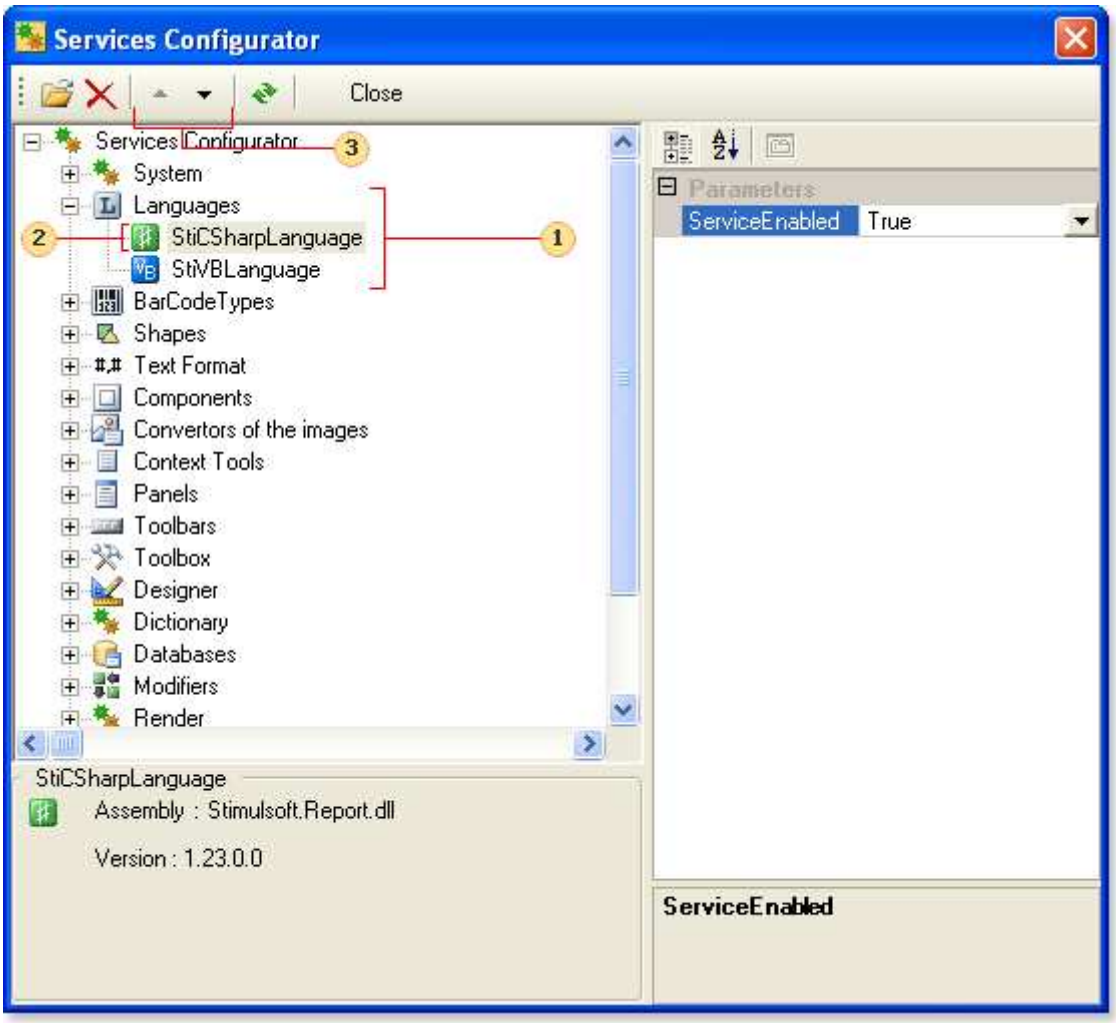
In the Language group select a new programming language and press Ok. The current programming language will then be changed.

Important: The underlying report code will have to be regenerated for the entire report, and any changes which have been made within the report code will be lost.

Changing The Default Language

It is not convenient to change the programming language each time a report is rendered, so Stimulsoft Reports allows you to set the default programming language used by all new reports.

To do this you should use the Services Configurator utility. All programming languages in Stimulsoft Reports are located under the Languages node. The language which is shown first in the list is the default programming language. For example, on the picture below the C# programming language is the default programming language. To make VB.Net the default programming language simply drag the StiCSharpLanguage service down one position with the mouse or use the up and down buttons to re-order the languages.



Services Configurator Dialog

	Supported programming languages
	Default programming language. When reports are rendered, the first programming language in the Languages node of the Services tree will be used as the default.
	Up and Down Buttons. To change the default programming language select a language and use these buttons to change its position in the tree.

Report Code

When you create a new report its source code, often called the report script, is generated automatically using either C# or VB.NET programming language depending on the currently selected default. You can use only one of these programming languages at a time

In the report code the structure and initialization of the report class, which itself inherits from the StiReport class, are described. When adding new pages, components or changing any parameters of a report, those changes are automatically recorded within the class. The report class therefore contains a description of all components, data, events, report properties, and data source structures for the report. Any events specified by the user are also added to the report code.

For the ultimate in power and flexibility Stimulsoft Reports allows direct editing of the report code - if you want something not provided by the available properties and features of the designer you can actually code your own features within the report. When writing events or another code in the report, you use the standard syntax of the selected .Net Framework programming language i.e. if the language is set to C# you write code using C# syntax.

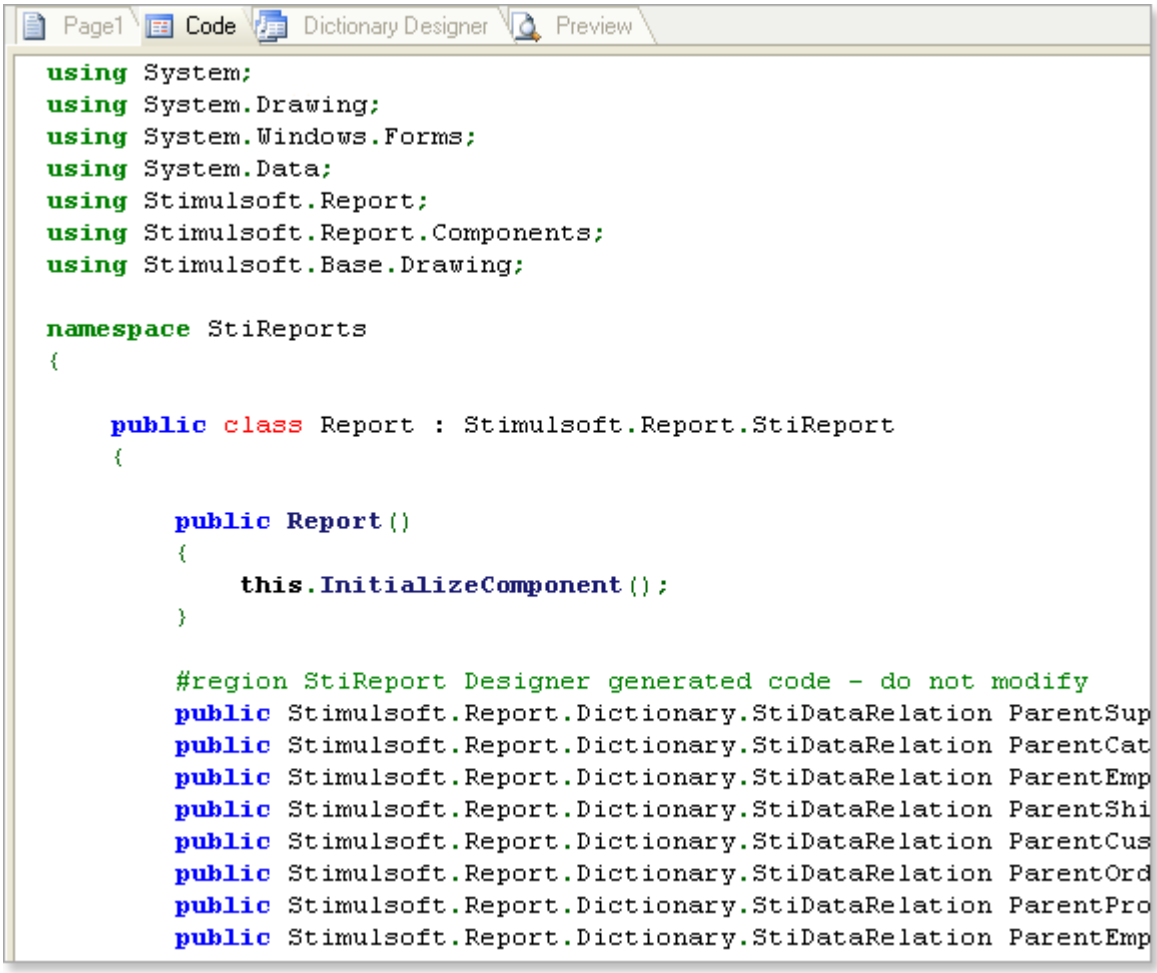
Note: The report code is generated in **C#** or **VB.Net** programming language. All events and any another code in this report must be written in the currently selected language.

When rendering reports, compilation of the report class occurs first. After that the compiled report is executed.

Note: The report code is compiled using the .NET Framework compiler.

Reviewing and Editing the Report Code

To see the report code click the Code tab in the designer. The code will then be displayed:



To edit the code, simply start typing in the appropriate place.

Important: Do not change preprocessor directives or automatically updated code.

Whilst Stimulsoft Reports allows you to directly edit the report code, it is important to remember that it is impossible to make changes in the parts of the report code which are automatically updated - such changes will be lost when the next update takes place. The automatically updated report code is enclosed in Region preprocessor directives:

At the beginning of the automatically updated code

```
#region StiReport Designer generated code - do not modify

Automatically updated code goes here
```

the end of the automatically updated code

```
#endregion StiReport Designer generated code - do not modify
```

Any code that you write within the report must be written outside these Regions.

Totals

In many reports it is necessary to calculate totals: totals by a page, number of rows in a group, average value etc. For all these calculations it is possible to use aggregate functions. Using aggregate functions, it is possible to calculate a sum, number of rows, average values, maximal values, minimal values, to get first values from the list, to get last values from the list etc. The ranges of rows can be all rows of a list, rows on one page, rows from one group, rows from one container etc.

Stimulsoft Reports supports the following aggregate functions:

- **Avg** - returns the average value of the specified expression;
- **Count** - returns the number of rows in the specified range;
- **CountDistinct** - returns the number of unique rows in the specified range;
- **First** - returns the first value in the specified range;
- **Last** - returns the last value in the specified range;
- **Max** - returns the maximal value of the specified expression;
- **Median** - returns the median of all values of the specified range;
- **Min** - returns the minimal value of the specified expression;
- **Mode -** returns the greatest value of the specified range;
- **Sum** - returns the sum of the specified expression.

Calculation of Totals Associated with Bands

Totals Output in Any Part of Report

Usually, components in what text expressions the aggregate function call is specified, must be put on the total band or on the **Data** band. If it is required to output the total on the **Header** band then it is either impossible or possible using the script. But in Stimulsoft Reports the component with the aggregate function can be placed in any kind of a band.

Also you can put an aggregate function on a page and other pages. For example, you can calculate the sum of values in the list and show it in the header of the list. Also you may calculate the number of strings and show its value in the beginning of a page. At that there is a restriction. You should specify to which the Data band does this aggregate function belong. For this you should specify the Data band as a function argument. For example:

- this expression will return the number of strings of the DataBand1 band.

Notice. Components with aggregate functions can be put in any part of a report.

Also it is possible to put an aggregate function on a page and other pages. For example, it is possible to calculate the sum of values by the list and show it in the header of the list. Also it is possible to calculate the number of rows and show its value in the beginning of a page. But there is a restriction. It is necessary to specify to which **Data** band does this aggregate function belong. For this, it is necessary to specify the **Data** band as a function argument. For example:

```
{Sum(DataBand1, Products. UnitsInStock)}
```

- this expression will return the sum of values of the **Products.UnitsInStock** column for every row of the **DataBand1**. Considering the **Count** aggregate function:

```
{Count(DataBand1)}
```

- this expression will return the number of rows of the **DataBand1**.

Type of Result of Total Function

By default, all total functions return the value of the **Decimal** type (except **Count** and **CountDistinct**). In addition Stimulsoft Reports allows making calculations using two types of data: **Double** and **Int64**. If you want the aggregate function to return the result of calculation using the **Double** type of data, write the Latin letter **D** in uppercase before the function name. For calculations using the **Int64** type it is necessary to add Latin letter **I** in uppercase. For example, for calculation sum using the **Int64** type it is necessary to write:

```
SumI
```

For the **Double** type:

```
SumD
```

For the **Decimal** type:

```
Sum
```

It allows preventing losses in totals calculation.

Aggregate function spelling	Type of return value
Sum	Decimal
SumD	Double
SumI	Int64

I, D letters can be added to any functions except for: **Count** and **CountDistinct**. These functions always return the Int64 type .

Notice. The **Count** and **CountDistinct** functions are always calculated using the **Int64** type**.**

Expression with Aggregate Functions

To sum up all values of one column it is enough to write the following text expression of a component:

```
{Sum(DataSource.Column)}
```

Also it is acceptable to use complex expressions:

```
{100 + Sum(DataSource.Column) * 2}
```

For example, it is necessary to output number of rows and the sum of values. For this, it is necessary to add the **Footer** band to the report. Put two **Text** components on this band. Write in the first component the following expression:

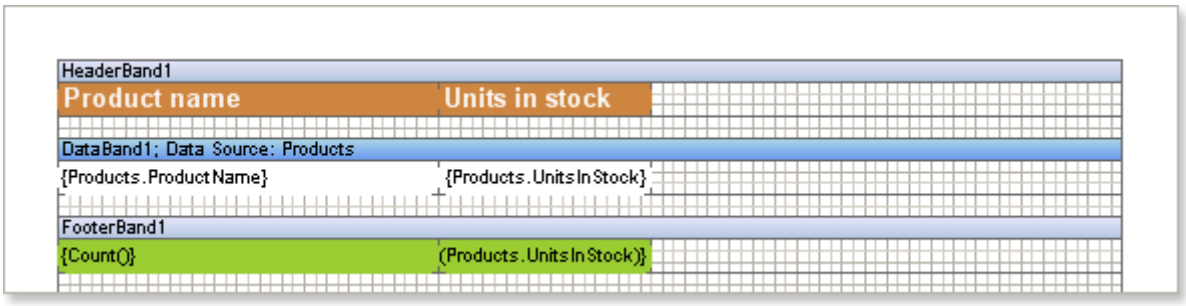
```
{Count()}
```

in this component the number of rows will be output.

Write in the second component the following expression:

```
{Sum(Products.UnitsInStock)}
```

in this component the sum of values of the **UnitsInStock** column will be output.



As one can see from the sample, there is no need in additional arguments for calculation of number of rows of the **Count** function. One argument was specified to the **Sum** function. It is the expression that should be summed up. In other words the report generator specified to which **Data** band all these aggregate functions belong to and how many times these functions must be called.

Product name	Units in stock
Alice Mutton	0
Aniseed Syrup	13
Boston Crab Meat	123
Camembert Pierrot	19
Carnarvon Tigers	42
5	197

This occurred because text components, in which aggregate functions was used, were placed on the total **Footer** band. This band belongs to a **Data** band. This enables the report generator to bind the aggregate functions and the **Data** band automatically.
There are several types of total bands in Stimulsoft Reports. They are as follow:

- **ReportSummaryBand** - this band is used to output totals of the whole report;
- **PageFooterBand** - this band is used to output totals by a page;
- **FooterBand** - this band is used to output totals by a list;
- **GroupFooterBand** - this band is used to output totals by a group.

Placing components and aggregate functions together allows the report generator to indicate to which **Data** band do these aggregate functions belong. Also, in addition, it is possible put the component with aggregate function on the **Data** band. In this case there will be an output of the result of an aggregate function calculation of all strings.

Notice. In most cases Stimulsoft Reports automatically binds the aggregate function and the **Data** band together.

Calculating Totals by Page

It is very simple to calculate totals of a container or a page. For this it is necessary to add before the name of an aggregate function the Latin letter **c** in lowercase. For example:

```
{cCount(DataBand1)}
```

- this expression will return the number of rows of one page.

Notice. A page is a container too. Therefore, calculation of totals of a page goes the same way as calculation of totals of a container.

When calculation of totals of a panel or of a page it is necessary specify the **Data** band by what the aggregate function will be calculated. It is necessary because more than one **Data** band can be placed on one page.

Notice. For calculation of an aggregate function by a panel it is enough to add Latin letter **c** before the name of this aggregate function.

Any number of aggregate functions can be used on one page or a panel. Stimulsoft Reports has no limit in it. It is possible to combine totals of a page with a condition. For example:

```
{CountIf(DataBand1, Products.UnitsInStock = 0)}
```

- the expression will return the number of elements equal zero on this page.

Calculating Totals by Column

Using Stimulsoft Reports it is possible to calculate total by a column. Just write **col** (from **column**) in lowercase before the aggregate function name. For example:

```
{colCount()}
```

- the expression will return the number of rows in one column.

22.Familia Arquibaldo	(11) 555-9857	66.Rattlesnake Canyon Grocery	(505) 555-5939
23.FISSA Fabrica Inter. Salchichas SA.	(91) 555 94 44	67.Reggiani Caseifici	0522-556721
24.Folies gourmandes	20.16.10.16	68.Ricardo Adocicados	(21) 555-3412
25.Folk och fä HB	0695-34 67 21	69.Richter Supermarkt	0897-034214
26.France restauration	40.32.21.21	70.Romero y tomillo	(91) 745 6200
27.Franchi S.p.A.	011-4988260	71.Santé Gourmet	07-98 92 35
28.Frankenversand	089-0877310	72.Save-a-lot Markets	(208) 555-8097
29.Furia Bacalhaue Frutos do Mar	(1) 354-2534	73.Seven Seas Imports	(171) 555-1717
30.Galeria del gastrónomo	(93) 203 4560	74.Simons bistro	31 12 34 56
31.Godos Cocina Típica	(95) 555 82 82	75.Spécialités du monde	(1) 47.55.60.10
32.Gourmet Lanchonetes	(11) 555-9482	76.Split Rail Beer & Ale	(307) 555-4680
33.Great Lakes Food Market	(503) 555-7555	77.Suprêmes délices	(071) 23 67 22 20
34.GROSELLA-Restaurante	(2) 283-2951	78.The Big Cheese	(503) 555-3612
35.Hanari Carnes	(21) 555-0091	79.The Cracker Box	(406) 555-5834
36.HILARION-Abastos	(5) 555-1340	80.Toms Spezialitäten	0251-031259
37.Hungry Coyote Import Store	(503) 555-6874	81.Tortuga Restaurante	(5) 555-2933
38.Hungry Owl All-Night Grocers	2967 542	82.Tradição Hipermercados	(11) 555-2167
39.Island Trading	(198) 555-8888	83.Trail's Head Gourmet Provisioners	(206) 555-8257
40.Königlich Essen	0555-09876	84.Vaffeljernet	86 21 32 43
41.La corne d'abondance	30.59.84.10	85.Victuailles en stock	78.32.54.86
42.Lamai sond'Asie	61.77.61.10	86.Vins et alcools Chevalier	26.47.15.10
43.Laughing Bacchus Wine Cellars	(604) 555-3392	87.Wartian Herkku	981-443655
44.Lazy K Kountry Store	(509) 555-7969	88.Wellington Importadora	(14) 555-8122
	44		44

Company	Phone
89.White Clover Markets	(206) 555-4112
90.Wilman Kala	90-224 8858
91.Wolski Zajazd	(26) 642-7012
	3

There is one restriction in calculation of totals by a column in Stimulsoft Reports. Totals can only be calculated by columns on a page. It is impossible to calculate totals by columns in the **Data** band.

Important! Totals can only be calculated by columns on a page. It is impossible to calculate totals by columns in the **Data** band.

When calculating totals by a column, the text component with aggregate function should be placed on **ColumnHeader**, **ColumnFooter**, **Header** of **Footer** bands.

Important. For calculation aggregate functions by column it is enough to add **col** before the name of an aggregate function.

It is possible to calculate unlimited number of totals by column. There are no limitations in this case. Also it is possible to combine totals by column with condition. For example:

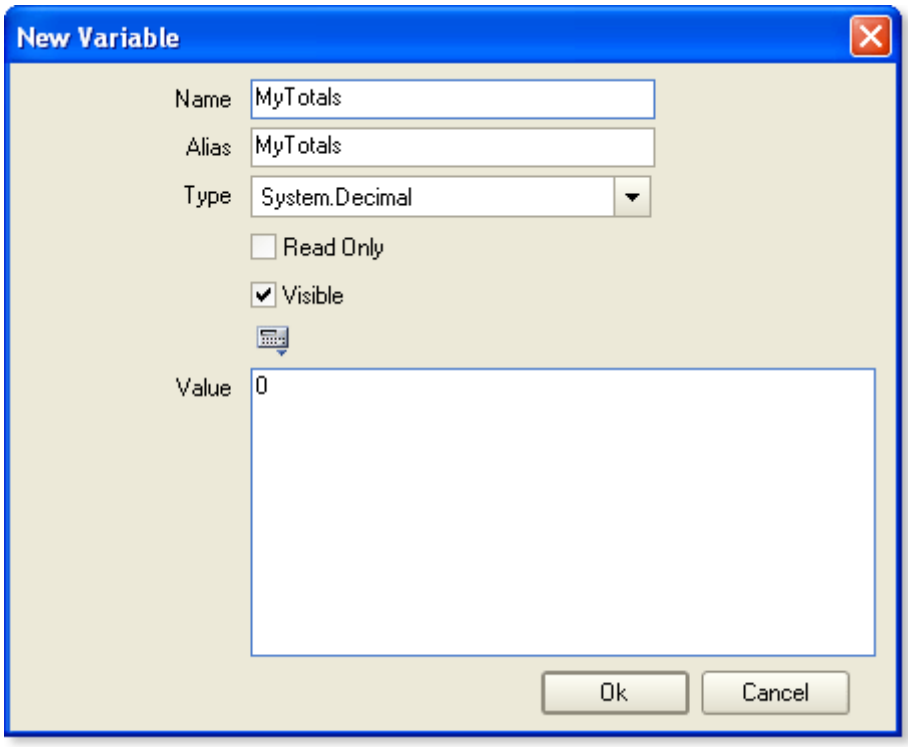
```
{colCountIf(DataBand1, Products.UnitsInStock = 0)}
```

- this expression will return the number of rows, which condition is set to **true**, by a column.

Calculating Totals in Code of Report Events

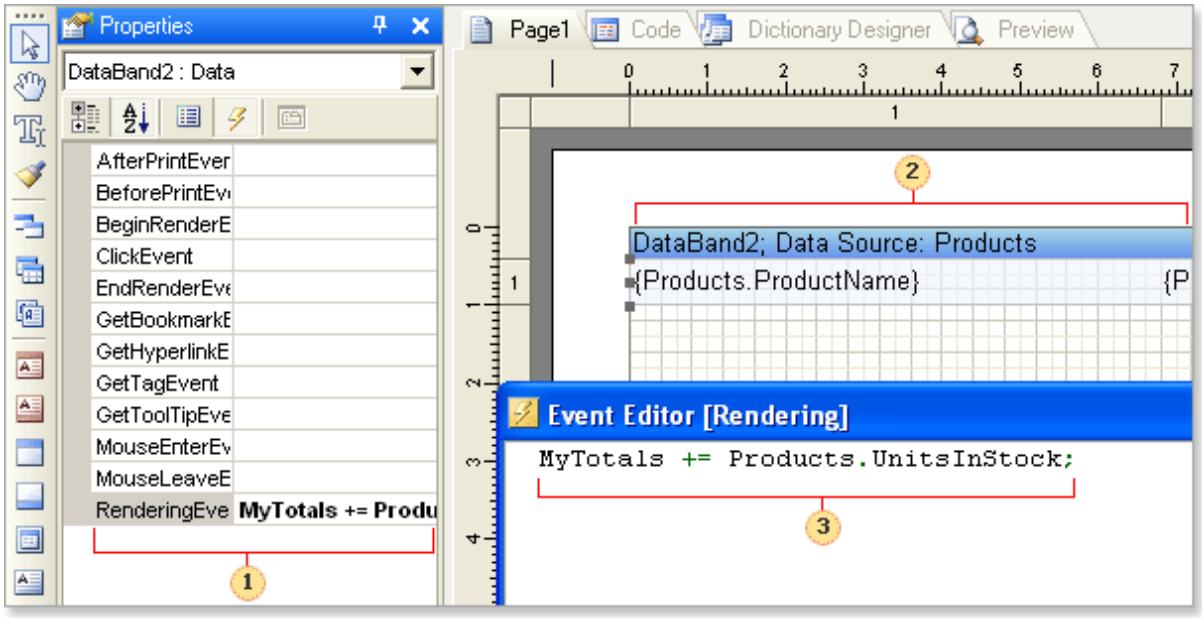
Using Stimulsoft Reports you may make calculation of aggregate functions in the code of events of a report. This allows calculating aggregate functions with complex logic or condition. Another advantage of Stimulsoft Reports is that, when calculating, you call the value to be calculated from the report code and make changes. If you want to make such a calculation, the variable which stores the value of an aggregate function is required. For this, you should create a new variable in the data dictionary.

Important! Using variables from the code to store the result of an aggregate function calculation is not allowed. It is necessary to use variables created in the dictionary.



Here you specify the type of a variable, for example **Decimal**, and the initial value is zero. Then in the **Render** event of the **Data** band you should set the code for a variable increment. For example, if you want to calculate the sum of variables of the **Products.ItemsInStock** field then the code will be as follows:

```
MyTotals += Products.ItemsInStock;
```



1. The **RenderingEvent** event
2. The **Data** band in what the **RenderingEvent** event calculation will be made
3. The code for calculating the sum

For calling the variable, that contains the value of a total, write in the following in the text expression:

```
{MyTotal}
```

If, after you have written the text expression, you run the report rendering. Then, when the report rendering will reach the component which contains the expression with the total variable, then there will be an output of this variable. Therefore, you should specify to the report generator that the component output must be done right after the whole report rendering - when the variable will be calculated completely. For this, you must set the **ProcessAtEnd** property of the text component to **true**.

Notice. Expressions of text components which the **ProcessAtEnd** properties are set to **true** are always calculated at the end of the report rendering.

As was written before the total will be calculated and shown in the proper part of a report.

Invisible Bands

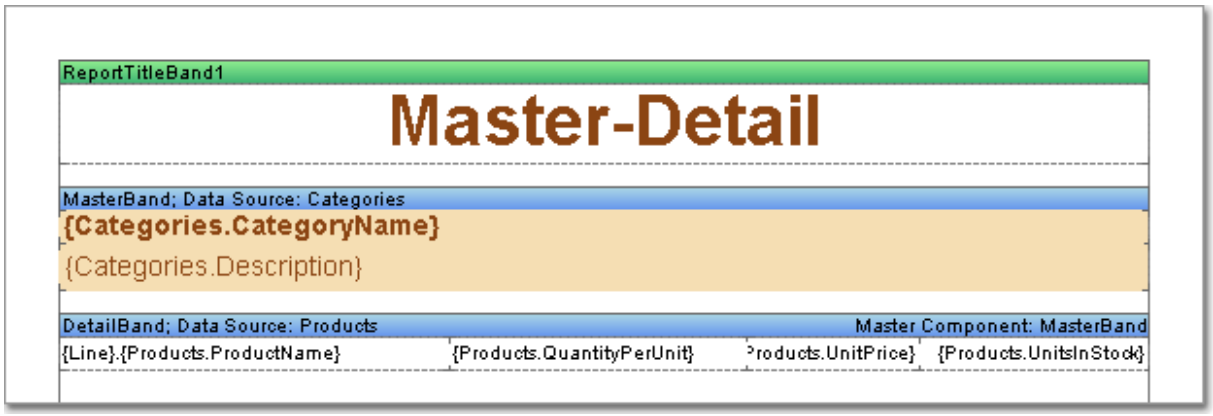
Many reports use invisible bands by a certain condition. By default, the report generator will not consider disabled **Data** bands. But it is necessary, when calculating totals, also to consider invisible **Data** bands then it is necessary to set the **CalcInvisible** property of the **Data** band to **true**. In this case only invisible **Data** bands will be output and, when calculating aggregate functions, all rows will be calculated.

Totals and Automatic Change of Size

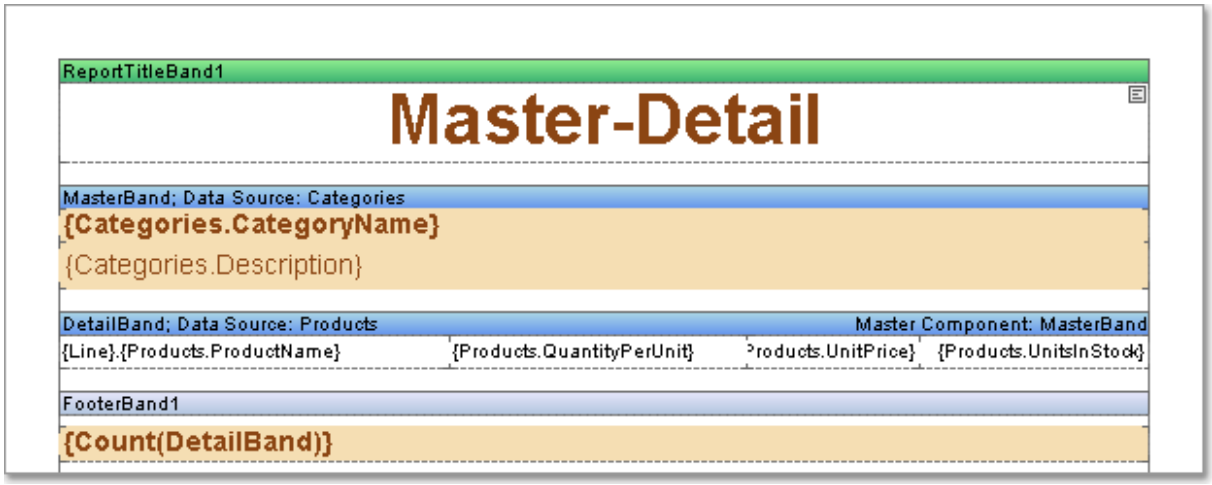
There is one feature when using automatic change of a size of a component that is responsible for the calculation of totals output. As a rule in the moment when the component size indication is in process, the result of the aggregate function is unknown. Therefore, the component cannot correct its size considering the result of an aggregate function. This feature should be considered when reports rendering.

Calculating Totals in Master-Detail Reports

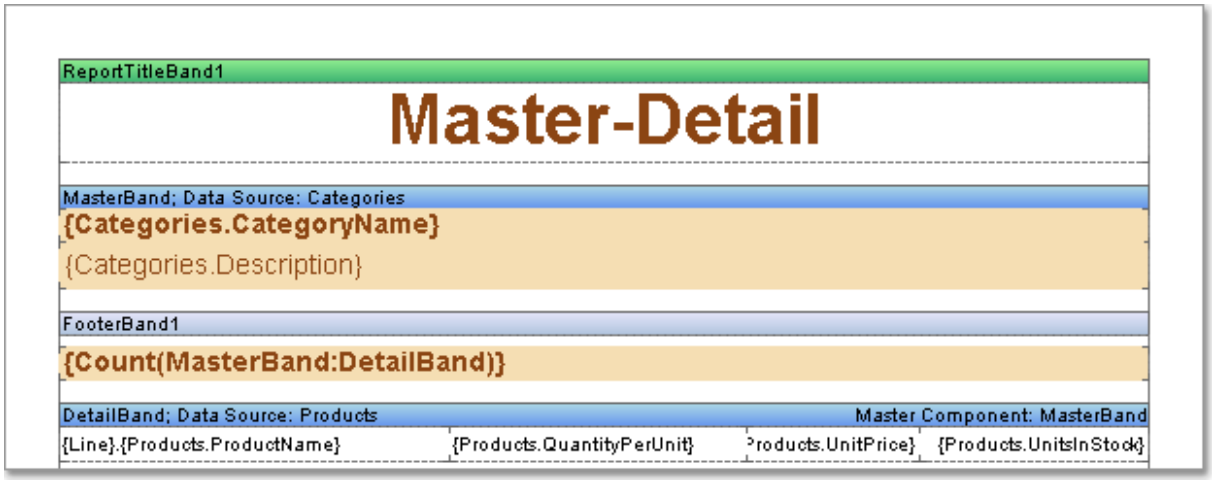
When building the **Master-Detail** reports you may meet the problem with calculation of totals in hierarchical reports. What is it? Suppose, you have a list of products which is output by categories. The report is built using **Master-Detail** relations. In other words we have a certain number of master strings (categories) and a certain number of detail strings (products).



So we need to count how many products are output in the report. If we add the **FooterBand** with the aggregate function to the **Detail** band, then we will get the total by each group.



If we add the **FooterBand** to the **MasterBand** then we will get the number of categories. In this case it is possible to use the calculation of totals. For this purpose you need to specify names of both **DataBands** in the aggregate function. In our case: **Count(MasterBand: DetailBand)**.



Seafood			
Seaweed and fish			
1.Ikura	12 - 200 ml jars	31,00p.	31,00
2.Konbu	2 kg box	6,00p.	24,00
3.Carnarvon Tigers	16 kg pkg.	62,50p.	42,00
4.Nord-Ost Matjeshering	10 - 200 g glasses	25,89p.	10,00
5.Inlagd Sill	24 - 250 g jars	19,00p.	112,00
6.Gravad lax	12 - 500 g pkgs.	26,00p.	11,00
7.Boston Crab Meat	24 - 4 oz tins	18,40p.	123,00
8.Rogede sild	1k pkg.	9,50p.	5,00
9.Spegesild	4 - 450 g glasses	12,00p.	95,00
10.Escargots de Bourgogne	24 pieces	13,25p.	62,00
11.Röd Kaviar	24 - 150 g jars	15,00p.	101,00
73			

The result of the **Count(MasterBand:DetailBand)** function is the number of products by all categories.

Totals with Condition

Often it is necessary to calculate totals with condition. For example, it is necessary to sum up all values greater than zero. Stimulsoft Reports allows adding a condition to an aggregate function. When a condition is added to an aggregate function one should add **If** to this aggregate function name and additional argument with condition. For example:

```
{SumIf(Products.UnitsInStock, Products.UnitsInStock > )}
```

- this expression will return the sum of elements, and each element should be greater than zero. For the **Count** function:

```
{CountIf(Products.UnitsInStock == 0)}
```

- this expression will return the number of elements equal zero. If it is necessary to make calculation using the **Double** type or the **Int64** type, then add the Latin letter **D** or **I**, and them the **If**. For example:

```
{SumDIf(Products.UnitsInStock, Products.UnitsInStock > 0)}
```

Syntax of Aggregated Functions

See the aggregate functions syntax by the example of the **Sum** function:

Sum(expression)
Sum(band, expression)

SumIf(band, expression, condition)

expression – an expression for calculation;
band – a name of a band for calculation;
condition – a condition of inclusion of an expression into the calculation.

When an aggregate function by a page or container is calculated, you should write a letter "**c**" first and then the aggregate function name. See the sample:

cSum(expression)
cSum(band, expression)
cSumIf(band, expression, condition)

For calculation of totals of a container you should write **col** first and then the aggregate function name:

colSum(expression)
colSum(band, expression)
colSumIf(band, expression, condition)

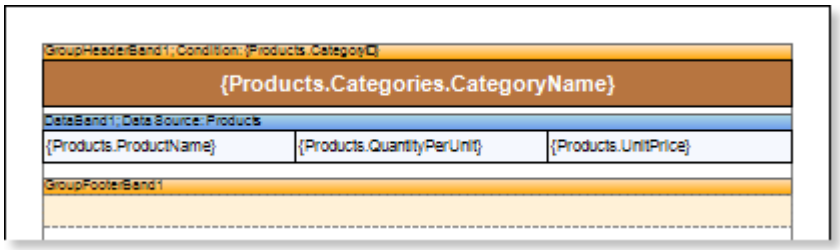
The **Count** function has a distinguishing feature from other aggregate functions. It does not have an expression for calculation. Syntax of this function see below:

Count()
CountIf (condition)
Count (band)
CountIf(band, condition)
cCount ()
cCount (band)
cCountIf(band, condition)
colCount ()
colCount (band)
colCountIf(band, condition)

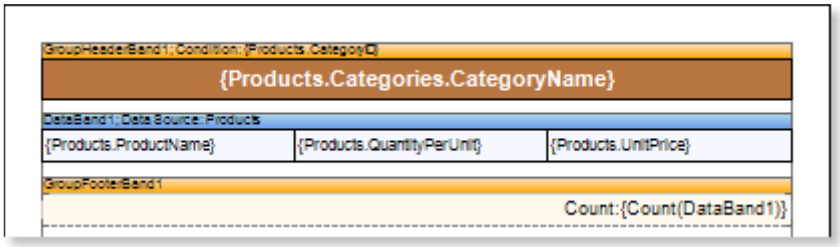
Important! When using C# programming language, all aggregate functions should be written considering the case.

Calculation of Totals not Associated with Bands

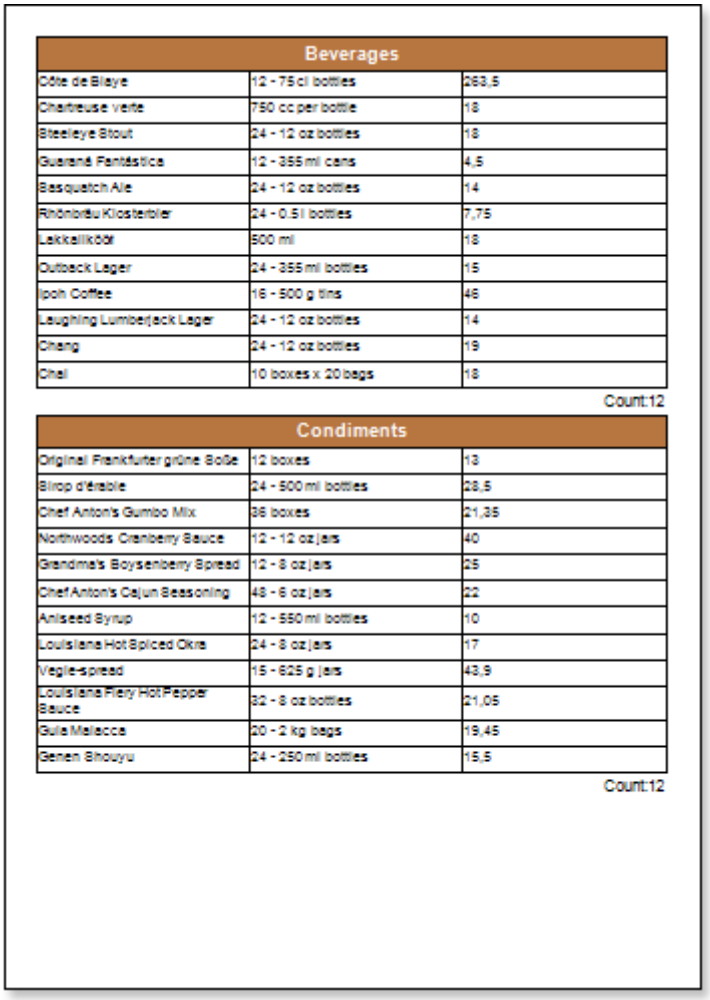
The calculation totals in a report can be done by specifying an expression, for example, **{Sum (DataBand1)}**. These totals are calculated when rendering the report: each time when a single operation with the **DataBand** is done, a single value is calculated. All calculated values makes a total value, which will be displayed. In Stimulsoft Reports calculation of totals can be produced in another way - instantly. You should use the special **Totals** prefix. Calculation of **totals** occurs where the function is called, in contrast to standard results, calculations are performed while report rendering. Consider the calculation of totals in a report by the example of a report with grouping. Suppose there is a report with grouping. The picture below shows a report template with the group:



In this example, we calculate the totals using the **Count** function. This function calculates the number of rows. Put the text component in the **GroupFooterBand** band with the following expression: **{Count (DataBand1)}**. The picture below shows a report template with the grouping and the **Count** function in the **GroupFooterBand**:



When rendering a report, the report generator renders a report with groups, and then calculates total values by groups and displays them. The picture below shows a report page displaying the totals by groups:



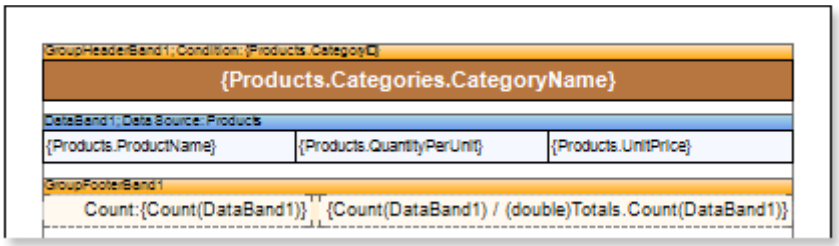
Beverages		
Côte de Blaye	12 - 75 cl bottles	263,5
Chartreuse verte	750 cc per bottle	18
Steeleye Stout	24 - 12 oz bottles	18
Guaraná Fantástica	12 - 355 ml cans	4,5
Sasquatch Ale	24 - 12 oz bottles	14
Rhönbräu Klosterbier	24 - 0.5 l bottles	7,75
Lakkalikööri	500 ml	18
Outback Lager	24 - 355 ml bottles	15
Ipoh Coffee	16 - 500 g tins	46
Laughing Lumberjack Lager	24 - 12 oz bottles	14
Chang	24 - 12 oz bottles	19
Chai	10 boxes x 20 bags	18

Count:12

Condiments		
Original Frankfurter grüne Soße	12 boxes	13
Sirop d'érable	24 - 500 ml bottles	28,5
Chef Anton's Gumbo Mix	36 boxes	21,35
Northwoods Cranberry Sauce	12 - 12 oz jars	40
Grandma's Boysenberry Spread	12 - 8 oz jars	25
Chef Anton's Cajun Seasoning	48 - 6 oz jars	22
Aniseed Syrup	12 - 550 ml bottles	10
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Vegie-spread	15 - 625 g jars	43,9
Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21,05
Gula Malacca	20 - 2 kg bags	19,45
Genen Shouyu	24 - 250 ml bottles	15,5

Count:12

Go back to the report template. Calculate the ratio of entries in the group to entries in the report and show the total in per cent. To do this, add a text component in the **GroupFooterBand** with the following **{Count (DataBand1) / (double) Totals.Count (DataBand1)}** expression, where the **Count (DataBand1)** function will count the number of rows in the group, and the **Totals.Count (DataBand1)** function will calculate the number of rows in the report. To show the total value in per cent, you should to set the **Text Format** property of the text component to **Percentage**. The picture below shows a report template with the added text component in the **GroupFooterBand**:



Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. Also the calculation of totals will be done. The picture below shows a rendered report with ratio (in per cent) of entries in the group to entries in the report:

Beverages		
Côte de Blaye	12 - 75 cl bottles	263,5
Chartreuse verte	750 cc per bottle	18
Steeleye Stout	24 - 12 oz bottles	18
Guaraná Fantástica	12 - 355 ml cans	4,5
Sasquatch Ale	24 - 12 oz bottles	14
Rhönbräu Klosterbier	24 - 0.5 l bottles	7,75
Lakkalikööri	500 ml	18
Outback Lager	24 - 355 ml bottles	15
Ipoh Coffee	16 - 500 g tins	46
Laughing Lumberjack Lager	24 - 12 oz bottles	14
Chang	24 - 12 oz bottles	19
Chai	10 boxes x 20 bags	18
Count:12		15,58%

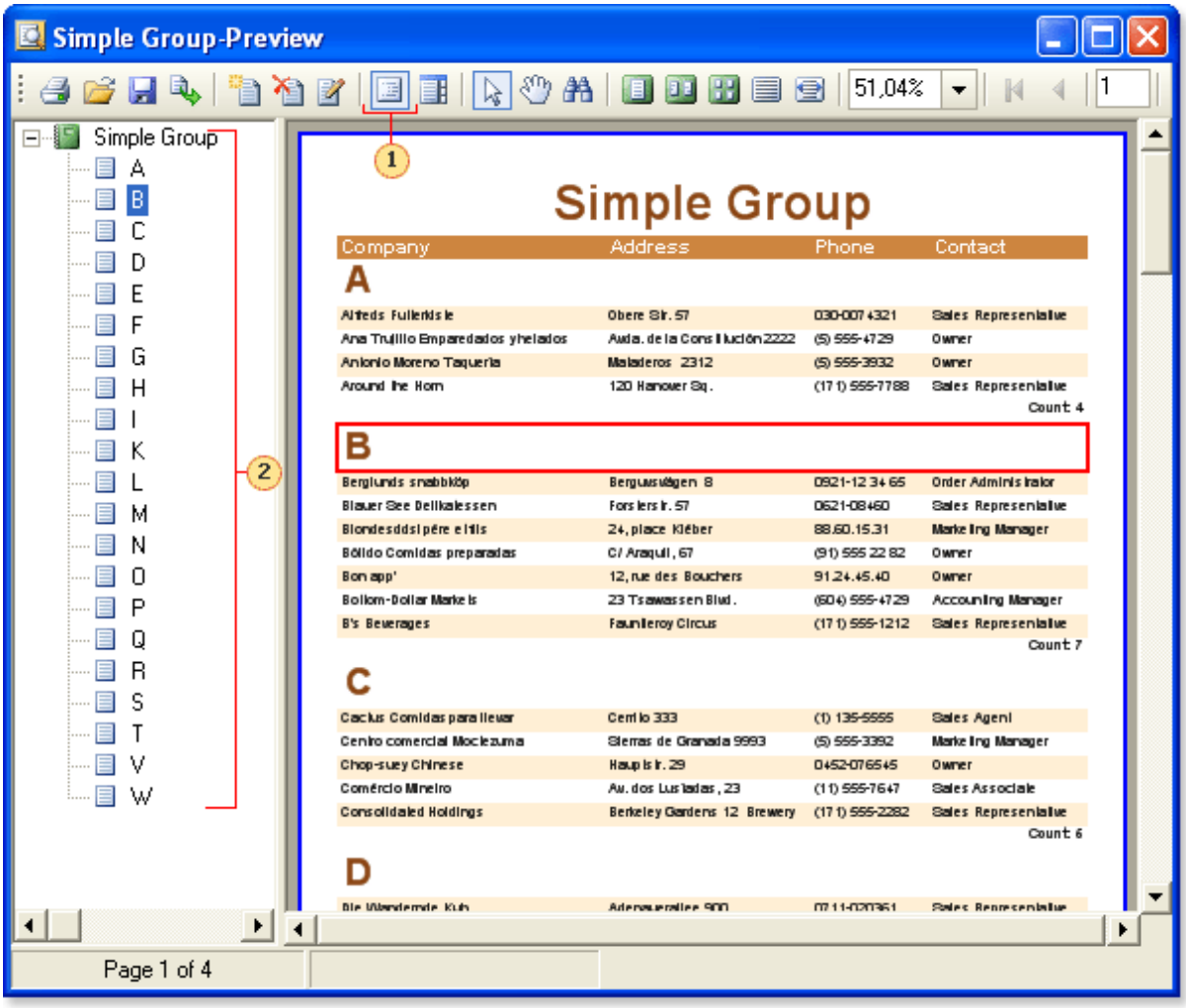
Condiments		
Original Frankfurter grüne Soße	12 boxes	13
Sirop d'érable	24 - 500 ml bottles	28,5
Chef Anton's Gumbo Mix	36 boxes	21,35
Northwoods Cranberry Sauce	12 - 12 oz jars	40
Grandma's Boysenberry Spread	12 - 8 oz jars	25
Chef Anton's Cajun Seasoning	48 - 6 oz jars	22
Aniseed Syrup	12 - 550 ml bottles	10
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Vegie-spread	15 - 625 g jars	43,9
Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21,05
Gula Malacca	20 - 2 kg bags	19,45
Genen Shouyu	24 - 250 ml bottles	15,5
Count:12		15,58%

Interaction

Stimulsoft Reports has a set of features to render interactive reports. They are bookmarks, hyperlinks, Drill-Down links, dynamic sorting, dynamic collapsing, editing reports in the window of preview. All these features are described in chapters below.

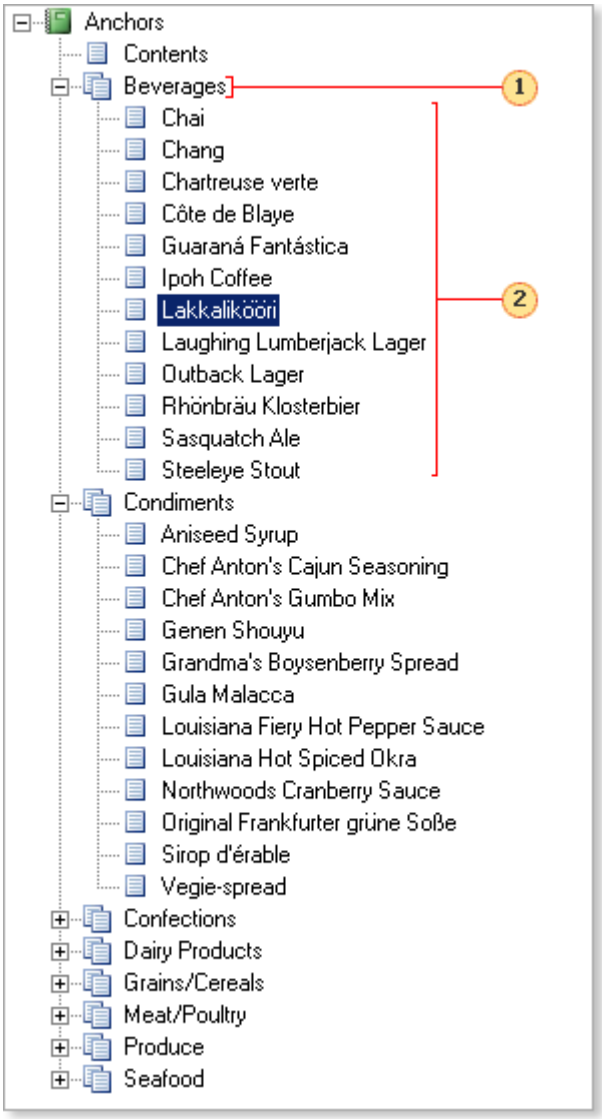
Bookmarks

Bookmarks are used to show the structure of a report. Also bookmarks are used to mark the component to make a reference on it using hyperlinks. All components have the **Interaction.Bookmark** property. The expression, specified in this property, is set in the **BookmarkValue** property. Setting occurs when the report rendering. This property is invisible in the **Properties** panel but it can be called from the report code or refer to it from the expression. Before showing a report in the window of preview, Stimulsoft Reports views all components of a rendered report and logs a tree of bookmarks.



Tree of Bookmarks

The tree of allows viewing the hierarchical structure of a report. For example, two bookmarks were specified: one on the **Master** band and the second on the **Detail** band. In this case, each element of the **Master** band bookmark fits to a node of the bookmarks tree. All elements of bookmarks from the **Detail** bands will be added to the proper node of the **Master** band.



Bookmarking Using Code

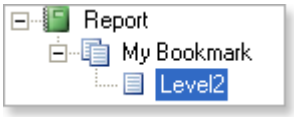
Using the **Interaction.Bookmark** property very complicated structure of bookmarks in a report can be formed. But sometimes it is not enough of this property. For example, it is necessary to add nodes to the tree of bookmarks without using the **Interaction.Bookmark** property. Or the bookmark should be placed on another level of nesting. The **Interaction.Bookmark** property of Stimulsoft Reports can be used. This is an invisible property and it is available only from the code. It is very simple to use this property. For example, to add the bookmark of the first level of nesting the following code can be used:

```
Bookmark.Add("My Bookmark");
```

This code will create this bookmark in the tree of bookmarks:

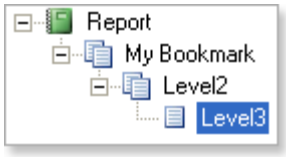
To add a bookmark of the second level to the tree it is necessary write the following code:

```
Bookmark["My Bookmark"].Add("Bookmark Level2");
```



...and for the third level:

```
Bookmark["My Bookmark"]["Level2"].Add("Bookmark Level3");
```



To create all three bookmarks the code sample shown above can be used. Stimulsoft Reports automatically checks presence of each bookmark in a tree and will add ones which should be added. Sometimes it is required to organize navigation using bookmarks. If it is necessary to find components the **Interaction.Bookmark** property of these components should be logged. The value of the **Interaction.Bookmark** property should be the same with the name of the created bookmark. For example, add the bookmark:

```
Bookmark.Add(Customers.CompanyName);
```

So the values of the **Interaction.Bookmark** property should be as follow:

```
{Customers.CompanyName}
```

As a result all components will be marked with bookmark with the company name. The same company name will be added to the report tree. And, when clicking on the bookmark node of the report tree, all components will be found.

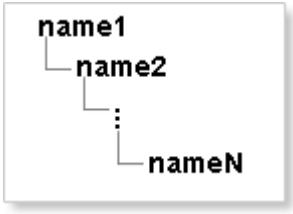
Creating Bookmarks Using Expression

Using the expression it is possible to form rather complex structure of bookmarks in a report. Even a flat report (containing no subordinate entries) can be represented as a hierarchy of bookmarks. General view of the expression with which one can submit any report as a hierarchy of bookmarks is as follows:

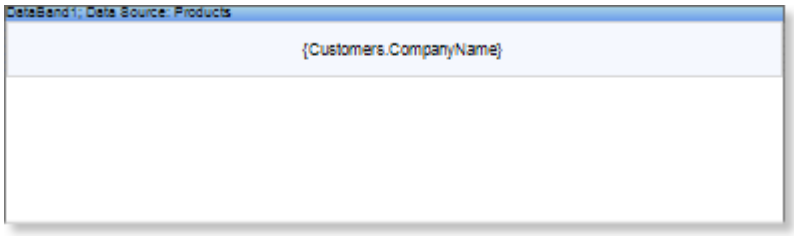
```
%\name1\name2...\nameN
```

where **name1** is a name of a highest level bookmark;
nameN is a name of the lowest level bookmark.

The picture below shows the expression hierarchy of a common type:



In the name of the bookmark the following things can be specified: function, expression, data source column, system variables, random names, aliases and more. To make a flat report with the hierarchy of bookmarks, create a single **Data** band, place the band on a text component with the **Company Name** data source column. The picture below shows an example of a report template:

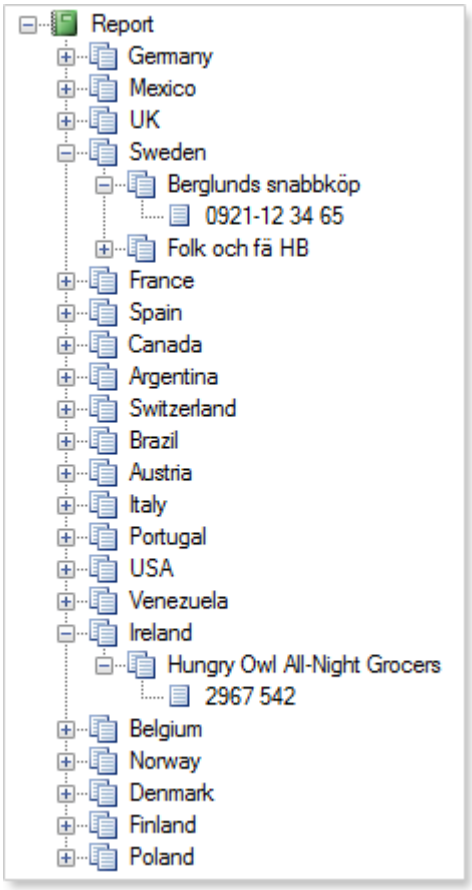


When rendering the report a list of companies will be built, but the tree of bookmarks will not be shown. To show the hierarchy of bookmarks it is necessary to specify an expression (see below an axample):

```
%\{Customers.Country}\{Customers.CompanyName}\{Customers.Phone}
```

As seen from the expression the hierarchy of bookmarks will be represented in three levels:
The highest level will be represented as bookmarks which correspond to the name of the country.
The middle level will be represented as bookmarks which correspond to the name of the company.
The lowest level will be represented as bookmarks which correspond to the phone number of the company.

The picture below shows an example hierarchy of tabs:



Bookmark Nesting

Nesting depends on which components generated bookmarks. For example, the page bookmark will always be one level higher then other bookmarks. The bookmark, created with the **GroupHeaderBand** component, is one level higher then the bookmark, created by the **DataBand** component, in this group. In the Master-Detail relation the Master bookmark will enable all Detail bookmarks. For example, we have a report with a group.

```
Group
--Data 1
--Data 2
--Data 3
Group
--Data 1
--Data 2
--Data 3
```

In this report groups include data. And bookmarks from the group will include bookmarks from data. As a result we get the same structure in the tree of bookmark. For example:

```
Group 1
--Group 2
----Data 1
----Data 2
----Data 3
```

Group 1
--Group 2
----Data 1
----Data 2
----Data 3

In the tree of bookmarks two nodes will be created. They are **Group 1**, **Group 1**. Each of these nodes will include the **Group 2** node. The **Group 2** nodes will include the **data** nodes. For example, the Master-Detail report:

Master-Data
--Data 1
--Data 2
--Data 3
Master-Data
--Data 1
--Data 2
--Data 3

In this example the nodes of the Master band form the Master-Data nodes. Each of these nodes will include nodes formed with the Detail band.

Hyperlinks

Hyperlinks are used in report navigation. Also it is possible to use the **Interaction.Bookmark** and **Interaction.Tag** properties for this. Hyperlink is set in the **Interaction.Hyperlink** property. When report rendering, the expression, specified in this property, is set in the **Interaction.HyperlinkValue** property. Setting occurs when report rendering. There are three ways of specifying hyperlinks. It is possible to use one of them.

Hyperlink to Another Component in Report Using Interaction.Bookmark

In this way you should put the # symbol before the hyperlink text. This makes the report generator to understand that this is a reference inside of a document. If, in the window of preview, a user clicks on this component then the report generator will start to search all bookmarks of this report. If the bookmark name concurs with the hyperlink name (the # symbol is skipped) then this component will be displayed in the window of preview. It is important to remember that a bookmark is shown in the tree of bookmarks.

Notice. The **Interaction.Bookmark** property contains the text marker by what this component will be found, when hyperlink processing.

Hyperlink to Another Component in Report Using Interaction.Tag

In this case it is necessary to add two # symbols before a hyperlink. In this case the search is executed using the **Interaction.Tag** property of components (two # symbols in the text of a hyperlink are skipped). **Interaction.Tag** properties are not shown in the structure of a report. If one want to make navigation without bookmarks showing in the structure of a report then one should use this way.

Notice. When using the **Interaction.Tag** property, one should not use the hyperlink to another component in a report in ASP.NET. In ASP.NET, when creating a report, it is impossible to use hyperlink to another component in a report, created using the **Interaction.Tag** property.

Hyperlink to External Documents

In this way any symbols to a hyperlink should not be added. The string of a hyperlink is directly sent to the OS for processing. For example, for Notepad start just write the following code:

```
notepad.exe
```

For jumping to the address in the Internet:

```
http://www.site.com
```

For email hyperlink:

```
mailto: mail@domain.com
```

Notice. When Web reports rendering, bookmarks can be put only on visible fields. For example, on a text, on an image. Otherwise this hyperlink will be ignored. This principle is to be considered when exporting reports to other formats

Reports with Contents

Often it is necessary to create a report with contents. In this case you should create the report structure first and then create the report on the whole. But there is a question. How to output page numbers, because at the moment, when contents rendering, numbers of pages, which elements of contents refer to, are unknown. Use the anchor in this case. The **AddAnchor** method is used for creating an anchor. When creating an anchor, the report generator saves the current page and compares it with the specified anchor. For example:

```
AddAnchor("MyAnchorName")
```

- in this line of the code a new anchor with "**MyAnchorName**" will be created. To get the anchor value it is necessary to use the **GetAnchorPageNumber** method. This method returns the number of a page according to the anchor name. If there is no the anchor with such a name the 0 is returned.

For example:

```
{GetAnchorPageNumber("MyAnchorName")}
```

- this text expression will return the number of a page according to "**MyAnchorName**". So having an anchor name you will know the number of a page on what this anchor was created. Using these two methods a contents building is organized. The contents is built first. Instead of numbers of pages hyperlinks to anchors are pasted. For all components which call a function for getting a page number via anchor you should set the **ProcessAtEnd** property to **true**. It is necessary to do because these components are to be processed in the end of report rendering when all numbers of pages are known.
After the contents has been created the whole report rendering is in process. Anchors are created while report building. After the report has been rendered, instead of hyperlinks, the real page numbers are put on anchors in the

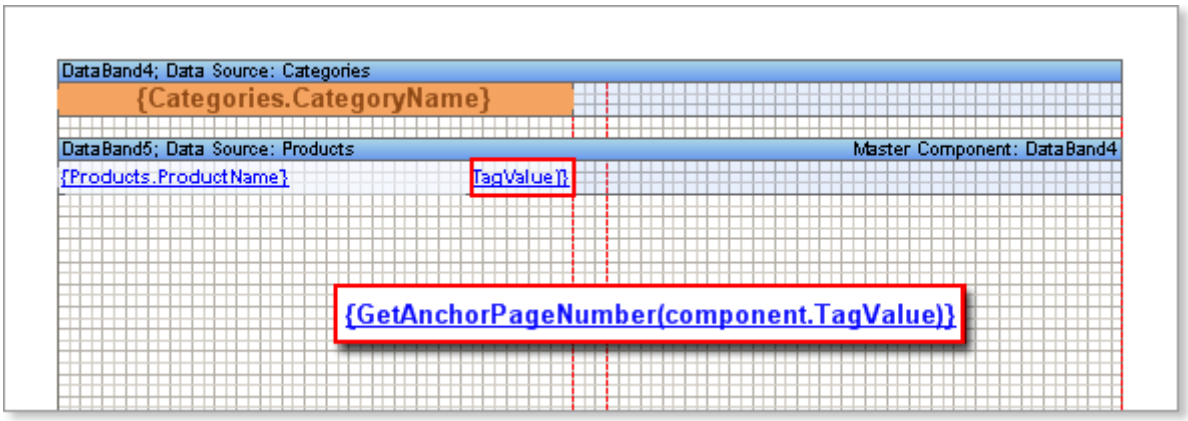
content. Let see the anchor usage in a template. Create the **Master-Detail-Detail** report that shows the list of products that is split with categories. For building of such a report you should have two pages. The first page for the contents and the second for the report. On the page of the contents we put two bands. Between them we set the **Master-Detail** link. Then, on the **Detail** band, we put the text component. This **ProcessAtEnd** text components property should be set to **true**.

Notice. You should enable the **ProcessAtEnd** property of the text component, which expression returns the number of a page. This property is used for the values of these text components to be processed after report rendering (when numbers of pages are known).

Specify the following text expression of the **Text** property:

```
{GetAnchorPageNumber(component.TagValue)}
```

- this text expression will return the number of a page using the anchor.



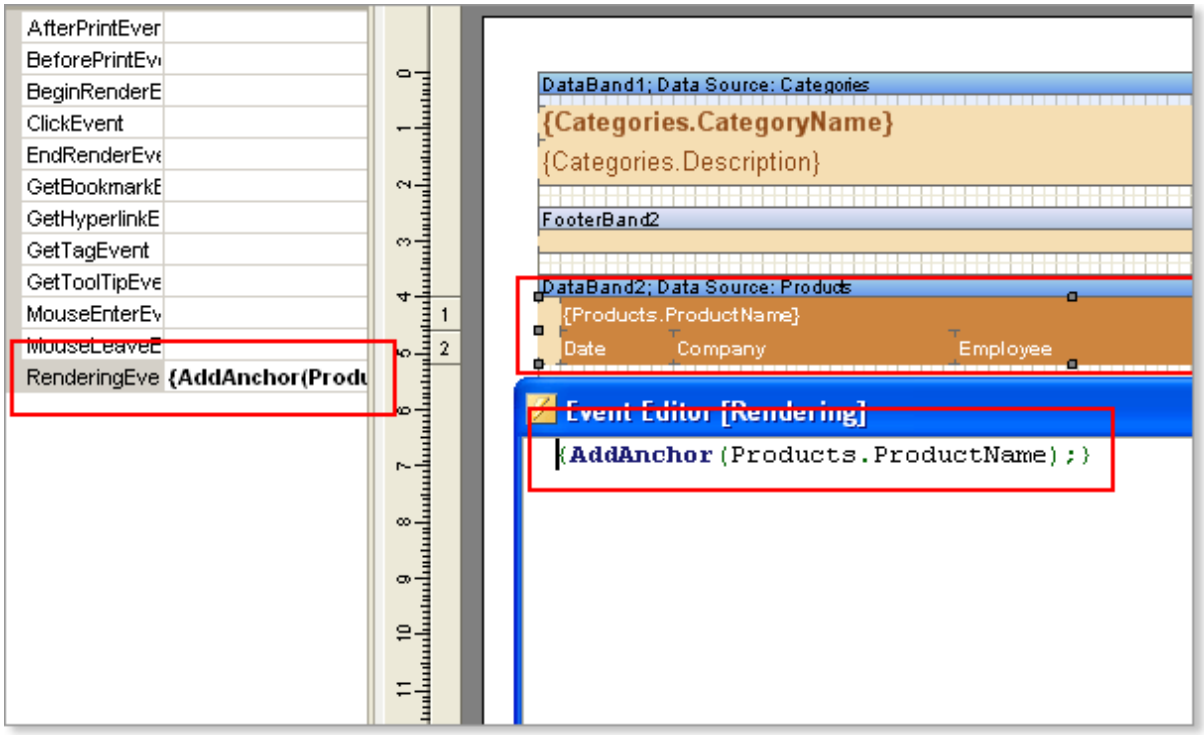
As an anchor name the value of the **Tag** property is used. For filling the **Tag** property the following expression is used:

```
{Products.ProductName}
```

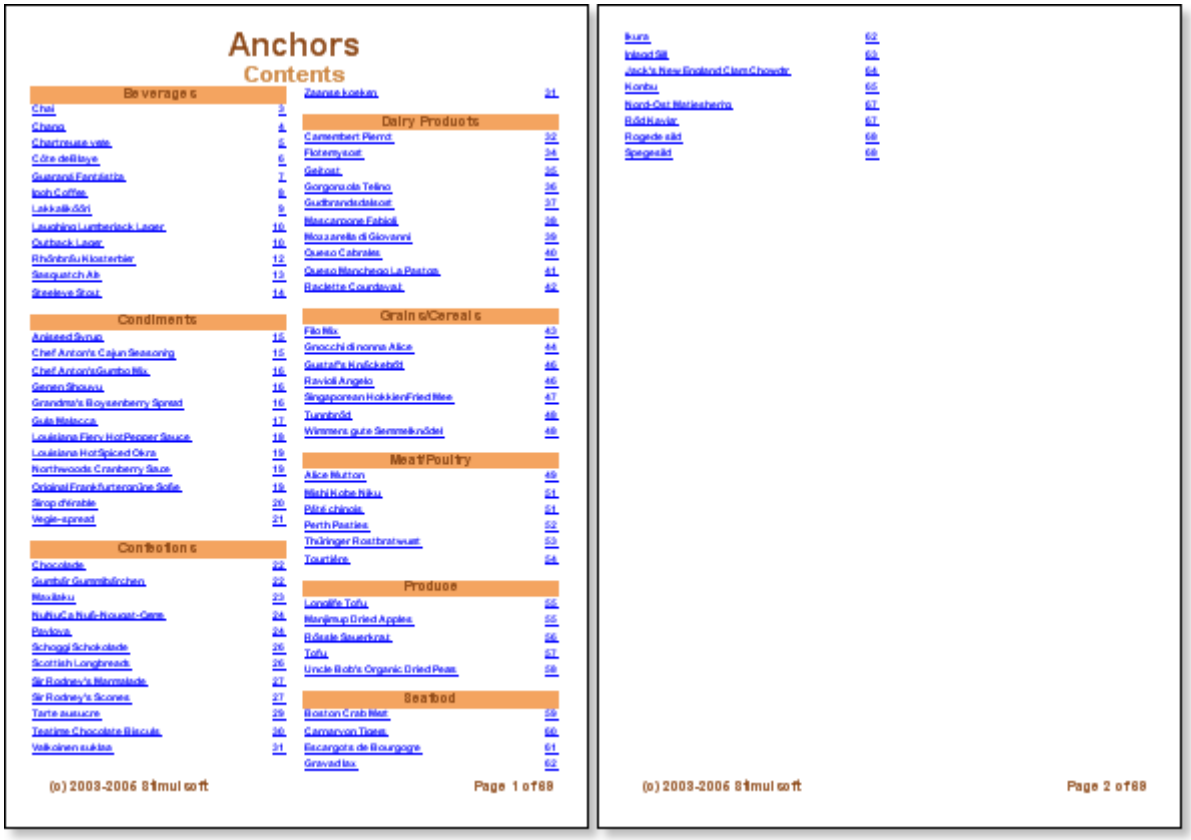
- in this expression the name of a product is used. Therefore, it is impossible to use the expression below:

```
{GetAnchorPageNumber(Products.ProductName)}
```

The component that contains an expression will be processed in the end of report building. So the value of the **Products.ProductName** field will be equal for all strings – the last in a list. That is why it is necessary to remember the value of the **Products.ProductName** field for every string when the content is being built. For this use the **Tag** property. On the second page the report is built. In the **Rendering** property of the **DataBand** component (used for the content building) the **AddAnchor** method is called. This method will return the current page in the moment of its calling.



The anchor name is the value of the **Products.ProductName** field. As a result, the page number is rendered first. Then the second page is rendered and numbers of pages are saved. After the report rendering the report generator engine returns to the first page and numbers all pages.



Getting Started

This section discusses basic steps for creating various reports and showing them in a viewer. These examples demonstrate basic functionality of the reporting tool, and provide you with step-by-step instructions on how to create reports. We suggest that you review the following tutorials first.

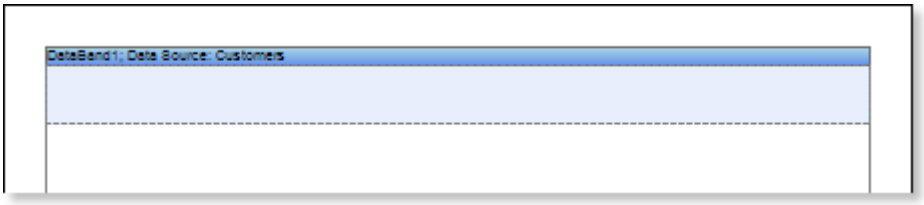
Simple List Report

For better understanding this step-by-step instruction, please watch the video file.

Do the following steps to create a simple list report:

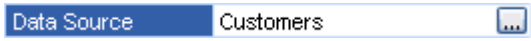
1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

3. Put a **DataBand** on a page of a report template.



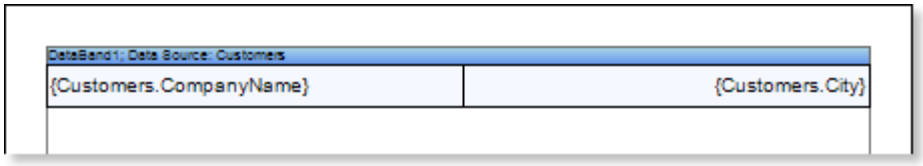
4. Edit **DataBand**:
   4.1. Align the **DataBand** by height;
   4.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
   4.3. Change the **DataBand** background;
   4.4. Enable **Borders** for the **DataBand**, if required;
   4.5. Change the border color.

5. Define the data source for the **DataBand** using the **Data Source** property:

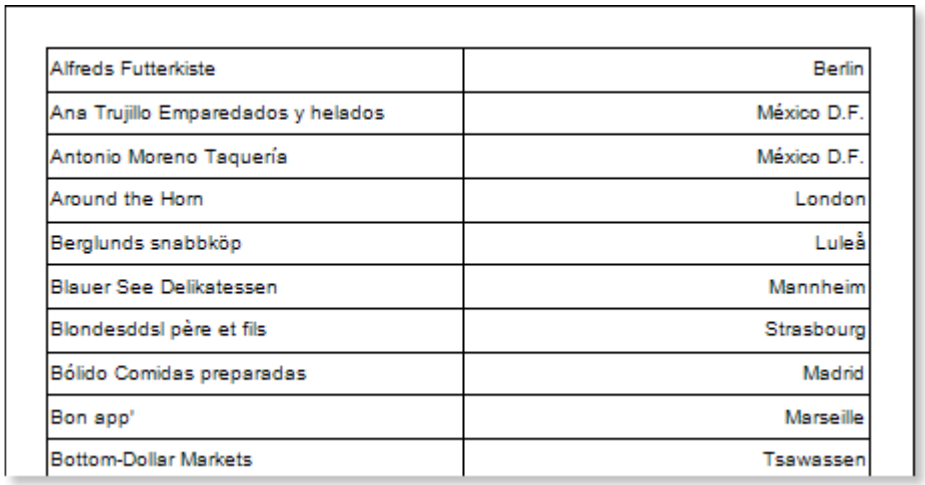


6. Put text components with expressions in the **DataBand**. Where expression is a reference to the data field. For example, put two text components with expressions: **{Customers.CompanyName}** and **{Customers.City}**;
7. Edit **Text** and **TextBox** component:
   7.1. Drag and drop the text component in the **DataBand**;
   7.2. Change parameters of the text font: size, type, color;
   7.3. Align the text component by width and height;
   7.4. Change the background of the text component;
   7.5. Align text in the text component;
   7.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   7.7. Enable **Borders** for the text component, if required.
   7.8. Change the border color.

The picture below shows a report template with the list:



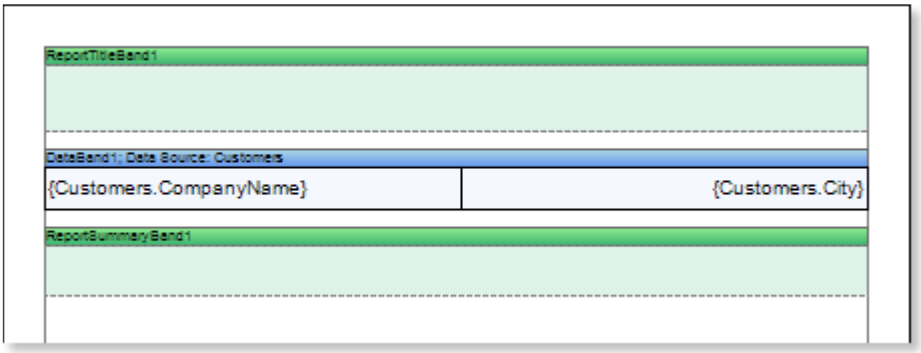
8. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of a simple list report:



Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondesddsl père et fils	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Markets	Tsawassen

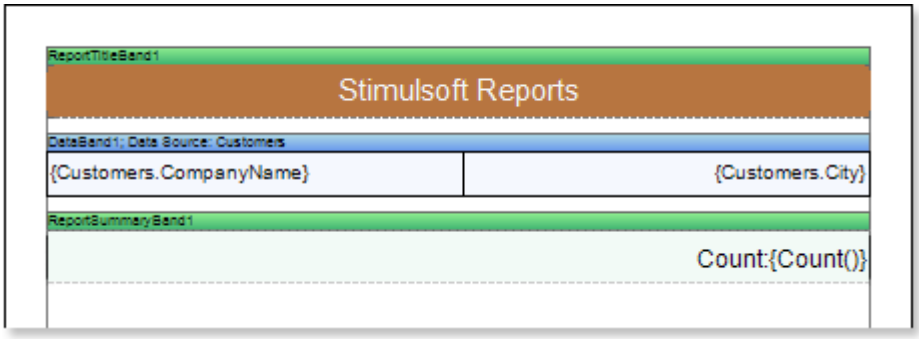
9. Go back to the report template;
10. If needed, add other bands to the report template, for example, **ReportTitleBand** and **ReportSummaryBand**;
11. Edit these bands:
    11.1. Align them by height;
    11.2. Change values of properties, if required;
    11.3. Change the background of bands;
    11.4. Enable **Borders**, if required;
    11.5. Set the border color.

The picture below shows a simple list report template with **ReportTitleBand** and **ReportSummaryBand:**

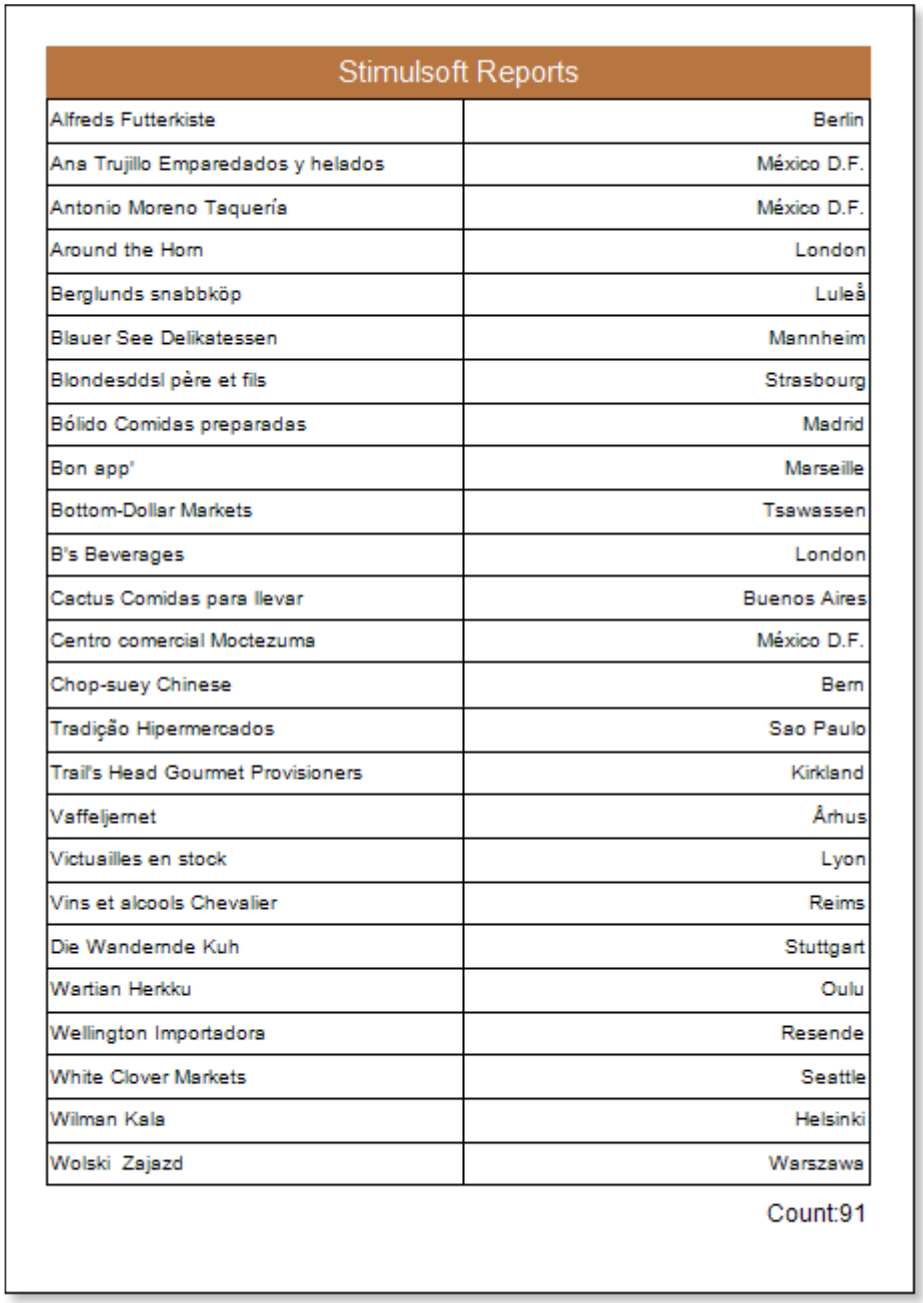


12. Put text components with expressions in the these bands. The expression in the text component is a title in the **ReportTitleBand**, and a summary in the **ReportSummaryBand**.
13. Edit text and text components:
    13.1. Drag and drop the text component in the band;
    13.2. Change font options: size, type, color;
    13.3. Align text component by height and width;
    13.4. Change the background of the text component;
    13.5. Align text in the text component;
    13.6. Change values of text component properties, if required;
    13.7. Enable **Borders** of the text component, if required;
    13.8. Set the border color.

The picture below shows a sample of the simple list report template:



14. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of a simple list report with the title and summary:



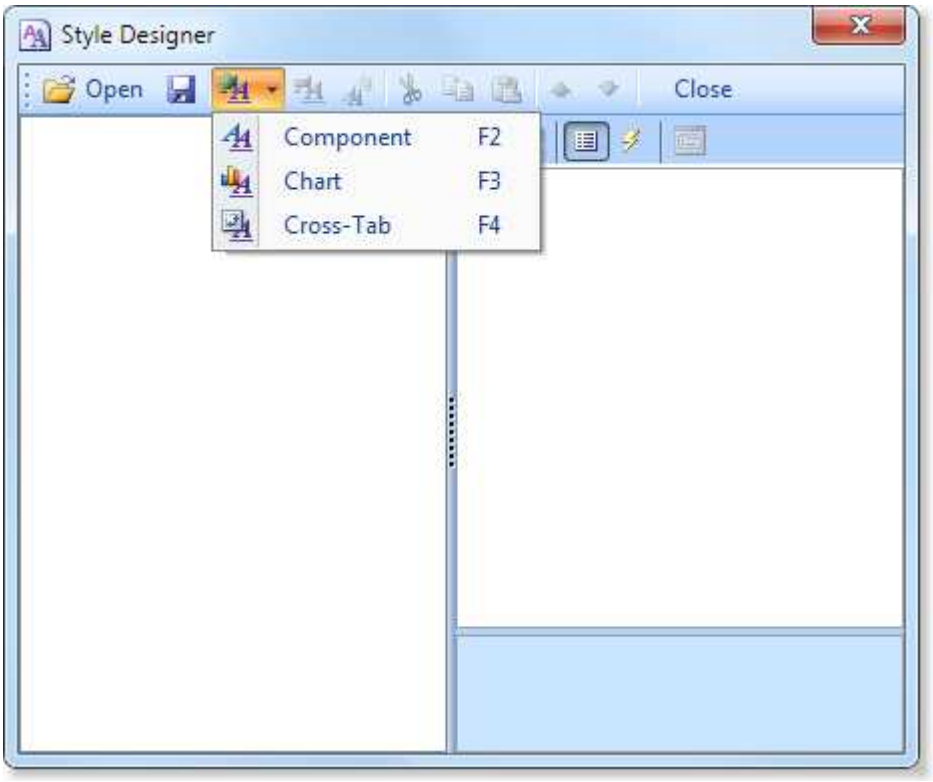
Stimulsoft Reports

Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondesddsl père et fils	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Markets	Tsawassen
B's Beverages	London
Cactus Comidas para llevar	Buenos Aires
Centro comercial Moctezuma	México D.F.
Chop-suey Chinese	Bern
Tradição Hipermercados	Sao Paulo
Trail's Head Gourmet Provisioners	Kirkland
Vaffeljernet	Århus
Victuailles en stock	Lyon
Vins et alcools Chevalier	Reims
Die Wandernde Kuh	Stuttgart
Wartian Herkku	Oulu
Wellington Importadora	Resende
White Clover Markets	Seattle
Wilman Kala	Helsinki
Wolski Zajazd	Warszawa

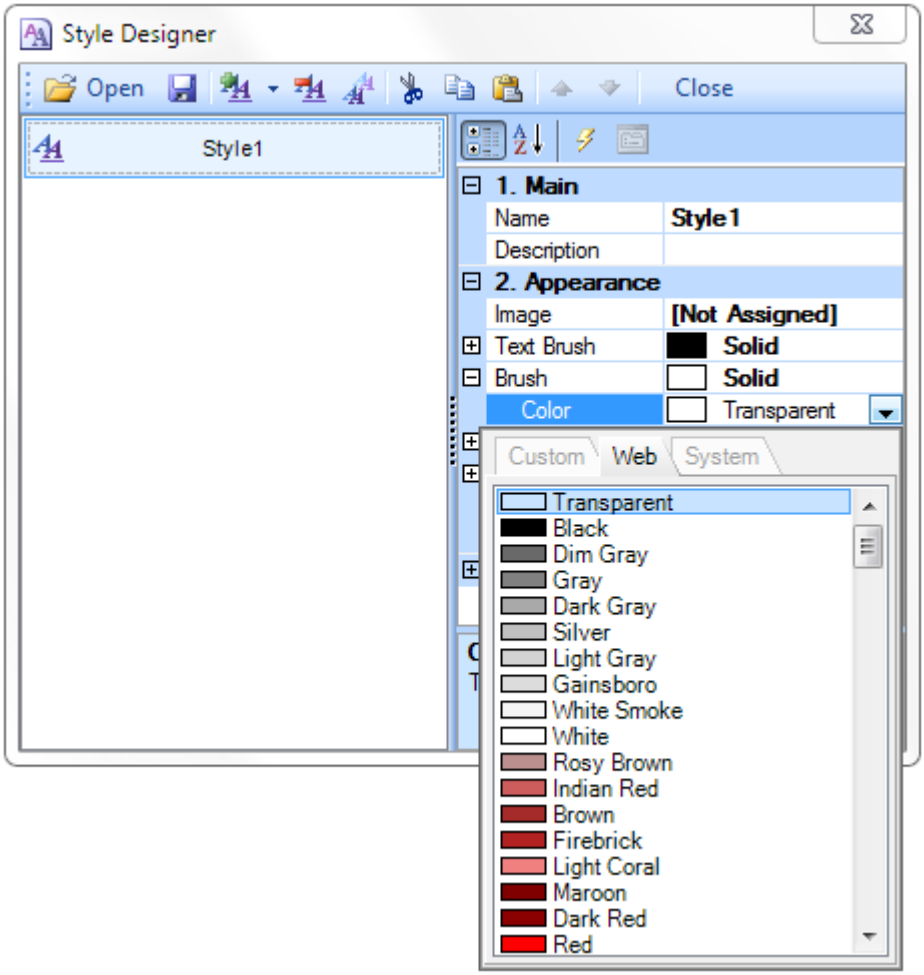
Count:91

Adding styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

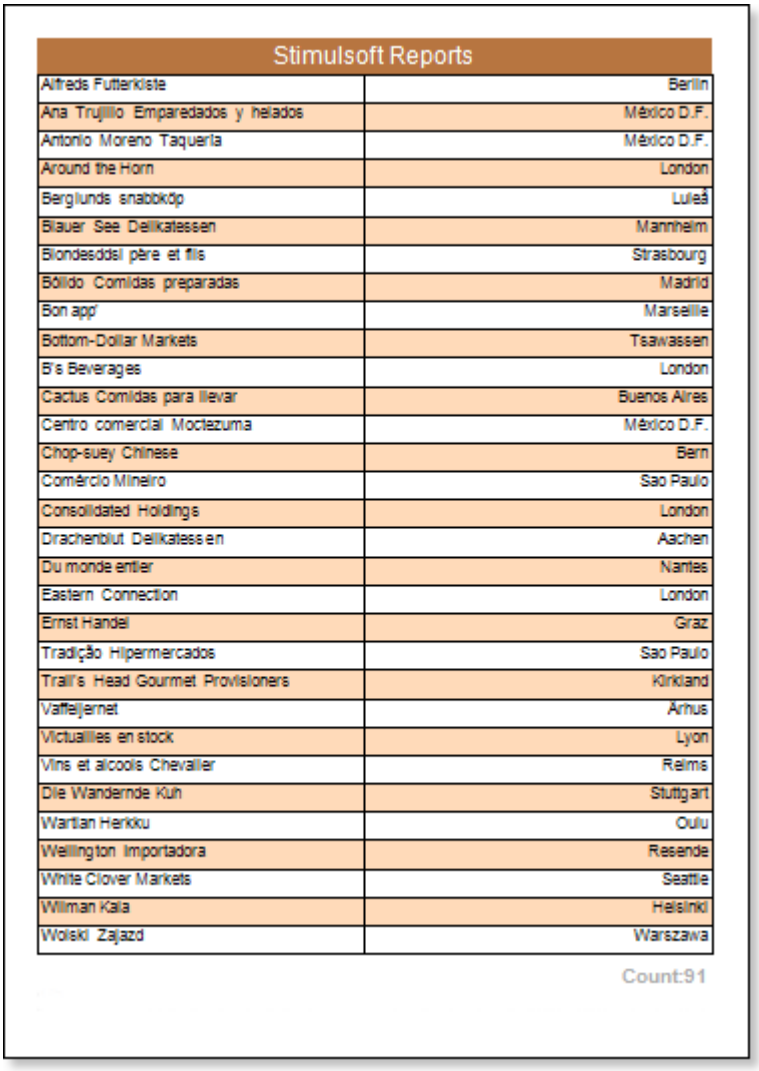


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered simple list report with alternative color of rows:



Stimulsoft Reports

Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondesddsl père et fils	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Markets	Tsawassen
B's Beverages	London
Cactus Comidas para llevar	Buenos Aires
Centro comercial Moctezuma	México D.F.
Chop-suey Chinese	Bern
Comércio Mineiro	Sao Paulo
Consolidated Holdings	London
Drachenblut Delikatessen	Aachen
Du monde entier	Nantes
Eastern Connection	London
Ernst Handel	Graz
Tradição Hipermercados	Sao Paulo
Trail's Head Gourmet Provisioners	Kirkland
Vaffeljernet	Århus
Victuailles en stock	Lyon
Vins et alcools Chevalier	Reims
Die Wandernde Kuh	Stuttgart
Wartian Herkku	Oulu
Wellington Importadora	Resende
White Clover Markets	Seattle
Wilman Kala	Helsinki
Wolski Zajazd	Warszawa

Count:91

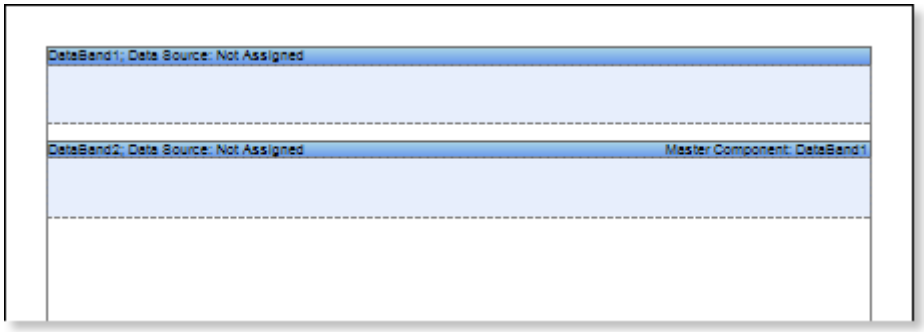
Master-Detail Report

For better understanding this step-by-step instruction, please watch the video file.

Do the following steps to create a master-detail report:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

3. Create **Relation** between data sources. If the relation will not be created and/or the **Relation** property of the **Detail** data source will not be filled, then, for **Master** entry, all **Detail** entries will be output;
4. Put two **DataBands** on a page of a report template**.**



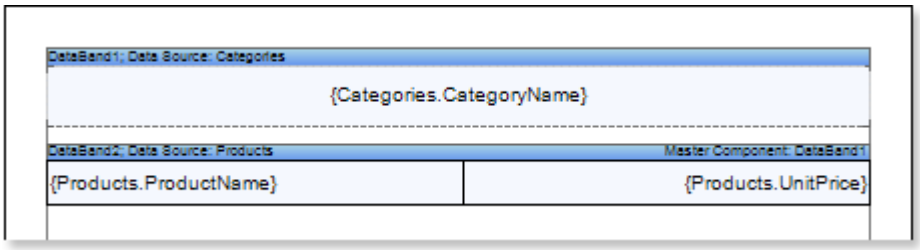
5. Edit **DataBand1** and **DataBand2**:
   5.1. Align them by height;
   5.2. Change values of required properties. For example, if to set the **PrintIfDetailEmpty** property of the **DataBand1** that is the **Master** component in the **Master-Detail** report to **true**, if it is necessary all **Master** entries be printed in any case, even if **Detail** entries not present. And set the **CanShrink** property of the **DataBand2** that is the **Detail** component in the **Master-Detail** report to **true**, if it is necessary to shrink this band;
   5.3. Change the background color of the **DataBand**;
   5.4. Enable **Borders** of the band, if required;

6. Define data sources for **DataBands**, a define the **Master** component. In our tutorial, the **Master** component is the **DataBand1**. This means that in the **Data Setup** window of the lower **DataBand2**, the **DataBand1** will be specified as the Master component in the **Master Component** tab;
7. Fill the **Data Relation** property of the **DataBand**, that is the **Detail** components. In our case this **DataBand2**:

Data Relation	Categories

8. Put text components with expressions on **DataBands**. Where expression is a reference to the data field. For example, put a text component with **{Customers.CompanyName}** expression on the **DataBand1**. Put a text component with **{Products.ProductName}** and **{Products.UnitPrice}** expressions in the **DataBand2**;
9. Edit **Text** and **TextBox** component:
   9.1. Drag and drop the text component in **DataBands**;
   9.2. Change parameters of the text font: size, type, color;
   9.3. Align the text component by width and height;
   9.4. Change the background of the text component;
   9.5. Align text in the text component;
   9.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   9.7. Enable **Borders** for the text component, if required.
   9.8. Change the border color.

The picture below shows the master-details report template.



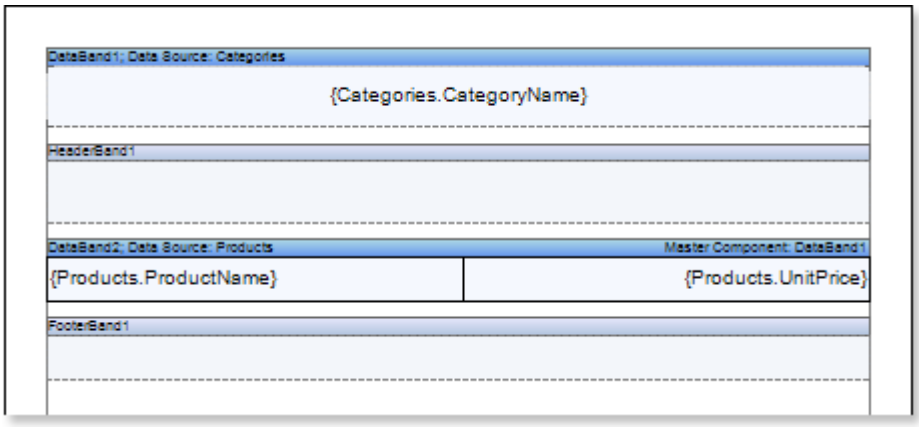
10. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of the master-detail report:

Beverages

Chai	18
Chang	19
Guaraná Fantástica	4,5
Sasquatch Ale	14
Steeleye Stout	18
Côte de Blaye	263,5
Chartreuse verte	18
Ipoh Coffee	46
Laughing Lumberjack Lager	14
Outback Lager	15
Rhönbräu Klosterbier	7,75
Lakkalikööri	18

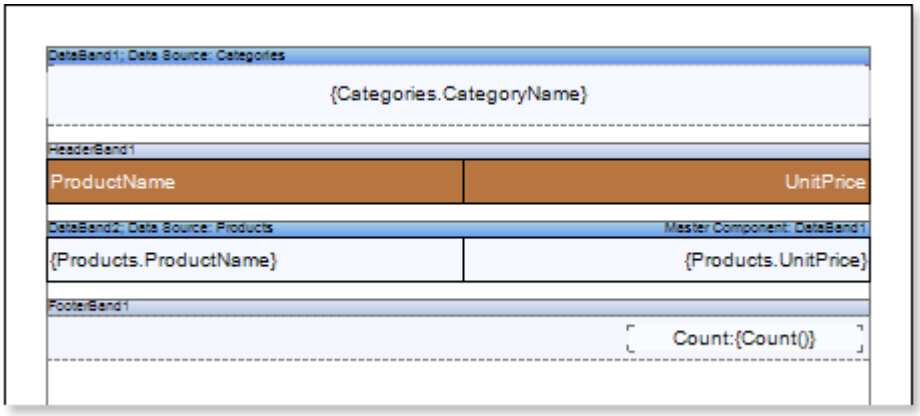
11. Go back to the report template;
12. If needed, add other bands to the report template, for example, **HeaderBand** and **FooterBand**;
13. Edit these bands:
    13.1. Align them by height;
    13.2. Change values of properties, if required;
    13.3. Change the background of bands;
    13.4. Enable **Borders**, if required;
    13.5. Set the border color.

The picture below shows a simple list report template with **HeaderBand** and **FooterBand**:

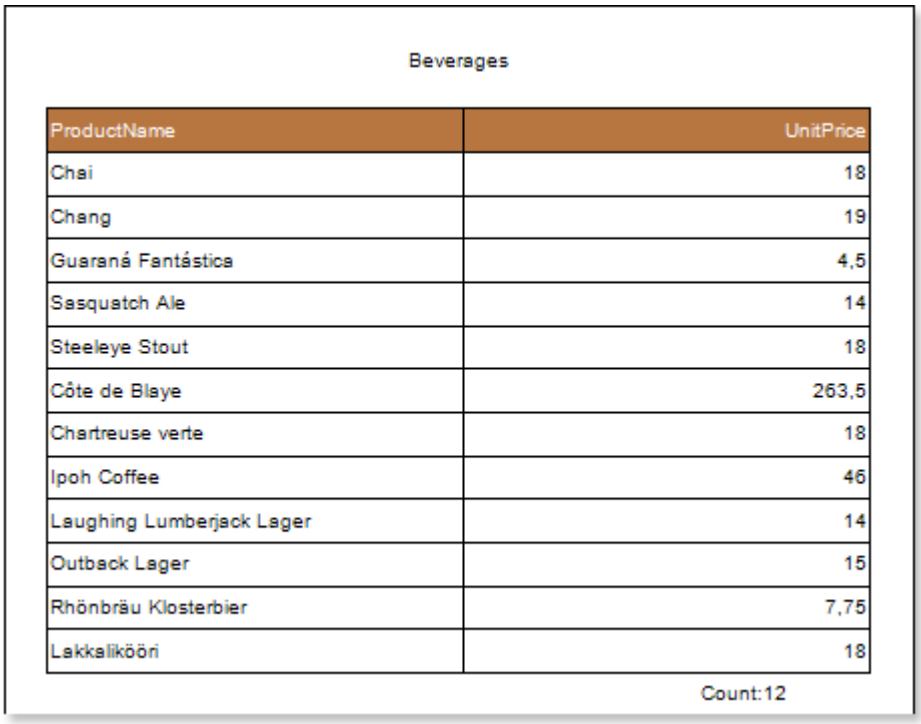


14. Put text components with expressions in the these bands. The expression in the text component is a header in the **HeaderBand**, and a footer in the **FooterBand**.
15. Edit text and text components:
    15.1. Drag and drop the text component in the band;
    15.2. Change font options: size, type, color;
    15.3. Align text component by height and width;
    15.4. Change the background of the text component;
    15.5. Align text in the text component;
    15.6. Change values of text component properties, if required;
    15.7. Enable **Borders** of the text component, if required;
    15.8. Set the border color.

The picture below shows a sample of the master-detail report template:



16. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of the master-detail report with header and footer:



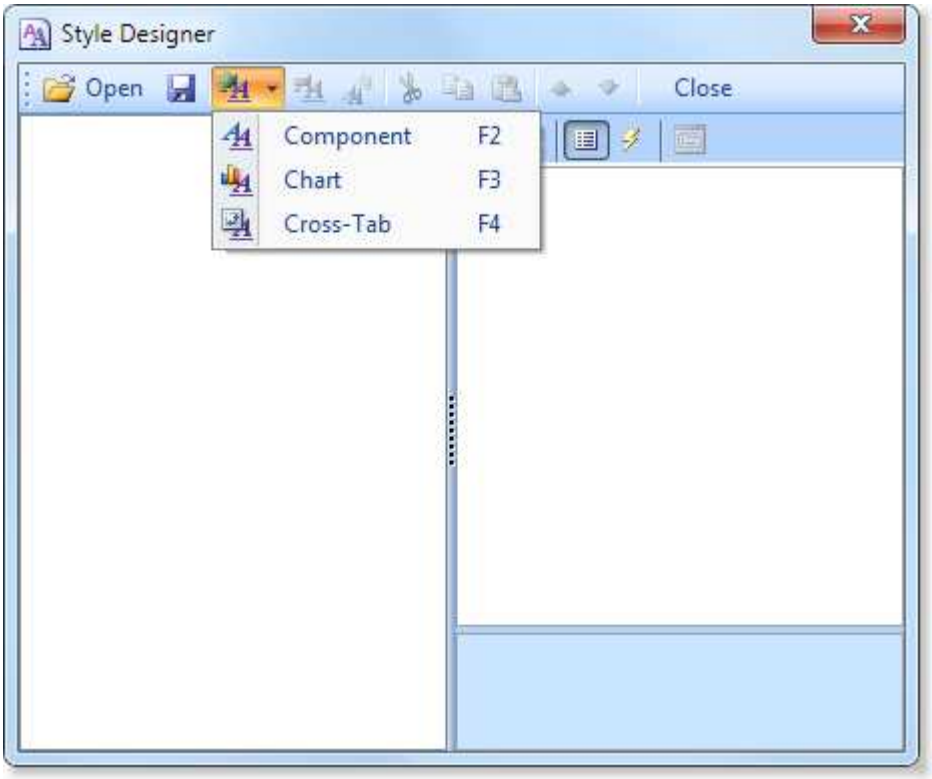
Beverages

ProductName	UnitPrice
Chai	18
Chang	19
Guaraná Fantástica	4,5
Sasquatch Ale	14
Steeleye Stout	18
Côte de Blaye	263,5
Chartreuse verte	18
Ipoh Coffee	46
Laughing Lumberjack Lager	14
Outback Lager	15
Rhönbräu Klosterbier	7,75
Lakkalikööri	18

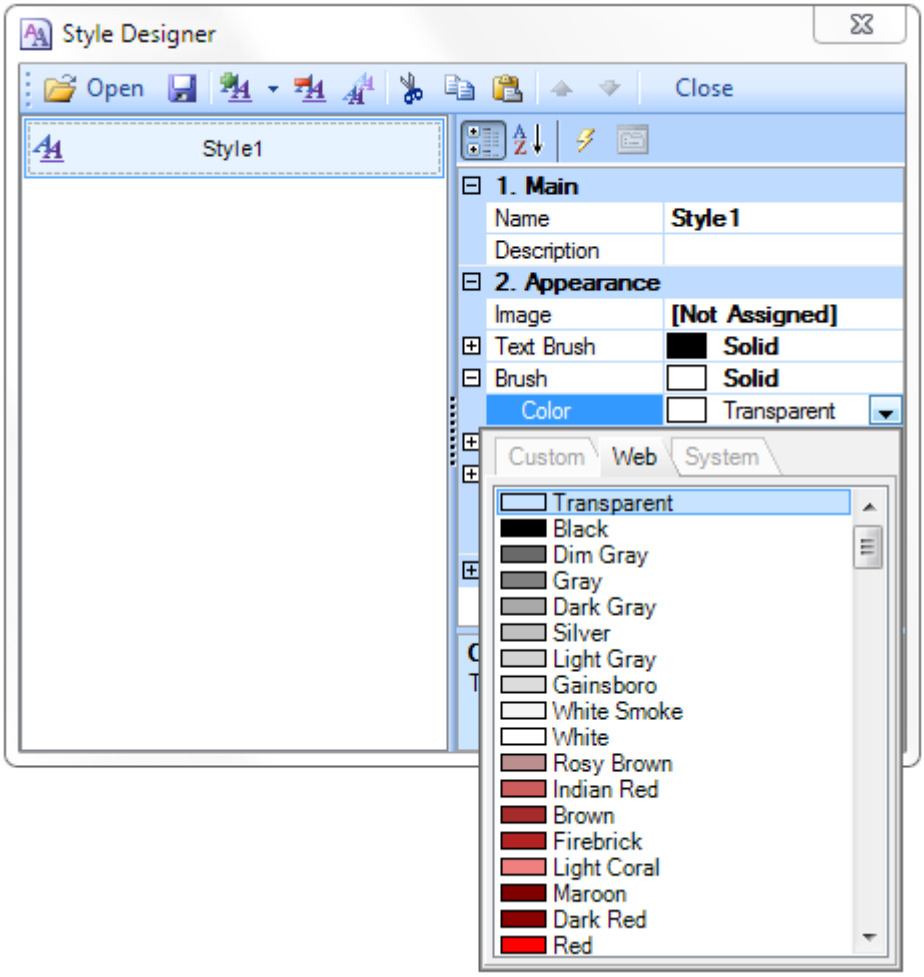
Count:12

Adding styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

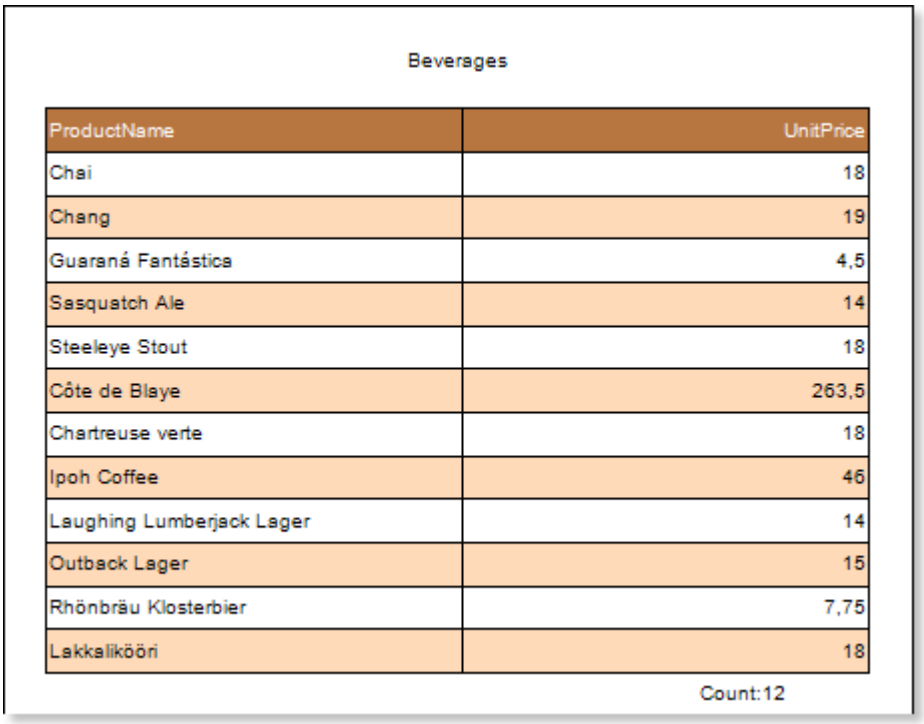


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property**:**



Click **Close**. Then in the list of **Even style** and **Odd style** properties a new value (a style of a list of odd and even rows).

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered master-detail report with alternative color of rows:



Beverages

ProductName	UnitPrice
Chai	18
Chang	19
Guaraná Fantástica	4,5
Sasquatch Ale	14
Steeleye Stout	18
Côte de Blaye	263,5
Chartreuse verte	18
Ipoh Coffee	46
Laughing Lumberjack Lager	14
Outback Lager	15
Rhönbräu Klosterbier	7,75
Lakkalikööri	18

Count:12

If to select the **DataBand1**, that is the **Master** component in the **Master-Detail** report, then it is possible to change values of **Even style**
and **Odd style** properties. In such a case, alternative row color will be applied only for **Master** entries.

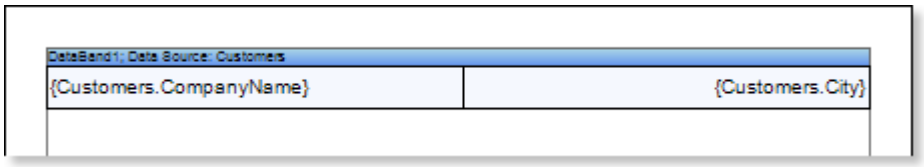
Report with Groups

For better understanding this step-by-step instruction, please watch the video file.

Do the following steps to create a report with grouping:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

3. Create a report or open already created one. For example, we can take a simple list report created in the chapter "Simple List Report".

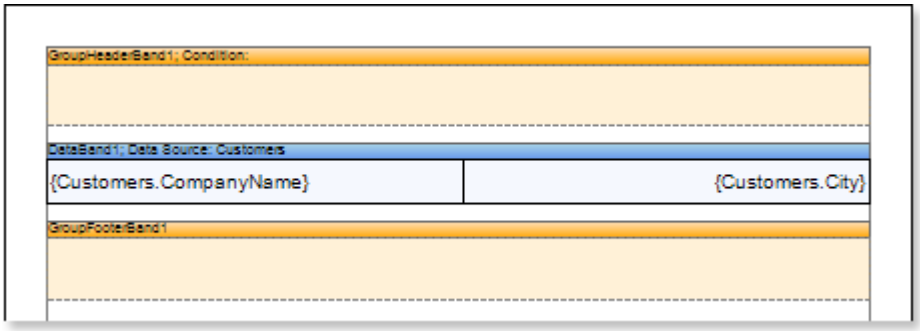


DataBand1; Data Source: Customers

{Customers.CompanyName}	{Customers.City}

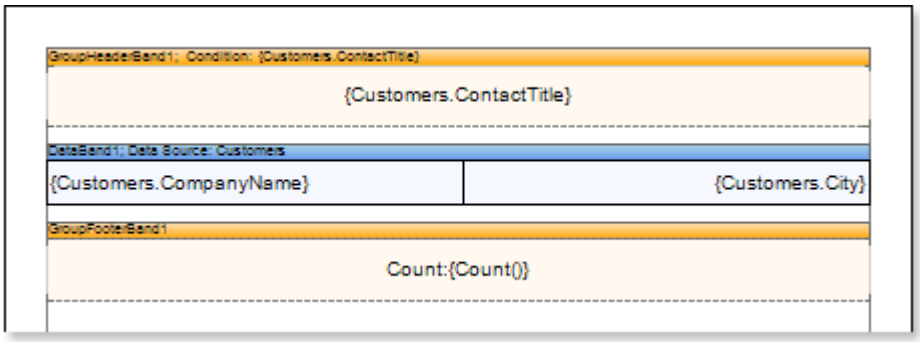
4. Add **GroupHeaderBand** and **GroupFooterBand** to the report template. The **GroupHeaderBand** should be placed higher than the **DataBand** to what it is related to. The **GroupFooterBand** is placed under the **Data to** what **GroupHeader** is related. Each **GroupFooter** corresponds to a specified **GroupHeader**. The **GroupFooter** band

will not output without **GroupHeader**. The picture below shows a report template with added **GroupHeaderBand** and **GroupFooterBand)**.



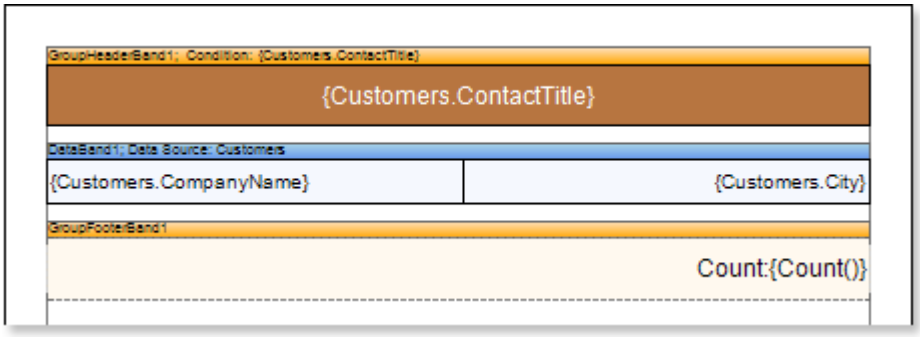
5. Edit **GroupHeaderBand** and **GroupFooterBand**:
   5.1. Align them be height;
   5.2. Change values of properties according to requirements. For example, set the **KeepGroupHeaderTogether** property for the **GroupHeaderBand to true**, it is necessary to keep the group header with the group. And for the **GroupFooterBand** set the **KeepFooterTogether** to **true**, if it is required to keep the footer with the group;
   5.3. Set the background of the **GroupHeaderBand**;
   5.4. Enable **Borders** of the **DataBand**, if required;

6. Set the condition data grouping in the report using the **Condition** property of the **GroupHeader** band. Condition of grouping can be set by setting the expression or by selecting the data column from the data source. In our tutorial, define the **{Customers.ContactTitle}** expression in the condition of grouping.
7. Put a text component in the **GroupHeaderBand** and put the **{Customers.ContactTitle}** expression into this text component. Put a text component in the **GroupFooterBand** and put the **{Count()}** expression into this text component. The **{Count()}** function will count summary by the amount of entries in each group. The picture below shows a report template with the condition of grouping set, and text components placed in **GroupHeaderBand** and **GroupFooterBand:**

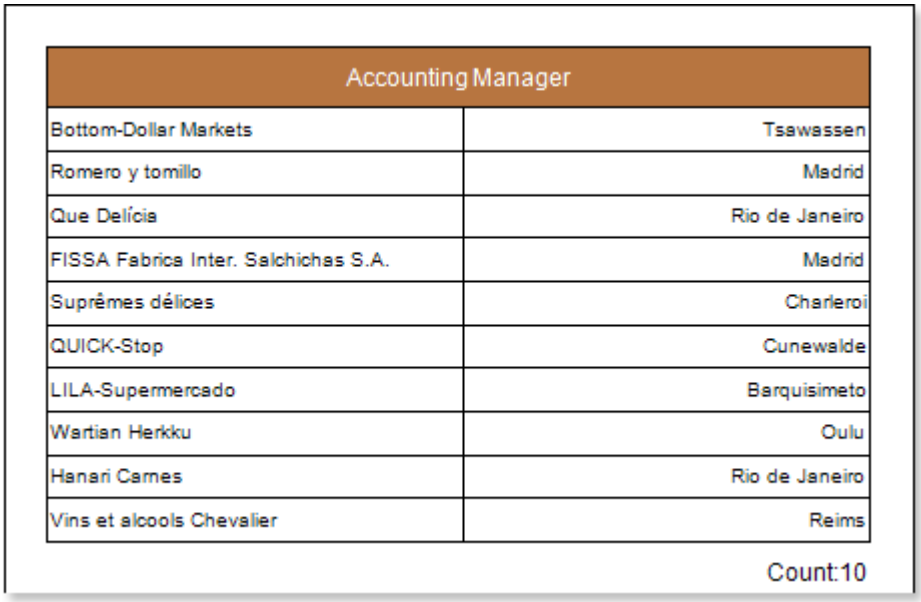


8. Edit expressions and text components:
   8.1. Drag and drop the text component in **GroupHeaderBand** and **GroupFooterBand**;
   8.2. Change parameters of the text font: size, type, color;
   8.3.. Align the text component by width and height;
   8.4. Change the background of the text component;
   8.5. Align text in the text component;
   8.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   8.7. Enable **Borders** for the text component, if required.
   8.8. Change the border color.

The picture below shows a sample of the edited report template with grouping:



9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of the report with grouping:

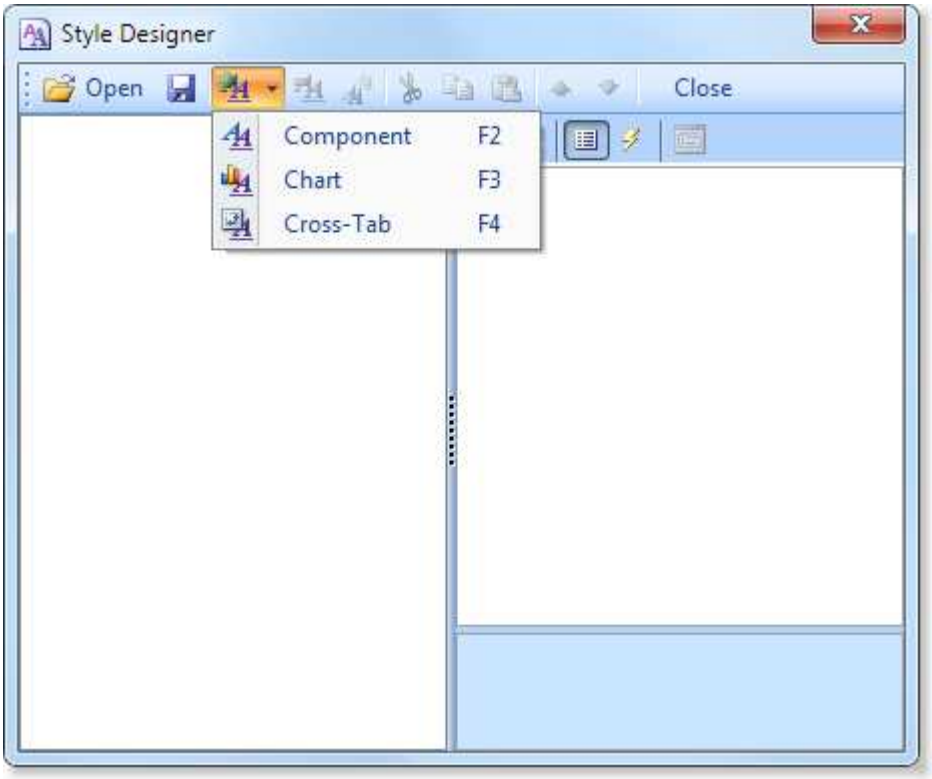


Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delícia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	Reims

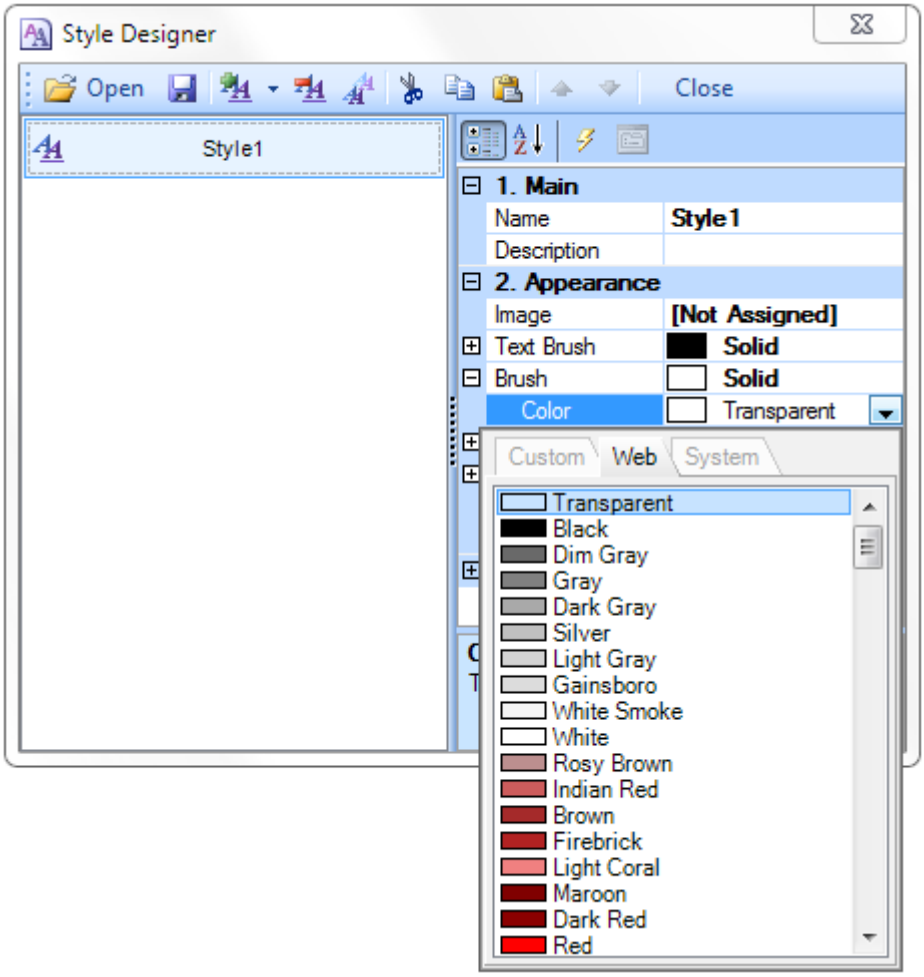
Count:10

Adding styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

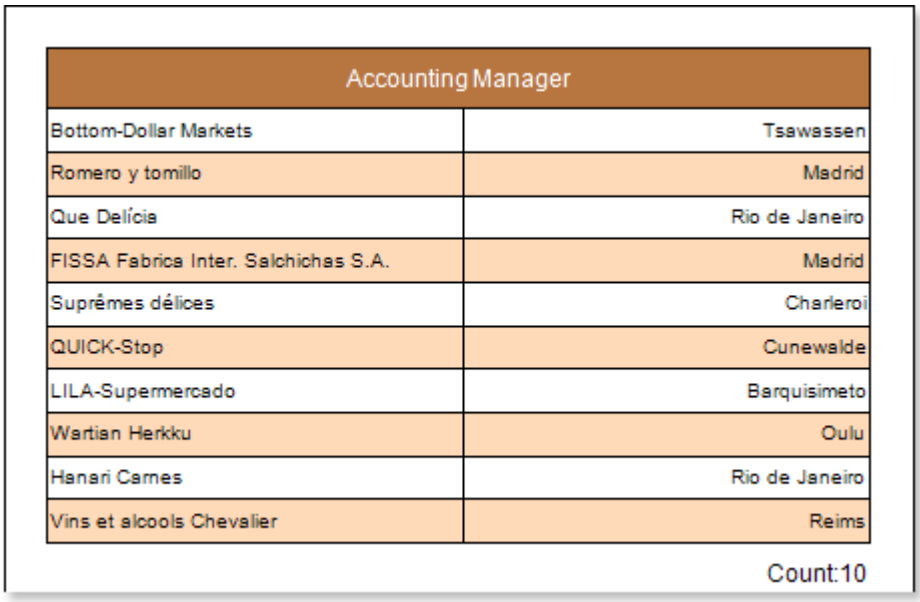


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then in the list of **Even style** and **Odd style** properties a new value (a style of a list of odd and even rows).

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered report with grouping and alternative color of rows:



Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delícia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	Reims

Count:10

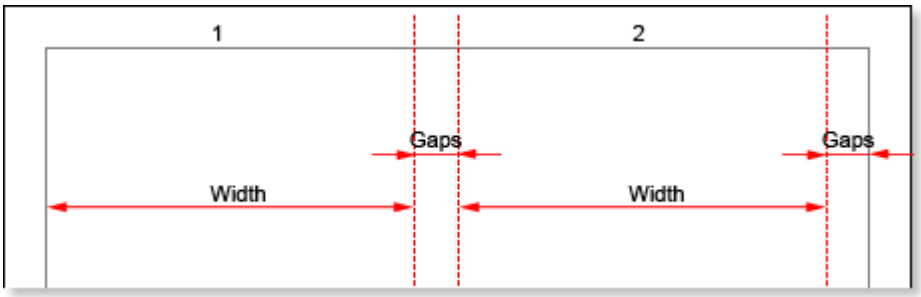
Report with Columns on Page

For better understanding this step-by-step instruction, please watch the video file.

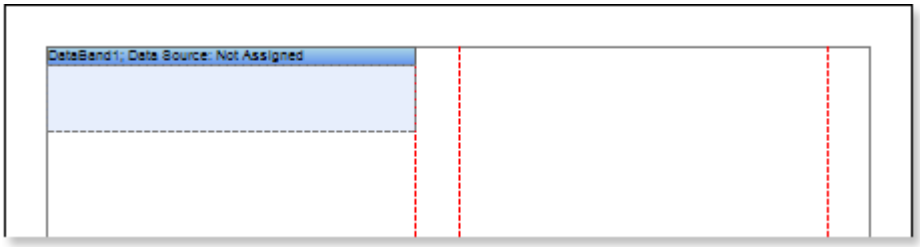
Do the following steps to create a report with columns on a page:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

3. Set column options: the number of columns, column width, and column gap. For example, set the number of columns equal to **2**, with the gap equal to **1**. The column width is created automatically. The picture below shows a sample of the report template with two columns:

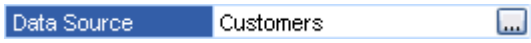


4. Put **DataBand** on a page.

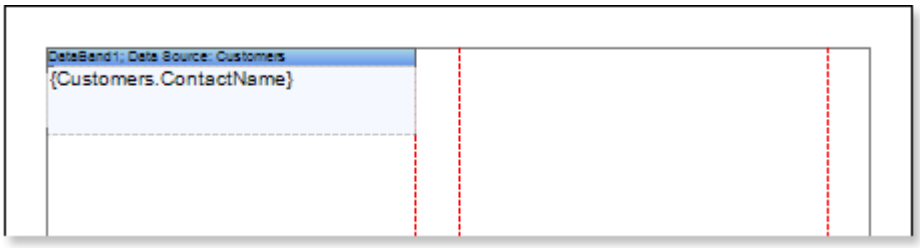


5. Edit **DataBand**:
   5.1. Align the **DataBand** by height;
   5.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
   5.3. Change the **DataBand** background;
   5.4. Enable **Borders** for the **DataBand**, if required;
   5.5. Change the border color.

6. Define the data source for the **DataBand** using the **Data Source** property**:**

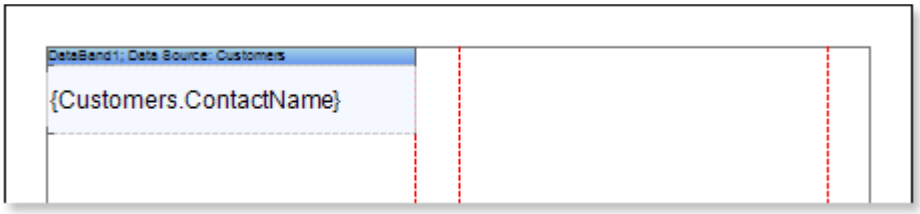


7. Put text components with expressions on the **DataBand**. Where expression is a reference to the data field. For example, put two text components with expressions: **{Customers.ContactName}**.



8. Edit expressions and text components:
   8.1. Drag and drop the text component in **DataBand**;
   8.2. Change parameters of the text font: size, type, color;
   8.3. Align the text component by width and height;
   8.4. Change the background of the text component;
   8.5. Align text in the text component;
   8.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   8.7. Enable **Borders** for the text component, if required.
   8.8. Change the border color.

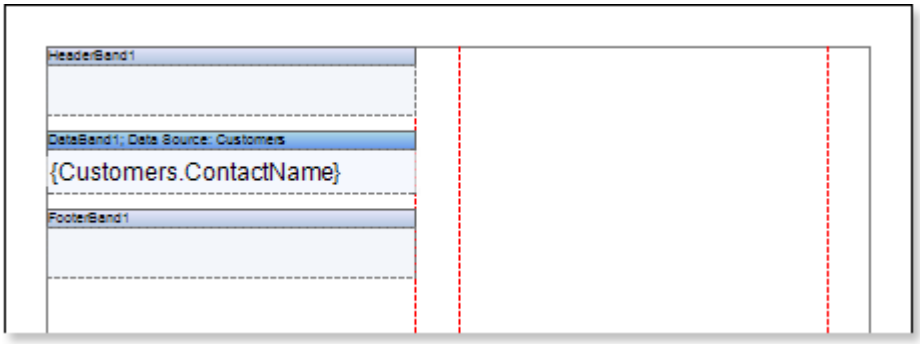
The picture below shows a report template with edited text component:



9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of the report with two columns on a page:

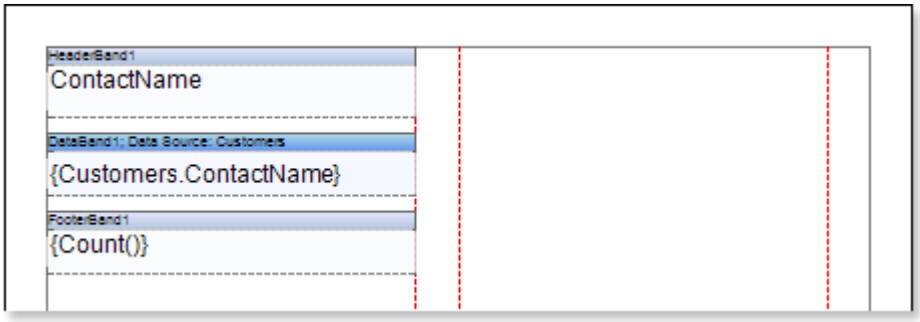
Step **3** and **4** can be changed in sequence of doing. So you may put **DataBand** first and then set the column options on page.

10. Go back to the report template;
11. If needed, add other bands to the report template, for example, **HeaderBand** and **FooterBand**;



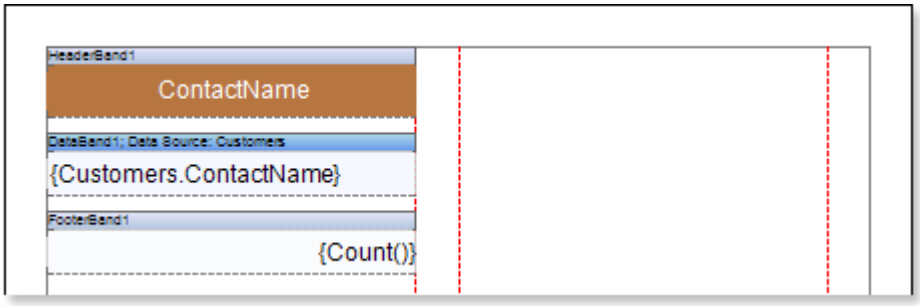
12. Edit these bands:
    12.1. Align them by height;
    12.2. Change values of properties, if required;
    12.3. Change the background of bands;
    12.4. Enable **Borders**, if required;
    12.5. Set the border color.

13. Put text components with expressions in the these bands. The expression in the text component is a header in the **HeaderBand**, and a footer in the **FooterBand**.

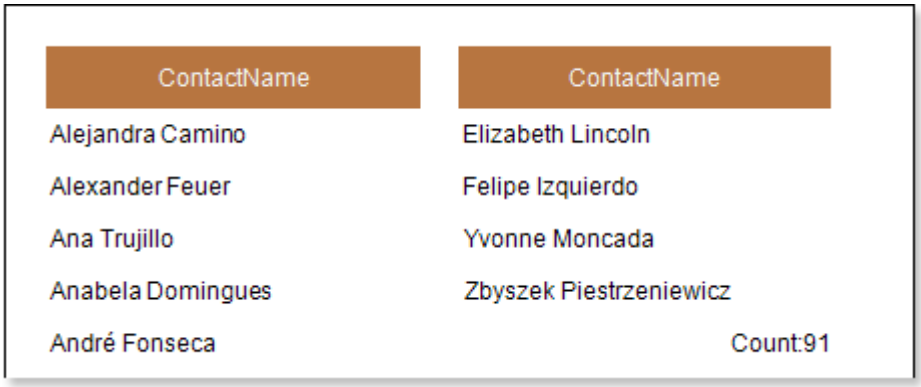


14. Edit text and text components:
    14.1. Drag and drop the text component in the band;
    14.2. Change font options: size, type, color;
    14.3. Align text component by height and width;
    14.4. Change the background of the text component;
    14.5. Align text in the text component;
    14.6. Change values of text component properties, if required;
    14.7. Enable **Borders** of the text component, if required;
    14.8. Set the border color.

The picture below shows a sample of the report with two columns on a page:

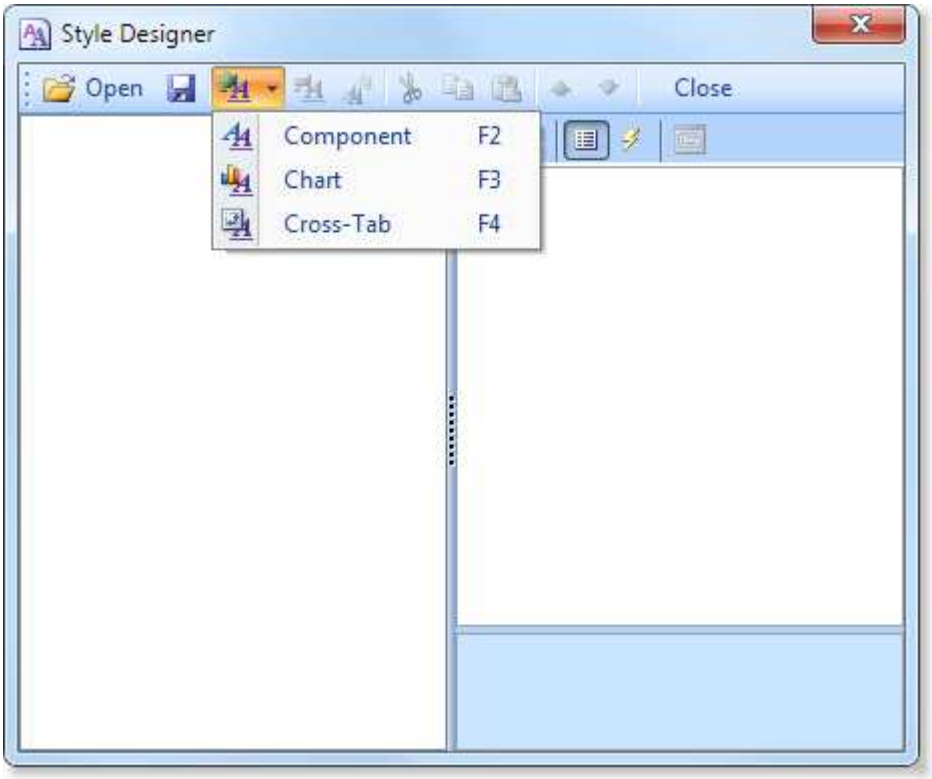


15. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of the report with a header and a footer:

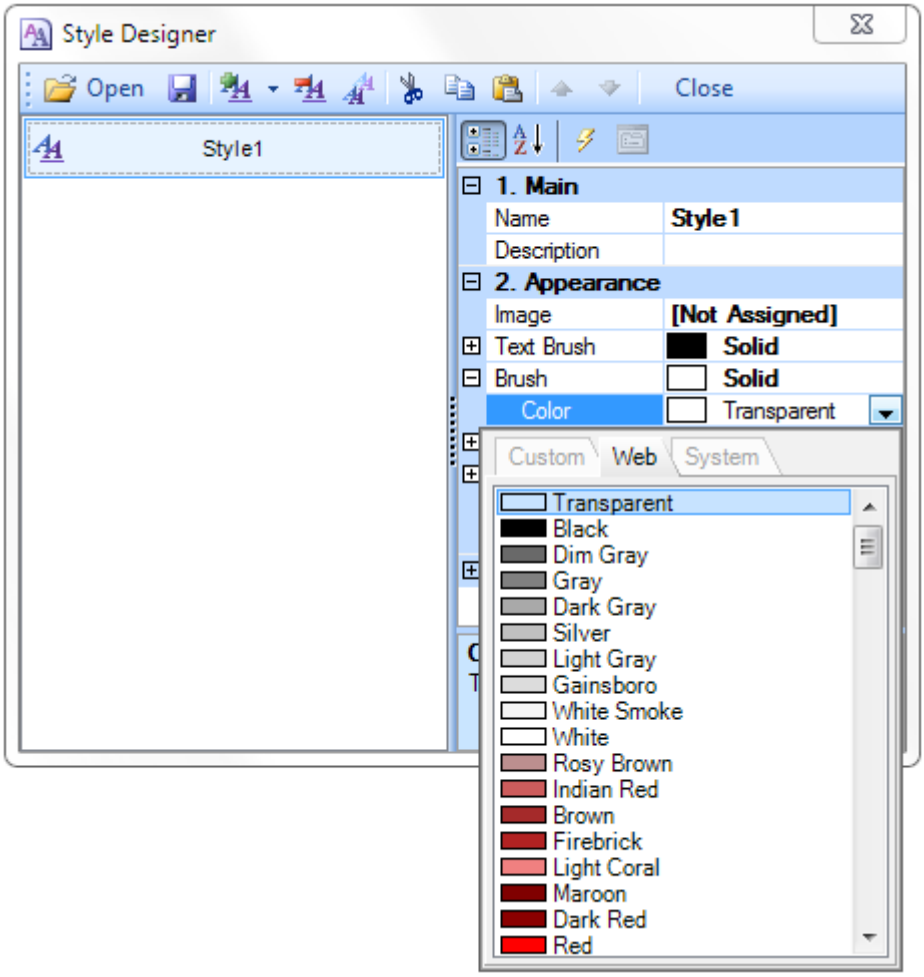


Adding styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:



Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then in the list of **Even style** and **Odd style** properties a new value (a style of a list of odd and even rows).

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered report with columns on a page and alternative color of rows:

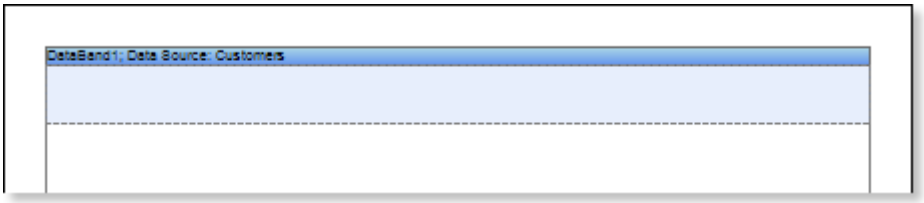
Report with Columns in DataBand

For better understanding this step-by-step instruction, please watch the video file.

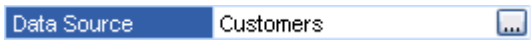
Do the following steps to create a report with columns in DataBand:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

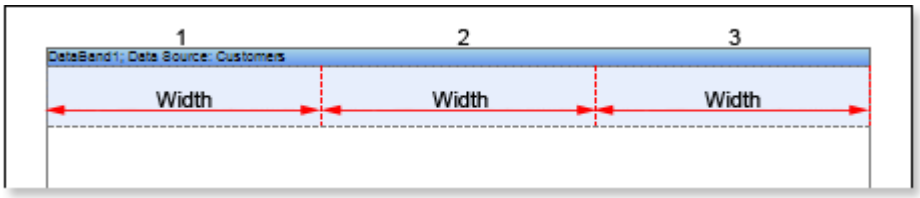
3. Put a **DataBand** on a page of a report template.



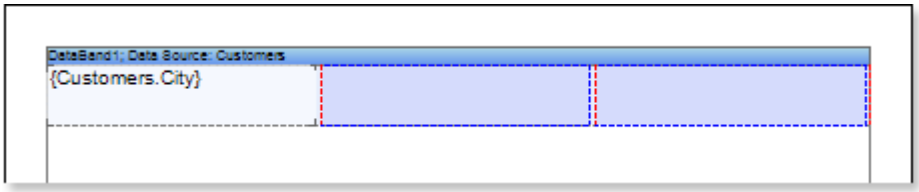
4. Define the data source for the **DataBand** using, for example, the **Data Source** property**:**



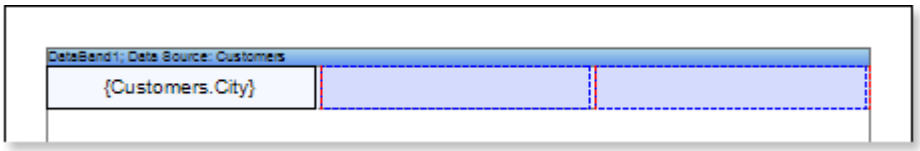
5. Set column options: the number of columns, column width, and column gap. For example, set the number of columns equal to **3**, with the gap equal to **0**. The column width is created automatically. The picture below shows a sample of the report template with two columns, placed in the **DataBand**:



6. Put a text component with expressions on the **DataBand**. Where expression is a reference to the data field. For example, put one text component with the **{Customers.City}** expression.



7. Edit expressions and text components:
   7.1. Drag and drop the text component in **DataBand**;
   7.2. Change parameters of the text font: size, type, color;
   7.3. Align the text component by width and height;
   7.4. Change the background of the text component;
   7.5. Align text in the text component;
   7.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   7.7. Enable **Borders** for the text component, if required.
   7.8. Change the border color.



8. Set the columns direction of data output using the **Column Direction** property. Read about this property in section Report Internals -> Columns.
9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows samples of reports with columns rendered using different values of the **Column Direction** property.

Down Then Across

1.Aachen	24.Elgin	47.Madrid
2.Albuquerque	25.Eugene	48.Madrid
3.Anchorage	26.Frankfurt a.M.	49.Madrid
4.Århus	27.Genève	50.Mannheim
5.Barcelona	28.Graz	51.Marseille
6.Barquisimeto	29.Helsinki	52.México D.F.
7.Bergamo	30.I. de Margarita	53.México D.F.
8.Berlin	31.Kirkland	54.México D.F.
9.Bern	32.Kobenhavn	55.México D.F.
10.Boise	33.Köln	56.México D.F.
11.Bräcke	34.Lander	57.Montréal
12.Brandenburg	35.Leipzig	58.München
13.Bruxelles	36.Lille	59.Münster
14.Buenos Aires	37.Lisboa	60.Nantes
15.Buenos Aires	38.Lisboa	61.Nantes
16.Buenos Aires	39.London	62.Oulu
17.Butte	40.London	63.Paris
18.Campinas	41.London	64.Paris
19.Caracas	42.London	65.Portland
20.Charleroi	43.London	66.Portland
21.Cork	44.London	67.Reggio Emilia
22.Cowes	45.Luleå	68.Reims
23.Cunewalde	46.Lyon	69.Resende

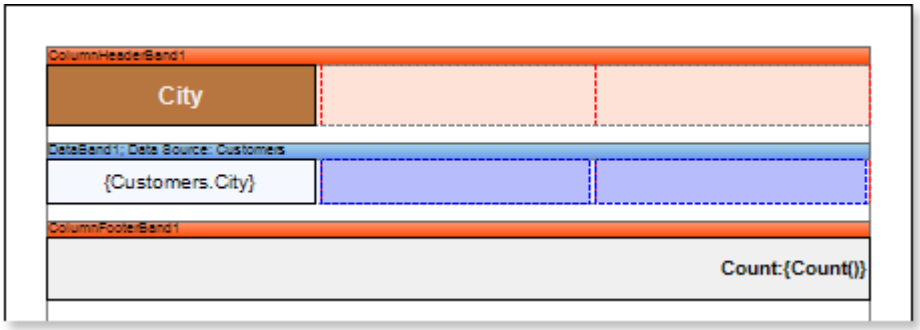
Across Then Down

1.Aachen	2.Albuquerque	3.Anchorage
4.Århus	5.Barcelona	6.Barquisimeto
7.Bergamo	8.Berlin	9.Bern
10.Boise	11.Bräcke	12.Brandenburg
13.Bruxelles	14.Buenos Aires	15.Buenos Aires
16.Buenos Aires	17.Butte	18.Campinas
19.Caracas	20.Charleroi	21.Cork
22.Cowes	23.Cunewalde	24.Elgin
25.Eugene	26.Frankfurt a.M.	27.Genève
28.Graz	29.Helsinki	30.I. de Margarita
31.Kirkland	32.Kobenhavn	33.Köln
34.Lander	35.Leipzig	36.Lille
37.Lisboa	38.Lisboa	39.London
40.London	41.London	42.London
43.London	44.London	45.Luleå
46.Lyon	47.Madrid	48.Madrid
49.Madrid	50.Mannheim	51.Marseille
52.México D.F.	53.México D.F.	54.México D.F.
55.México D.F.	56.México D.F.	57.Montréal
58.München	59.Münster	60.Nantes
61.Nantes	62.Oulu	63.Paris
64.Paris	65.Portland	66.Portland
67.Reggio Emilia	68.Reims	69.Resende

10. Go back to the report template;
11. If needed, add other bands to the report template, for example, **ColumnHeaderBand** and **ColumnFooterBand**.

12. Edit these bands:
    12.1. Align them by height;
    12.2. Change values of properties, if required;
    12.3. Change the background of bands;
    12.4. Enable **Borders**, if required;
    12.5. Set the border color.
13. Put text components with expressions in the these bands. Where expression of the text component in the **ColumnHeaderBand** is the column name and the expression of the text component in the **ColumnFooterBand** is the data footer.

14. Edit **Text** and **TextBox** component:
    14.1. Drag and drop the text component in **ColumnHeaderBand** and **ColumnFooterBand**;
    14.2. Change parameters of the text font: size, type, color;
    14.3. Align the text component by width and height;
    14.4. Change the background of the text component;
    14.5. Align text in the text component;
    14.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
    14.7. Enable **Borders** for the text component, if required.
    14.8. Change the border color.



15. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows samples of reports with column headers.

Down Then Across

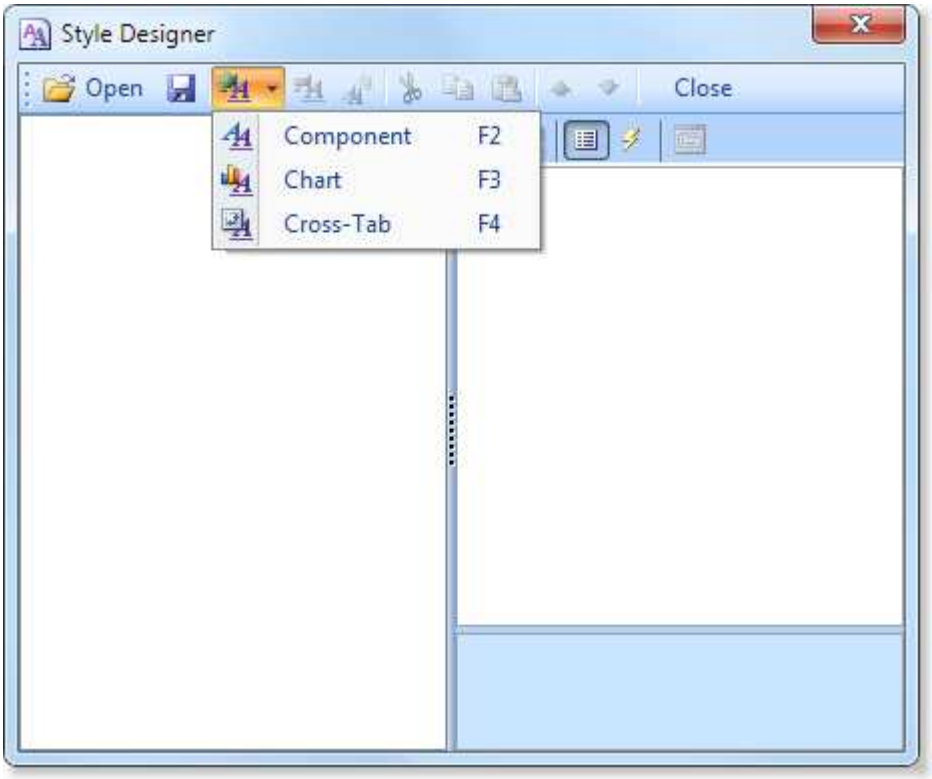
City	City	City
1.Aachen	22.Cowes	43.London
2.Albuquerque	23.Cunewalde	44.London
3.Anchorage	24.Elgin	45.Luleå
4.Århus	25.Eugene	46.Lyon
5.Barcelona	26.Frankfurt a.M.	47.Madrid
6.Barquisimeto	27.Genève	48.Madrid
7.Bergamo	28.Graz	49.Madrid
8.Berlin	29.Helsinki	50.Mannheim
9.Bern	30.I. de Margarita	51.Marseille
10.Boise	31.Kirkland	52.México D.F.
11.Bräcke	32.Kobenhavn	53.México D.F.
12.Brandenburg	33.Köln	54.México D.F.
13.Bruxelles	34.Lander	55.México D.F.
14.Buenos Aires	35.Leipzig	56.México D.F.
15.Buenos Aires	36.Lille	57.Montréal
16.Buenos Aires	37.Lisboa	58.München
17.Butte	38.Lisboa	59.Münster
18.Campinas	39.London	60.Nantes
19.Caracas	40.London	61.Nantes
20.Charleroi	41.London	62.Oulu
21.Cork	42.London	63.Paris

Across Then Down

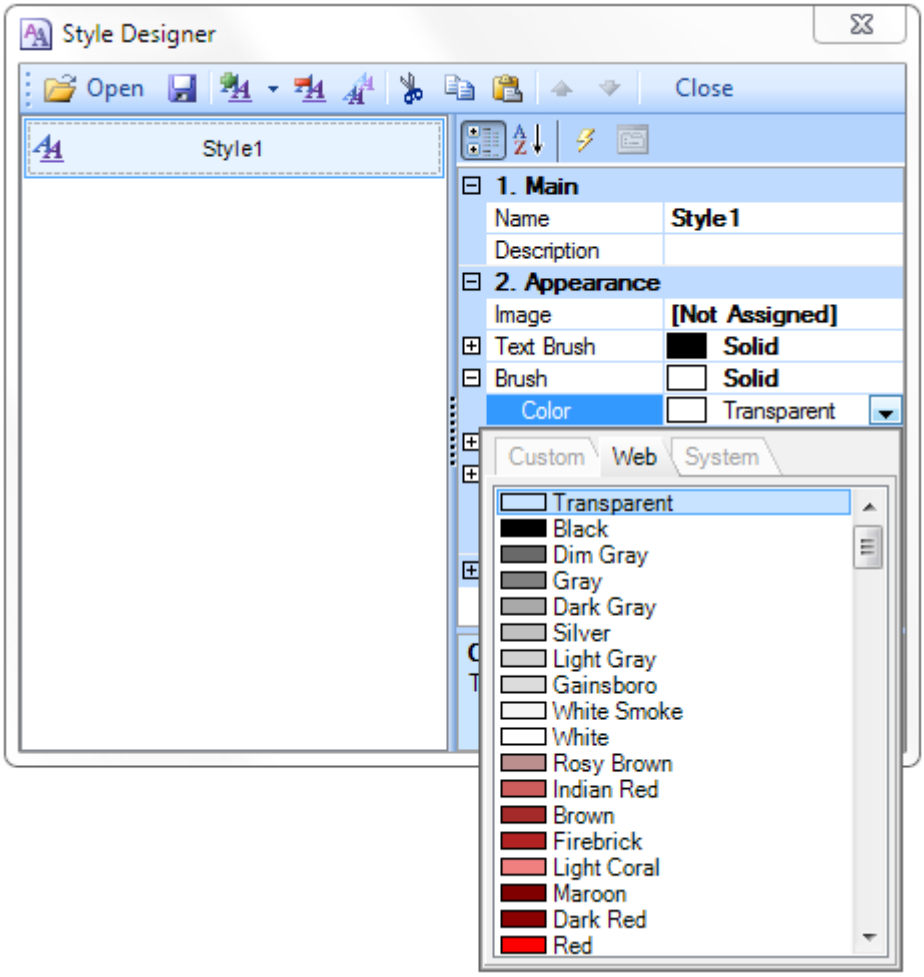
City	City	City
1.Aachen	2.Albuquerque	3.Anchorage
4.Århus	5.Barcelona	6.Barquisimeto
7.Bergamo	8.Berlin	9.Bern
10.Boise	11.Bräcke	12.Brandenburg
13.Bruxelles	14.Buenos Aires	15.Buenos Aires
16.Buenos Aires	17.Butte	18.Campinas
19.Caracas	20.Charleroi	21.Cork
22.Cowes	23.Cunewalde	24.Elgin
25.Eugene	26.Frankfurt a.M.	27.Genève
28.Graz	29.Helsinki	30.I. de Margarita
31.Kirkland	32.Kobenhavn	33.Köln
34.Lander	35.Leipzig	36.Lille
37.Lisboa	38.Lisboa	39.London
40.London	41.London	42.London
43.London	44.London	45.Luleå
46.Lyon	47.Madrid	48.Madrid
49.Madrid	50.Mannheim	51.Marseille
52.México D.F.	53.México D.F.	54.México D.F.
55.México D.F.	56.México D.F.	57.Montréal
58.München	59.Münster	60.Nantes
61.Nantes	62.Oulu	63.Paris

Adding styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:



Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then in the list of **Even style** and **Odd style** properties a new value (a style of a list of odd and even rows).

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered report with columns on a page and alternative color of rows:

Down Then Across

City	City	City
1.Aachen	22.Cowes	43.London
2.Albuquerque	23.Cunewalde	44.London
3.Anchorage	24.Elgin	45.Luleå
4.Århus	25.Eugene	46.Lyon
5.Barcelona	26.Frankfurt a.M.	47.Madrid
6.Barquisimeto	27.Genève	48.Madrid
7.Bergamo	28.Graz	49.Madrid
8.Berlin	29.Helsinki	50.Mannheim
9.Bern	30.I. de Margarita	51.Marseille
10.Boise	31.Kirkland	52.México D.F.
11.Bräcke	32.Kobenhavn	53.México D.F.
12.Brandenburg	33.Köln	54.México D.F.
13.Bruxelles	34.Lander	55.México D.F.
14.Buenos Aires	35.Leipzig	56.México D.F.
15.Buenos Aires	36.Lille	57.Montréal
16.Buenos Aires	37.Lisboa	58.München
17.Butte	38.Lisboa	59.Münster
18.Campinas	39.London	60.Nantes
19.Caracas	40.London	61.Nantes
20.Charleroi	41.London	62.Oulu
21.Cork	42.London	63.Paris

Across Then Down

City	City	City
1.Aachen	2.Albuquerque	3.Anchorage
4.Århus	5.Barcelona	6.Barquisimeto
7.Bergamo	8.Berlin	9.Bern
10.Boise	11.Bräcke	12.Brandenburg
13.Bruxelles	14.Buenos Aires	15.Buenos Aires
16.Buenos Aires	17.Butte	18.Campinas
19.Caracas	20.Charleroi	21.Cork
22.Cowes	23.Cunewalde	24.Elgin
25.Eugene	26.Frankfurt a.M.	27.Genève
28.Graz	29.Helsinki	30.I. de Margarita
31.Kirkland	32.Kobenhavn	33.Köln
34.Lander	35.Leipzig	36.Lille
37.Lisboa	38.Lisboa	39.London
40.London	41.London	42.London
43.London	44.London	45.Luleå
46.Lyon	47.Madrid	48.Madrid
49.Madrid	50.Mannheim	51.Marseille
52.México D.F.	53.México D.F.	54.México D.F.
55.México D.F.	56.México D.F.	57.Montréal
58.München	59.Münster	60.Nantes
61.Nantes	62.Oulu	63.Paris

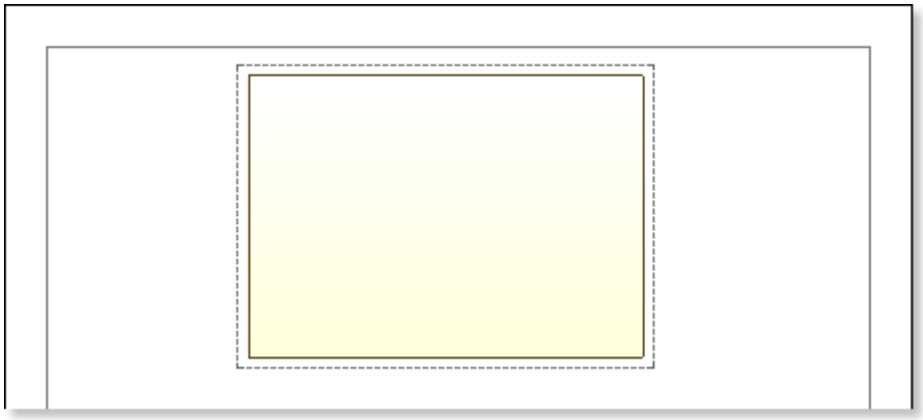
Report with Chart on Page

For better understanding this step-by-step instruction, please watch the video file.

Do the following steps to create a report with charts:

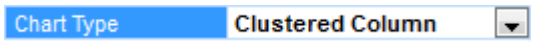
1. Run the designer;
2. Connect data:
    2.1. Create **New Connection**;
    2.2. Create **New Data Source**;

3. Put the **Chart** component on a page as seen on a picture below.

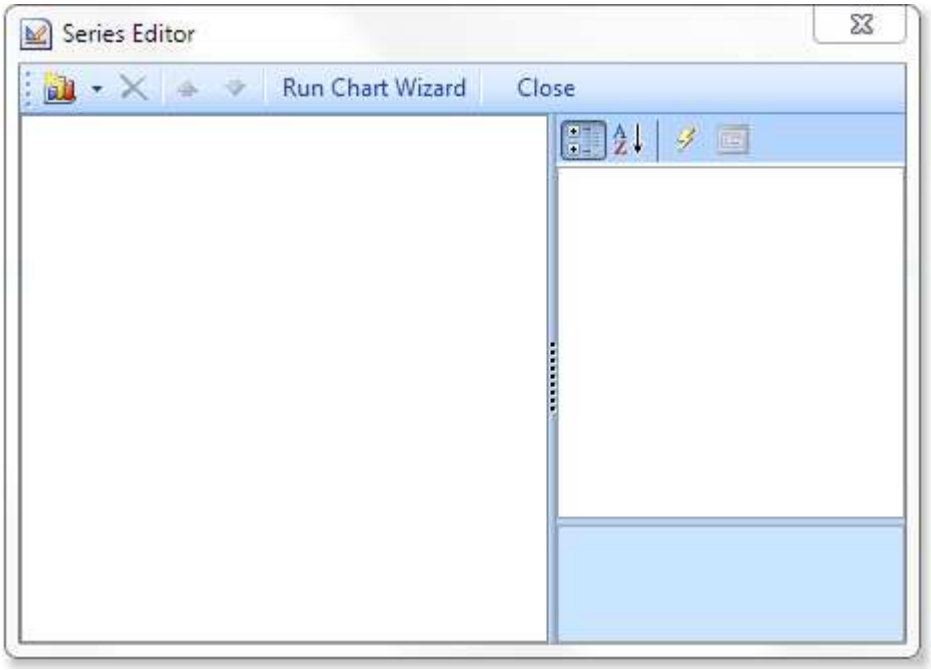


4. Edit the **Chart** component:
   4.1. Align it by width;
   4.2. Change properties of the **Chart** component. For example, set the **GrowToHeight** property to **true**, if it is required the Chart component be grown by height;
   4.3. Set **Borders**, if required, for the **Chart** component;
   4.4. Change the border color.
   4.5. Edit the chart area. For example, change the **Area.Brush.Color** property, if it is required to change the color of a chart area.

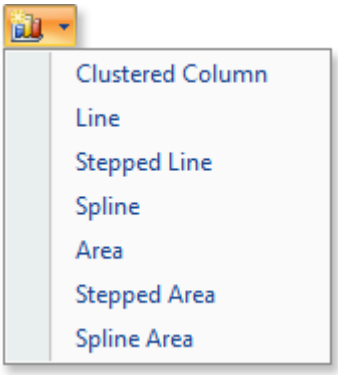
5. Change the type of a chart using the **Chart Type** property. For example, set it to **Clustered Column**:



6. Add series. Invoke the **Series Editor**, for example, by double-clicking the **Chart**.



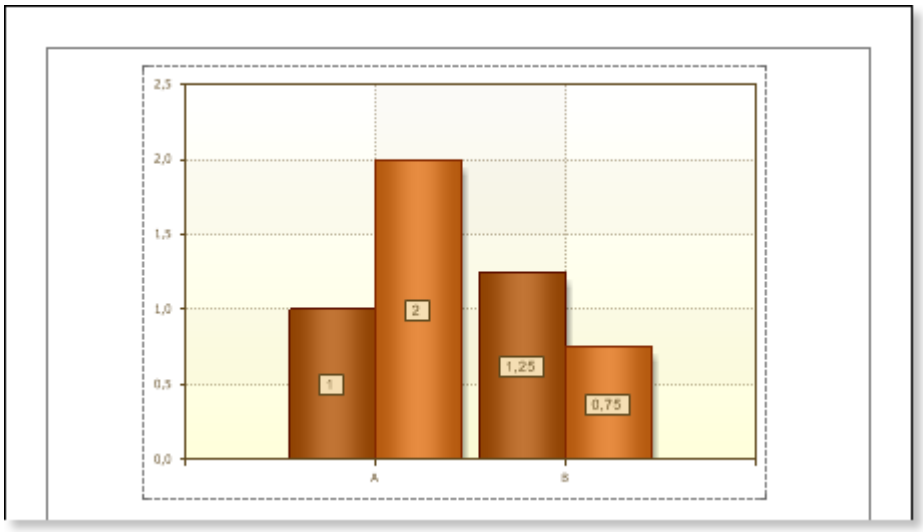
Click the **Add Series** button to add a series and select the type of series in the menu. The picture below shows the menu of the **Add Series** button:



It should be noted that the type of number should match the type of chart, i.e. if the **Clustered Column** chart type, then the series must be of the **Clustered Column** type.

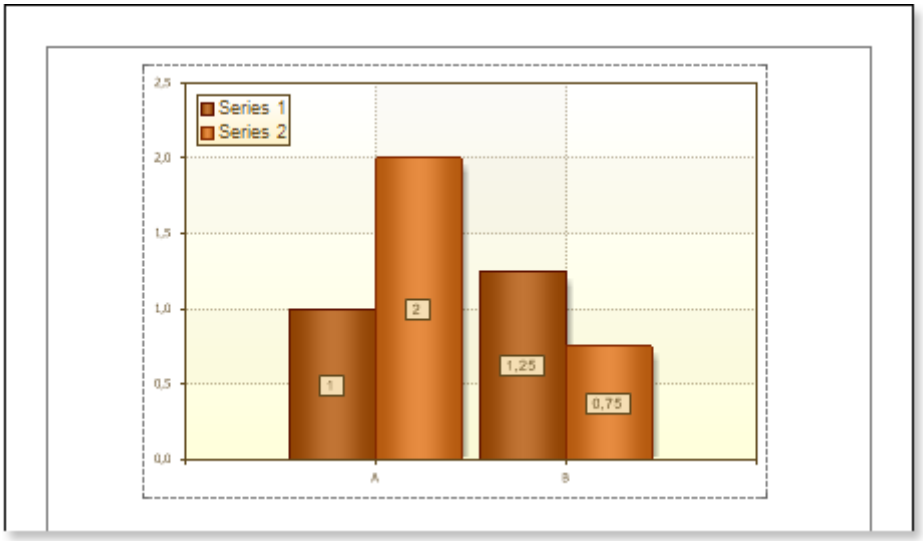
7. Setup chart series:
   7.1. Get the data for **Value** and for the **Argument** of series. There are three ways to get data for the series: set the column data from the dictionary, or specify an expression, or manually specify values for the series as a list, through the ',' separator. For example, create two rows, and manually define the values for these series as a list, with the ";" delimiter: arguments for **Series 1** - **A**; **B**, the values - **1**; **1.25**; for arguments **Series 2** - **A**; **B**, the value - **2**, **0.75**.
   7.2. Change the values of the series properties. For example, set the **Show Zeros** property to **false**, if it is necessary to hide zero values;
   7.3. Enable or disable **Series Labels**;
   7.4. Edit headers of rows: align, change the style, font, type of value, etc.;
   7.5. Change the design of series, by setting values of the following properties: **Border Color**, **Brush**, **Show Shadow**.

The picture below shows an example of a report template with the chart:



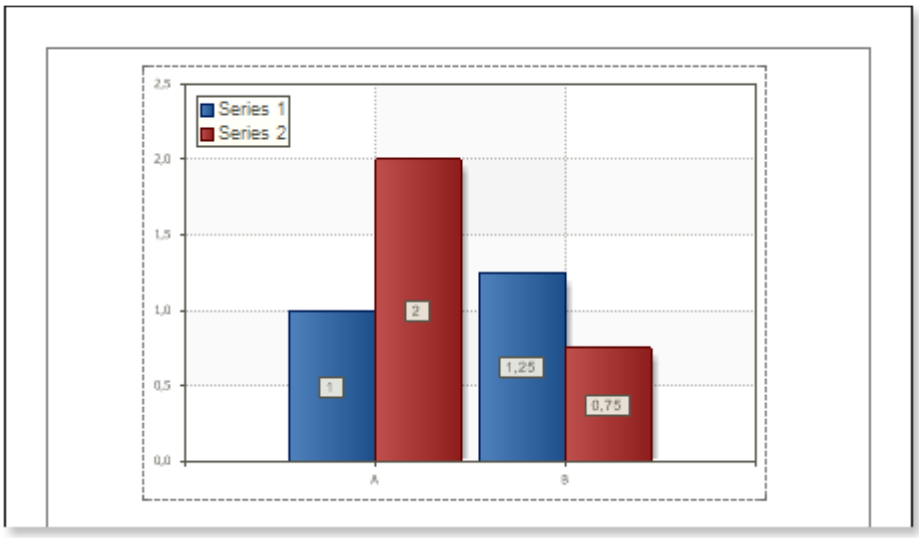
8. Edit **Legend**:
   8.1. Enable or disable the visibility of **Legends**. You can do it by setting the value of the **Legend.Visible** property to **true** or **false**, respectively;
   8.2. Align the legend horizontally and vertically;
   8.3. Change the legends design, etc.

The picture below shows an example of a report template with the chart displaying the legend:

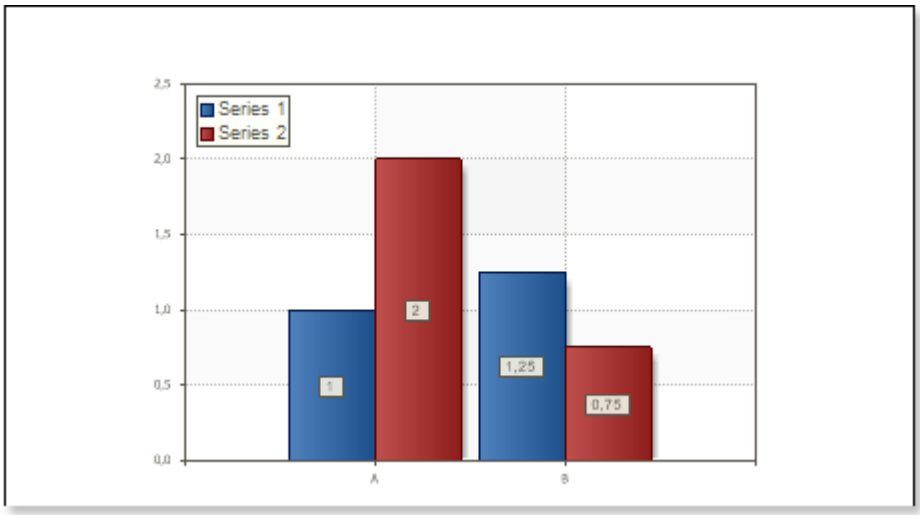


9. Change the style of the chart, i.e. completely change the appearance of the chart:
   9.1. Change the **Style** property. Where the value of the property is a chart style;
   9.2. Set the **AllowApplyStyle** to the **true**. If the **AllowApplyStyle** property is set to **false**, then the report generator, when rendering, will take into account the values of the appearance of the series.

The picture below shows an example of a report template of the chart with a changed style:



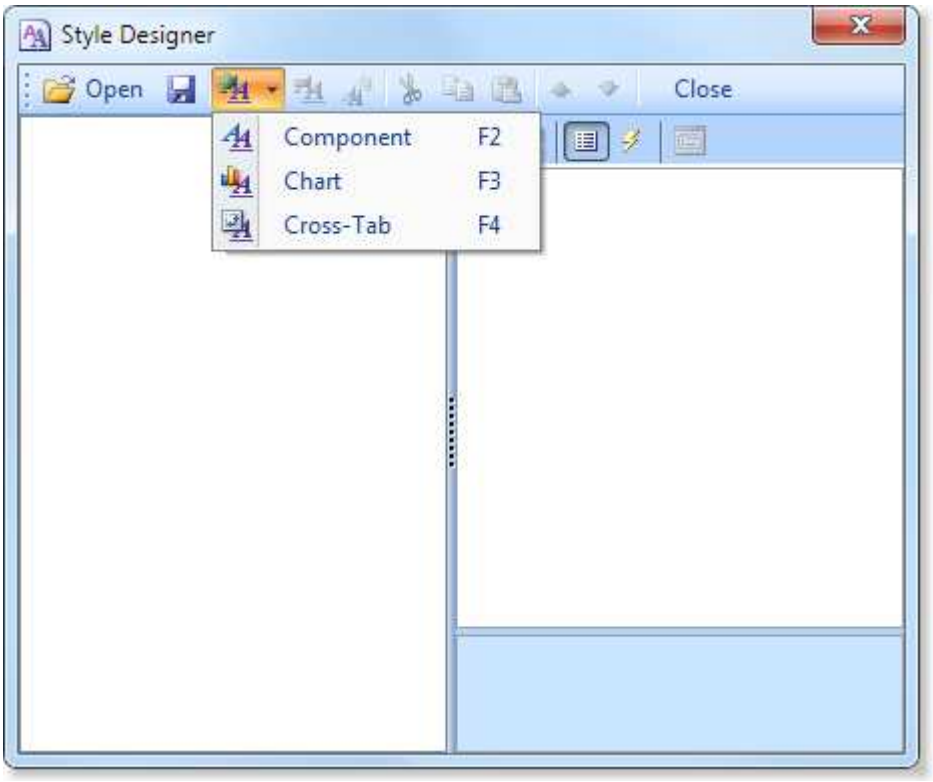
10. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows samples of reports with the chart:



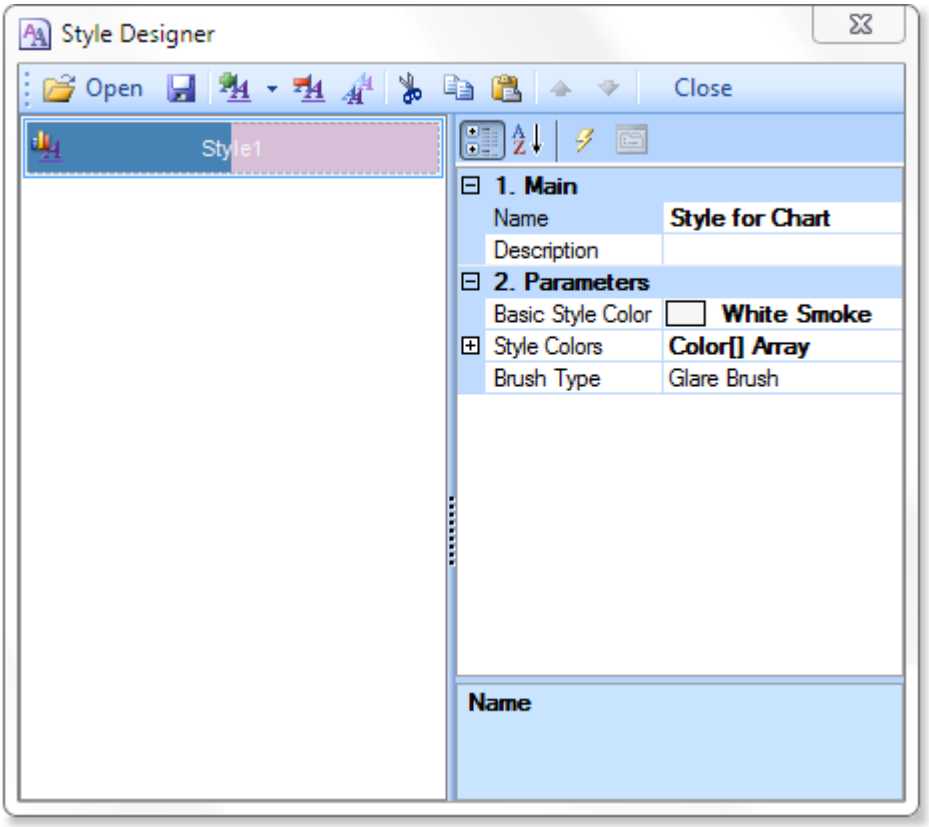
Adding styles

1. Go back to the report template;
2. Call the **Style Designer**;

The picture below shows the **Style Designer**:

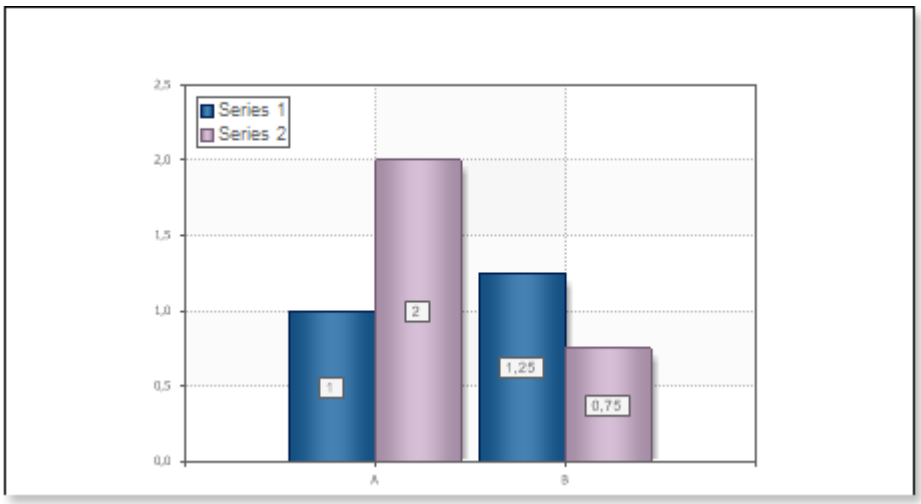


Click the **Add Style** button to start creating a style. Select **Chart** from the drop down list. Set the style using **Basic Color Style**, **Brush Type** and **Style Colors** group of properties.



Click **Close**. In the list of values of the **Style** property of the chart component a custom style will be displayed. In our case, the value is **Style for Chart**. Select this value;

3. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows samples of reports with the chart with a style applied:



Report with Chart in DataBand

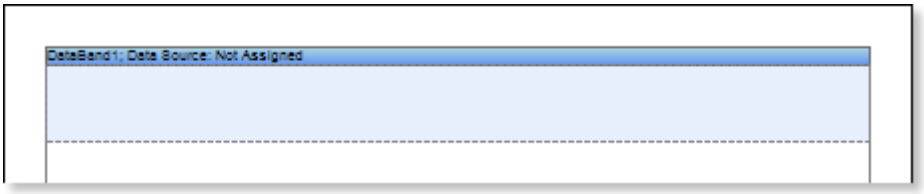
For better understanding this step-by-step instruction, please watch the video file.

Suppose a **Chart** component is placed on the page of the report, then, for a report, this component will be rendered as a page item. If the **Chart** component is placed in the **DataBand**, then, when rendering a report, this component will be rendered as part of the **DataBand**. Since the **Chart** component placed in the **DataBand**, is rendered as a part of the

DataBand, and will be printed as many times as the **DataBand** will be output. An example of designing a report with a chart in the **DataBand** will be described below. In this example, the chart will graphically display the detailed data of the data source in the **DataBand**. Follow the steps below in order to render a report with the **Chart** in the **DataBand**:

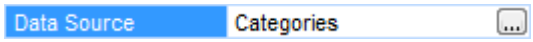
1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

3. Create a **Relation** between data sources. In this case, the **Parent Data Source** is the **Categories** data source, and the **Child Data Source** is the **Products** data source;
4. Put the **DataBand** on a report template page:

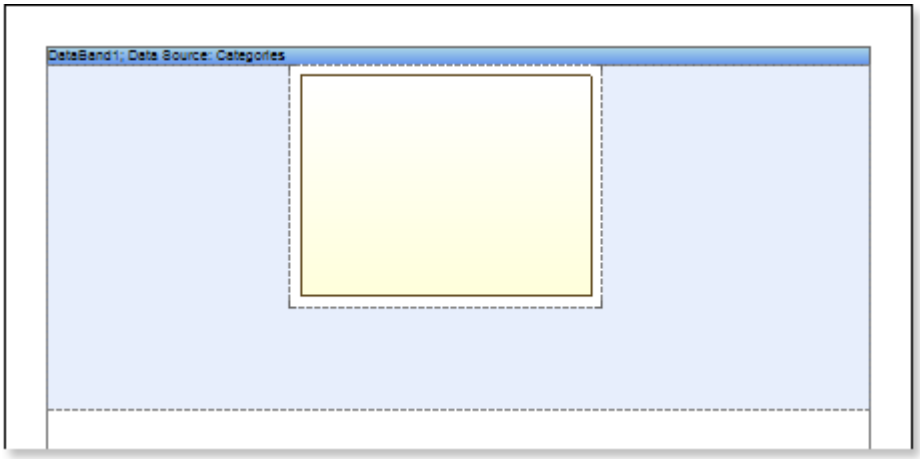


5. Edit **DataBand**:
   5.1. Align the **DataBand** by height;
   5.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
   5.3. Change the **DataBand** background;
   5.4. Enable **Borders** for the **DataBand**, if required;
   5.5. Change the border color.

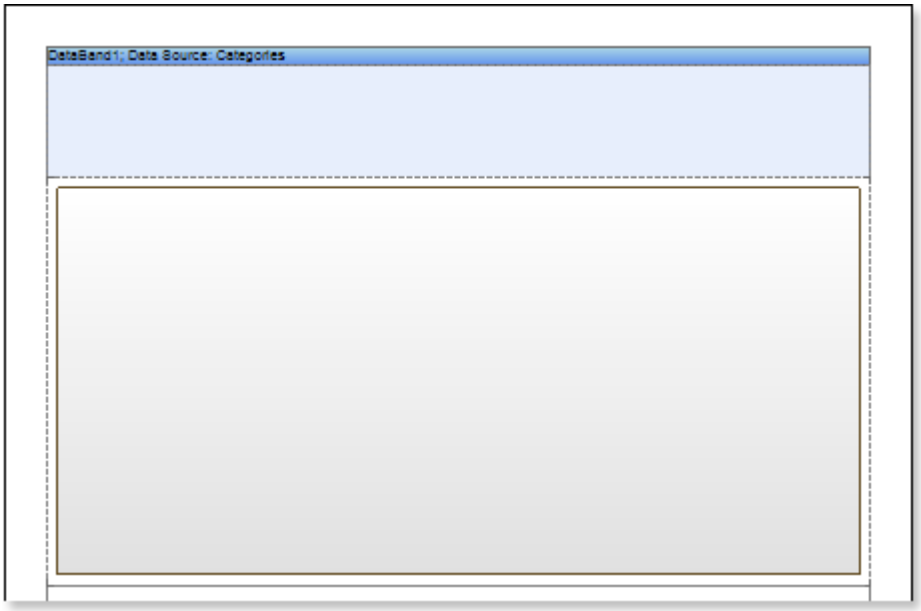
6. Define the data source for the **DataBand** using the **Data Source** property:



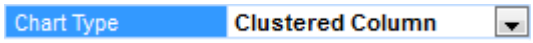
7. Put the **Chart** component in the **DataBand** as seen on a picture below:



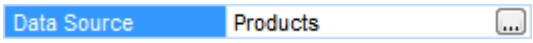
8. Edit the **Chart** component:
   8.1. Align it by width;
   8.2. Change properties of the **Chart** component. For example, set the **GrowToHeight** property to **true**, if it is required the Chart component be grown by height;
   8.3. Set **Borders**, if required, for the **Chart** component;
   8.4. Change the border color.
   8.5. Edit the chart area. For example, change the **Area.Brush.Color** property, if it is required to change the color of a chart area.



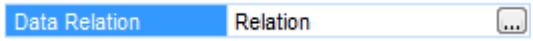
9. Change the type of a chart using the **Chart Type** property. For example, set it to **Clustered Column**:



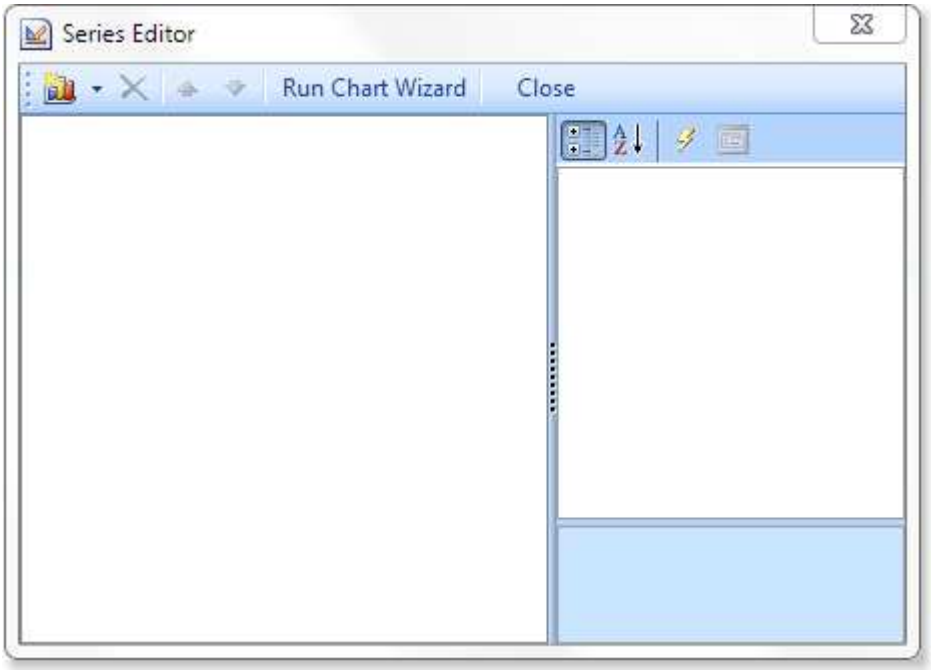
10. Define the data source for the **Chart** component using the **Data Source** property



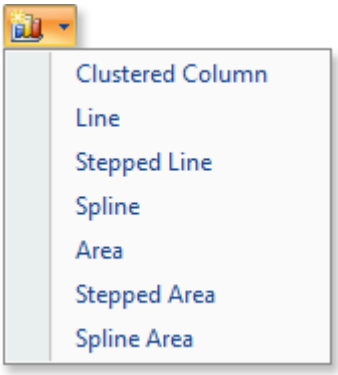
11. Define the relation between data sources, using the **DataRelation** property of the **Chart** component:



12. Add series. Invoke the **Series Editor**, for example, by double-clicking the **Chart**:



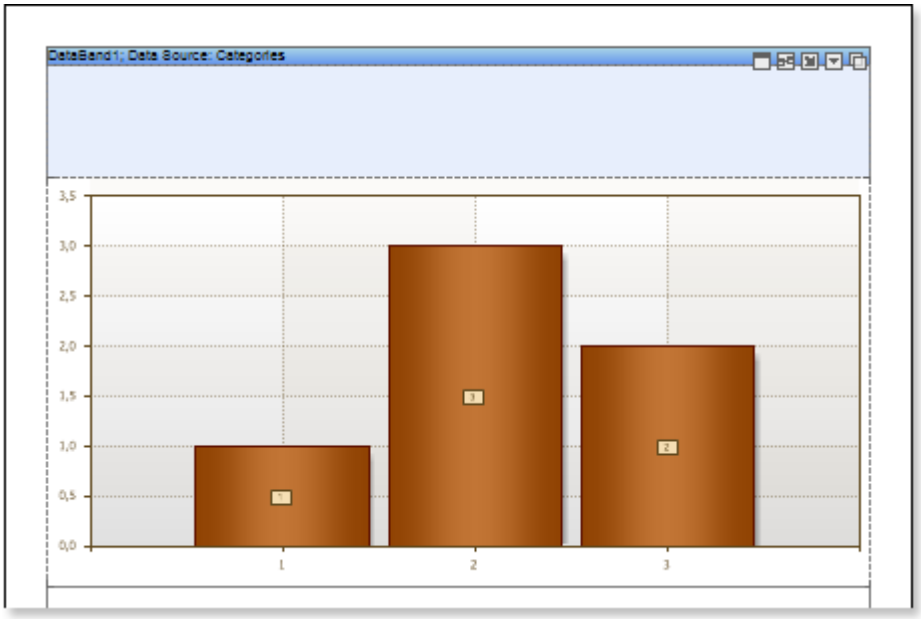
Click the **Add Series** button to add a series and select the type of series in the menu. The picture below shows the menu of the **Add Series** button:



It should be noted that the type of number should match the type of chart, i.e. if the **Clustered Column** chart type, then the series must be of the **Clustered Column** type.

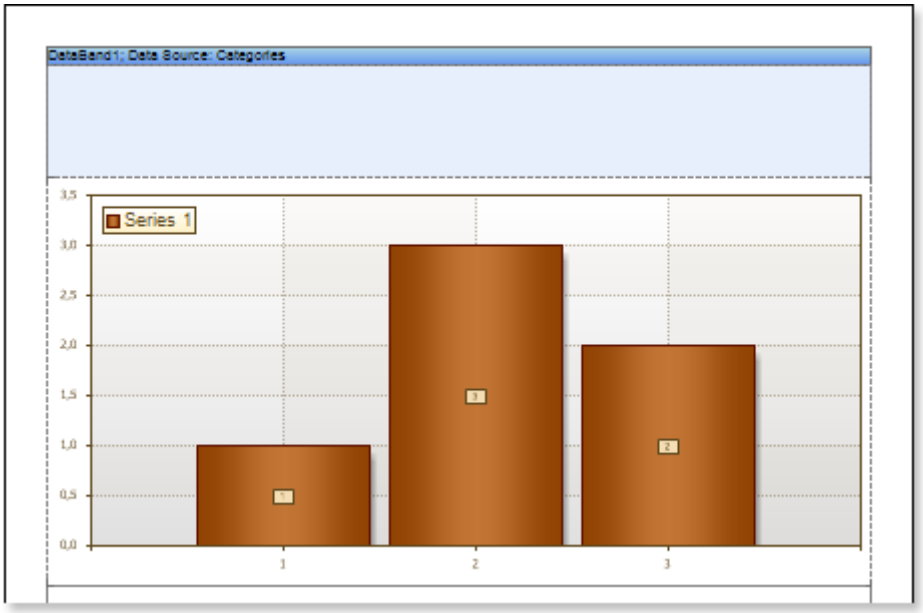
13. Setup chart series:
    13.1. Get the data for **Value** and for the **Argument** of series. There are three ways to get data for the series: set the column data from the dictionary, or specify an expression, or manually specify values for the series as a list, through the ',' separator. For example, create a series and specify columns from the dictionary: define the **Products.ProductName** for the **Argument** and **Products.UnitPrice** for the **Value**;
    13.2. Change the values of the series properties. For example, set the **Show Zeros** property to **false**, if it is necessary to hide zero values;
    13.3. Enable or disable **Series Labels**;
    13.4. Edit headers of rows: align, change the style, font, type of value, etc.;
    13.5. Change the design of series, by setting values of the following properties: **Border Color**, **Brush**, **Show Shadow**.

The picture below shows an example of a report template with the chart:



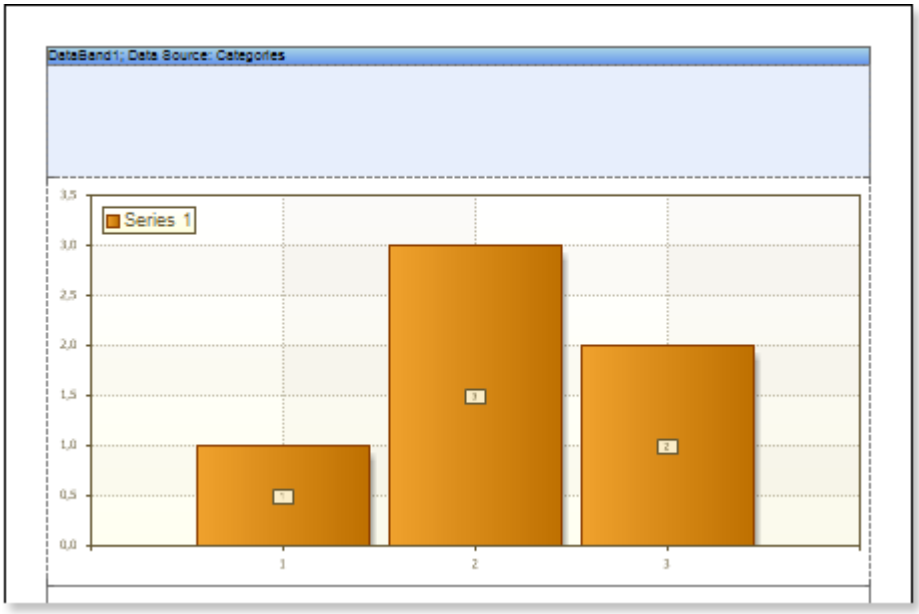
14. Edit **Legend**:
    14.1. Enable or disable the visibility of **Legends**. You can do it by setting the value of the **Legend.Visible** property to **true** or **false**, respectively;
    14.2. Align the legend horizontally and vertically;
    14.3. Change the legends design, etc.

The picture below shows an example of a report template with the chart displaying the legend:

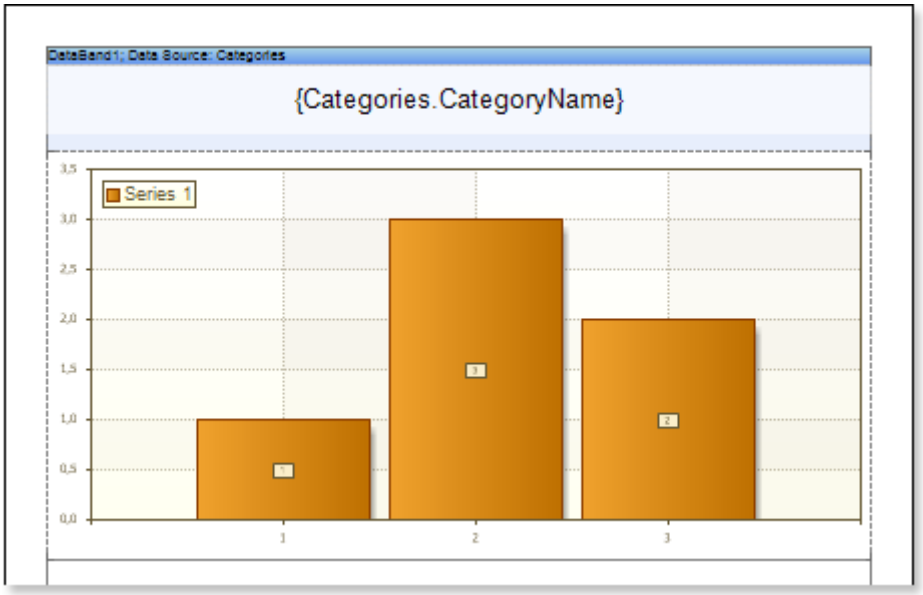


15. Change the style of the chart, i.e. completely change the appearance of the chart:
    15.1. Change the **Style** property. Where the value of the property is a chart style;
    15.2. Set the **AllowApplyStyle** to the **true**. If the **AllowApplyStyle** property is set to **false**, then the report generator, when rendering, will take into account the values of the appearance of the series.

The picture below shows an example of a report template of the chart with a changed style:

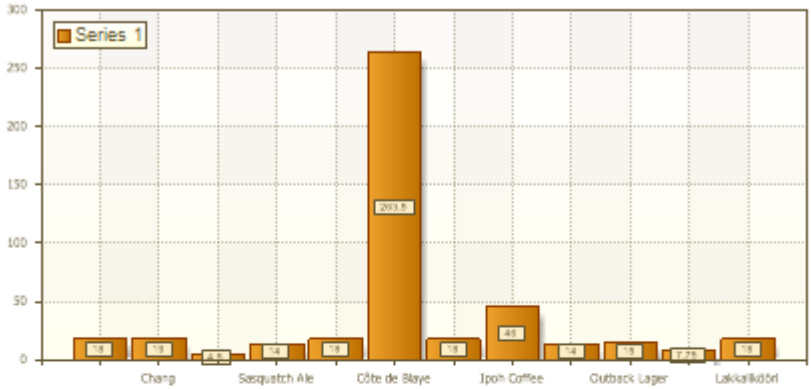


16. Put text components with an expression in the **DataBand**. Where the expression is a reference to the data field. For example, put a text component with the expression: **{Categories.CategoryName}**;
17. Edit **Text** and **TextBox** component:
    17.1. Drag and drop the text component in the **DataBand**;
    17.2. Change parameters of the text font: size, type, color;
    17.3. Align the text component by width and height;
    17.4. Change the background of the text component;
    17.5. Align text in the text component;
    17.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
    17.7. Enable **Borders** for the text component, if required.
    17.8. Change the border color.

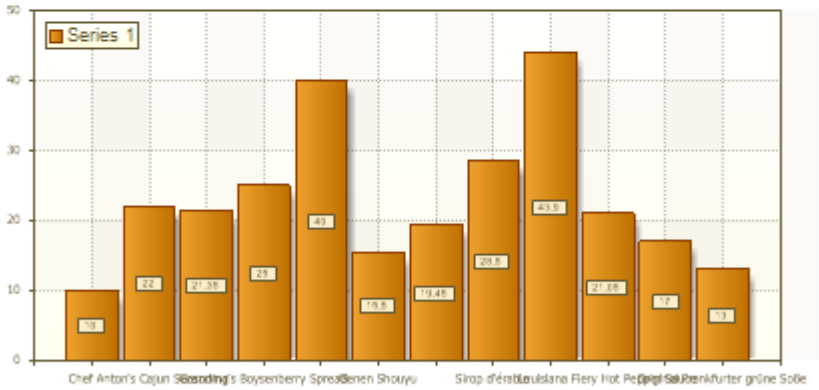


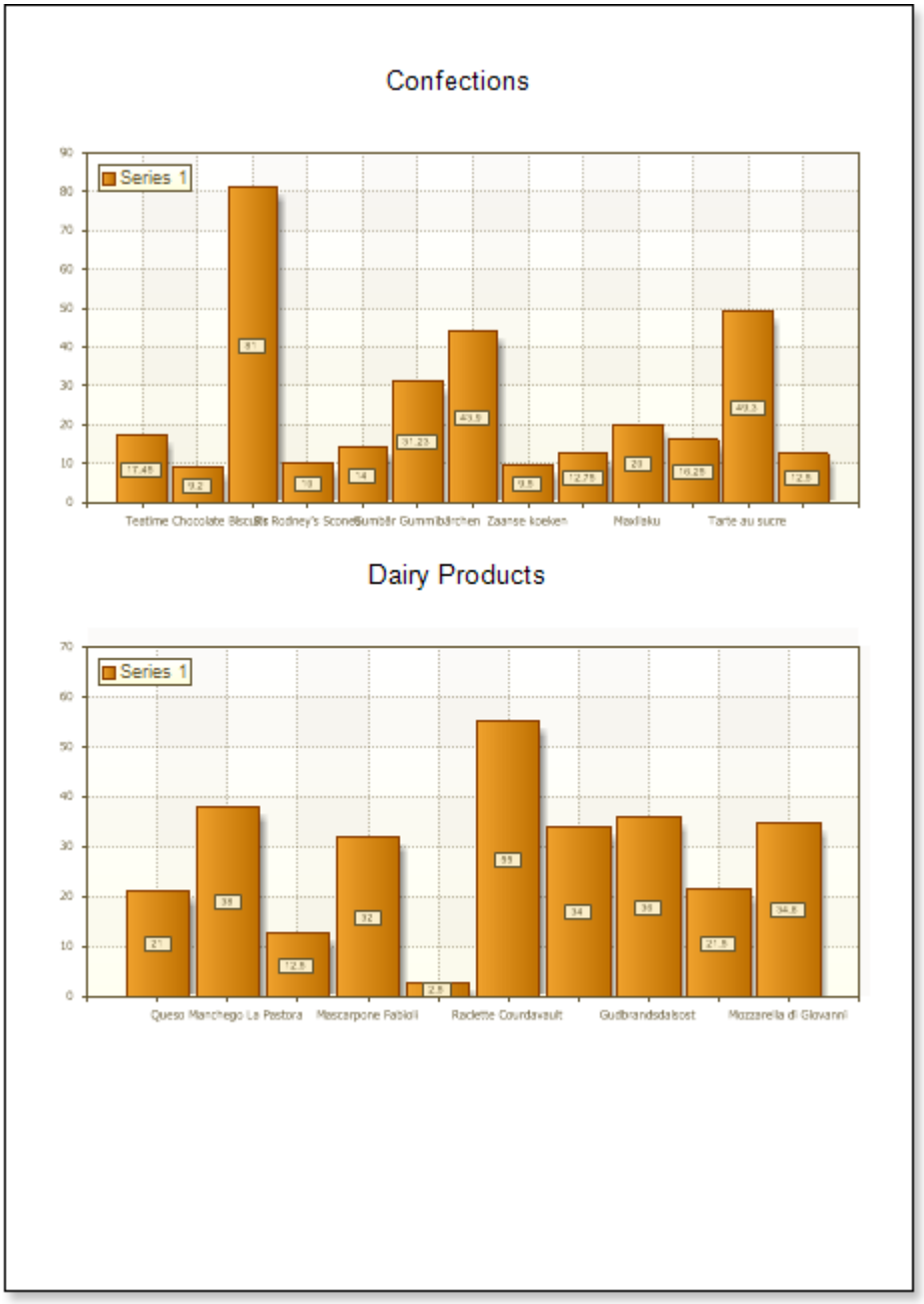
18. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of the report with the chart in the **DataBand**:

Beverages



Condiments

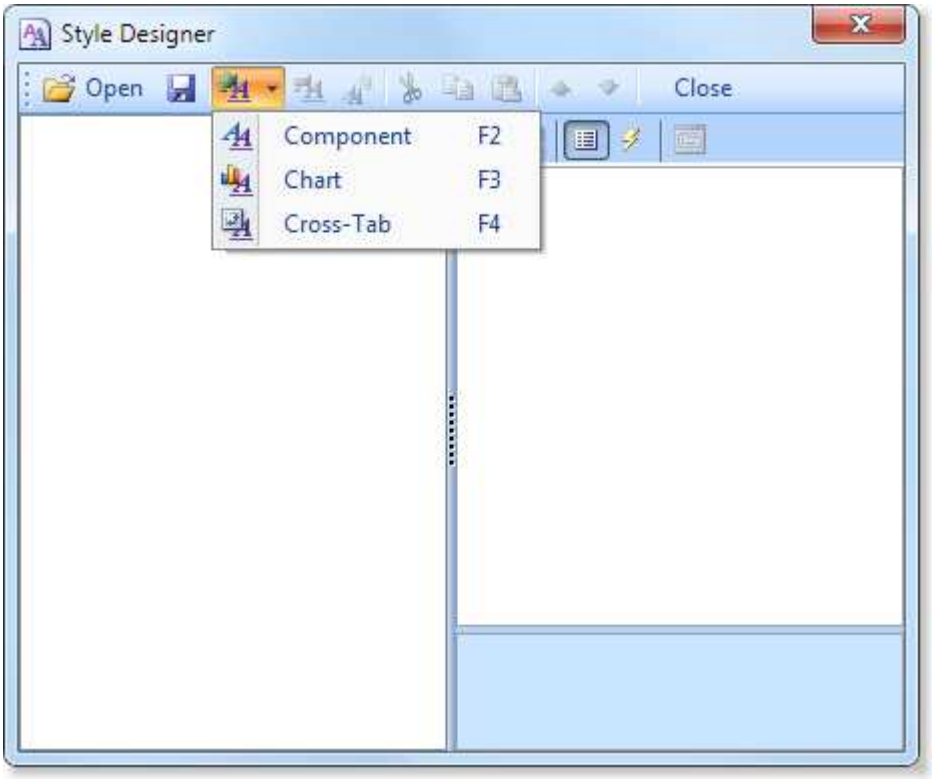




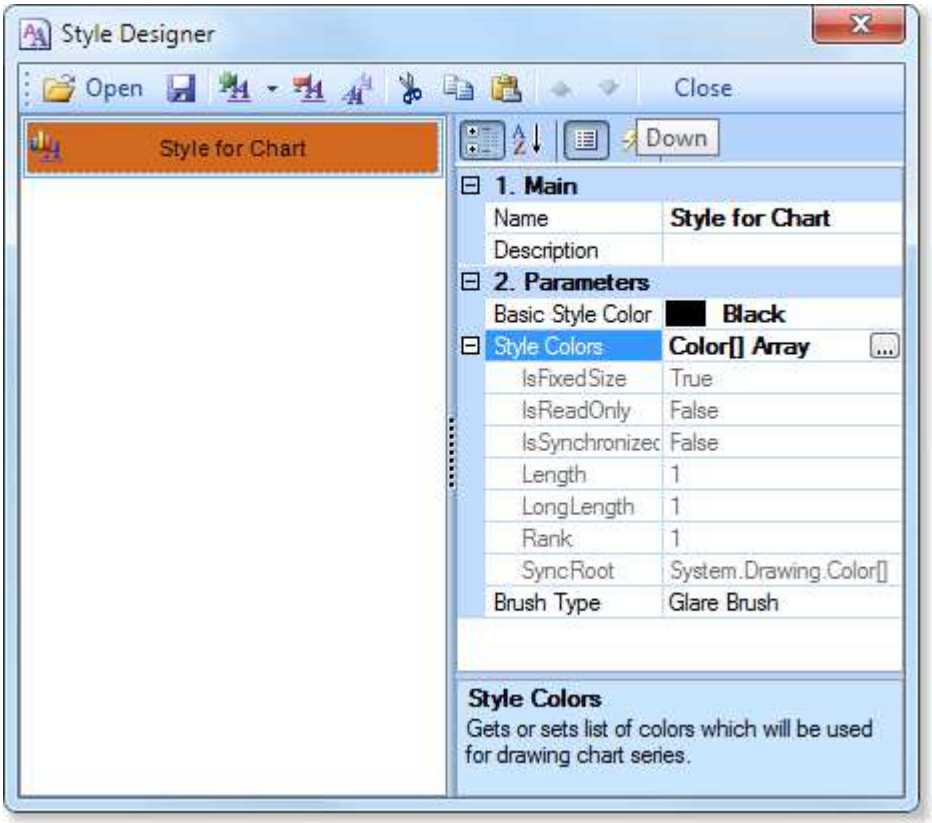
Adding styles

1. Go back to the report template;
2. Call the **Style Designer**;

The picture below shows the **Style Designer**:



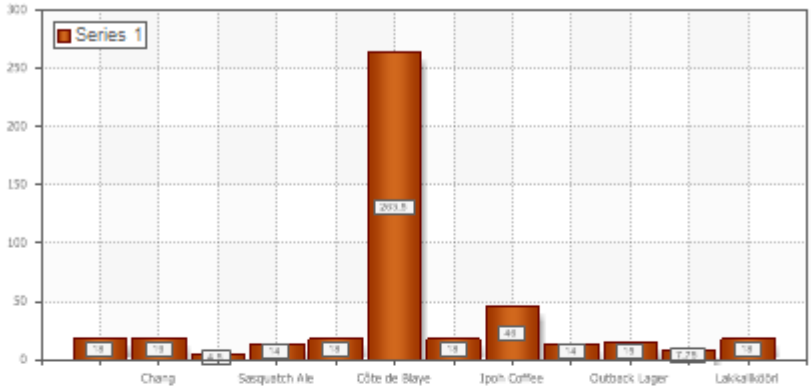
Click the **Add Style** button to start creating a style. Select **Chart** from the drop down list. Set the style using **Basic Color Style**, **Brush Type** and **Style Colors** group of properties.



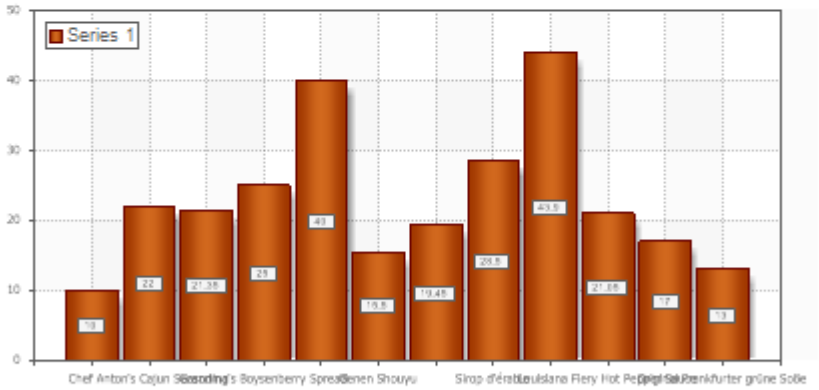
Click **Close**. In the list of values of the **Style** property of the chart component a custom style will be displayed. In our case, the value is **Style for Chart**. Select this value;

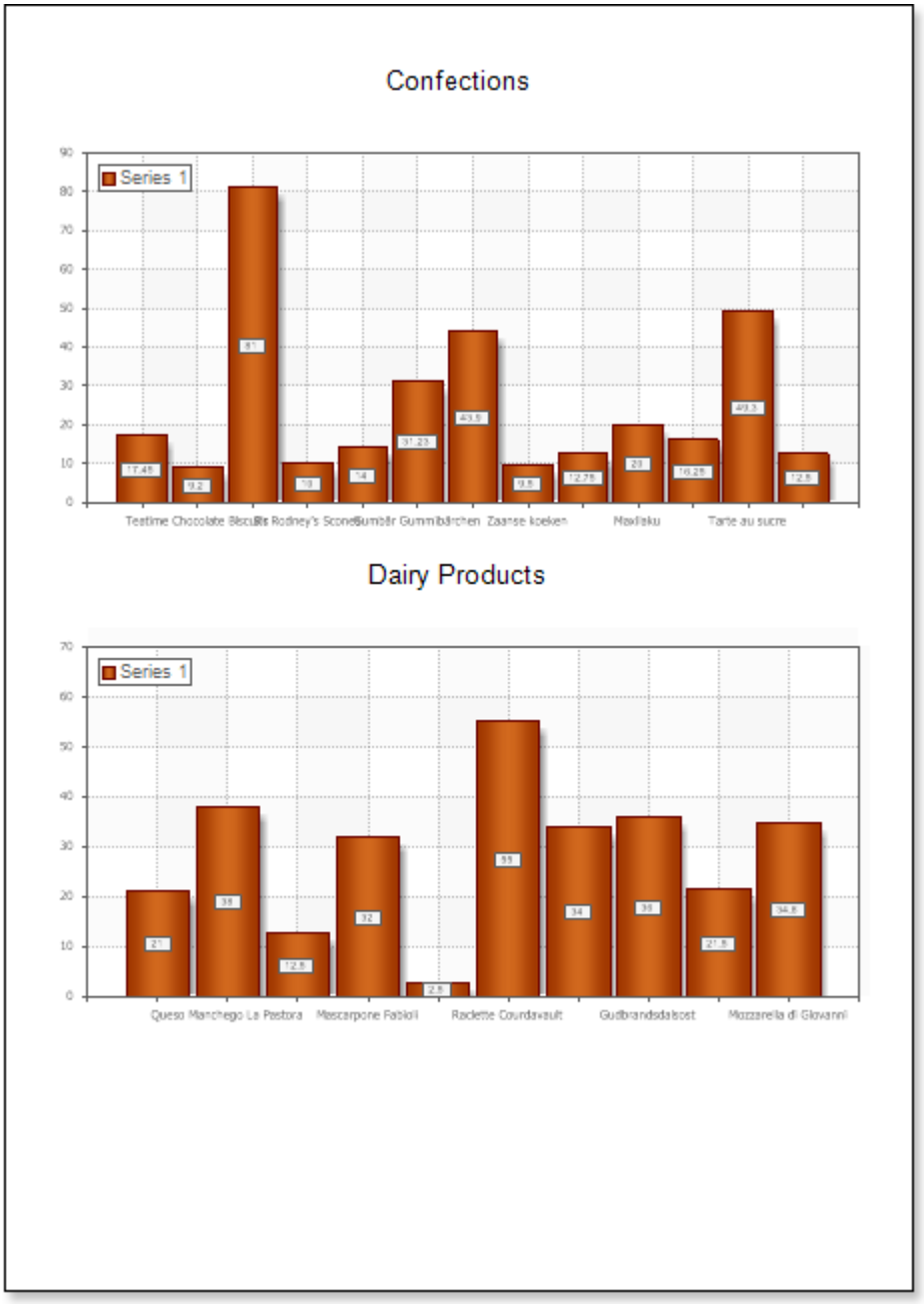
3. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows samples of reports with the chart with a style applied:

Beverages



Condiments





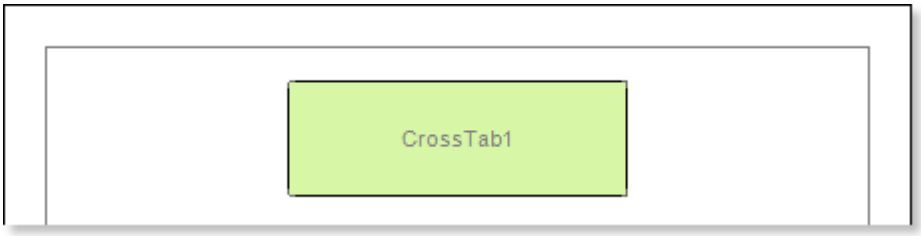
Report with Cross-Tab on Page

For better understanding this step-by-step instruction, please watch the video file.

Do the following steps to create a report with the cross table:

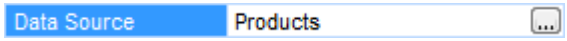
1. Run the designer;
2. Connect data:
    2.1. Create **New Connection**;
    2.2. Create **New Data Source**;

3. Put the **Cross-Tab** component on a page of the report template.

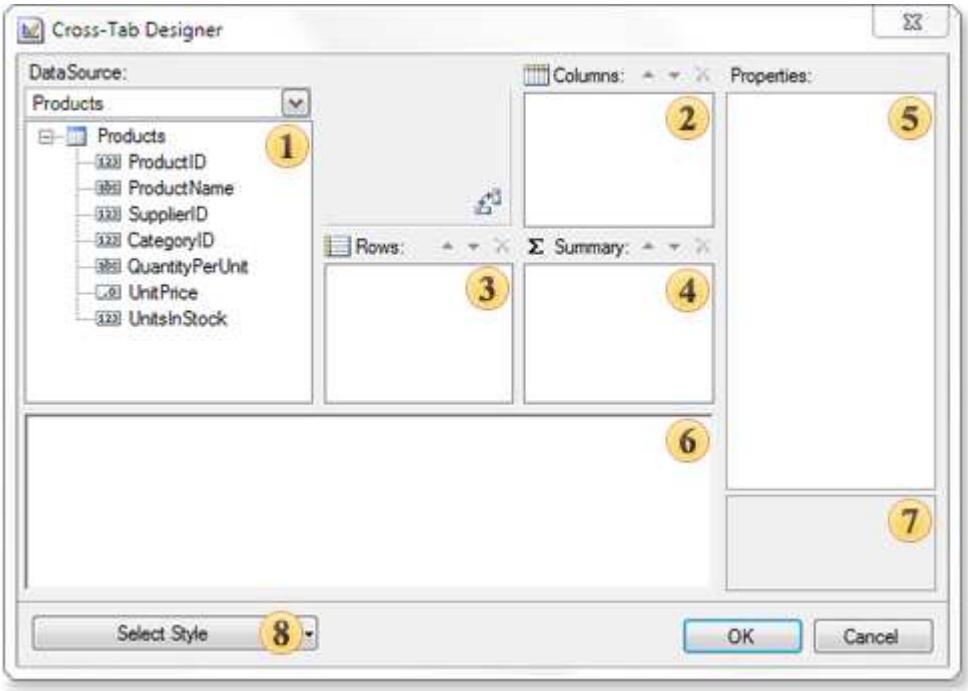


4. Edit the **Cross-Tab** component:
   4.1. For example, set the **GrowToHeight** property to **true**, to allow the **Cross-Tab** component to grow by height;

5. Define the data source for the **Cross-Tab** component of the band, for example, using the **Data Source** property:



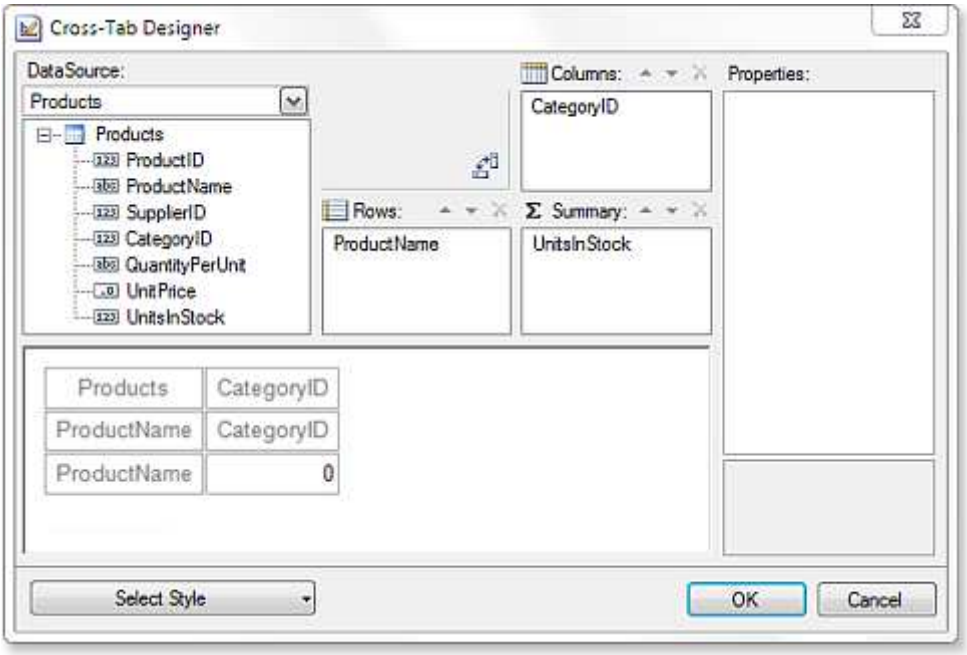
6. Invoke the **Cross-Tab Designer**, for example, clicking the **Design...** item of the context menu of the cross table component. The picture below shows the **Cross-Tab Designer** window:



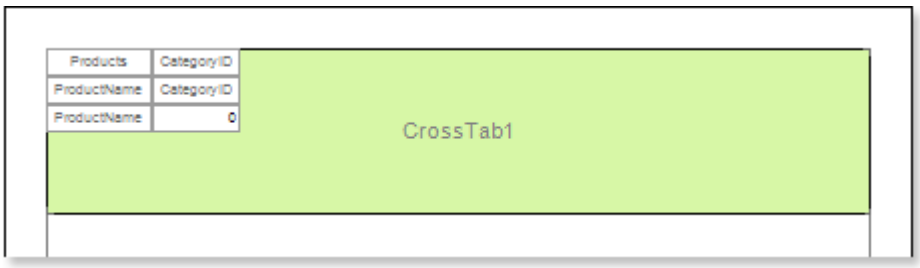
1 The **DataSource** field shows the data columns of the selected data source;
2 The **Columns** field shows a list of columns of the data source by what the columns in the cross table will be created;
3 The **Rows** field shows a list of rows of the data source by what the rows in the cross table will be created;
4 The **Summary** field shows a list of columns of the data source by what the summary in the cross table will be created;
5 The **Properties** field shows the properties of the selected item of the cross table;
6 The **Cross-Tab Cells** field shows cells of the cross table;
7 The **Description** field shows a brief description of the selected property of the cross table item;
8 The **Select Style** button. When clicking the drop down list of styles for the cross table appear.

7. Do the following steps in the **Cross-Tab Designer**:
   7.1. Add the data column from the 1 **DataSource** to the 2 **Columns** field of the cross-tab. For example, add the **CategoryID** data column to the **Columns** field of the cross-tab. Hence one entry from this data column will correspond to one column in the rendered cross-table, i.e. the number of entries in this data column will be equal to the number of columns in the cross-table;
   7.2. Add a column of the data source from 1 the **DataSource** field to 3 the **Rows** of the cross-table. For example, add the **ProductName** data column to the **Rows** field of the cross-table, and then one entry from this data column will correspond to one row in the rendered cross-table, i.e. the number of entries in this data column will be equal to the number of rows in the cross-table;

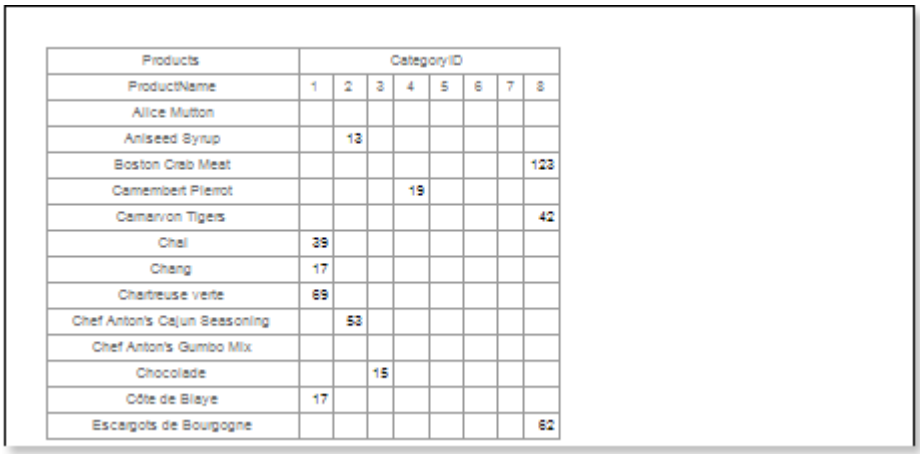
7.3. Add a data column from 1 the **DataSource** field to the 4 **Summary** field of the cross-table. For example, add the **UnitInStock** data column to the **Summary** field of the cross-table, i.e. all entries in this data column will be summary entries in the cross-table;



8. Press the **OK** button in order to save your changes and go back to the report template with cross-table.



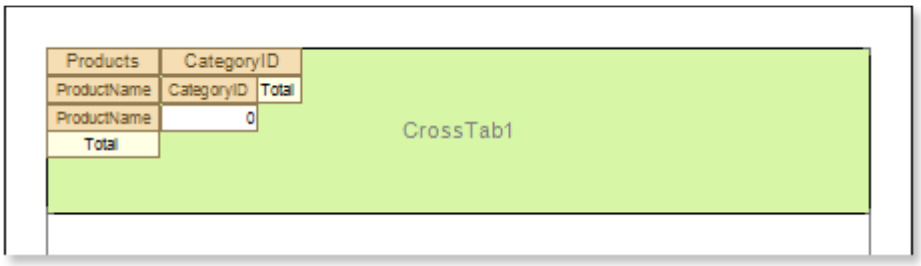
9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a rendered cross-tab report:



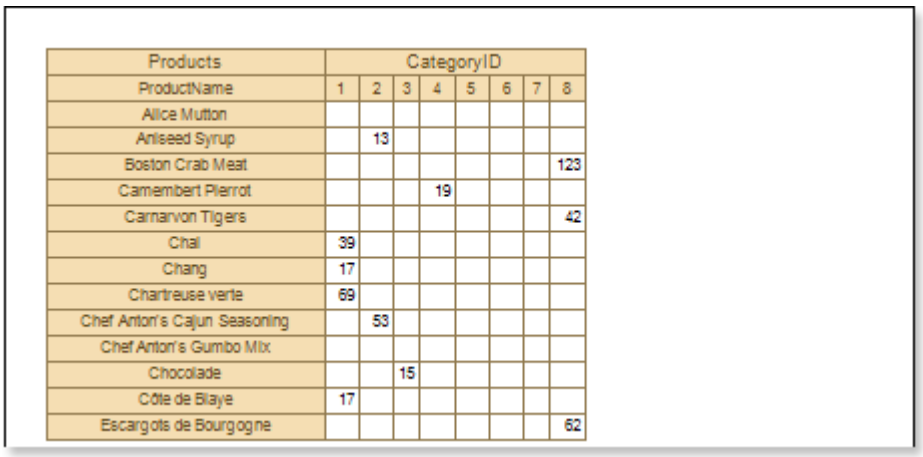
Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123
Camembert Pierrot				19				
Camarvon Tigers								42
Chai	39							
Chang	17							
Chartreuse verte	69							
Chef Anton's Cajun Seasoning		53						
Chef Anton's Gumbo Mix								
Chocolade			15					
Côte de Blaye	17							
Escargots de Bourgogne								62

10. Go back to the report template;
11. Edit cells in the report template:
    11.1. Set the font settings: type, style, size;
    11.2. Set the background of cells;
    11.3. Set the **Word Wrap** property to **true** if it is necessary to wrap text;
    11.4. Switch on/off **Borders**;

11.5. Set the border color;
11.6. Set the background of cells etc.



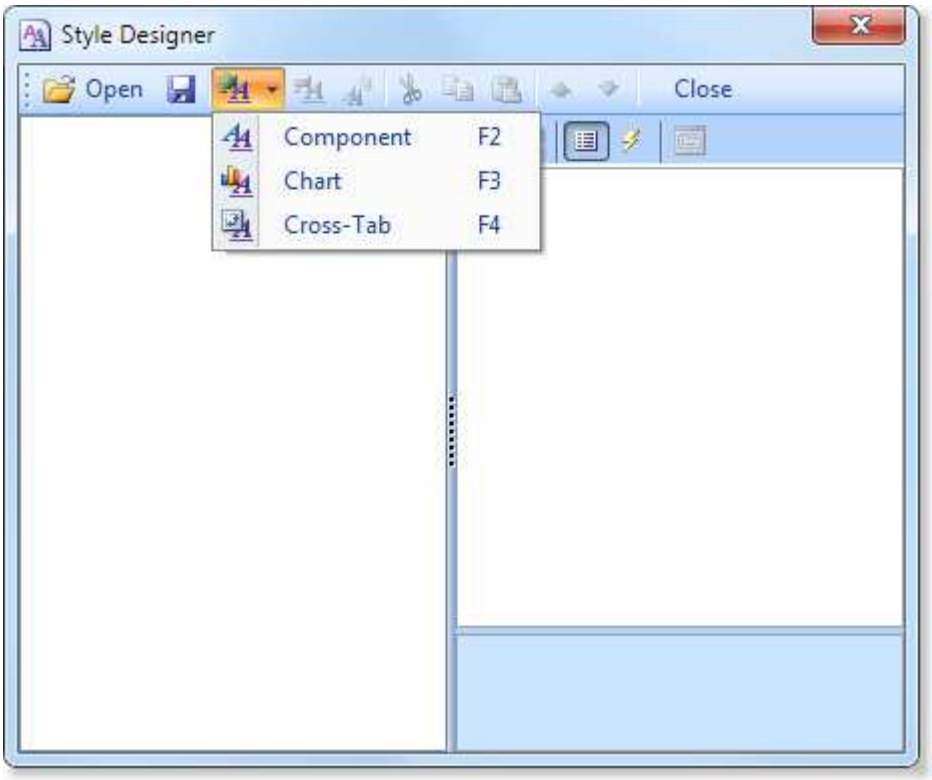
12. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a report of the rendered report with the cross table after editing report template cells:



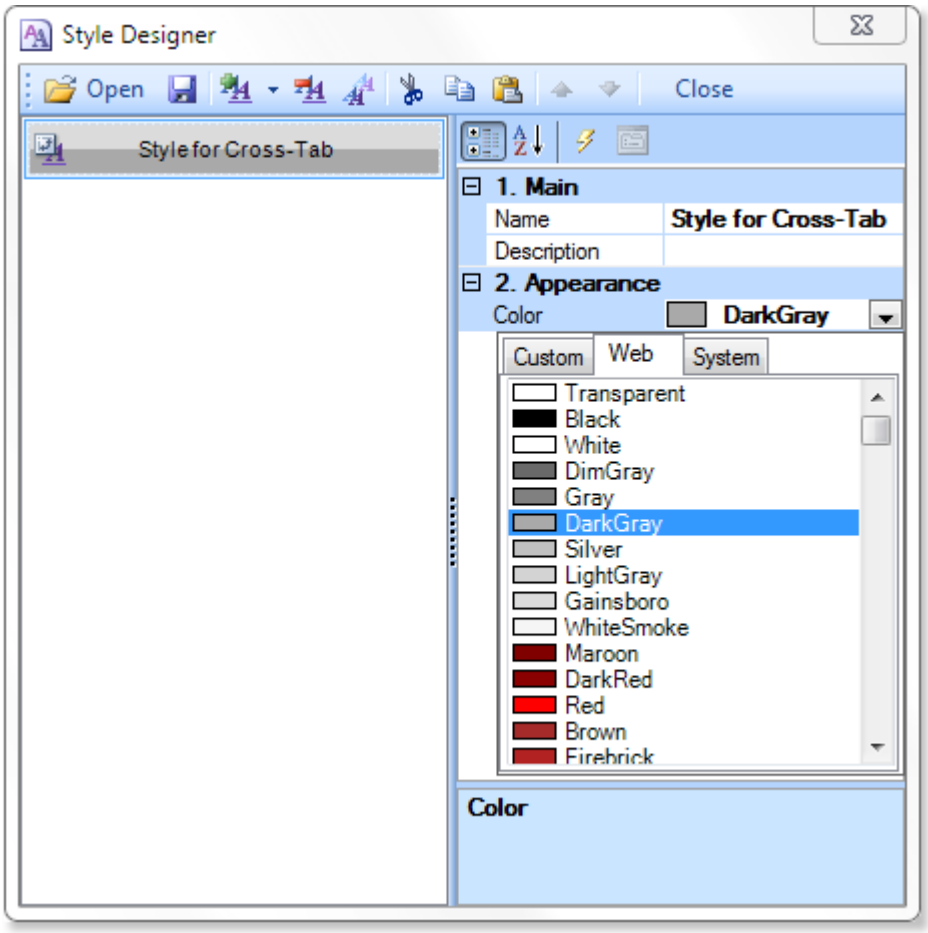
Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123
Camembert Pierrot				19				
Carnarvon Tigers								42
Chai	39							
Chang	17							
Chartreuse verte	69							
Chef Anton's Cajun Seasoning		53						
Chef Anton's Gumbo Mix								
Chocolade			15					
Côte de Blaye	17							
Escargots de Bourgogne								62

Adding styles

1. Go back to the report template;
2. Call the **Style Designer**;
The picture below shows the **Style Designer**:

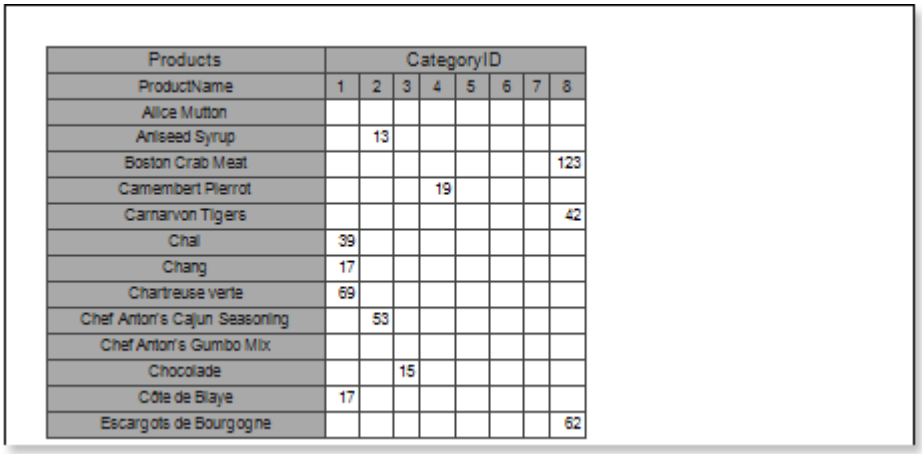


Click the **Add Style** button to start creating a style. Select **Cross-Tab** from the drop down list. To create the custom style, set the **Color** property. The picture below shows a sample of the **Style Designer** with created custom style**:**



Click **Close**. In the list of values of the **Select Style** button in the cross-table editor, a custom style will be displayed. In our case, the name is **Style for Cross-Tab**. Select this value;

3. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of the rendered cross-table report using the custom style:



Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123
Camembert Pierrot				19				
Carnarvon Tigers								42
Chai	39							
Chang	17							
Chartreuse verte	69							
Chef Anton's Cajun Seasoning		53						
Chef Anton's Gumbo Mix								
Chocolade			15					
Côte de Blaye	17							
Escargots de Bourgogne								62

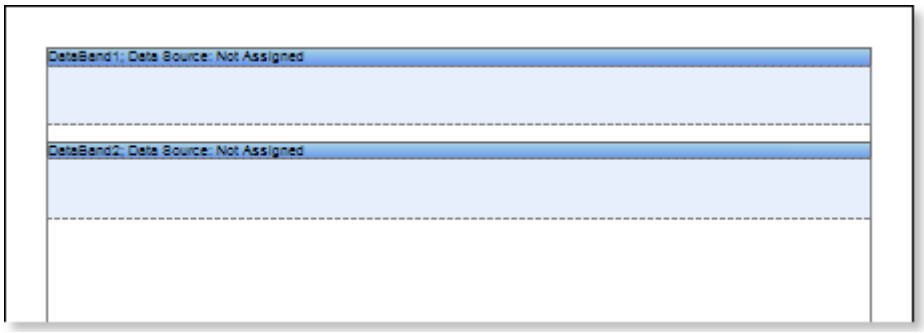
Cross-Tab Report in DataBand

For better understanding this step-by-step instruction, please watch the video file.

If the **Cross-Tab** component is placed in the **DataBand**, then when designing a report, this component will be constructed as part of the **DataBand**. Because the **Cross-Tab** component placed in the **DataBand** is designed as an element of the **DataBand**, then, when designing a report, this component will be printed as many times as the **DataBand**. Consider an example of building a report with the **Cross-Tab** in the **DataBand**. In this example, **Cross-Tab** will display the detailed entries in the **Master-Detail** report. Do the following steps in order to build a report with the **Cross-Tab** in the **DataBand**:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

3. Create the **Relation** between data sources. If the **Relation** is not created and/or the **Relation** property will be not filled for the **Detail** data source, then, for each **Master** entries, all **Detail** entries will not be output;
4. Put two **DataBands** on a page of a report template;



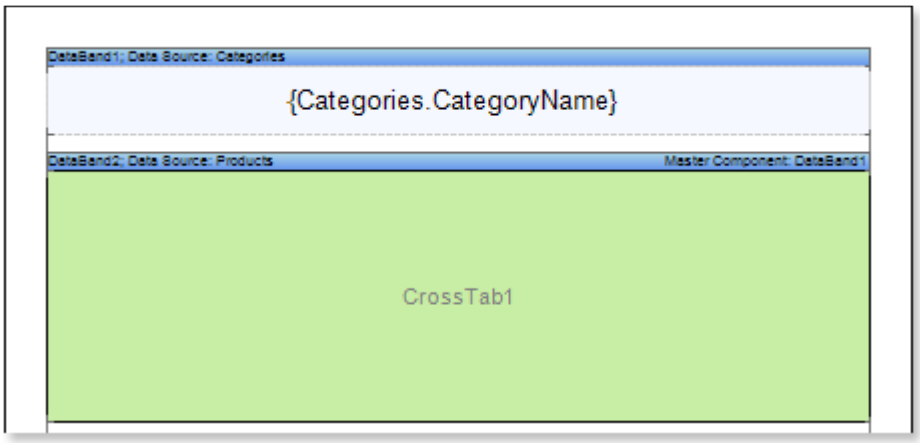
5. Edit **DataBand1** and **DataBand2**:
   5.1 Align the **DataBands** vertically;
   5.2 Change the value of the required properties. For example, for the **DataBand1**, which is a **Master** component in the **Master-Detail** report, set the **Print If Detail Empty** property to **true**, if you want the **Master** entries be printed in any case, even if the **Detail** entries are not available. And for the **DataBand2**, which is a **Detail** component in the **Master-Detail** report, set the **CanShrink** property to **true**, if it is necessary for this band to be shrunk;
   5.3 Change the background color of the **DataBand**;
   5.4 If necessary, set **Borders** of the **DataBand**;

6. Specify data sources for **DataBands**, as well as assign the **Master** component. In our example, the **Master** component is the upper **DataBand1**, and hence indicate the **DataBand1** in the **Master Component** tab of the **Data Setup** dialog box of the lower **DataBand2** as the **Master** component;
7. Fill in the **Data Relation** property of the **DataBand**, which is the **Detail** component, in our case, this is the **DataBand2**:

Data Relation	Categories

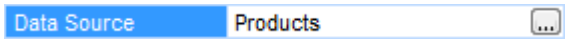
8. Put the text component with an expression. Where the expression is a reference to the data field. For example: the **DataBand1**, that is the **Master** component, put the text component with the **{Categories.CategoryName}** expression;
9. Edit text and text components located in the **DataBand**:
   9.1. Drag the text component to the required place in the **DataBand**;
   9.2. Align the text in a text component;
   9.3. Change the value of the required properties. For example to set the **Word Wrap** property to **true**, if you want the text be wrapped;
   9.4. Set **Borders** of a text component, if required.
   9.5. Change the border color.

10. Put the **Cross-Tab** component in the **DataBand**. In this case, the **Cross-Tab** component will be located on the **DataBand2**, that is the **Detail** component of the report.

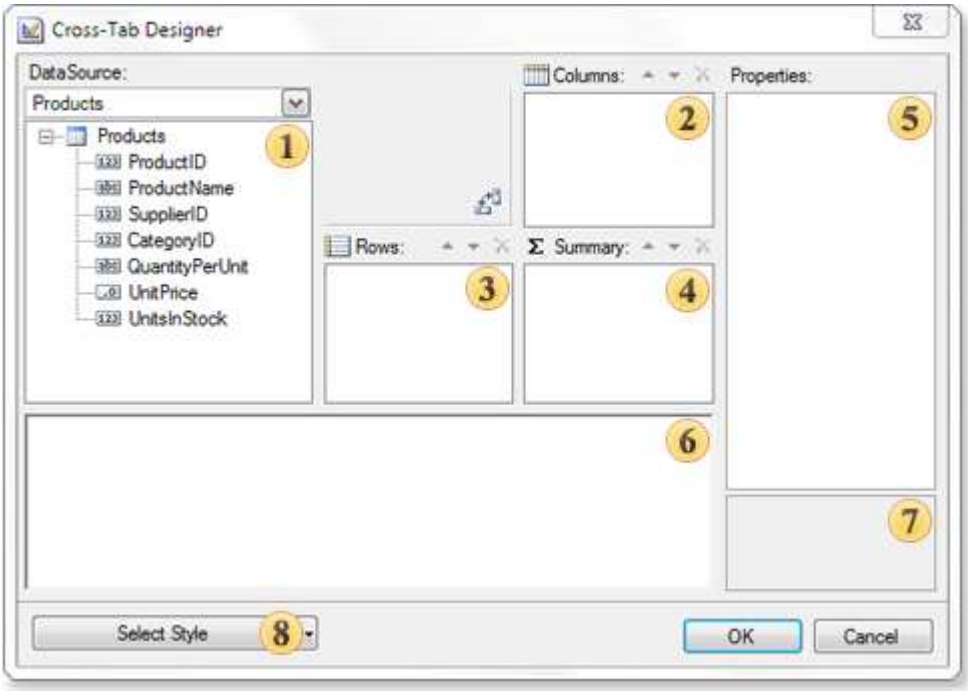


11. Edit the **Cross-Tab** component:
    11.1 Change values of the **Cross-Tab** properties. For example, set the **Can Shrink** property to **true**, if you want the **Cross-Tab** component be shrunk;

12. Specify the data source for the band of the **Cross-Tab** component, for example, using the **Data Source:**



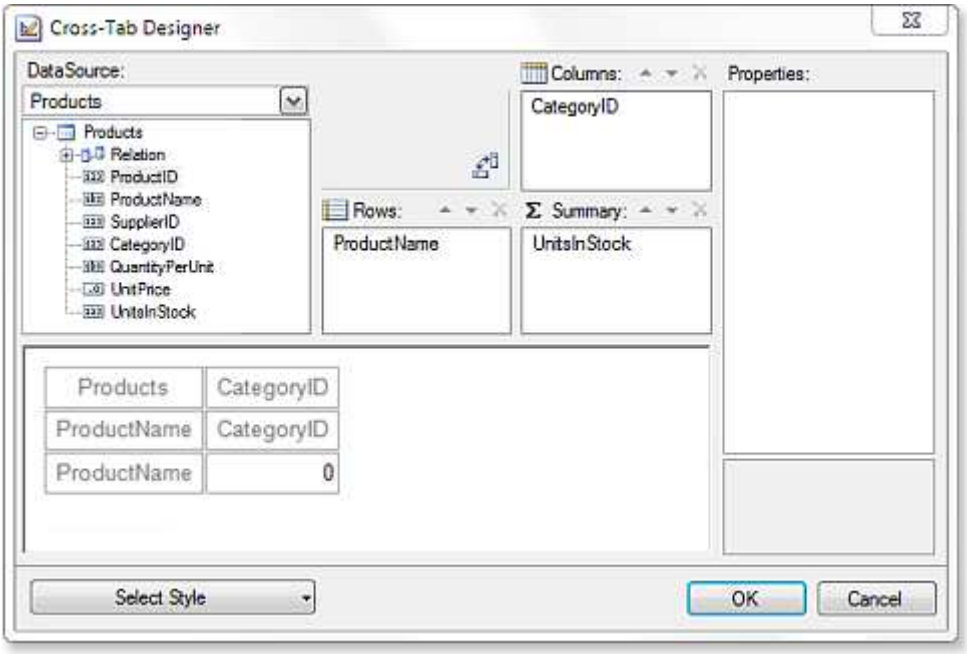
13. Call the **Cross-Tab Designer**, for example, by selecting **Edit ..** (**Design..**) of the context menu of the cross-table component.



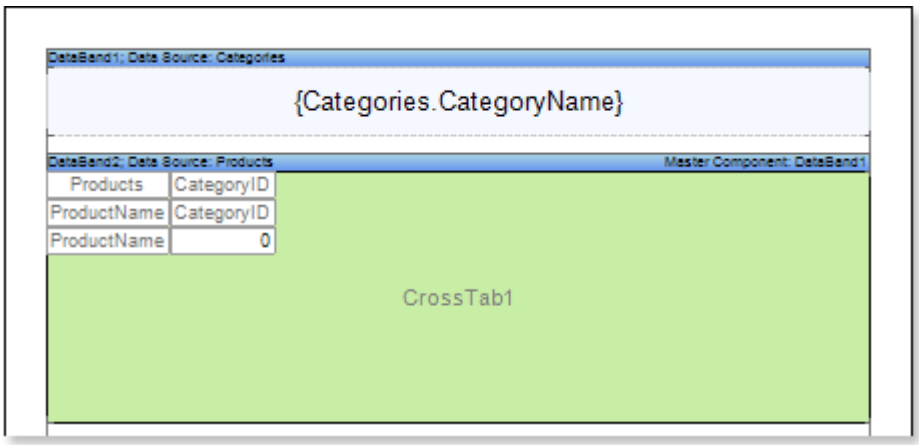
1 The **DataSource** field. This field displays data columns of the selected data source;
2 The **Columns** field. This field displays a list of columns of the data source for the entries by which columns in the cross-table will be formed;
3 The **Rows** field. This field displays a list of columns of the data source for the entries by which lines in the cross-table will be formed;
4 The **Summary** field. This field displays a list of columns of the data source for the entries by which summaries in the cross-table will be formed;
5 The **Properties** field. This field displays the properties of the selected element of cross-table;
6 The **Cross-Tab Cells** field. This field displays cells of the cross-table;
7 The **Description** field. This field displays a short description of the selected properties of the cross-table item;

8 The **Select Style** button. When you click, the drop-down list of styles appears for the cross-table.

14. Do the following in the **Cross-Tab Designer** editor:
    14.1. Add a data column from the 1 **DataSource** field to the 2 **Columns** field of the cross-table. Add a data column from the **DataSource** field to the **Columns** field of the cross-table. For example, add the **CategoryID** data column of data to the **Columns** field of the cross-table, and then one entry from this data column will correspond to one column in the rendered cross-table;
    14.2. Add a data column of the data source from the 1 **DataSource** field to the 3 **Rows** field of the cross-table. For example, add the **ProductName** data column to the **Rows** field of the cross-table, and then one entry from this data column will correspond to one row in the rendered cross-table, i.e. the number of entries in this data column will be equal to the number of rows in the cross-table;
    14.3. Add a data column from the 1 **DataSource** field to the 4 **Summary** field of the cross-table. For example, add the **UnitInStock** data column to the **Summary** field of the cross-table, i.e. entries in this data column will be summary entries in the cross-table;



15. Press the **OK** button in order to save your changes and go back to the report template with the cross-table.



16. Render a report. Click the **Preview** button or call the **Viewer** by pressing **F5** or select the **Preview** of the menu item. The picture below shows an example of the cross-table report:

Beverages

Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123
Camembert Pierrot				19				
Carnarvon Tigers								42
Chai	39							
Chang	17							
Chartreuse verte	69							
Chef Anton's Cajun Seasoning		53						
Chef Anton's Gumbo Mix								
Chocolade			15					
Côte de Blaye	17							
Escargots de Bourgogne								62
Filo Mix					38			
Flotemysost				26				
Geitost				112				
Genen Shouyu		39						
Gnocchi di nonna Alice					21			
Gorgonzola Telino								
Grandma's Boysenberry Spread		120						
Gravad lax								11
Guaraná Fantástica	20							
Gudbrandsdalsost				26				
Gula Malacca		27						
Gumbär Gummibärchen			15					
Gustaf's Knäckebröd					104			
Ikura								31
Inlagd Sill								112
Ipoh Coffee	17							
Jack's New England Clam Chowder								85
Konbu								24
Lakkalikööri	57							
Laughing Lumberjack Lager	52							
Longlife Tofu							4	
Louisiana Fiery Hot Pepper Sauce		76						
Louisiana Hot Spiced Okra		4						
Manjimup Dried Apples							20	
Mascarpone Fabioli				9				
Maxilaku			10					
Mishi Kobe Niku						29		
Mozzarella di Giovanni				14				

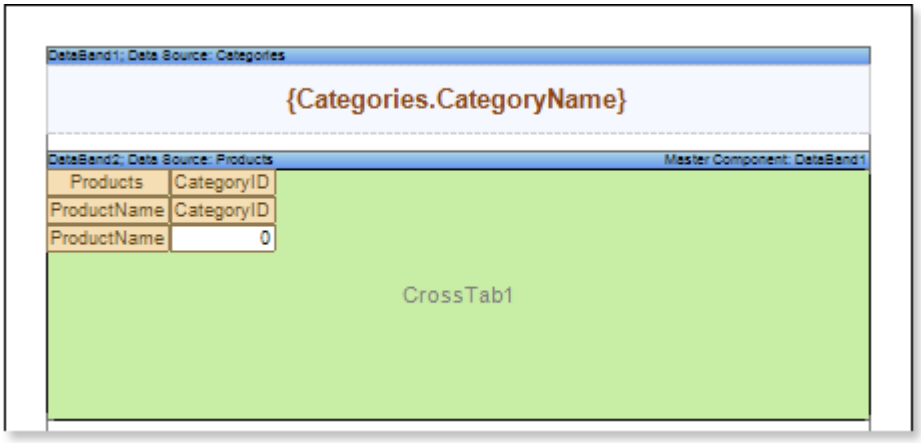
Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Nord-Ost Matjeshering								10
Northwoods Cranberry Sauce		6						
NuNuCa Nuß-Nougat-Creme			76					
Original Frankfurter grüne Soße		32						
Outback Lager	15							
Pâté chinois						115		
Pavlova			29					
Perth Pasties								
Queso Cabrales				22				
Queso Manchego La Pastora				86				
Raclette Courdavault				79				
Ravioli Angelo					36			
Rhönbräu Klosterbier	125							
Röd Kaviar								101
Rogede sild								5
Rössle Sauerkraut							26	
Sasquatch Ale	111							
Schoggi Schokolade			49					
Scottish Longbreads			6					
Singaporean Hokkien Fried Mee					26			
Sir Rodney's Marmalade			40					
Sir Rodney's Scones			3					
Sirop d'érable		113						
Spegesild								95
Steeleye Stout	20							
Tarte au sucre			17					
Teatime Chocolate Biscuits			25					
Thüringer Rostbratwurst								
Tofu							35	
Tourtière						21		
Tunnbröd					61			
Uncle Bob's Organic Dried Pears							15	
Valkoinen suklaa			65					
Vegie-spread		24						
Wimmers gute Semmelknödel					22			
Zaanse koeken			36					

Condiments

Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123

17. Go back to the report template;
18. If necessary, edit the text component in the **DataBand**:
    18.1. Change the background color of the text component;
    18.2. Change the style, color, and text type.

19. Edit cells in the report template:
    19.1. Change the font settings: type, style, size;
    19.2. Change the background color of a cell;
    19.3. Set the **Word Wrap** property to **true**, if you want the text to be wrapped;
    19.4. Set **Borders** if necessary;
    19.5. Change the border color.
    19.6. Change the background color of cells, etc.



20. Render a report. Click the **Preview** button or call the **Viewer** by pressing **F5** or select the **Preview** of the menu item. The picture below shows an example of the cross-table report after editing cells of the report template:

Beverages

Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123
Camembert Pierrot				19				
Carnarvon Tigers								42
Chai	39							
Chang	17							
Chartreuse verte	69							
Chef Anton's Cajun Seasoning		53						
Chef Anton's Gumbo Mix								
Chocolade			15					
Côte de Blaye	17							
Escargots de Bourgogne								62
Filo Mix					38			
Flotemysost				26				
Geitost				112				
Genen Shouyu		39						
Gnocchi di nonna Alice					21			
Gorgonzola Telino								
Grandma's Boysenberry Spread		120						
Gravad lax								11
Guaraná Fantástica	20							
Gudbrandsdalsost				26				
Gula Malacca		27						
Gumbär Gummibärchen			15					
Gustaf's Knäckebröd					104			
Ikura								31
Inlagd Sill								112
Ipoh Coffee	17							
Jack's New England Clam Chowder								85
Konbu								24
Lakkalikööri	57							
Laughing Lumberjack Lager	52							
Longlife Tofu							4	
Louisiana Fiery Hot Pepper Sauce		76						
Louisiana Hot Spiced Okra		4						
Manjimup Dried Apples							20	
Mascarpone Fabioli				9				
Maxilaku			10					
Mishi Kobe Niku						29		
Mozzarella di Giovanni				14				

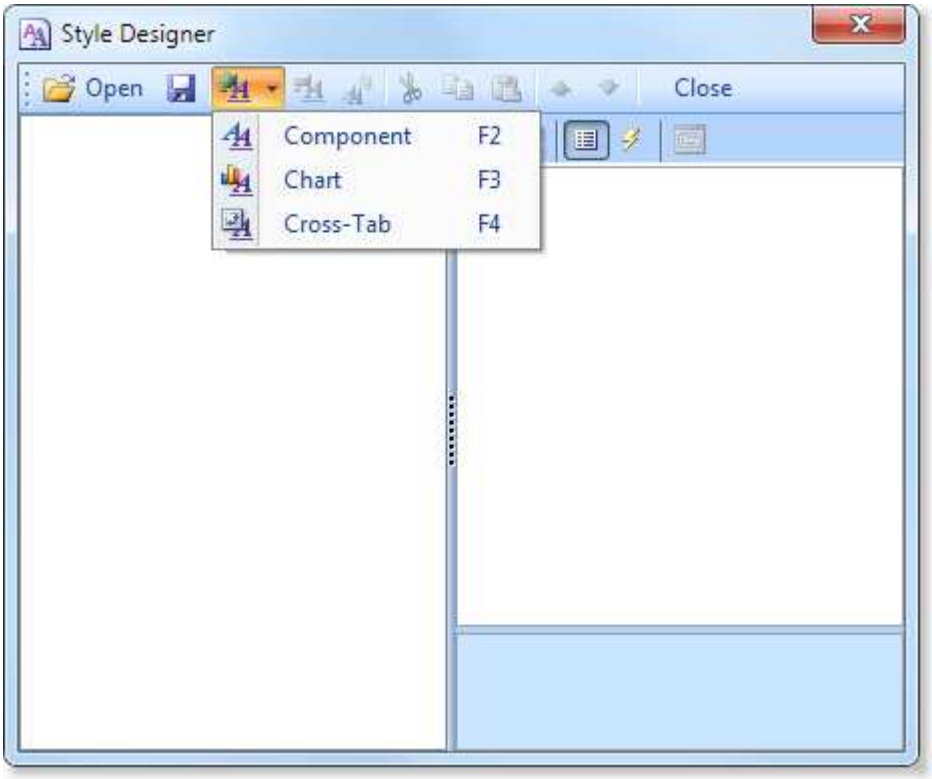
Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Nord-Ost Matjeshering								10
Northwoods Cranberry Sauce		6						
NuNuCa Nuß-Nougat-Creme			76					
Original Frankfurter grüne Soße		32						
Outback Lager	15							
Pâté chinois						115		
Pavlova			29					
Perth Pasties								
Queso Cabrales				22				
Queso Manchego La Pastora				86				
Raclette Courdavault				79				
Ravioli Angelo					36			
Rhönbräu Klosterbier	125							
Röd Kaviar								101
Rogede sild								5
Rössle Sauerkraut							26	
Sasquatch Ale	111							
Schoggi Schokolade			49					
Scottish Longbreads			6					
Singaporean Hokkien Fried Mee					26			
Sir Rodney's Marmalade			40					
Sir Rodney's Scones			3					
Sirop d'érable		113						
Spegesild								95
Steeleye Stout	20							
Tarte au sucre			17					
Teatime Chocolate Biscuits			25					
Thüringer Rostbratwurst								
Tofu							35	
Tourtière						21		
Tunnbröd					61			
Uncle Bob's Organic Dried Pears							15	
Valkoinen suklaa			65					
Vegie-spread		24						
Wimmers gute Semmelknödel					22			
Zaanse koeken			36					

Condiments

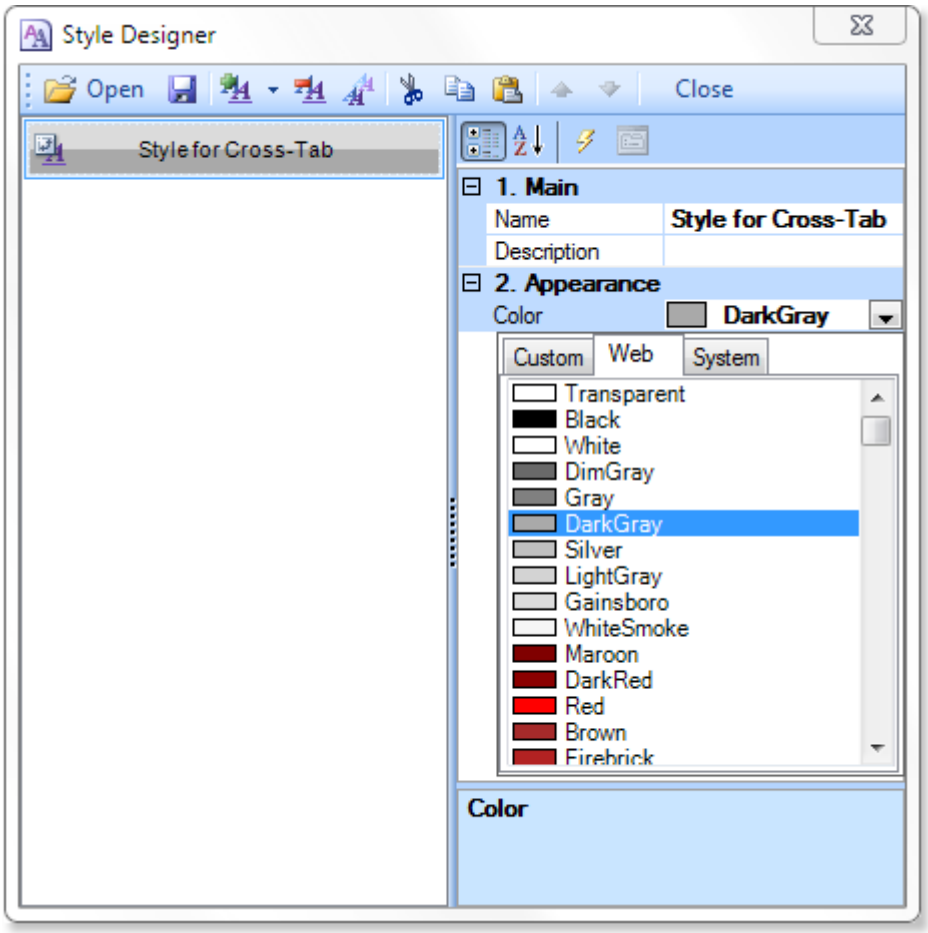
Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123

Adding styles

1. Go back to the report template;
2. Invoke the **Style Designer**;



Click the **Add Style** button to start creating a style. Select **Cross-Tab** from the drop down list. Call the new style as **Style for Cross-Tab**. To create a custom style it is necessary to change the **Color** property, where the value of this property and is a color scheme.



After the style is created, press the **Close** button. In the list of values of the **Select Style** button in the editor of the cross-table, a custom style will be displayed. In our case, this is the **Style for Cross-Tab**. Select this value;

3. Render a report. Click the **Preview** button or call the **Viewer** by pressing **F5** or select the **Preview** of the menu item. Now you can see the result of the rendered report:

Beverages

Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123
Camembert Pierrot				19				
Carnarvon Tigers								42
Chai	39							
Chang	17							
Chartreuse verte	69							
Chef Anton's Cajun Seasoning		53						
Chef Anton's Gumbo Mix								
Chocolade			15					
Côte de Blaye	17							
Escargots de Bourgogne								62
Filo Mix					38			
Flotemysost				26				
Geitost				112				
Genen Shouyu		39						
Gnocchi di nonna Alice					21			
Gorgonzola Telino								
Grandma's Boysenberry Spread		120						
Gravad lax								11
Guaraná Fantástica	20							
Gudbrandsdalsost				26				
Gula Malacca		27						
Gumbär Gummibärchen			15					
Gustaf's Knäckebröd					104			
Ikura								31
Inlagd Sill								112
Ipoh Coffee	17							
Jack's New England Clam Chowder								85
Konbu								24
Lakkalikööri	57							
Laughing Lumberjack Lager	52							
Longlife Tofu							4	
Louisiana Fiery Hot Pepper Sauce		76						
Louisiana Hot Spiced Okra		4						
Manjimup Dried Apples							20	
Mascarpone Fabioli				9				
Maxilaku			10					
Mishi Kobe Niku						29		
Mozzarella di Giovanni				14				

Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Nord-Ost Matjeshering								10
Northwoods Cranberry Sauce		6						
NuNuCa Nuß-Nougat-Creme			76					
Original Frankfurter grüne Soße		32						
Outback Lager	15							
Pâté chinois						115		
Pavlova			29					
Perth Pasties								
Queso Cabrales				22				
Queso Manchego La Pastora				86				
Raclette Courdavault				79				
Ravioli Angelo					36			
Rhönbräu Klosterbier	125							
Röd Kaviar								101
Rogede sild								5
Rössle Sauerkraut							26	
Sasquatch Ale	111							
Schoggi Schokolade			49					
Scottish Longbreads			6					
Singaporean Hokkien Fried Mee					26			
Sir Rodney's Marmalade			40					
Sir Rodney's Scones			3					
Sirop d'érable		113						
Spegesild								95
Steeleye Stout	20							
Tarte au sucre			17					
Teatime Chocolate Biscuits			25					
Thüringer Rostbratwurst								
Tofu							35	
Tourtière						21		
Tunnbröd					61			
Uncle Bob's Organic Dried Pears							15	
Valkoinen suklaa			65					
Vegie-spread		24						
Wimmers gute Semmelknödel					22			
Zaanse koeken			36					

Condiments

Products	CategoryID							
ProductName	1	2	3	4	5	6	7	8
Alice Mutton								
Aniseed Syrup		13						
Boston Crab Meat								123

Hierarchical Report

For better understanding this step-by-step instruction, please watch the video file.

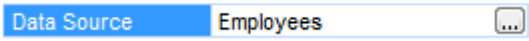
Do the following steps to create a hierarchical report:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

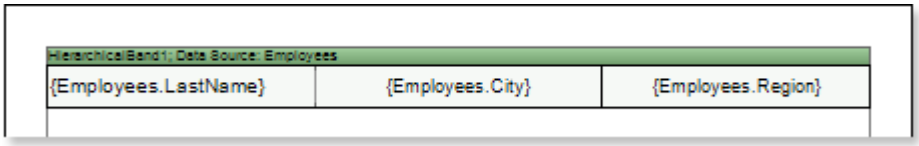
3. Put the **HierarchicalBand** on a page of the report template**.**

4. Edit the **HierarchicalBand**:
   4.1. Align the **HierarchicalBand** by height;
   4.2. Set the properties of the **HierarchicalBand**. For example, set the **Can Break** property to **true**, if it is necessary for the **HierarchicalBand** to be broken;
   4.3. Set the background of the **HierarchicalBand**;
   4.4. Set the **Borders** of the **HierarchicalBand**;
   4.5. Set the border color.

5. Set the data source of the **HierarchicalBand** using the **Data Source** property:



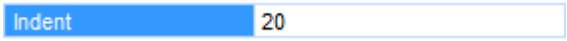
6. Put text components with expressions in the **HierarchicalBand**. Where the expression is a reference to the data field. For example, put three text component with expressions: **{Employees.LastName}, {Employees.City}**, and **{Employees.Region}**;
7. Edit text (**Text**) and text components (**TextBox**):
   7.1. Drag the text component to the required place in the **HierarchicalBand**;
   7.2. Set the font of the text: the size, style, color;
   7.3. Align the text component vertically and horizontally;
   7.4. Set the background color of the text component;
   7.5. Align text in the text component;
   7.6. Set values of the properties of a text component. For example, set the **Word Wrap** property to **true**, if you want the text to be wrapped;
   7.7. Set **Borders** of a text component.
   7.8. Set the border color.



8. Set the **KeyDataColum** property, i.e. select a data column on which an identification number of the data row will be assigned. In this case, select the **EmployeeID** data column:

9. Set the **MasterKeyDataColum** property, i.e. select a data column on which the reference to the table's primary key of the parent entry will be specified. In this case, select the **ReportsTo** data column:

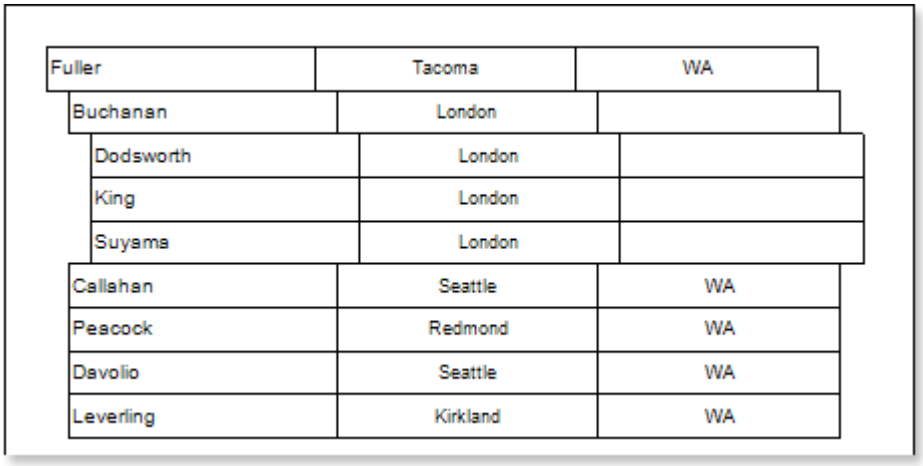
10. Set the **Indent** property, i.e. set an offset of the detail entry in relation to the parent one. In this example, the **Indent** property will be 20 units in the report (centimeters, inches, hundredths of inches, pixels);



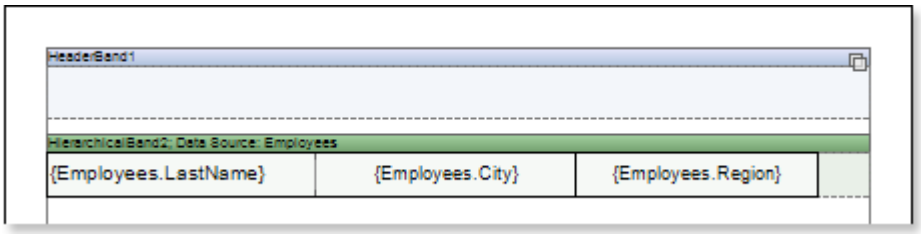
11. Set the **ParentValue** property, i.e. indicate the entry, which will be a parent for all rows. If this property is not specified, the default value is used. By default, the **Parent Value** property is set to **null**. In this case, the value of the **ParentValue** property is not specified, so the default value is used:

12. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item. After rendering a report, all references to data sources will be replaced with data from these sources. Data will be taken sequentially from the

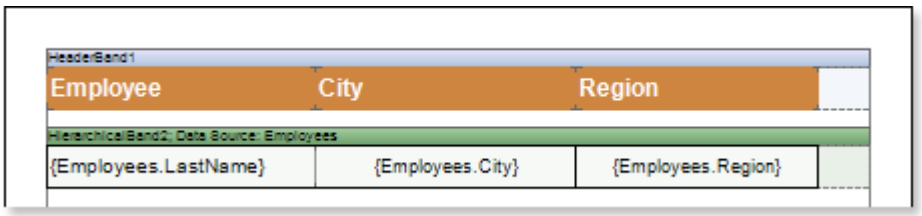
data source, which has been specified for this band. Number of copies of the **DataBand** in the report is equal to the number of rows in the data source.



13. Go back to the report template;
14. If necessary, add other bands into the report template, for example, **HeaderBand**;
15. Edit this band:
    15.1. Align the **HeaderBand** vertically;
    15.2. Set properties of the **HeaderBand**, if necessary;
    15.3. Set the background color of the **HeaderBand**;
    15.4. If necessary, set the **Borders**;
    15.5. Change the border color.



16. Put text components with the expressions. Where expressions in text components in the **HeaderBand** will be the data headers;
17. Edit text and text components:
    17.1. Drag the text component to the required place in the band;
    17.2. Set the font settings: size, style, color;
    17.3. Align the text component vertically and horizontally;
    17.4. Set the background color of the text component;
    17.5. Align the text in a text component;
    17.6. Set the value of properties of a text component, if necessary;
    17.7. If necessary, set **Borders** of a text component;
    17.8. Set the border color.

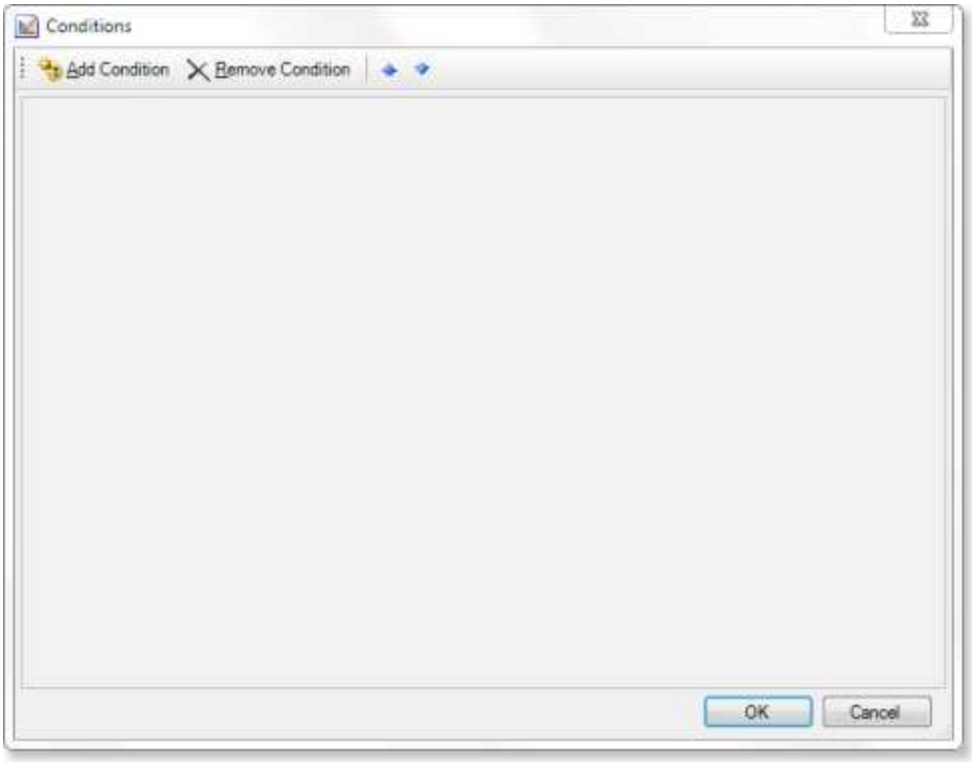


18. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item. After rendering a report, all references to data sources will be replaced with data from these sources:

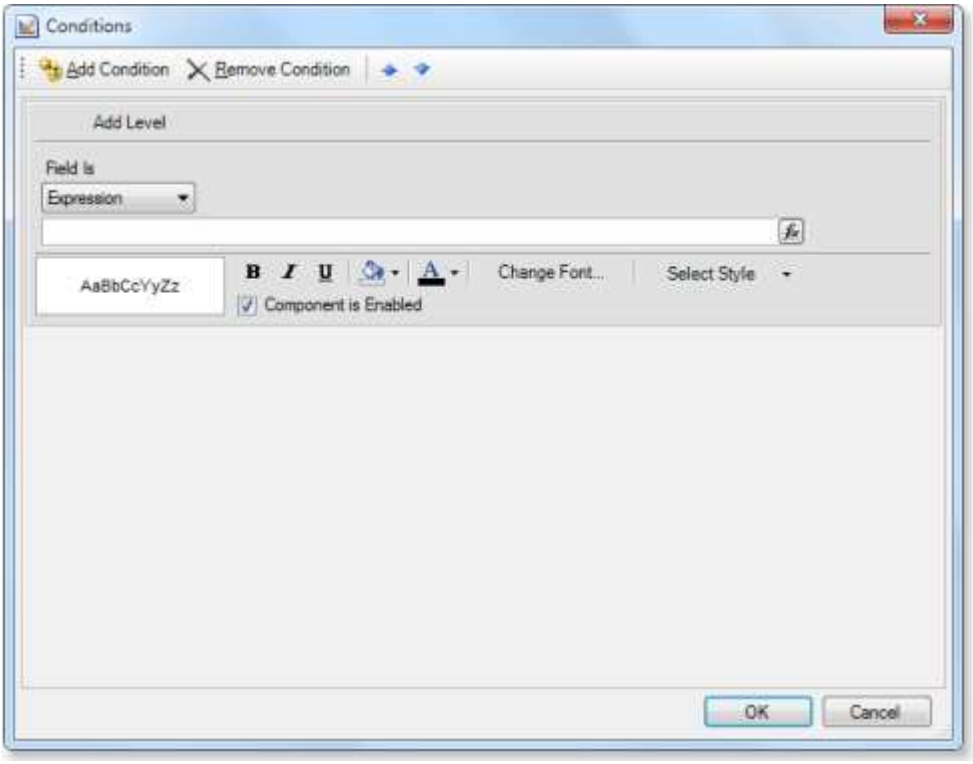
Employee	City	Region
Fuller	Tacoma	WA
Buchanan	London	
Dodsworth	London	
King	London	
Suyama	London	
Callahan	Seattle	WA
Peacock	Redmond	WA
Davolio	Seattle	WA
Leverling	Kirkland	WA

Adding styles

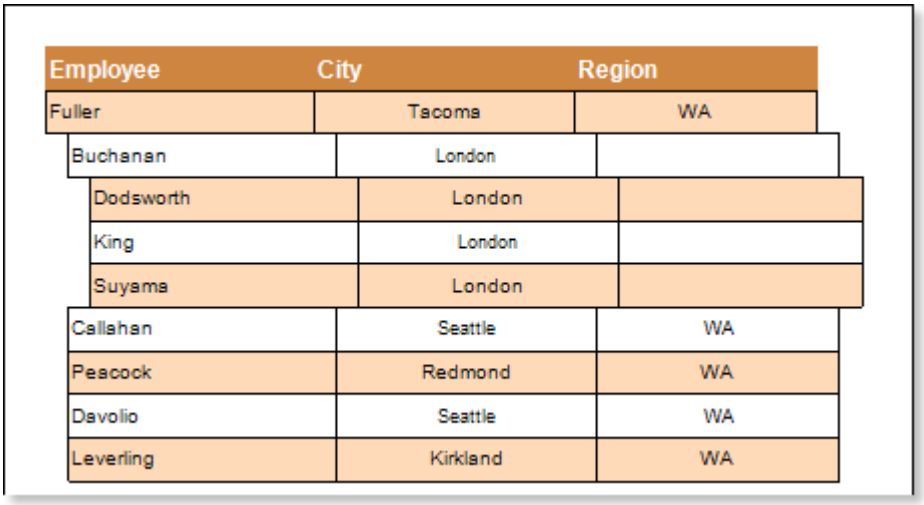
1. Go back to the report template;
1. Select component. In our case this is the text component;
2. Invoke the **Conditions** dialog box. For example, click the **Conditions** button on the control panel.



3. To get started, you must click the **Add Condition** button and in the **Conditions** dialog box the condition and formatting options will be displayed. The condition can be of two types: **Value** and **Expression**. In this case, consider an example of a condition, such as **Expression**. The picture below shows an example of **Conditions** dialog box with options and conditions of formatting:



4. Specify the options of conditional formatting. In this case, to specify the condition means to specify the expression. For example, **Line% 2 == 1**, and set the formatting means to change the background, for example, by pressing the **Back Color** button and selecting the drop-down list of values of the background color.
5. Click **OK**. It should also be noted that in order to odd and even rows have different styles, it is necessary to make a conditional formatting of each text component;
6. Render a report by clicking on the **Preview** tab or call the **Viewer** pressing **F5** or clicking **Preview** menu item.



Employee	City	Region
Fuller	Tacoma	WA
Buchanan	London	
Dodsworth	London	
King	London	
Suyama	London	
Callahan	Seattle	WA
Peacock	Redmond	WA
Davolio	Seattle	WA
Leverling	Kirkland	WA

Report with Sub-Report

For better understanding this step-by-step instruction, please watch the video file.

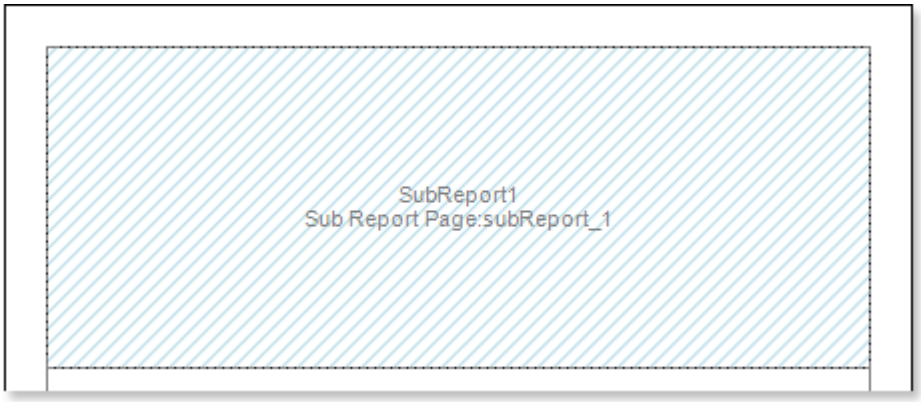
Do the following steps to create a sub-report:

1. Run the designer;
2. Connect data:

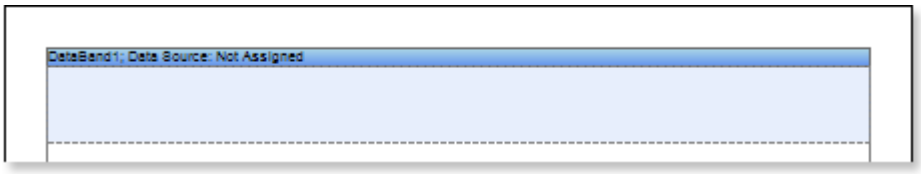
2.1. Create **New Connection**;
2.2. Create **New Data Source**;

3. Add the **Sub-Report** component to a report on a page of the report template:

4. Edit the **Sub-Report** component:
4.1. Stretch the **Sub-Report** component as seen on the picture below;
4.2. Change the value of properties of **Sub-Report**. For example, set the **Keep Sub-Report Together** property to **true**, if you want the sub-report to be kept together;;
4.3. Change the background color of the component.

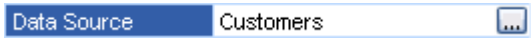


5. Go to the sub-report page;
6. Add to the **DataBand** to the sub-report page**;**



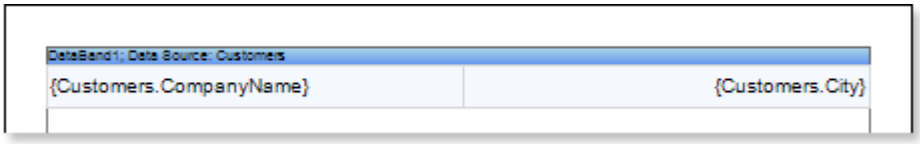
7. Edit the **DataBand**:
7.1. Align the **DataBand** vertically;
7.2. Change values of properties of the **DataBand**. For example, set the **CanBreak** property to **true**, if you want this band to be broken;
7.3. Change background color of the band;
7.4. Set **Borders**, if necessary;
7.5. Change the border color.

8. Specify the data source for the **DataBand** using the **Data Source** property:

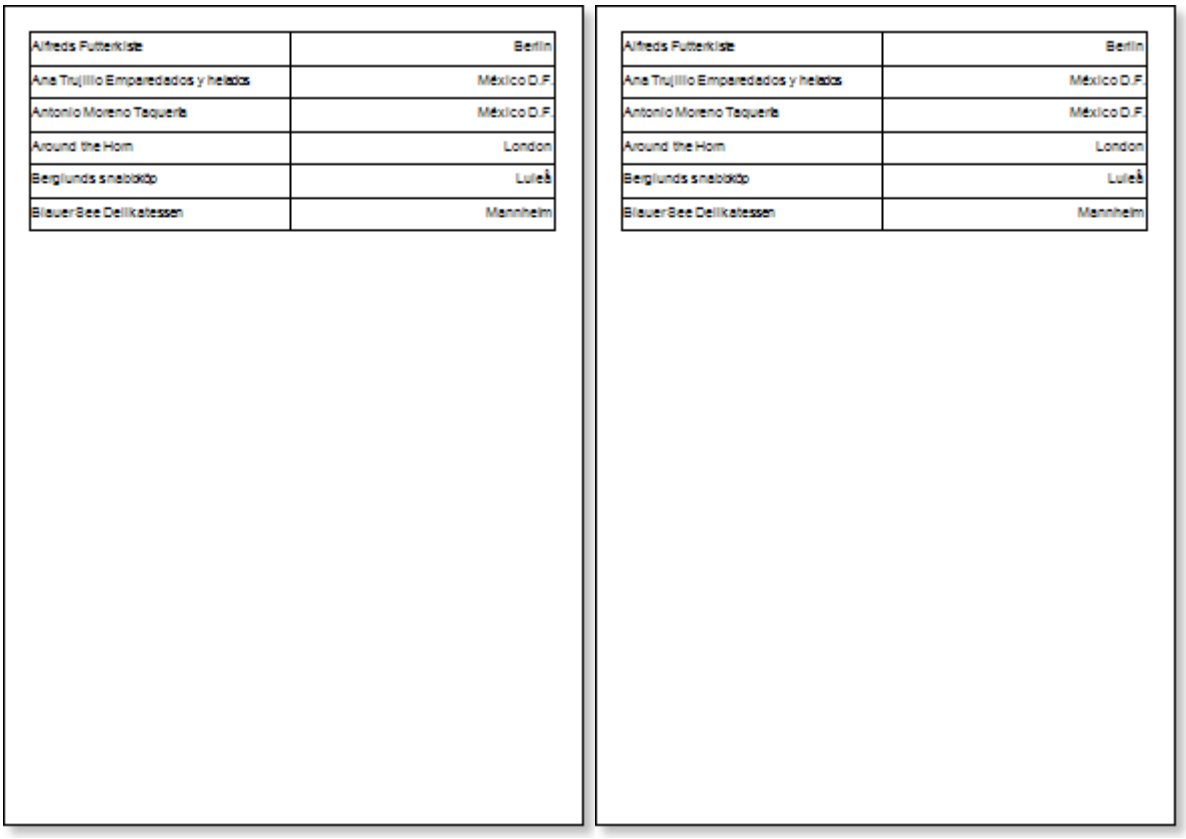


9. Put text components with expressions in the **DataBand**. Where an expression is a reference to a data field. For example, put the following expressions to two text components: **{Customers.CompanyName}** and **{Customers.City}**;
10. Edit **Text** and **TextBoxes**:
10.1. Drag the text component to the required place in the **DataBand**;
10.2. Set the text font: size, style, color;

10.3. Align text component vertically and horizontally;
10.4. Set the background color of the text component;
10.5. Align text in the component;
10.6. Set values of the properties of text components. For example to set the **Word Wrap** property to **true**, if you want the text to be wrapped;
10.7. Set **Borders** of a text component.
10.8. Set the border color.



11. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item to see how the report will look like.



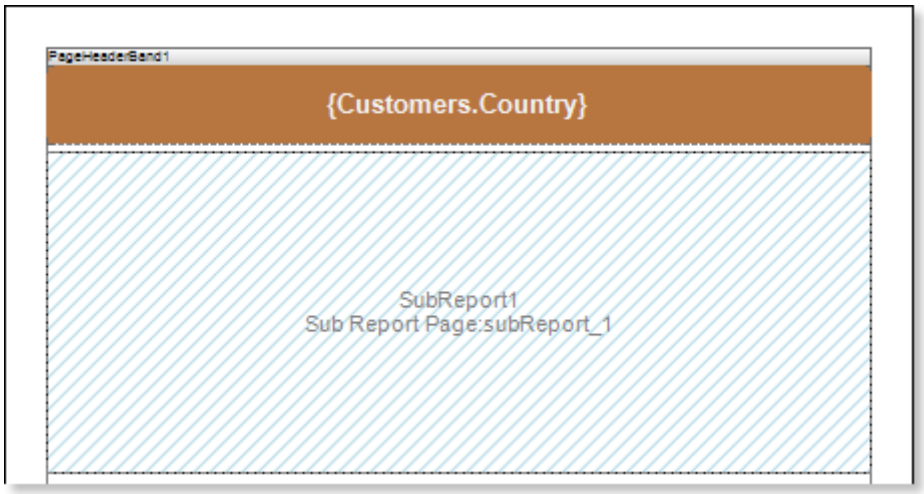
Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim

Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim

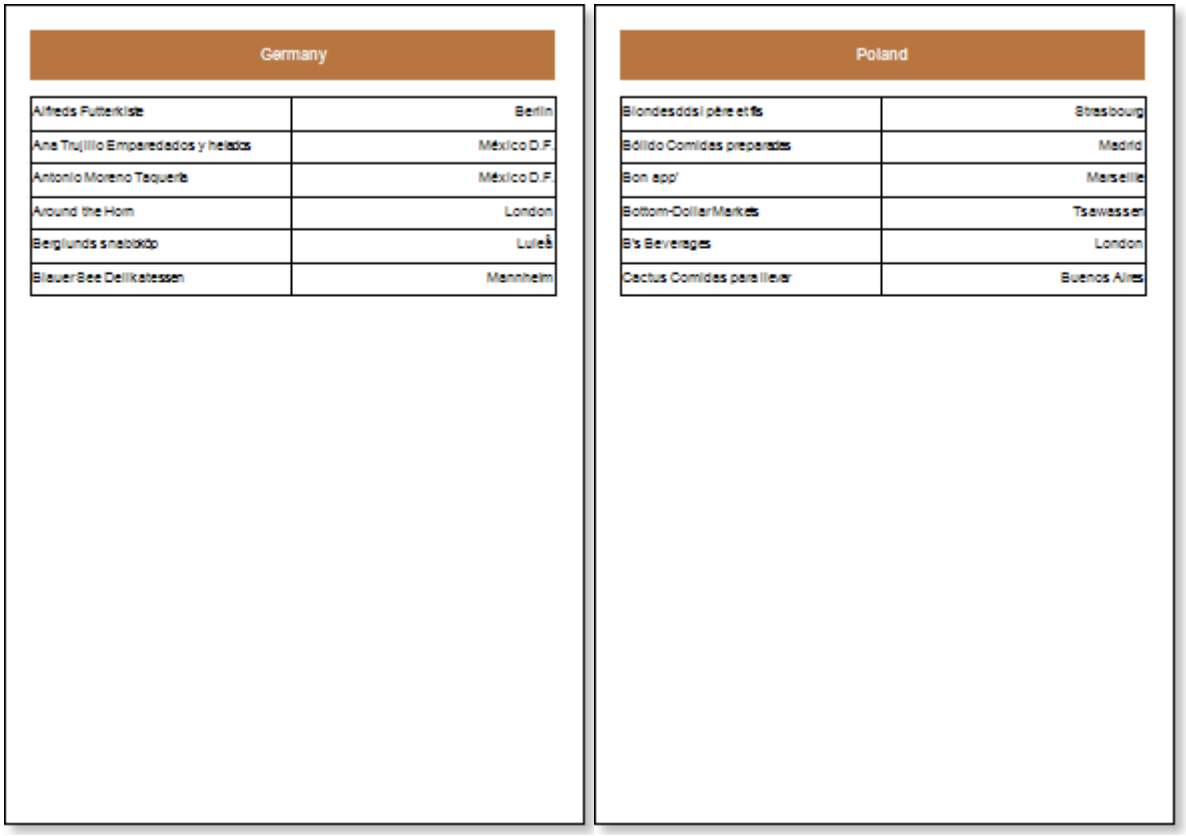
As can be seen from the picture above, the report generator rendered the report, which was located in the nested page and placed it on the report page but not in the Sub-Report component.

12. Go back to the report template;
13. If necessary, add some bands to the report template, for example, the **PageHeaderBand**;
14. Edit this band:
    14.1. Align vertically this band;
    14.2. Set values of the properties of the **PageHeaderBand**, if necessary;
    14.3. Set the background color;
    14.4. Set **Borders** of a text component.
    14.5. Set the border color.

15. Put a text component with expression where the expression of the text component in the **PageHeaderBand** will be the page title.
16. Edit the text component:
    16.1. Drag the text component to the required place in the band;
    16.2. Set the text font: size, style, color;
    16.3. Align text component vertically and horizontally;
    16.4. Set the background color of the text component;
    16.5. Align text in the component;
    16.6. Set values of the properties of text components;
    16.7. Set **Borders** of a text component.
    16.8. Set the border color.



17. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item to see how the report will look like.



Germany

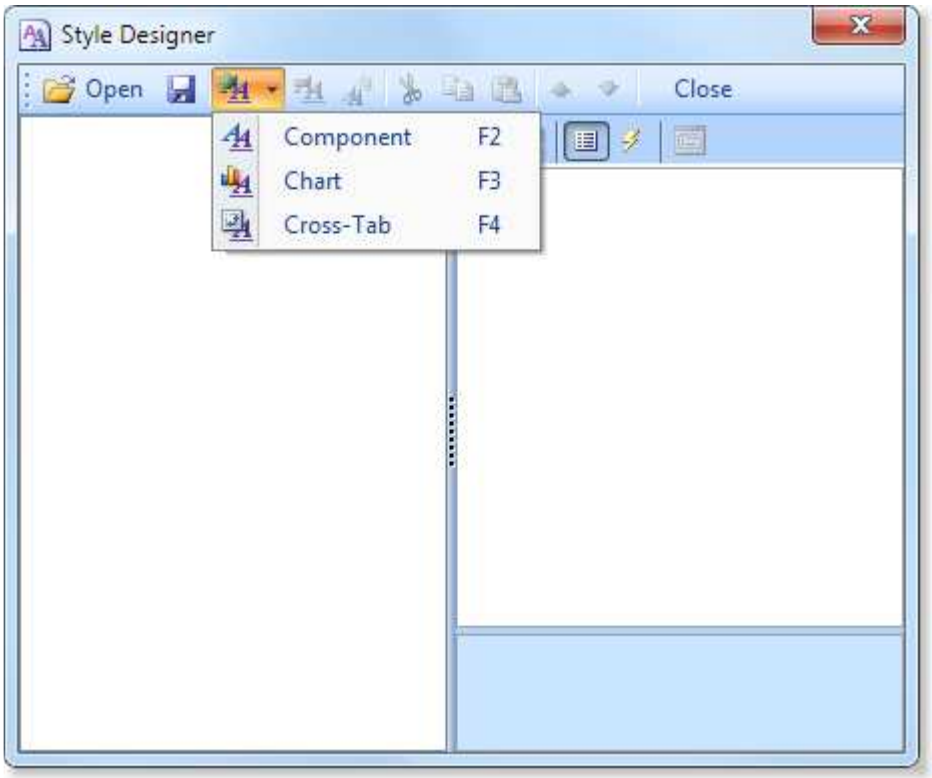
Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim

Poland

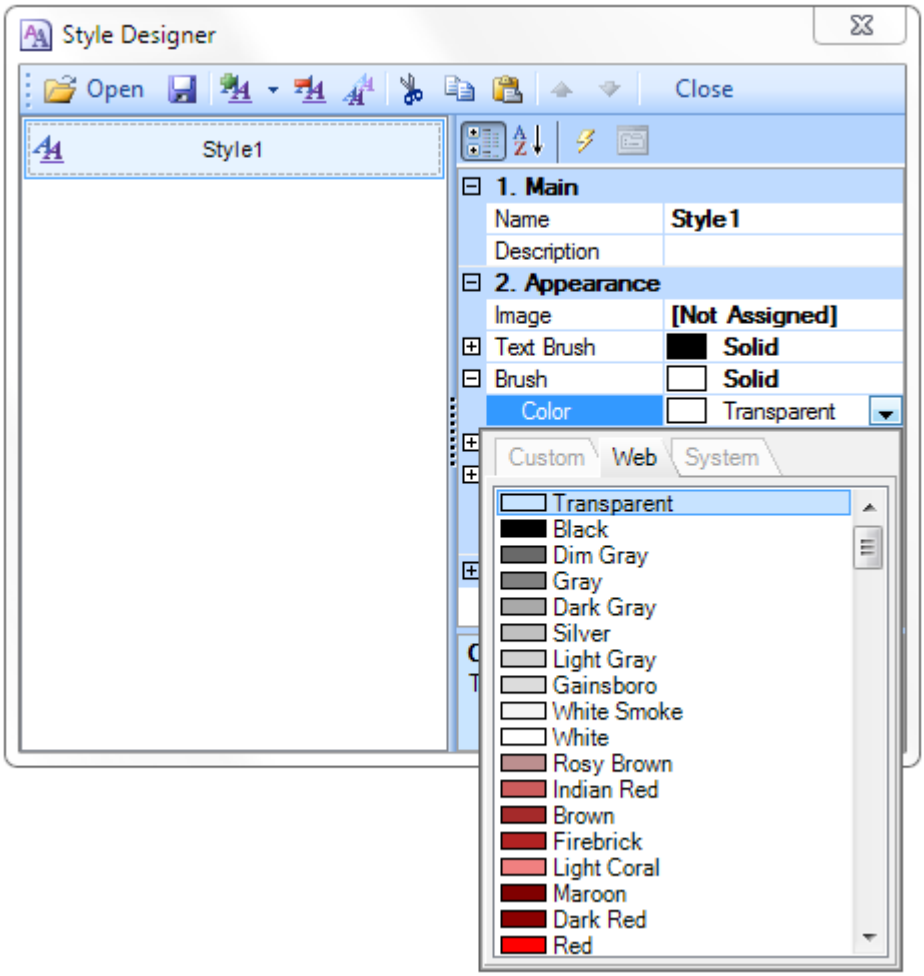
Blondesddsl père et fils	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Markets	Tsawassen
B's Beverages	London
Cactus Comidas para llevar	Buenos Aires

Adding styles

1. Go back to the report template;
2. Select the sub-report;
3. Select the **DataBand**;
4. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**

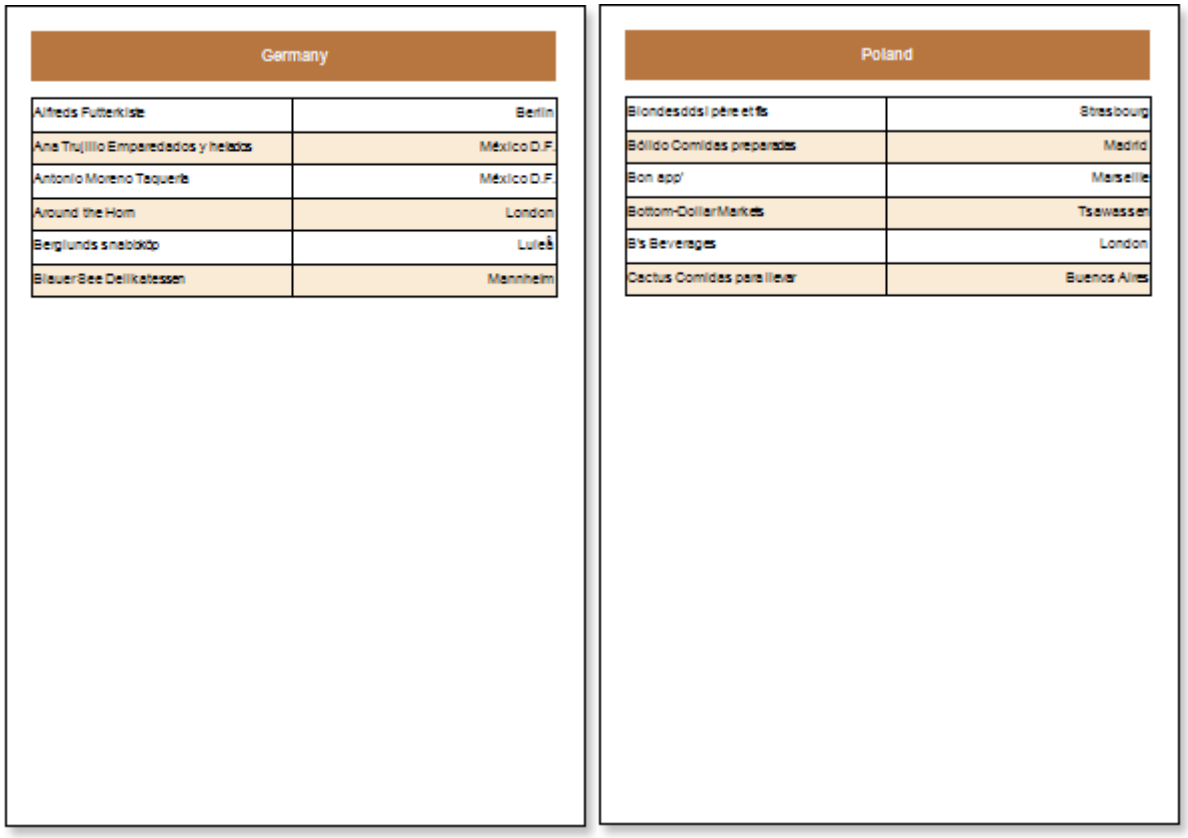


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

5. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered sub-report with alternative color of rows:



Germany

Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim

Poland

Blondesddsl père et fils	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Markets	Tsawassen
B's Beverages	London
Cactus Comidas para llevar	Buenos Aires

Side-by-Side Report

For better understanding this step-by-step instruction, please watch the video file.

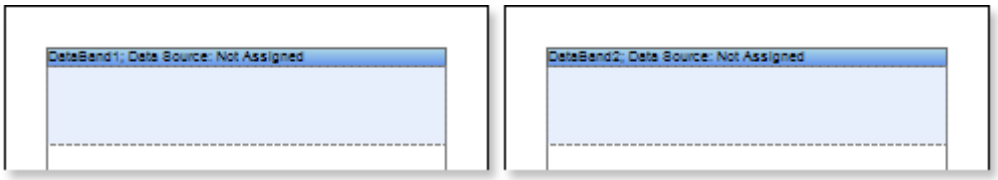
The **Side-by-side** report is a type of independent data lists, located side by side. Do the following steps to create such a report:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

3. Add **Sub-Report** components to a report on a page of the report template:

4. Edit **Sub-Report** components:
   4.1. Stretch **Sub-Report** components as seen on the picture below;
   4.2. Change the value of properties of **Sub-Report**. For example, set the **Keep Sub-Report Together** property to **true**, if you want the sub-report to be kept together;
   4.3. Change the background color of the component.

5. Go to the sub-report page;
6. Add two **DataBands** to the sub-report page. Add **DataBand1** to the **Sub Report1** and **DataBand2** to the **Sub Report2**;

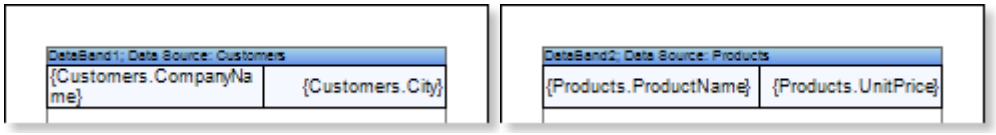


7. Edit the **DataBands**:
   7.1. Align the **DataBands** vertically;
   7.2. Change values of properties of the **DataBands**.
   7.3. Change background color of the band;
   7.4. Set **Borders**, if necessary;
   7.5. Change the border color.

8. Specify the data source for the **DataBand** using the **Data Source** property. For example, set the **Customers** data source for the **DataBand1**, and the **Products** data source for the **DataBand2**:

9. Put text components with expressions in the **DataBands**. Where an expression is a reference to a data field. For example, put the following expressions to the **DataBand1**: **{Customers.CompanyName}** and **{Customers.City}**. put the following expressions to the **DataBand2**: **{Products.ProductName}** and **{Products.UnitPrice}**;

10. Edit **Text** and **TextBoxes**:
   10.1. Drag the text component to the required place in the **DataBand**;
   10.2. Set the text font: size, style, color;
   10.3. Align text component vertically and horizontally;
   10.4. Set the background color of the text component;
   10.5. Align text in the component;
   10.6. Set values of the properties of text components. For example to set the **Word Wrap** property to **true**, if you want the text to be wrapped;
   10.7. Set **Borders** of a text component.
   10.8. Set the border color.



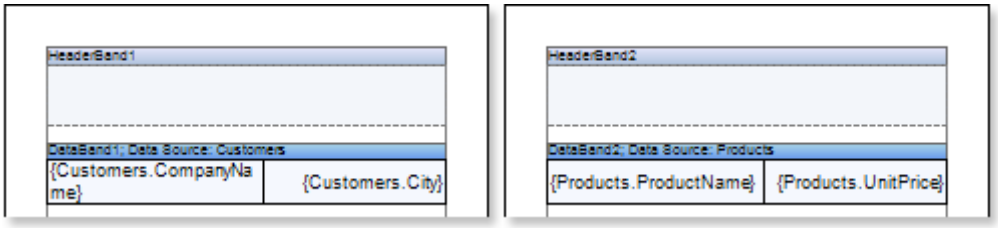
11. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item to see how the report will look like:

Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondesddsl père et fils	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Markets	Tsawassen
B's Beverages	London
Cactus Comidas para llevar	Buenos Aires
Centro comercial Moctezuma	México D.F.
Chop-suey Chinese	Bern
Comércio Mineiro	Sao Paulo

Chai	18
Chang	19
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Uncle Bob's Organic Dried Pears	30
Northwoods Cranberry Sauce	40
Mishi Kobe Niku	97
Ikura	31
Queso Cabrales	21
Queso Manchego La Pastora	38
Konbu	6
Tofu	23,25
Genen Shouyu	15,5

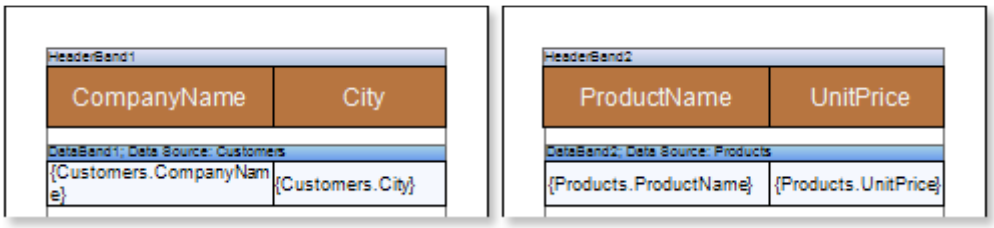
As can be seen from the picture above, the report generator rendered the report, which was located in the nested page and placed it on the report page but not in the Sub-Report component.

12. Go back to the report template;
13. If necessary, add some bands to the report template, for example, the **HeaderBand**;
14. Edit this band:
    14.1. Align vertically this band;
    14.2. Set values of the properties of the **HeaderBand**, if necessary;
    14.3. Set the background color;
    14.4. Set **Borders** of a text component.
    14.5. Set the border color.

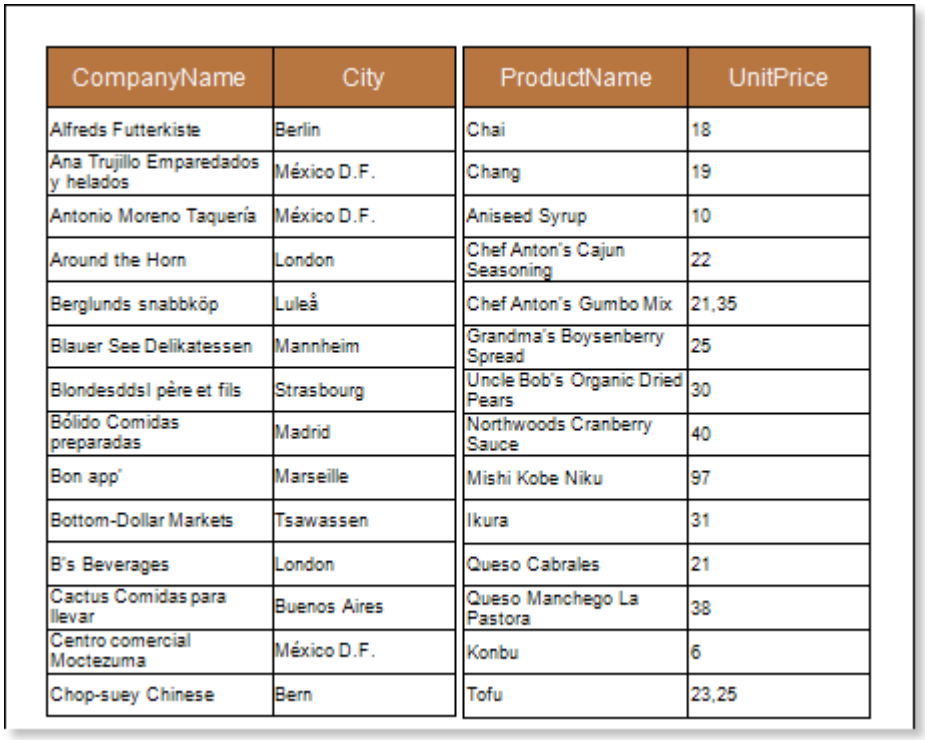


15. Put a text component with expression where the expression of the text component in the **HeaderBand** will be the page title.
16. Edit the text component:
    16.1. Drag the text component to the required place in the band;
    16.2. Set the text font: size, style, color;
    16.3. Align text component vertically and horizontally;
    16.4. Set the background color of the text component;
    16.5. Align text in the component;
    16.6. Set values of the properties of text components;

16.7. Set **Borders** of a text component.
16.8. Set the border color.



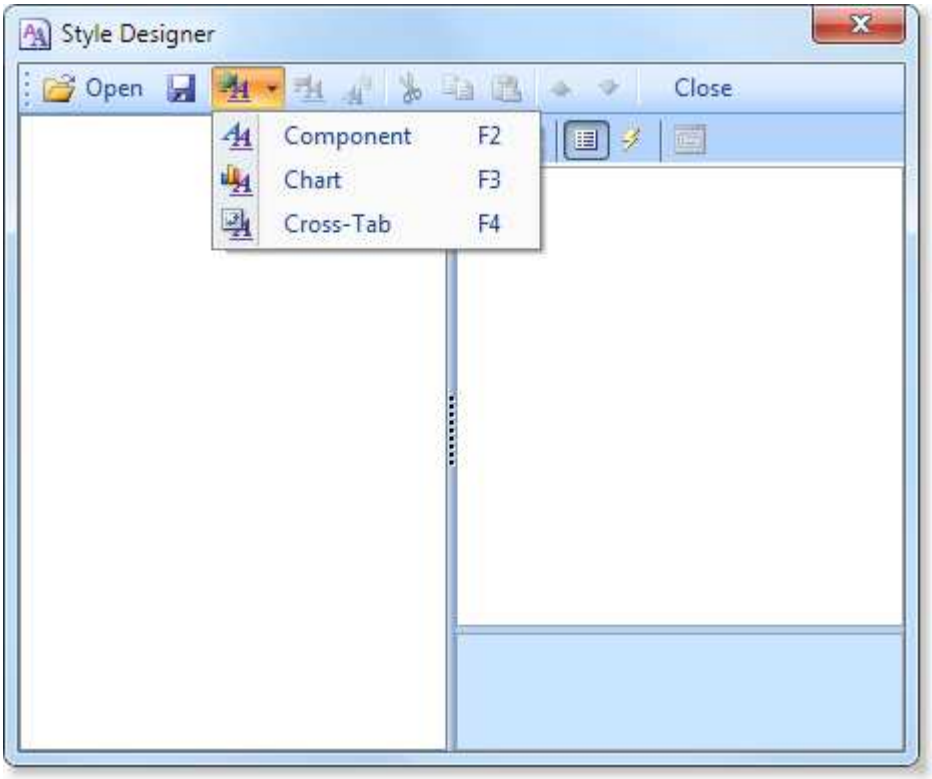
17. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item to see how the report will look like:



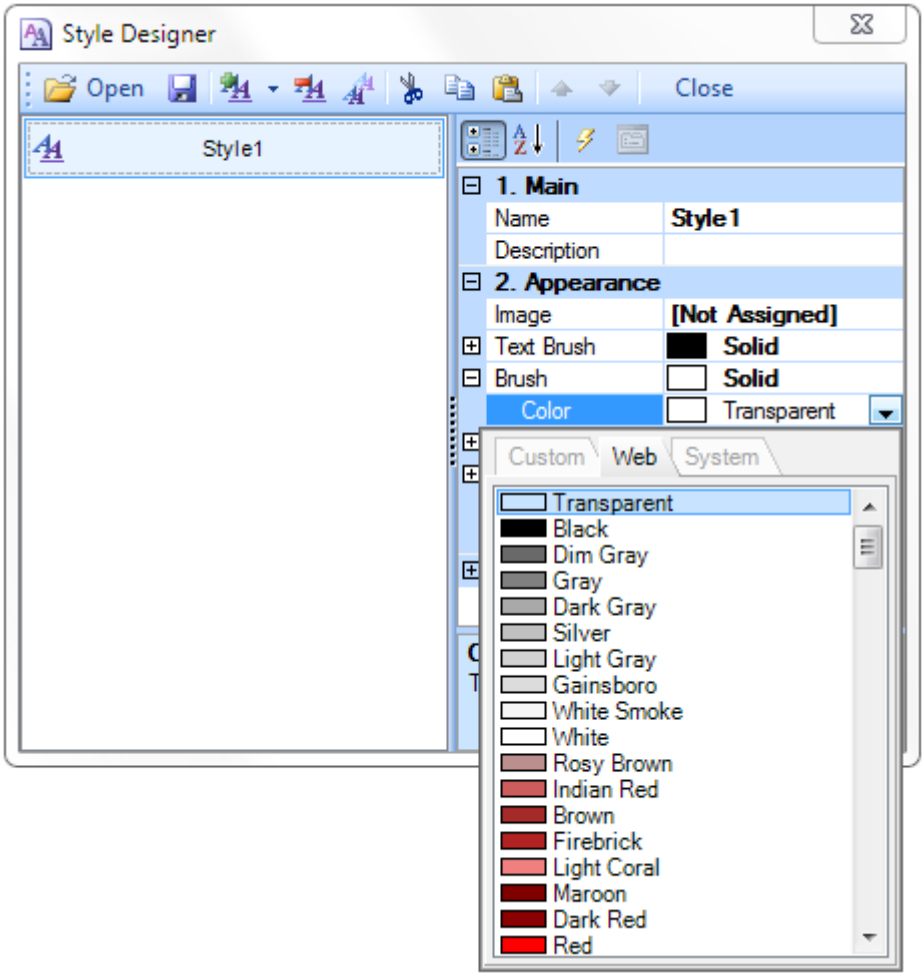
CompanyName	City	ProductName	UnitPrice
Alfreds Futterkiste	Berlin	Chai	18
Ana Trujillo Emparedados y helados	México D.F.	Chang	19
Antonio Moreno Taquería	México D.F.	Aniseed Syrup	10
Around the Horn	London	Chef Anton's Cajun Seasoning	22
Berglunds snabbköp	Luleå	Chef Anton's Gumbo Mix	21,35
Blauer See Delikatessen	Mannheim	Grandma's Boysenberry Spread	25
Blondesddsl père et fils	Strasbourg	Uncle Bob's Organic Dried Pears	30
Bólido Comidas preparadas	Madrid	Northwoods Cranberry Sauce	40
Bon app'	Marseille	Mishi Kobe Niku	97
Bottom-Dollar Markets	Tsawassen	Ikura	31
B's Beverages	London	Queso Cabrales	21
Cactus Comidas para llevar	Buenos Aires	Queso Manchego La Pastora	38
Centro comercial Moctezuma	México D.F.	Konbu	6
Chop-suey Chinese	Bern	Tofu	23,25

Adding styles

1. Go back to the report template;
2. Select the sub-report;
3. Select the **DataBand**;
4. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**.



Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

5. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered side-by-side report with alternative color of rows:

CompanyName	City	ProductName	UnitPrice
Alfreds Futterkiste	Berlin	Chai	18
Ana Trujillo Emparedados y helados	México D.F.	Chang	19
Antonio Moreno Taquería	México D.F.	Aniseed Syrup	10
Around the Horn	London	Chef Anton's Cajun Seasoning	22
Berglunds snabbköp	Luleå	Chef Anton's Gumbo Mix	21,35
Blauer See Delikatessen	Mannheim	Grandma's Boysenberry Spread	25
Blondesddsl père et fils	Strasbourg	Uncle Bob's Organic Dried Pears	30
Bólido Comidas preparadas	Madrid	Northwoods Cranberry Sauce	40
Bon app'	Marseille	Mishi Kobe Niku	97
Bottom-Dollar Markets	Tsawassen	Ikura	31
B's Beverages	London	Queso Cabrales	21
Cactus Comidas para llevar	Buenos Aires	Queso Manchego La Pastora	38
Centro comercial Moctezuma	México D.F.	Konbu	6
Chop-suey Chinese	Bern	Tofu	23,25

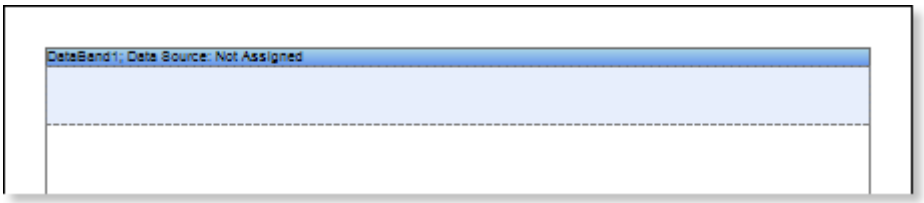
Report with Sub-Reports in DataBand

For better understanding this step-by-step instruction, please watch the video file.

Do the following steps to create a simple list report:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;

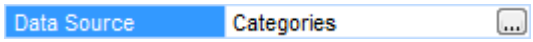
3. Put the **DataBand** on a page of a report template.



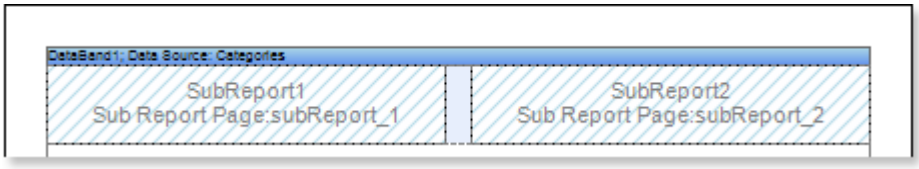
4. Edit **DataBand**:
   4.1. Align the **DataBand** by height;
   4.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
   4.3. Change the **DataBand** background color;
   4.4. Enable **Borders** for the **DataBand**, if required;

4.5. Change the border color.

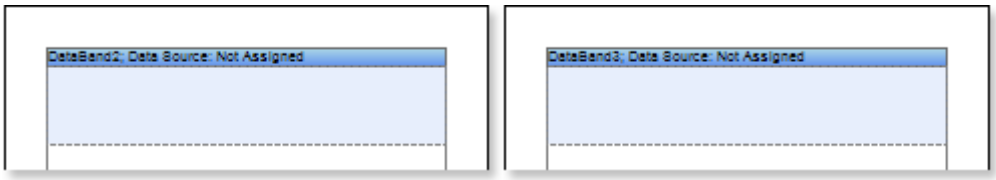
5. Define the data source for the **DataBand** using the **Data Source** property. For example, define the **Categories** data source for the **DataBand**:



6. Put **Sub-Report** components in the **DataBand**;
7. Edit the **Sub-Report** components:
   7.1. Stretch the **Sub-Report** components as seen on the picture below;
   7.2. Change the value of properties of **Sub-Reports**. For example, set the **Keep Sub-Report Together** property to **true**, if you want the sub-report to be kept together;;
   7.3. Change the background color of the components.



8. Go to the sub-report page;
9. Add two **DataBands** to the sub-report page. Add **DataBand1** to the **Sub Report1** and **DataBand2** to the **Sub Report2**;

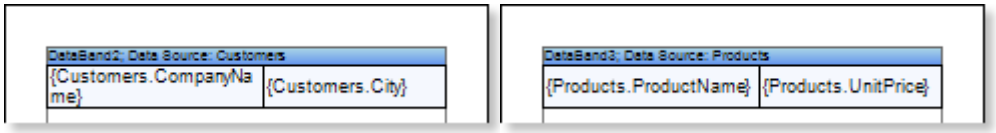


10. Edit the **DataBands**:
    10.1. Align the **DataBands** vertically;
    10.2. Change values of properties of the **DataBands**.
    10.3. Change background color of the band;
    10.4. Set **Borders**, if necessary;
    10.5. Change the border color.

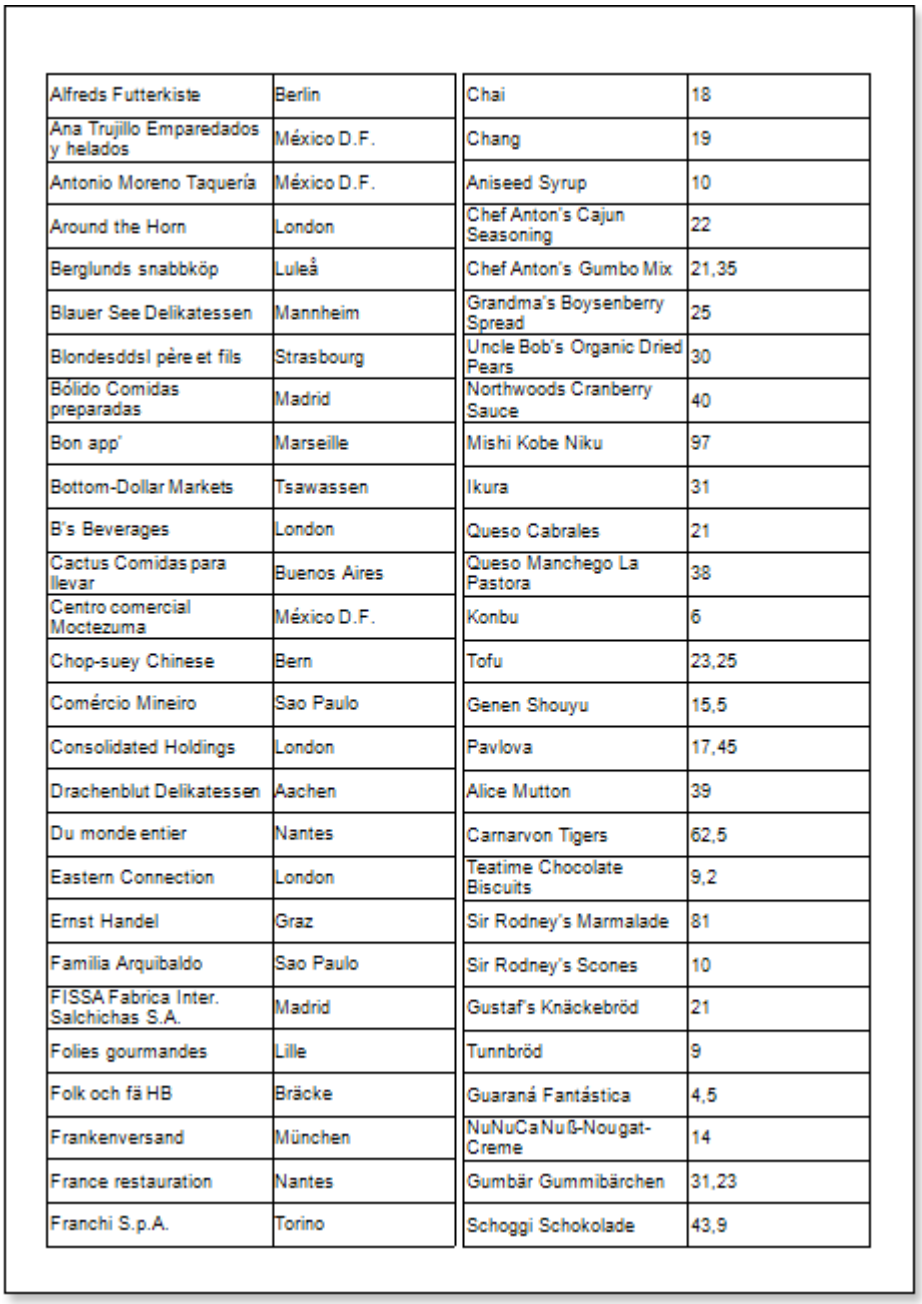
11. Specify the data source for the **DataBand** using the **Data Source** property. For example, set the **Customers** data source for the **DataBand1**, and the **Products** data source for the **DataBand2**:

12. Put text components with expressions in the **DataBands**. Where an expression is a reference to a data field. For example, put the following expressions to the **DataBand1**: **{Customers.CompanyName}** and **{Customers.City}**. put the following expressions to the **DataBand2**: **{Products.ProductName}** and **{Products.UnitPrice}**;

13. Edit **Text** and **TextBoxes**:
    13.1. Drag the text component to the required place in the **DataBand**;
    13.2. Set the text font: size, style, color;
    13.3. Align text component vertically and horizontally;
    13.4. Set the background color of the text component;
    13.5. Align text in the component;
    13.6. Set values of the properties of text components. For example to set the **Word Wrap** property to **true**, if you want the text to be wrapped;
    13.7. Set **Borders** of a text component.
    13.8. Set the border color.



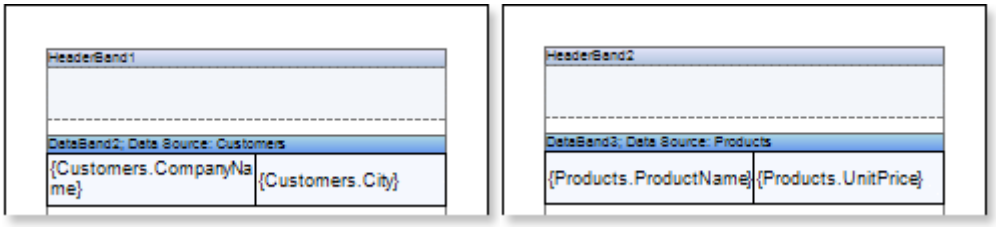
14. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item to see how the report will look like:



Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondesddsl père et fils	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Markets	Tsawassen
B's Beverages	London
Cactus Comidas para llevar	Buenos Aires
Centro comercial Moctezuma	México D.F.
Chop-suey Chinese	Bern
Comércio Mineiro	Sao Paulo
Consolidated Holdings	London
Drachenblut Delikatessen	Aachen
Du monde entier	Nantes
Eastern Connection	London
Ernst Handel	Graz
Familia Arquibaldo	Sao Paulo
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Folies gourmandes	Lille
Folk och fä HB	Bräcke
Frankenversand	München
France restauration	Nantes
Franchi S.p.A.	Torino

Chai	18
Chang	19
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Uncle Bob's Organic Dried Pears	30
Northwoods Cranberry Sauce	40
Mishi Kobe Niku	97
Ikura	31
Queso Cabrales	21
Queso Manchego La Pastora	38
Konbu	6
Tofu	23,25
Genen Shouyu	15,5
Pavlova	17,45
Alice Mutton	39
Carnarvon Tigers	62,5
Teatime Chocolate Biscuits	9,2
Sir Rodney's Marmalade	81
Sir Rodney's Scones	10
Gustaf's Knäckebröd	21
Tunnbröd	9
Guaraná Fantástica	4,5
NuNuCaNuß-Nougat-Creme	14
Gumbär Gummibärchen	31,23
Schoggi Schokolade	43,9

15. Go back to the report template;
16. If necessary, add some bands to the report template, for example, the **HeaderBand**;
17. Edit this band:
    17.1. Align vertically this band;
    17.2. Set values of the properties of the **HeaderBand**, if necessary;
    17.3. Set the background color;
    17.4. Set **Borders** of a text component.
    17.5. Set the border color.



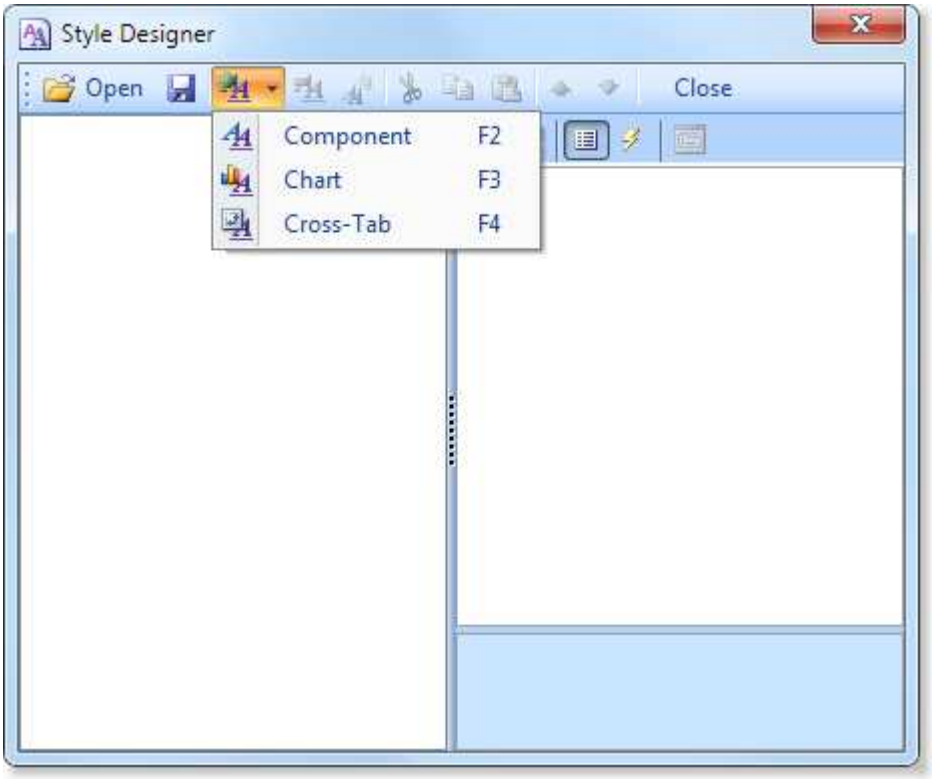
18. Put a text component with expression where the expression of the text component in the **HeaderBand** will be the page title.
19. Edit the text component:
    19.1. Drag the text component to the required place in the band;
    19.2. Set the text font: size, style, color;
    19.3. Align text component vertically and horizontally;
    19.4. Set the background color of the text component;
    19.5. Align text in the component;
    19.6. Set values of the properties of text components;
    19.7. Set **Borders** of a text component.
    19.8. Set the border color.

20. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item to see how the report will look like:

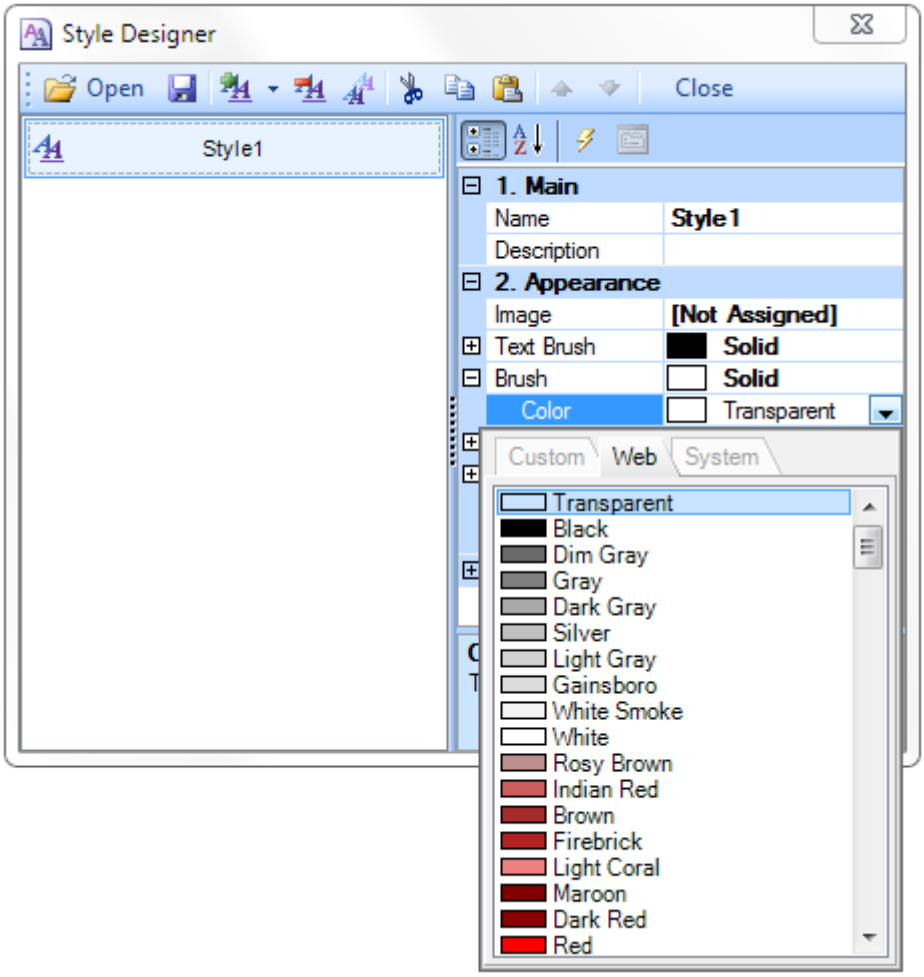
CompanyName	City	ProductName	UnitPrice
Alfreds Futterkiste	Berlin	Chai	18
Ana Trujillo Emparedados y helados	México D.F.	Chang	19
Antonio Moreno Taquería	México D.F.	Aniseed Syrup	10
Around the Horn	London	Chef Anton's Cajun Seasoning	22
Berglunds snabbköp	Luleå	Chef Anton's Gumbo Mix	21,35
Blauer See Delikatessen	Mannheim	Grandma's Boysenberry Spread	25
Blondesddsl père et fils	Strasbourg	Uncle Bob's Organic Dried Pears	30
Bólido Comidas preparadas	Madrid	Northwoods Cranberry Sauce	40
Bon app'	Marseille	Mishi Kobe Niku	97
Bottom-Dollar Markets	Tsawassen	Ikura	31
B's Beverages	London	Queso Cabrales	21
Cactus Comidas para llevar	Buenos Aires	Queso Manchego La Pastora	38
Centro comercial Moctezuma	México D.F.	Konbu	6
Chop-suey Chinese	Bern	Tofu	23,25
Comércio Mineiro	Sao Paulo	Genen Shouyu	15,5
Consolidated Holdings	London	Pavlova	17,45
Drachenblut Delikatessen	Aachen	Alice Mutton	39
Du monde entier	Nantes	Carnarvon Tigers	62,5
Eastern Connection	London	Teatime Chocolate Biscuits	9,2
Ernst Handel	Graz	Sir Rodney's Marmalade	81
Familia Arquibaldo	Sao Paulo	Sir Rodney's Scones	10
FISSA Fabrica Inter. Salchichas S.A.	Madrid	Gustaf's Knäckebröd	21
Folies gourmandes	Lille	Tunnbröd	9
Folk och fä HB	Bräcke	Guaraná Fantástica	4,5
Frankenversand	München	NuNuCa Nuß-Nougat-Creme	14

Adding styles

1. Go back to the report template;
2. Select the sub-report;
3. Select the **DataBand**;
4. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**.



Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

5. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered report with sub-report and alternative color of rows:

CompanyName	City	ProductName	UnitPrice
Alfreds Futterkiste	Berlin	Chai	18
Ana Trujillo Emparedados y helados	México D.F.	Chang	19
Antonio Moreno Taquería	México D.F.	Aniseed Syrup	10
Around the Horn	London	Chef Anton's Cajun Seasoning	22
Berglunds snabbköp	Luleå	Chef Anton's Gumbo Mix	21,35
Blauer See Delikatessen	Mannheim	Grandma's Boysenberry Spread	25
Blondesddsl père et fils	Strasbourg	Uncle Bob's Organic Dried Pears	30
Bólido Comidas preparadas	Madrid	Northwoods Cranberry Sauce	40
Bon app'	Marseille	Mishi Kobe Niku	97
Bottom-Dollar Markets	Tsawassen	Ikura	31
B's Beverages	London	Queso Cabrales	21
Cactus Comidas para llevar	Buenos Aires	Queso Manchego La Pastora	38
Centro comercial Moctezuma	México D.F.	Konbu	6
Chop-suey Chinese	Bern	Tofu	23,25
Comércio Mineiro	Sao Paulo	Genen Shouyu	15,5
Consolidated Holdings	London	Pavlova	17,45
Drachenblut Delikatessen	Aachen	Alice Mutton	39
Du monde entier	Nantes	Carnarvon Tigers	62,5
Eastern Connection	London	Teatime Chocolate Biscuits	9,2
Ernst Handel	Graz	Sir Rodney's Marmalade	81
Familia Arquibaldo	Sao Paulo	Sir Rodney's Scones	10
FISSA Fabrica Inter. Salchichas S.A.	Madrid	Gustaf's Knäckebröd	21
Folies gourmandes	Lille	Tunnbröd	9
Folk och fä HB	Bräcke	Guaraná Fantástica	4,5
Frankenversand	München	NuNuCa Nuß-Nougat-Creme	14

Master-Detail Report and Sub-Reports

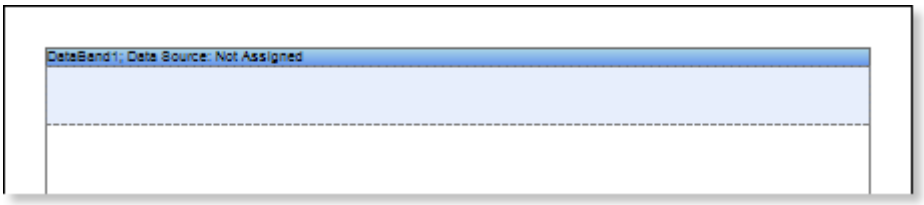
For better understanding this step-by-step instruction, please watch the video file.

Do the following steps to create a **Master-Detail** report with sub-reports:

1. Run the designer;
2. Connect data:
    2.1. Create **New Connection**;

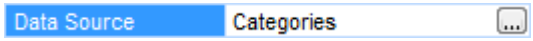
2.2. Create **New Data Source**;

3. Create **Relation** between data sources. If the relation will not be created and/or the **Relation** property of the **Detail** data source will not be filled, then, for **Master** entry, all **Detail** entries will be output;
4. Put the **DataBand1** on a page of a report template:



5. Edit **DataBand1**:
   5.1. Align the **DataBand1** by height;
   5.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
   5.3. Change the **DataBand1** background color;
   5.4. Enable **Borders** for the **DataBand1**, if required;
   5.5. Change the border color.

6. Define the data source for the **DataBand1** using the **Data Source** property. For example, define the **Categories** data source for the **DataBand2:**

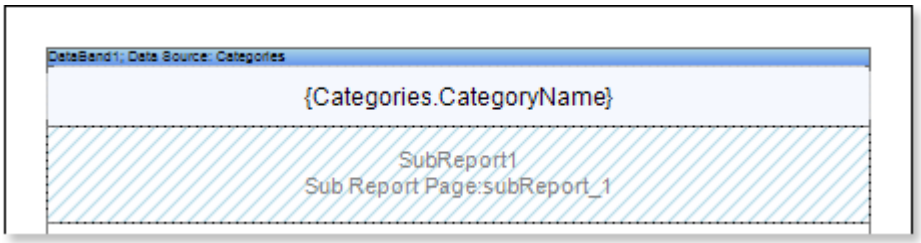


7. Put text components with expressions in the **DataBand1**. Where an expression is a reference to a data field. For example, put the text component with the following expression in the **DataBand1** (**Master** component): **{Categories.CategoryName}**;

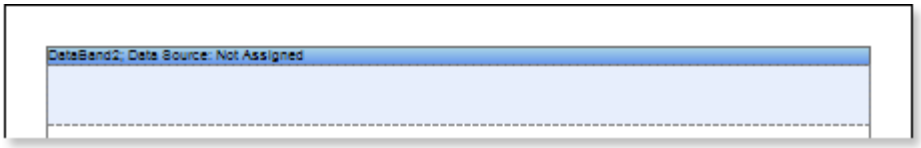
8. Edit **Text** and **TextBoxes**:
   8.1. Drag the text component to the required place in the **DataBand1**;
   8.2. Set the text font: size, style, color;
   8.3. Align text component vertically and horizontally;
   8.4. Set the background color of the text component;
   8.5. Align text in the component;
   8.6. Set values of the properties of text components. For example to set the **Word Wrap** property to **true**, if you want the text to be wrapped;
   8.7. Set **Borders** of a text component.
   8.8. Set the border color.

9. Put a **Sub-Report** component in the **DataBand1**;

10. Edit the **Sub-Report** components:
   10.1. Stretch the **Sub-Report** components as seen on the picture below;
   10.2. Change the value of properties of **Sub-Reports**. For example, set the **Keep Sub-Report Together** property to **true**, if you want the sub-report to be kept together;;
   10.3. Change the background color of the components.

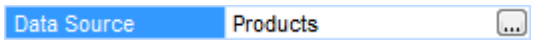


11. Go to the sub-report page;
12. Add to the **DataBand2** to the sub-report page.

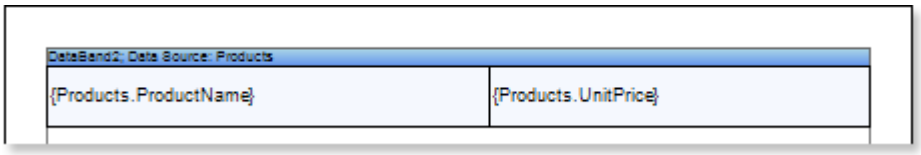


13. Edit **DataBand2**:
    13.1. Align the **DataBand2** by height;
    13.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
    13.3. Change the **DataBand2** background color;
    13.4. Enable **Borders** for the **DataBand2**, if required;
    13.5. Change the border color.

14. Define the data source for the **DataBand1** using the **Data Source** property. For example, define the **Products** data source for the **DataBand2**:



15. Define the **Master** component in a report. In our case set the **DataBand1** as a **Master** component for the **DataBand2**;
16. Fill the **Data Relation** property of the **DataBand**, that is the **Detail** component, i.e. in this case for the **DataBand2**;
17. Put text components with expressions in the **DataBand1**. Where an expression is a reference to a data field. For example, put the text component with the following expression in the **DataBand2**: **{Products.ProductName}** and **{Products.UnitPrice}**;
18. Edit **Text** and **TextBoxes**:
    18.1. Drag the text component to the required place in the **DataBand2**;
    18.2. Set the text font: size, style, color;
    18.3. Align text component vertically and horizontally;
    18.4. Set the background color of the text component;
    18.5. Align text in the component;
    18.6. Set values of the properties of text components. For example to set the **Word Wrap** property to **true**, if you want the text to be wrapped;
    18.7. Set **Borders** of a text component.
    18.8. Set the border color.



19. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item to see how the report will look like:

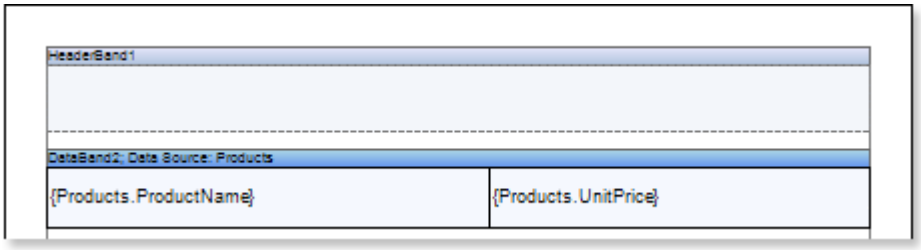
Beverages

Chai	18
Chang	19
Guaraná Fantástica	4,5
Sasquatch Ale	14
Steeleye Stout	18
Côte de Blaye	263,5
Chartreuse verte	18
Ipoh Coffee	46
Laughing Lumberjack Lager	14
Outback Lager	15
Rhönbräu Klosterbier	7,75
Lakkalikööri	18

Condiments

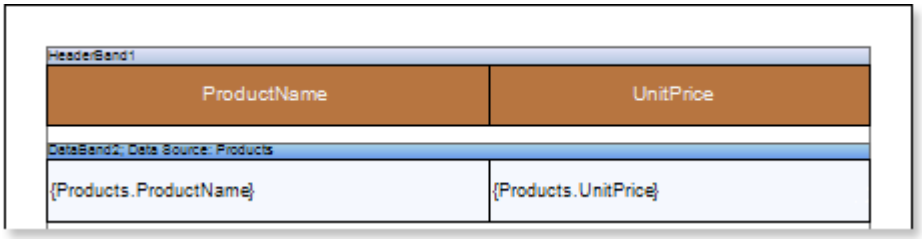
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Northwoods Cranberry Sauce	40
Genen Shouyu	15,5
Gula Malacca	19,45

20. Go back to the report template;
21. If necessary, add some bands to the report template, for example, the **HeaderBand**;
22. Edit this band:
    22.1. Align vertically this band;
    22.2. Set values of the properties of the **HeaderBand**, if necessary;
    22.3. Set the background color;
    22.4. Set **Borders** of a text component.
    22.5. Set the border color.



23. Put a text component with expression where the expression of the text component in the **HeaderBand** will be the page title.

24. Edit the text component:
    24.1. Drag the text component to the required place in the band;
    24.2. Set the text font: size, style, color;
    24.3. Align text component vertically and horizontally;
    24.4. Set the background color of the text component;
    24.5. Align text in the component;
    24.6. Set values of the properties of text components;
    24.7. Set **Borders** of a text component.
    24.8. Set the border color.



25. Click the **Preview** button or call **Viewer**, using an **F5** hot key or the **Preview** menu item to see how the report will look like:

Beverages

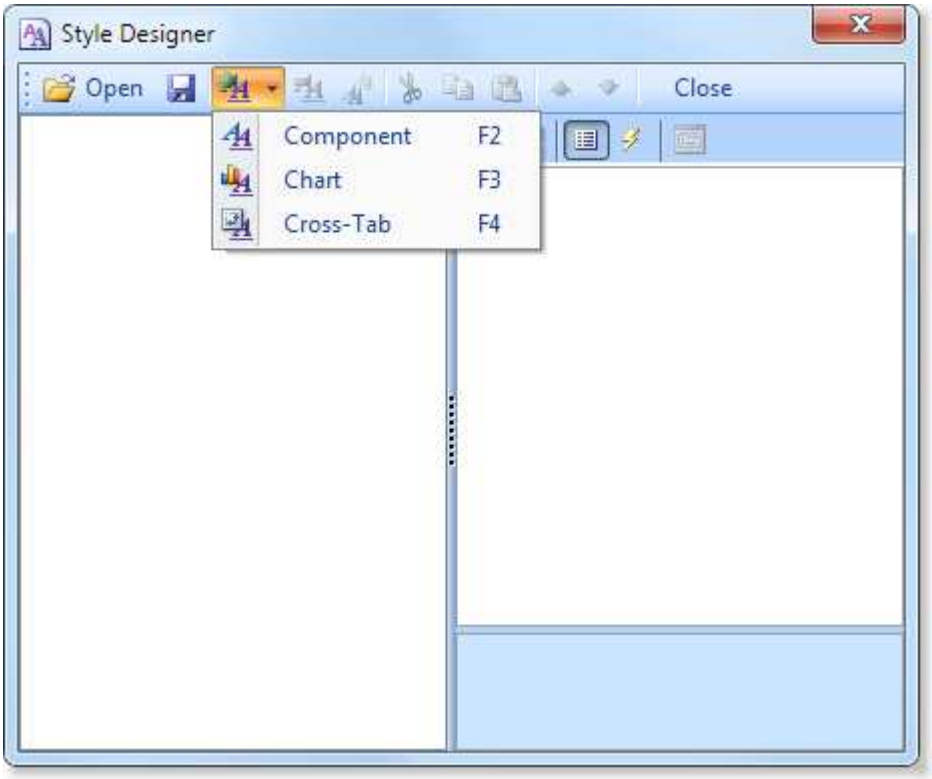
ProductName	UnitPrice
Chai	18
Chang	19
Guaraná Fantástica	4,5
Sasquatch Ale	14
Steeleye Stout	18
Côte de Blaye	263,5
Chartreuse verte	18
Ipoh Coffee	46
Laughing Lumberjack Lager	14
Outback Lager	15
Rhönbräu Klosterbier	7,75
Lakkalikööri	18

Condiments

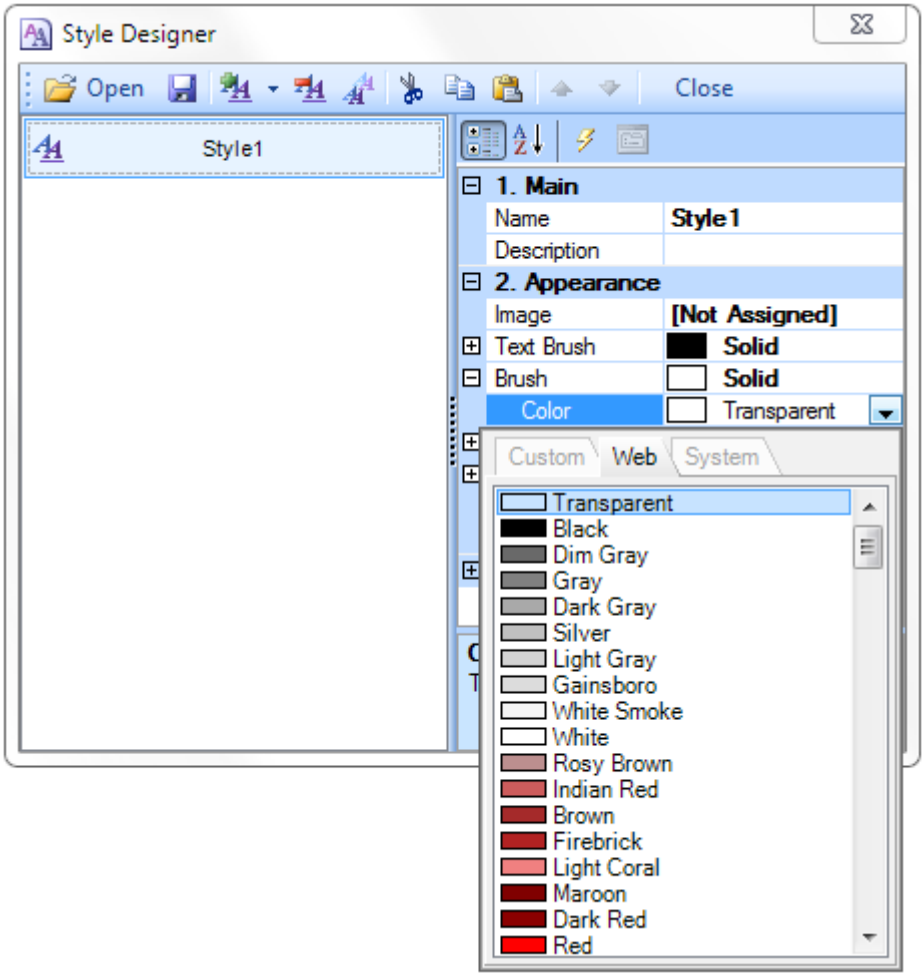
ProductName	UnitPrice
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Northwoods Cranberry Sauce	40
Genen Shouyu	15,5
Gula Malacca	19,45

Adding styles

1. Go back to the report template;
2. Select the sub-report;
3. Select the **DataBand**;
4. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**.



Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

5. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered "**master-detail report with sub-report**" with alternative color of rows:

Beverages

ProductName	UnitPrice
Chai	18
Chang	19
Guaraná Fantástica	4,5
Sasquatch Ale	14
Steeleye Stout	18
Côte de Blaye	263,5
Chartreuse verte	18
Ipoh Coffee	46
Laughing Lumberjack Lager	14
Outback Lager	15
Rhönbräu Klosterbier	7,75
Lakkalikööri	18

Condiments

ProductName	UnitPrice
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Northwoods Cranberry Sauce	40
Genen Shouyu	15,5
Gula Malacca	19,45

Report with EmptyBand

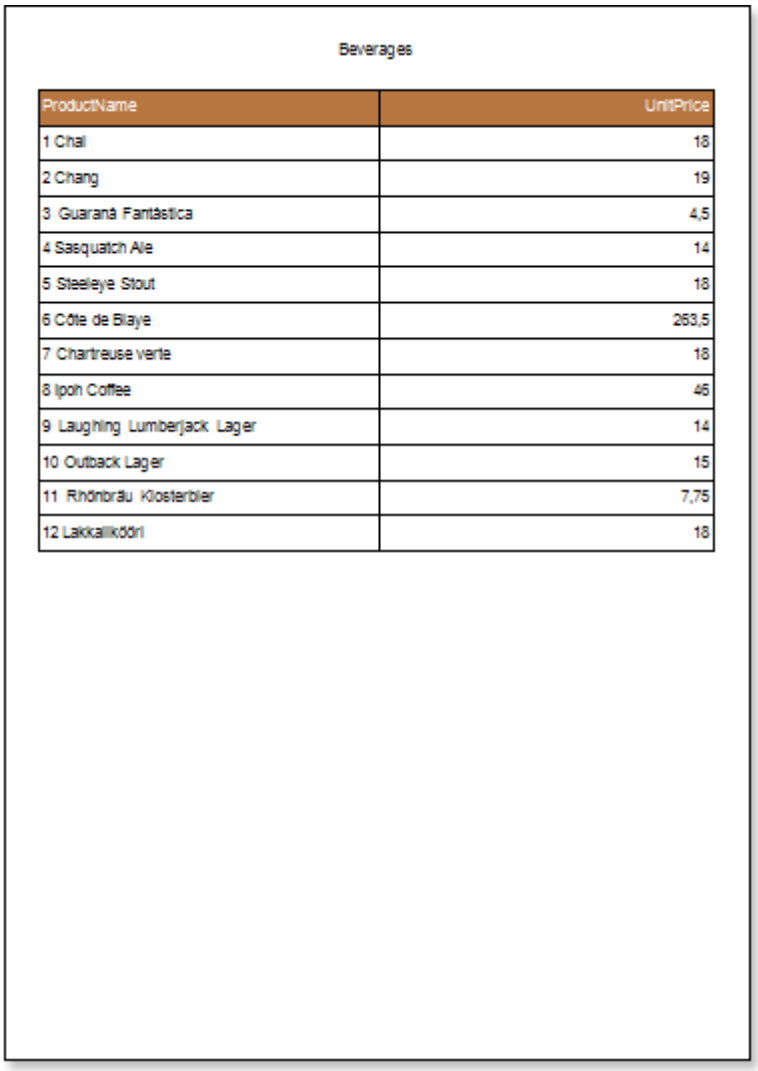
For better understanding this step-by-step instruction, please watch the video file.

The **EmptyBand** is used to fill free space at the bottom of a page. This tutorial describes how to create a report with the **EmptyBand**:

1. Run the designer;

2. Connect the data:
   2.1. Create a **New Connection**;
   2.2. Create a **New Data Source**;

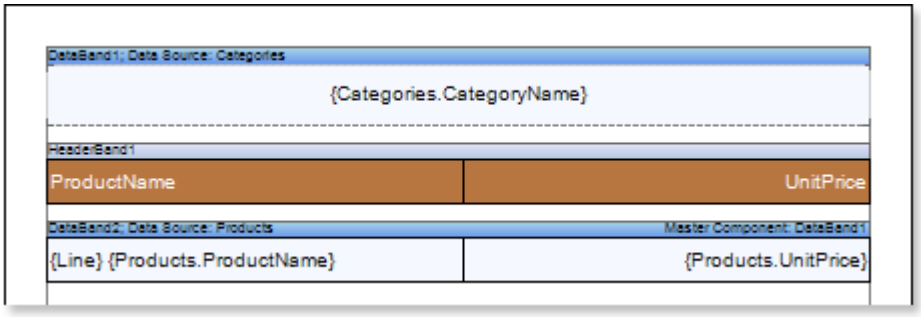
3. Design a report or load a previously saved one. Consider creating a report with the **EmptyBand** on the base of the **Master-Detail** report. Suppose there is a **Master-Detail** report in which data is printed on half of a page, then to fill the empty space you can use the **EmptyBand**. The picture below shows the rendered **Master-Detail** report:



Beverages

ProductName	UnitPrice
1 Chai	18
2 Chang	19
3 Guaraná Fantástica	4,5
4 Sasquatch Ale	14
5 Steeleye Stout	18
6 Côte de Blaye	263,5
7 Chartreuse verte	18
8 Ipoh Coffee	46
9 Laughing Lumberjack Lager	14
10 Outback Lager	15
11 Rhönbräu Klosterbier	7,75
12 Lakkalikööri	18

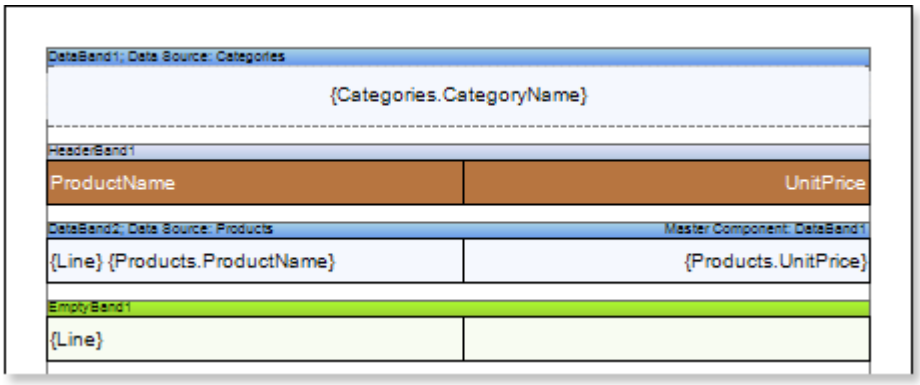
4. Go back to the **Master-Detail** report template.



5. Add the **EmptyBand** in the report template;
6. Edit the **EmptyBand**:
6.1. Align it by height;
6.2. Change the value of required properties. For example, set the **CanGrow** property to **true**, if you want the band be grown;
6.3. Set the background color of the **EmptyBand**;

6.4. If necessary, set **Borders** of the EmptyBand);

7. Put text components with an expression in the **EmptyBand**. Where the expression is a reference to the data field. For example, put a text component with the expression: **{Line}**;
8. Edit **Text** and **TextBox** component:
   8.1. Drag and drop the text component in the **EmptyBand**;
   8.2. Change parameters of the text font: size, type, color;
   8.3. Align the text component by width and height;
   8.4. Change the background of the text component;
   8.5. Align text in the text component;
   8.6. Change the value of properties of the text component. For example, set the **WordWrap** property to **true**, if you need a text to be wrapped;
   8.7. Enable **Borders** for the text component, if required.
   8.8. Change the border color.



9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of the report:

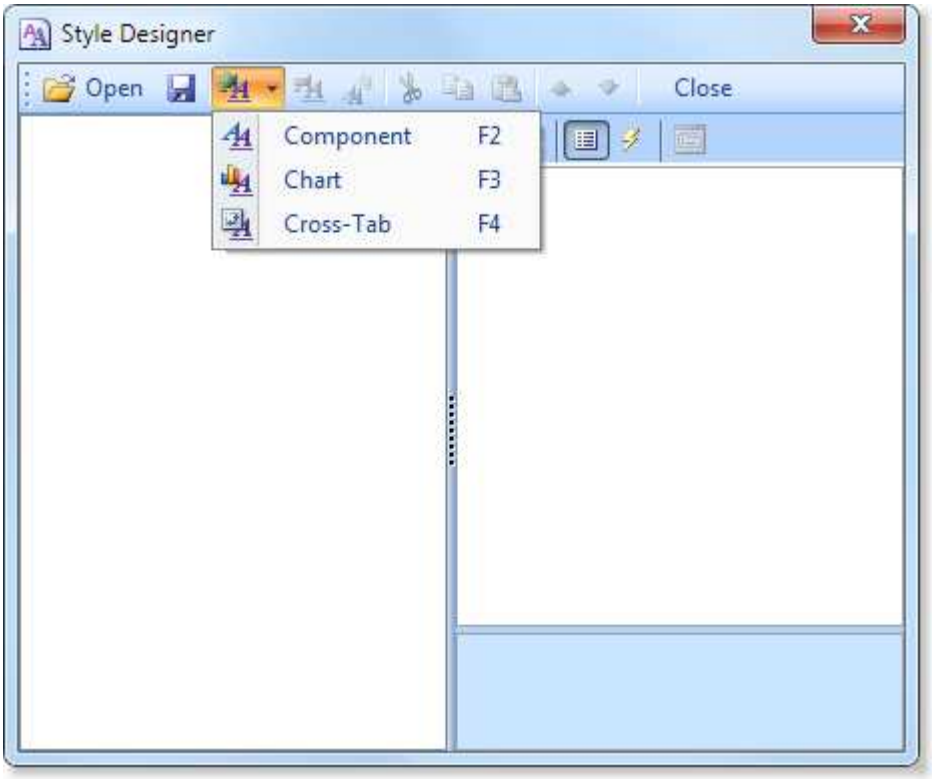
Beverages

ProductName	UnitPrice
1 Chai	18
2 Chang	19
3 Guaraná Fantástica	4,5
4 Sasquatch Ale	14
5 Steeleye Stout	18
6 Côte de Blaye	263,5
7 Chartreuse verte	18
8 Ipoh Coffee	46
9 Laughing Lumberjack Lager	14
10 Outback Lager	15
11 Rhönbräu Klosterbier	7,75
12 Lakkalikööri	18
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	

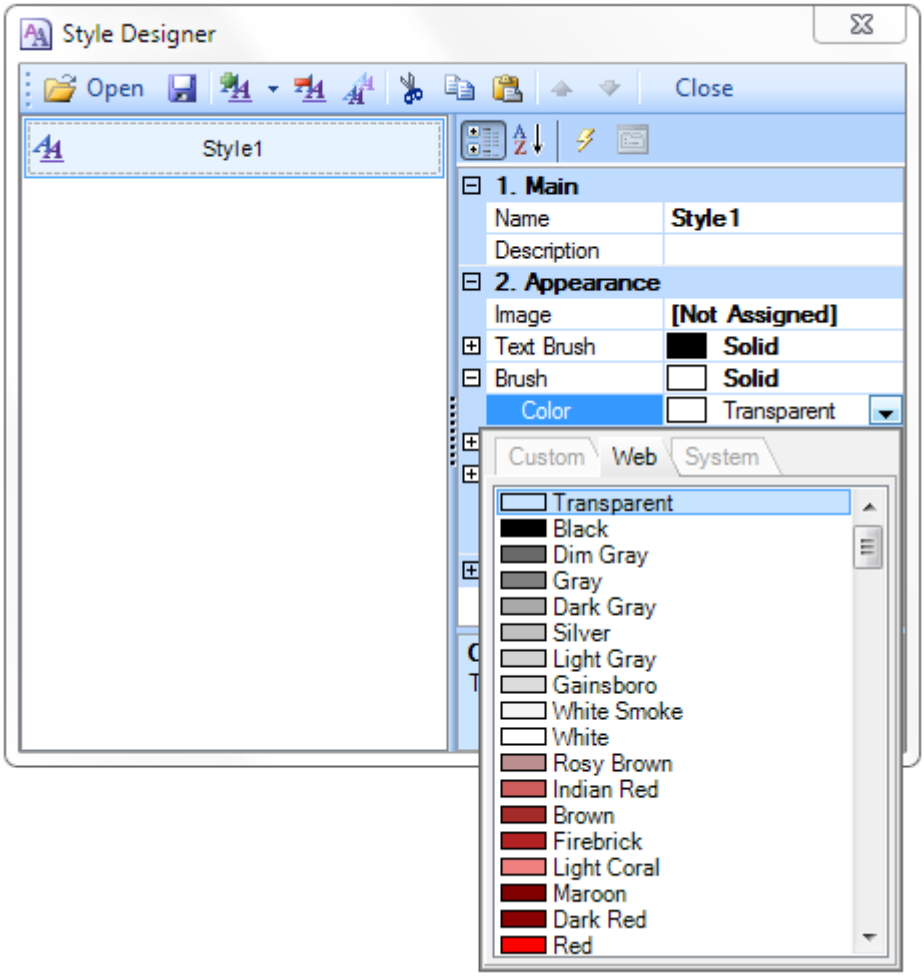
As can be seen in the picture above blank lines will be numbered and output in the report.

Adding styles

1. Go back to the report template;
2. Select the **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**.

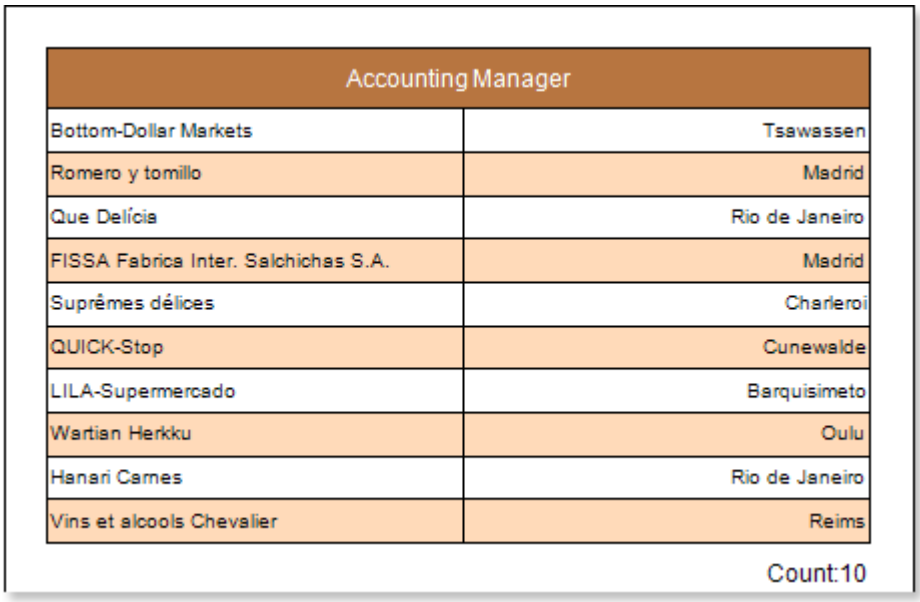


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

5. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows a sample of a rendered report:



Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delícia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	Reims

Count:10

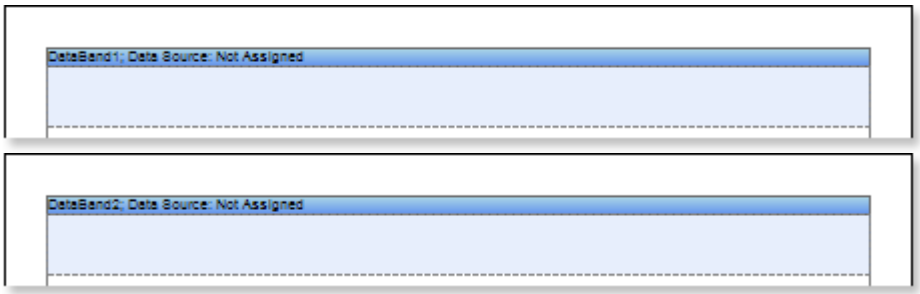
Drill-Down Report Using Page in Report

For better understanding this step-by-step instruction, please watch the video file.

The **Drill-Down** report using the pages in the report is an interactive report in what detailed data are placed on the page of a report and the relation between master and detailed data in the report is organized with the help of the **Interaction.Drill-Down Page** property. This type of report must contain at least two pages: a one with master data, and a second with detailed ones. Follow the steps below in order to design the report:

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Put the **DataBand1** on the **Page1** and **DataBand2** on **Page2** of a report. In this case, the master data will be located on the first page, and detailed - on the second page.

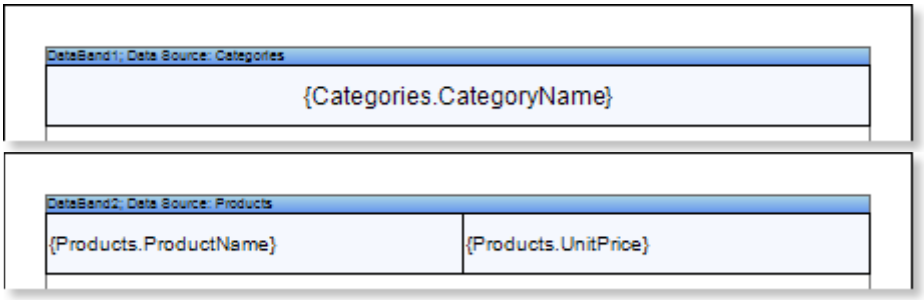


4. Edit **DataBand1** and **DataBand2**:
    4.1. Align the **DataBands** vertically;
    4.2. Change the value of the required properties;

4.3. Change the background color of the **DataBand**;
4.4. If necessary, set **Borders** of the **DataBand**;

5. Define a data source for **DataBands** using the **Data Source** property:

6. Put the text components with expressions. Where the expression is a reference to the data field. For example: put the text component with the **{Categories.CategoryName}** expression in the **DataBand1**, and put two text components with the **{Products.ProductName}** and **{Products.UnitePrice}** expressions in the **DataBand2**;
7. Edit text and text components located in the **DataBands**:
7.1. Drag the text component to the required place in the **DataBands**;
7.2. Align the text in a text component;
7.3. Change the value of the required properties. For example to set the **Word Wrap** property to **true**, if you want the text be wrapped;
7.4. Set **Borders** of a text component, if required.
7.5. Change the border color.



8. Select a text component in the **DataBand1**;
9. Set the **Interaction.Drill-Down Enabled** to **true**;
10. Set the Interaction.Drill-Down Page to **Page2**;
11. Set filter in the **DataBand2**, in this case, we specify the **(int) this ["CategoryID"] == Products.CategoryID** expression;
12. Edit **Drill-Down Parameter 1** for the text component of the **DataBand 1**:
12.1. The **Name** property should be set to **CategoryID**;
12.2. The **Expression** property should be set to **Categories.CategoryID**;

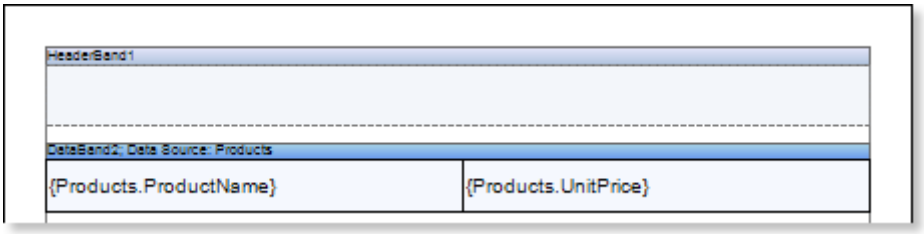
13. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of a report:

Beverages
Condiments
Confections
Dairy Products
Grains/Cereals
Meat/Poultry
Produce
Seafood

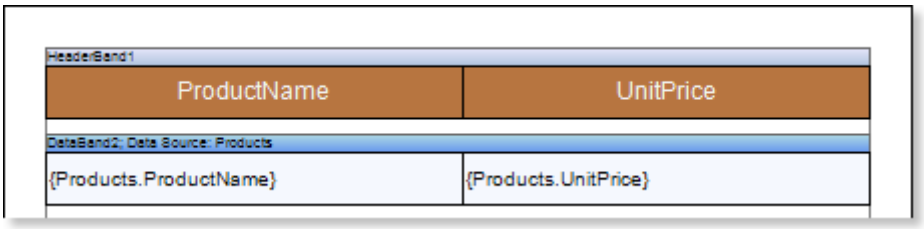
When you click the **Beverages**, the user will see the detailed data that correspond to filtering conditions and parameters of detailing. The picture below shows a page of a rendered report with detailed data of the **Beverages** entry:

Chai	18
Chang	19
Guaraná Fantástica	4,5
Sasquatch Ale	14
Steeleye Stout	18
Côte de Blaye	263,5
Chartreuse verte	18
Ipoh Coffee	46
Laughing Lumberjack Lage	14
Outback Lager	15
Rhönbräu Klosterbier	7,75
Lakkalikööri	18

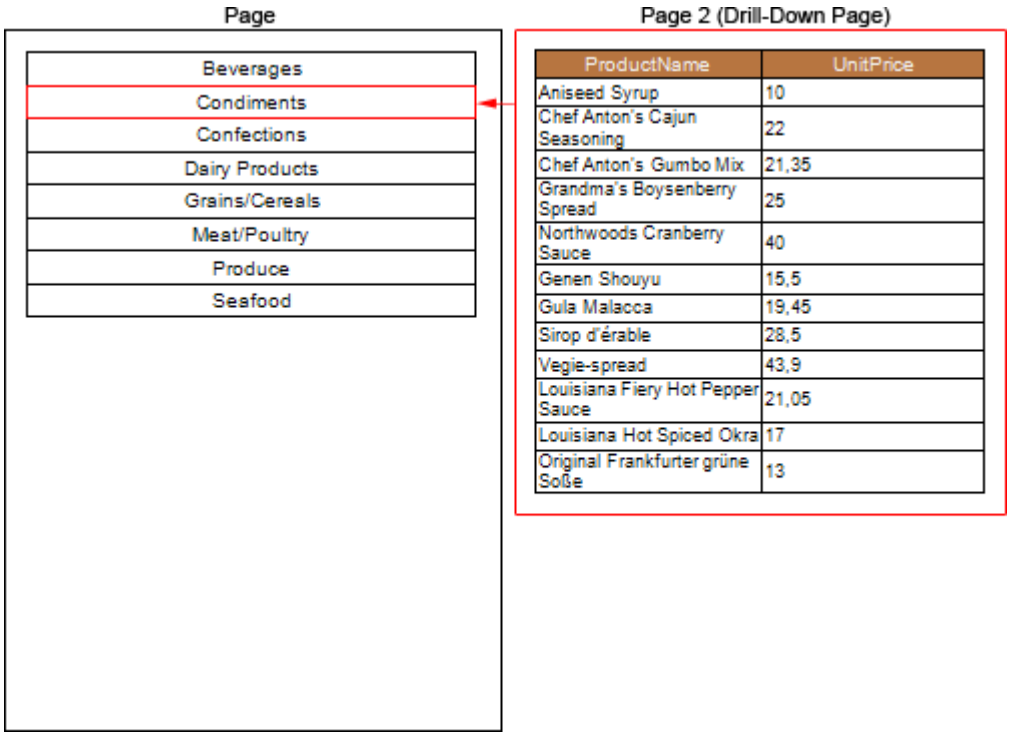
14. Go back to the report template;
15. Add other bands to a report template, for example, add the **HeaderBand** to the **Page2** of a report;
16. Edit the band:
    16.1. Align it by height;
    16.2. Change values of properties, if required;
    16.3. Change the background of the band;
    16.4. Enable **Borders**, if required;
    16.5. Set the border color.



17. Put a text component with an expression in this band. The expression in the text component is a header in the **HeaderBand**.
18. Edit text and text components:
    18.1. Drag and drop the text component in the band;
    18.2. Change font options: size, type, color;
    18.3. Align text component by height and width;
    18.4. Change the background of the text component;
    18.5. Align text in the text component;
    18.6. Change values of text component properties, if required;
    18.7. Enable **Borders** of the text component, if required;
    18.8. Set the border color.



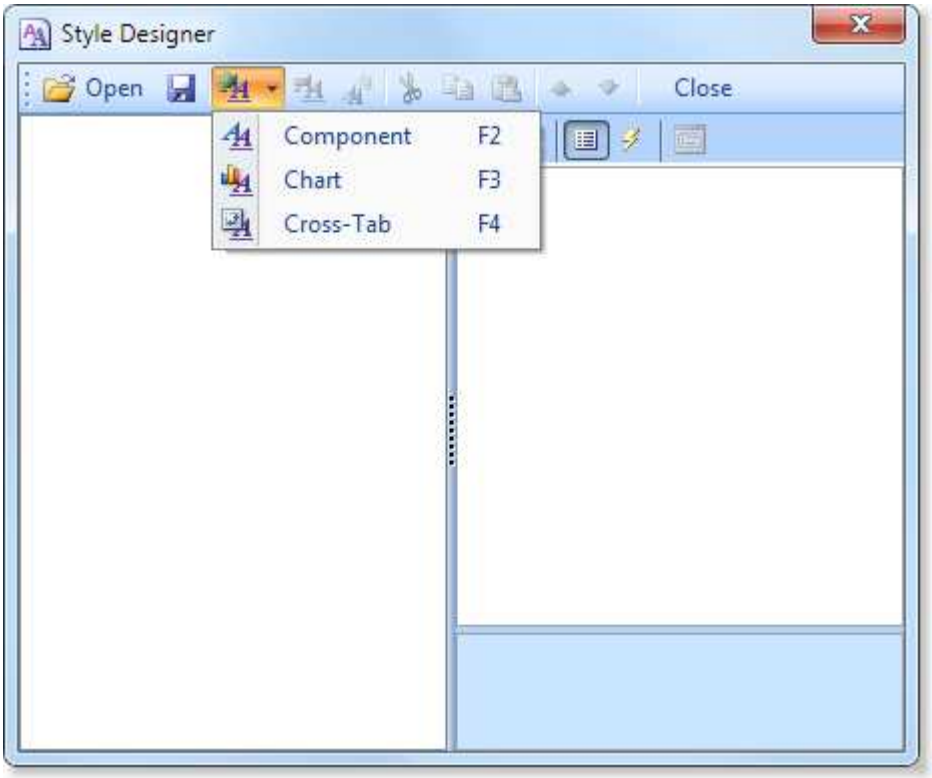
19. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows the structure of a report, i.e. shows the ratio of detailed data to the master **Condiments** entry:



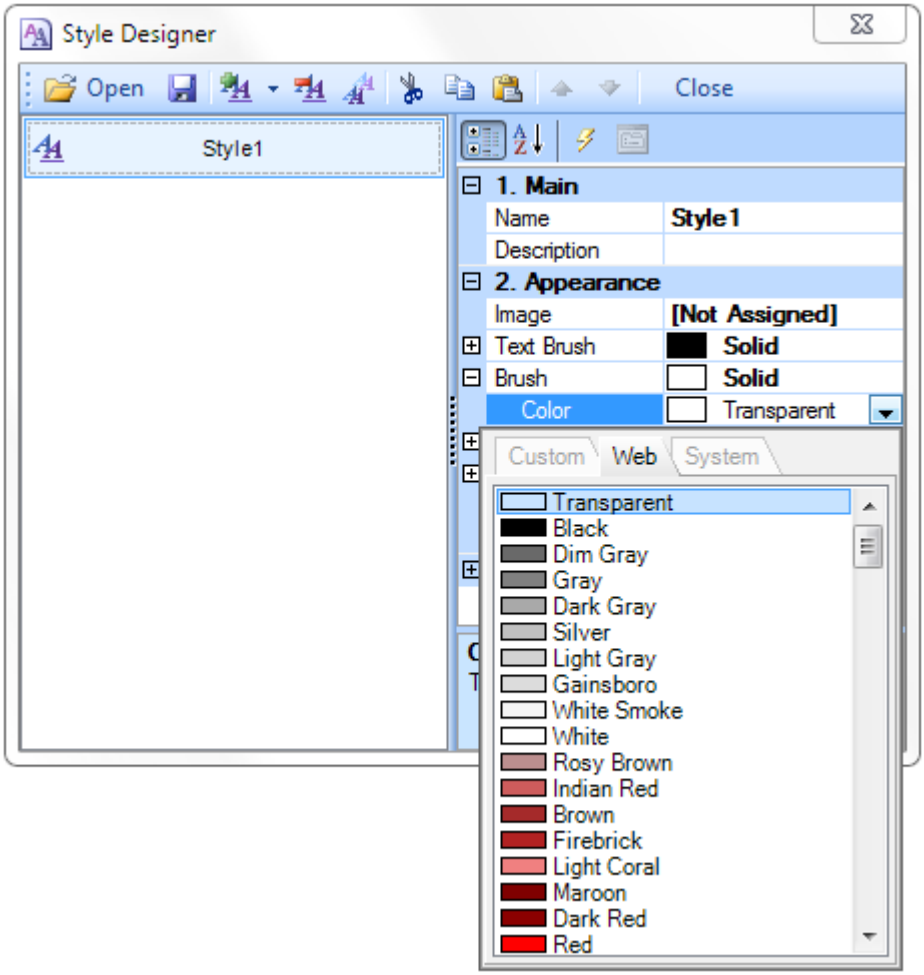
Adding styles

1. Go back to the report template;

2. Select the **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**.

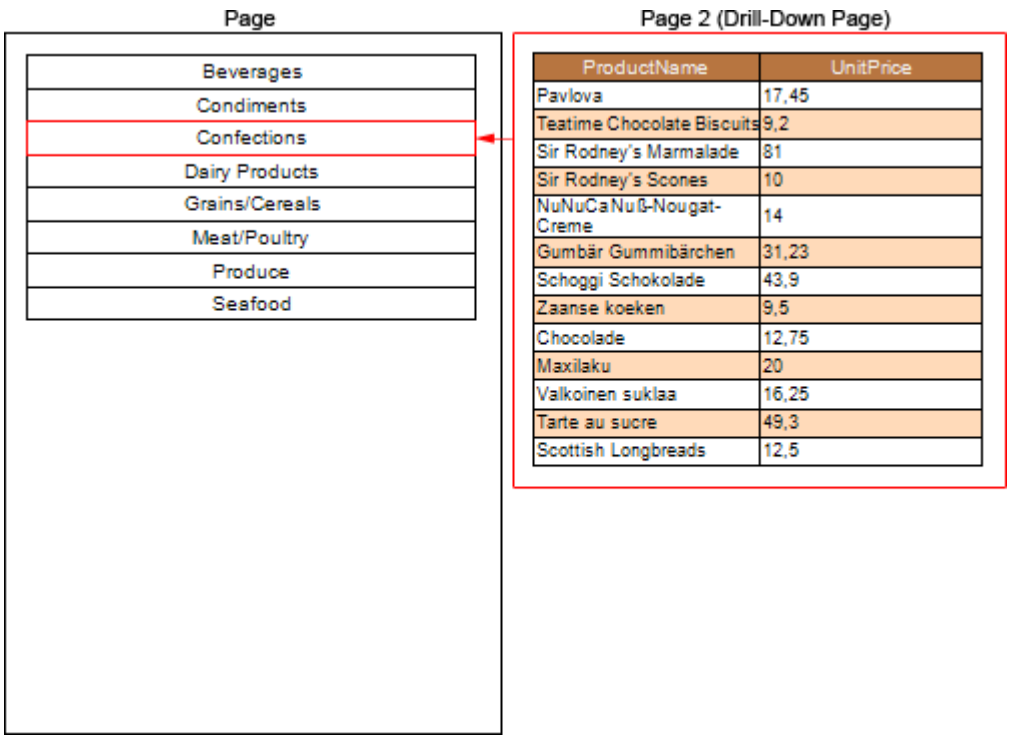


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

5. The picture below shows the structure of a report, i.e. shows the ratio of detailed data to the **Confections** master entry with different styles even/odd rows of the **DataBand**:



Drill-Down Report Using External Report

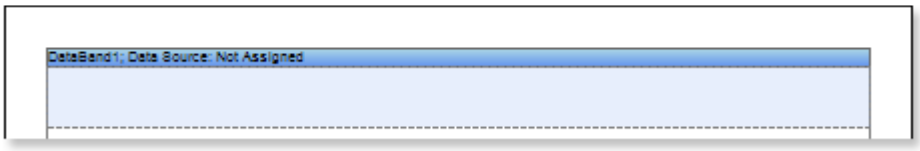
For better understanding this step-by-step instruction, please watch the video file.

Drill-Down report using external report is an interactive report in what detailed data are placed in an external report and the relationship between master and detailed data in reports is organized using the **Interaction.Drill-Down Report** property. Follow the steps below in order to design the report:

Creating a report with detailed data

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

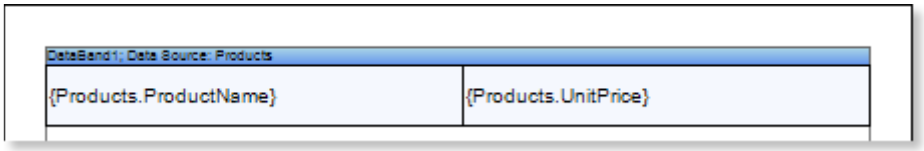
3. Put the **DataBand** on a report page:



4. Edit the **DataBand**:
    4.1. Align the **DataBand**;
    4.2. Change the values of properties;
    4.3. Set the background color of the **DataBand**;
    4.4. Set **Borders**, if required;
    4.5. Set the border color.

5. Specify the data source in **DataBand** using the **Data Source** property:

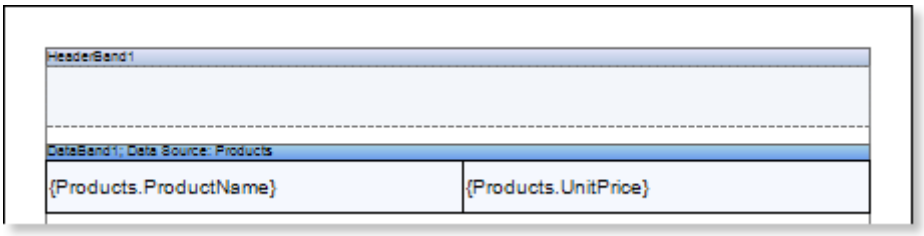
6. Put text components with expressions in the **DataBand**. Where the expression is a reference to the data field. For example: put two text components with the **{Products.ProductName}** and **{Products.UnitePrice}** expressions in the **DataBand**;
7. Edit text and text components located in the **DataBand**:
    7.1. Drag the text component to the required place in the **DataBand**;
    7.2. Align the text in a text component;
    7.3. Change the value of the required properties. For example to set the **Word Wrap** property to **true**, if you want the text be wrapped;
    7.4. Set **Borders** of a text component, if required;
    7.5. Change the border color.



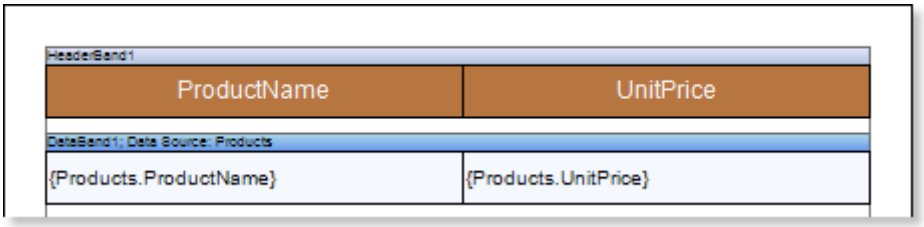
8. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of a report:

Chai	18
Chang	19
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Uncle Bob's Organic Dried Pears	30
Northwoods Cranberry Sauce	40
Mishi Kobe Niku	97
Ikura	31
Queso Cabrales	21
Queso Manchego La Pastora	38
Konbu	6
Tofu	23,25
Genen Shouyu	15,5
Pavlova	17,45
Alice Mutton	39
Carnarvon Tigers	62,5
Teatime Chocolate Biscuits	9,2
Sir Rodney's Marmalade	81
Sir Rodney's Scones	10
Gustaf's Knäckebröd	21
Tunnbröd	9

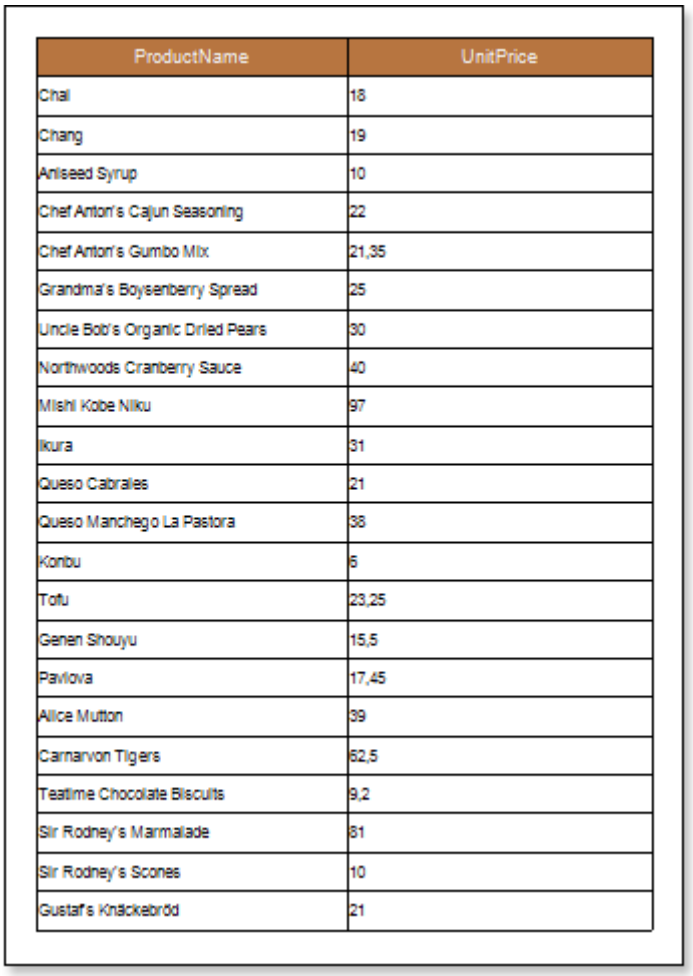
9. Go back to the report template;
10. Add other bands to a report template, for example, add the **HeaderBand** to the report page;
11. Edit the band:
    11.1. Align it by height;
    11.2. Change values of properties, if required;
    11.3. Change the background of the band;
    11.4. Enable **Borders**, if required;
    11.5. Set the border color.



12. Put a text component with an expression in this band. The expression in the text component is a header in the **HeaderBand**.
13. Edit text and text components:
    13.1. Drag and drop the text component in the band;
    13.2. Change font options: size, type, color;
    13.3. Align text component by height and width;
    13.4. Change the background of the text component;
    13.5. Align text in the text component;
    13.6. Change values of text component properties, if required;
    13.7. Enable **Borders** of the text component, if required;
    13.8. Set the border color.



14. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of a report:



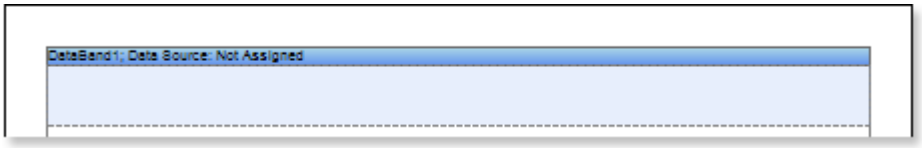
ProductName	UnitPrice
Chai	18
Chang	19
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Uncle Bob's Organic Dried Pears	30
Northwoods Cranberry Sauce	40
Mishi Kobe Niku	97
Ikura	31
Queso Cabrales	21
Queso Manchego La Pastora	38
Konbu	6
Tofu	23,25
Genen Shouyu	15,5
Pavlova	17,45
Alice Mutton	39
Carnarvon Tigers	62,5
Teatime Chocolate Biscuits	9,2
Sir Rodney's Marmalade	81
Sir Rodney's Scones	10
Gustaf's Knäckebröd	21

15. Go back to the report template;;
16. Set filtering in the **DataBand**. For example, set the following expression: **CategoryID == Products.CategoryID**;
17. Save the report. For example, save the report with detailed data on a local disk in the root directory D:\\, with the **Drill-Down Report** name, i.e. full path to the file will be **D:\\ Drill-Down Report.mrt**.

Creating a report with master data

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

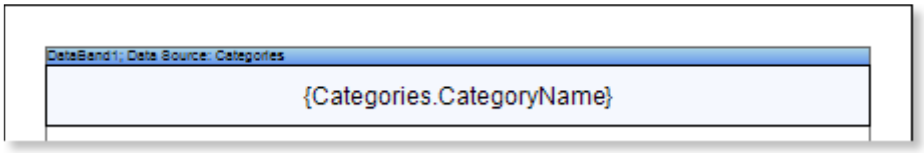
3. Put the **DataBand** on a report page:



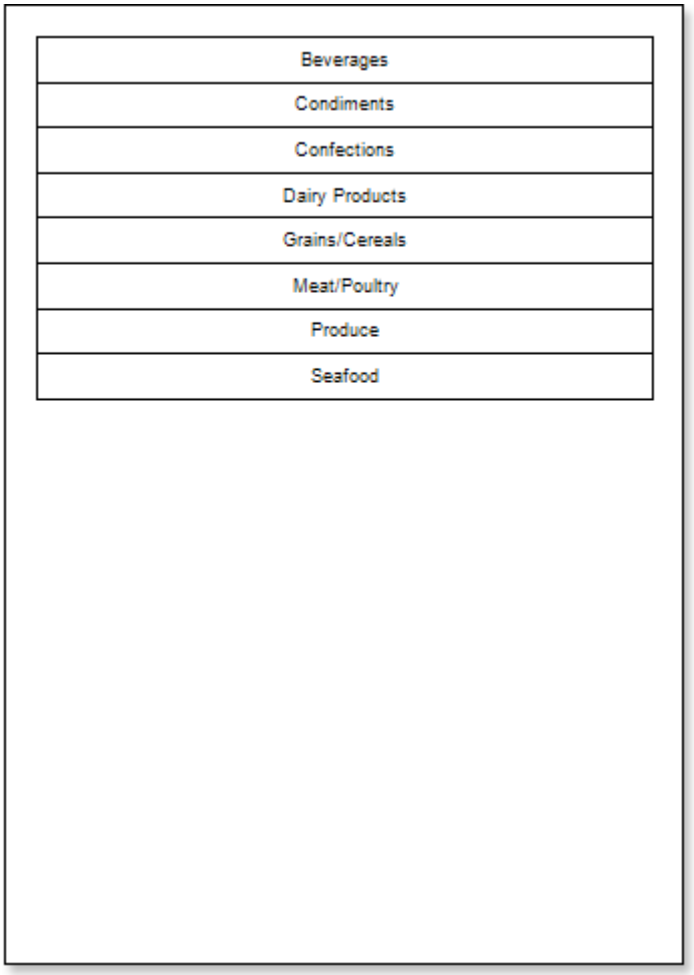
4. Edit the **DataBand**:
   4.1. Align the **DataBand**;
   4.2. Change the values of properties;
   4.3. Set the background color of the **DataBand**;
   4.4. Set **Borders**, if required;
   4.5. Set the border color.

5. Specify the data source in **DataBand** using the **Data Source** property:

6. Put a text component with expressions in the **DataBand**. Where the expression is a reference to the data field. For example: put the text component with the **{Categories.CategoryName}** expression in the **DataBand**;
7. Edit text and text components located in the **DataBand**:
   7.1. Drag the text component to the required place in the **DataBand**;
   7.2. Align the text in a text component;
   7.3. Change the value of the required properties. For example to set the **Word Wrap** property to **true**, if you want the text be wrapped;
   7.4. Set **Borders** of a text component, if required;
   7.5. Change the border color.



8. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of a report:



Creating an interactive report

1. Go back to the report template with the master data;
2. Select a text component in the **DataBand**;
3. Set the **Interaction.Drill-Down Enabled** property to **true**;
4. Set the **Interaction.Drill-Down Report** property. Where the value of this property is the full path to the report with detailed data. In our tutorial, the **Interaction.Drill-Down Report** property will be set to **D:\\Drill-Down Report.mrt**;
5. Edit **Drill-Down Parameter 1**:
    5.1. The **Name** property should be set to **CategoryID**;
    5.2. The **Expression** property should be set to **Categories.CategoryID**;

6. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database. The picture below shows a sample of a report:

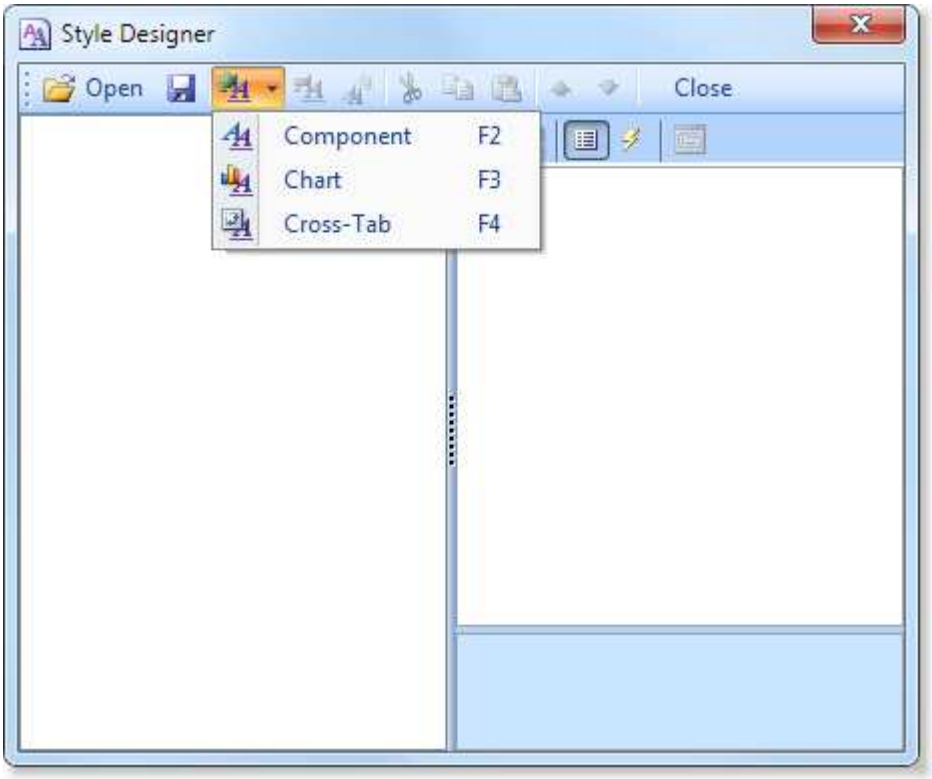
Beverages
Condiments
Confections
Dairy Products
Grains/Cereals
Meat/Poultry
Produce
Seafood

When you click the **Beverages**, the user will see the detailed data that correspond to filtering conditions and parameters of detailing. The picture below shows a page of a rendered report with detailed data of the **Beverages** entry:

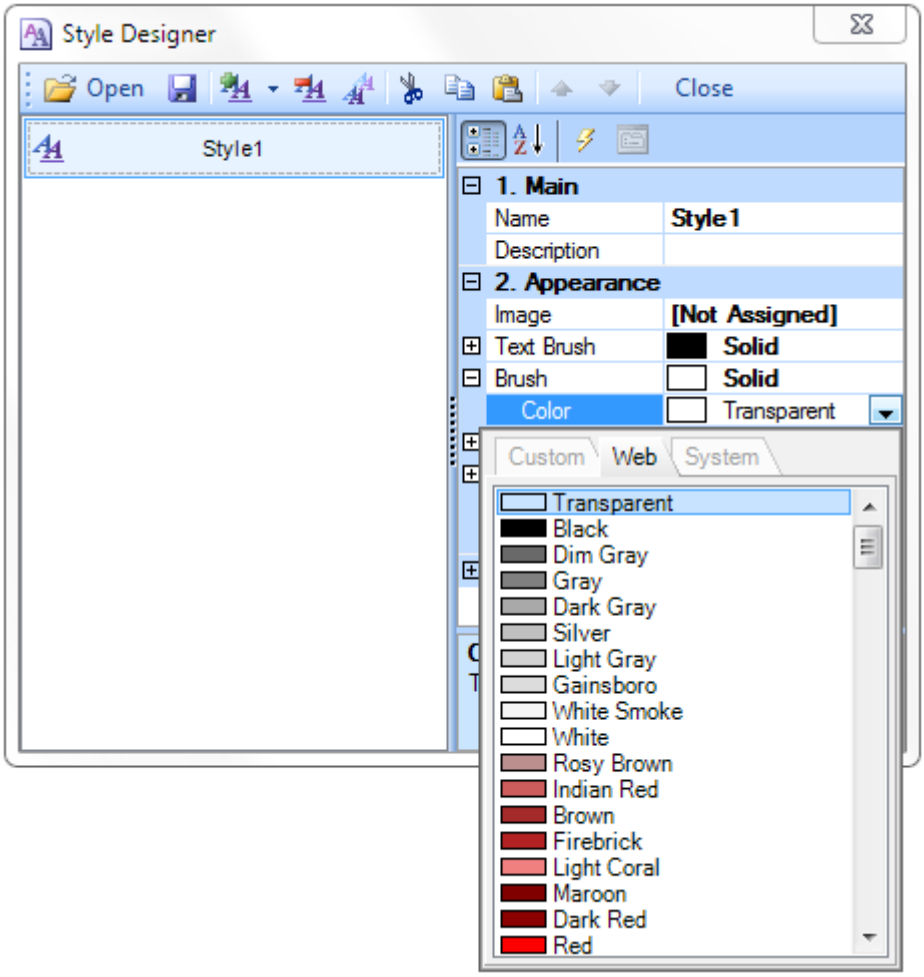
ProductName	UnitPrice
Chai	18
Chang	19
Guaraná Fantástica	4,5
Sasquatch Ale	14
Steeleye Stout	18
Côte de Blaye	263,5
Chartreuse verte	18
Ipoh Coffee	46
Laughing Lumberjack Lager	14
Outback Lager	15
Rhönbräu Klosterbier	7,75
Lakkalikööri	18

Adding styles

1. Go back to the report template;
2. Select the **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**.

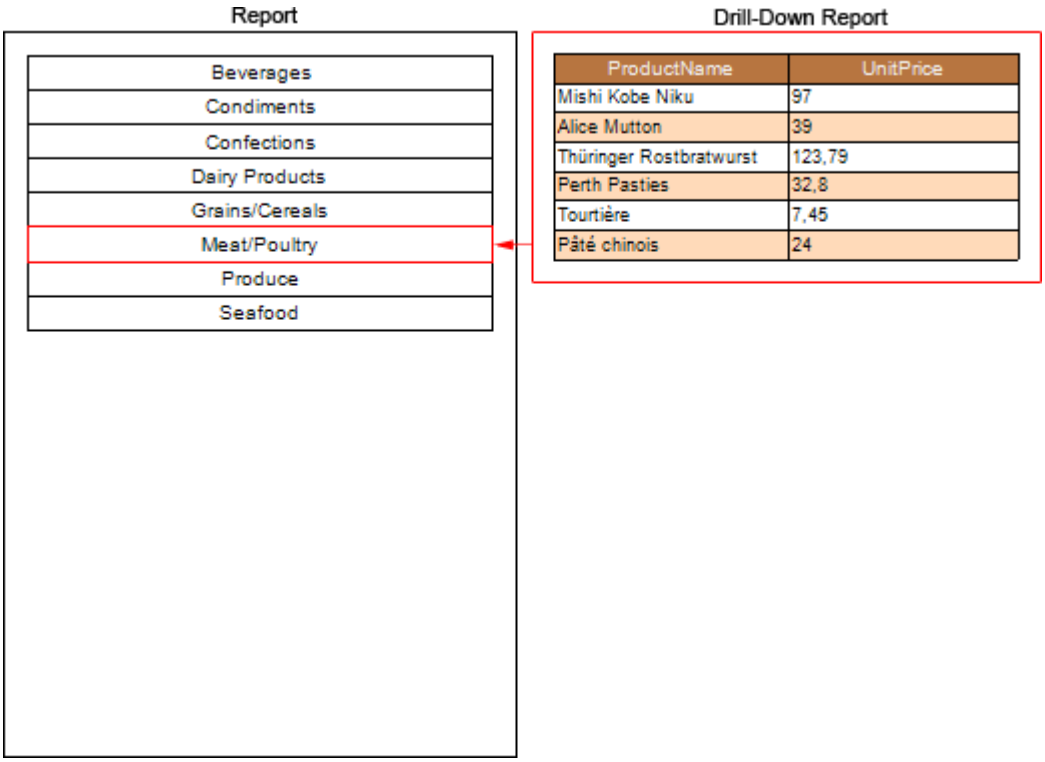


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

1. Save changes in the detailed report by clicking the **Save** button;
2. Open the report with master data in the designer;
3. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. The picture below shows the structure of the report, i.e. shows the ratio of the detailed data to the **Meat/Poultry** master entries with different styles of even/odd rows of the **DataBand** in the detailing report:



Report without Bands

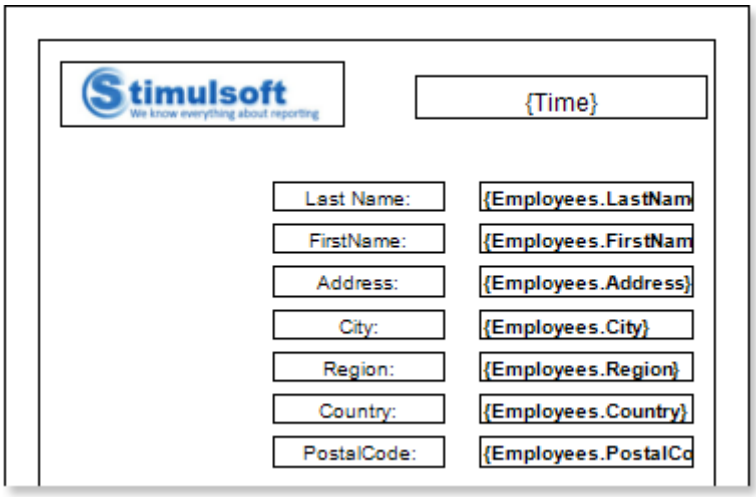
For better understanding this step-by-step instruction, please watch the video file.

If it is necessary to display data from only one entry of the data source or data from variables or other data sources that are not lists, the report can be created without the bands. In this case, components are placed directly on a report page.

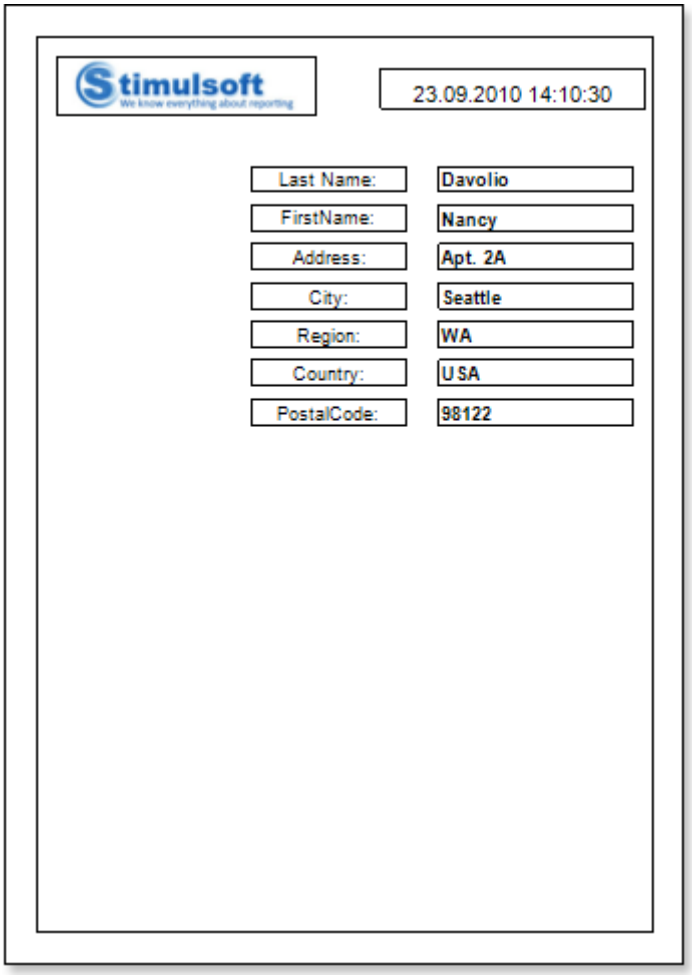
1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Put the **Image** component with the image on a page;
4. Edit the **Image** component and an image:
    4.1. Drag and drop the **Image** component on the report page;
    4.2. Align the **Image** component by height and width;
    4.3. Set the background color of the **Image** component;
    4.4. Align the image in the component;
    4.5. Change values of the properties of the **Image** component. For example to set the **Print** property to **true**, if you want this component be printed;
    4.6. If necessary, set **Borders** of the **Image** component;
    4.7. Set the border color.

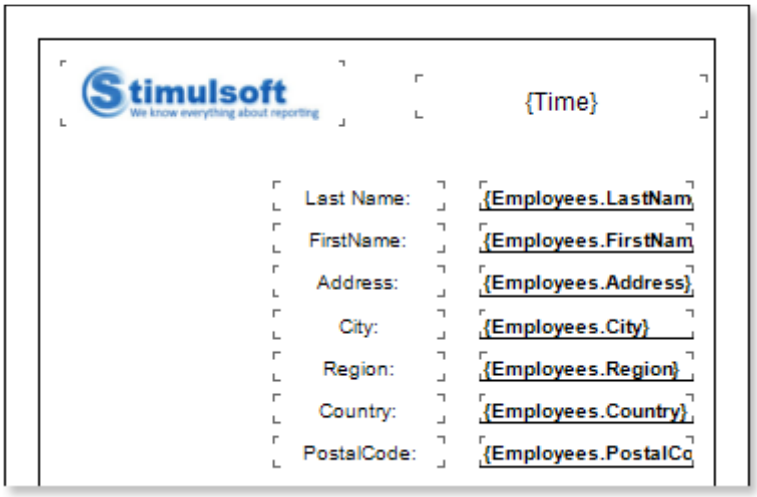
5. Put **TextBoxes** with the text on a page. In this report, put 15 Text components. The **TextBox1** contains the **{Time}** system variable, which will display the current time and date. **2-8 TextBoxes** contain the row names in the address box, and **9-15 TextBoxes** will include links to data sources;
6. Edit text and text components:
    6.1. Drag and drop the text component in the band;
    6.2. Change font options: size, type, color;
    6.3. Align text component by height and width;
    6.4. Change the background of the text component;
    6.5. Align text in the text component;
    6.6. Change values of text component properties, if required;
    6.7. Enable **Borders** of the text component, if required;
    6.8. Set the border color.



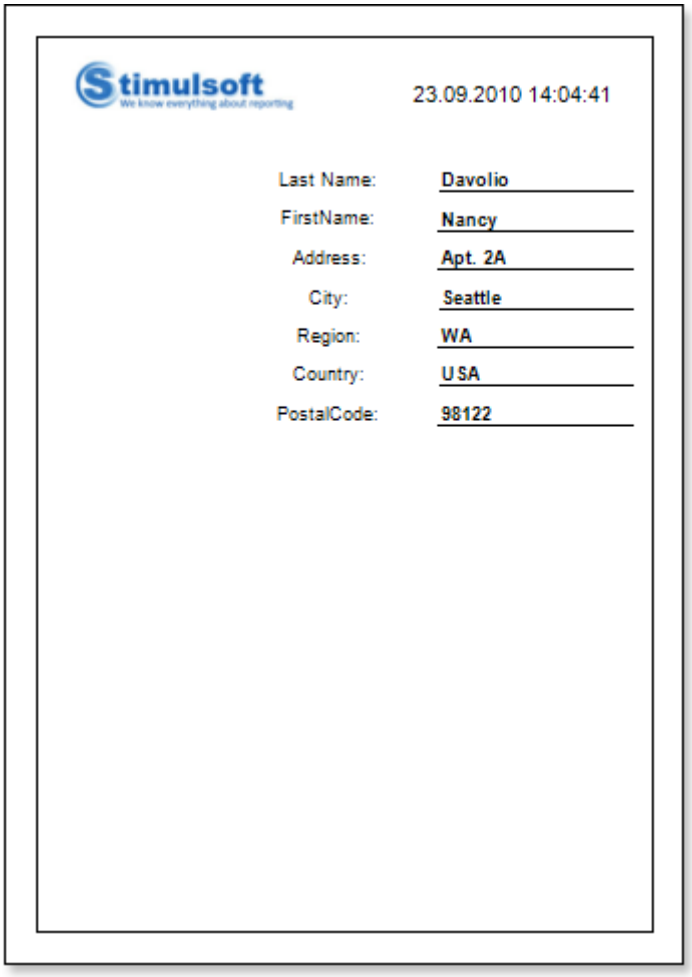
7. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item:



8. Go back to the report template;
9. Disable **Borders** of all components. Enable bottom borders for **9-15 TextBoxes**:



10. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.



Report with Multiple Pages in Template

For better understanding this step-by-step instruction, please watch the video file.

If you want to design a report, for example, with the cover page, the report template will consist of minimum two pages: the cover page and page with data. Creating a report with several pages in the template includes the following steps:

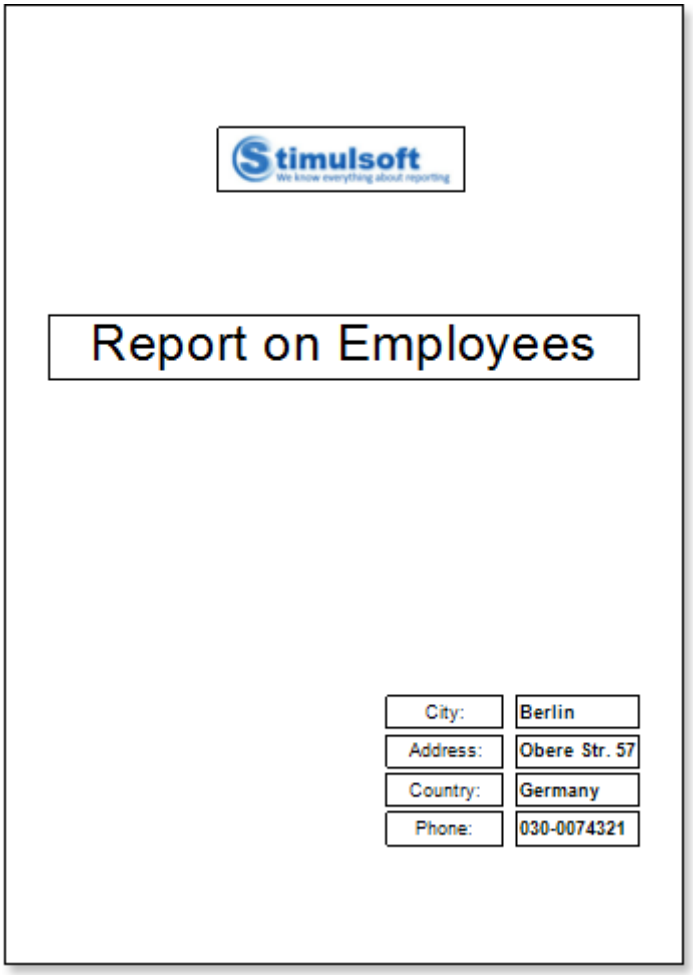
Creating a cover page

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Put an Image component on a report page;
4. Edit the Image component:
    4.1. Drag the **Image** component to the desired location on the report page;
    4.2. Align the **Image** component by height and width;
    4.3. Set the background color of the component;
    4.4. Align the image in the Image component;
    4.5. Set properties of the **Image** component. For example, set the **Print** property to **true**, if you want this component be printed;
    4.6. Set **Borders** of the component, if required;
    4.7. Set the border color.

5. On the report page Text components should be placed. We put 9 text components on this page. **TextBox1** will contain the **Report on Employees** text, which is the title of the report. **TextBoxes 2-5** will contain names in the address box, and **TextBoxes 6-9** will contain references to the source data;
6. Edit text and text components:
    6.1. Drag and drop the text component in the band;
    6.2. Change font options: size, type, color;
    6.3. Align text component by height and width;
    6.4. Change the background of the text component;
    6.5. Align text in the text component;
    6.6. Change values of text component properties, if required;
    6.7. Enable **Borders** of the text component, if required;
    6.8. Set the border color.

7. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item:

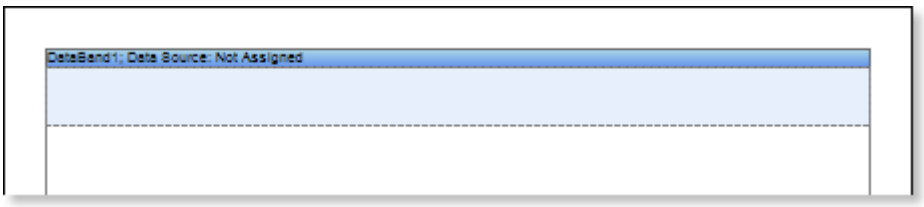


8. Go back to the report template;
9. Disable **Borders** for all components. Enable only the bottom borders in **TextBoxes 6-9**. The figure below submitted revised report template:

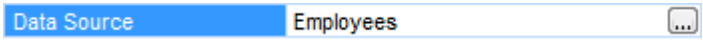
10. Create a second page in a report template and start editing it;

Creating a page with data

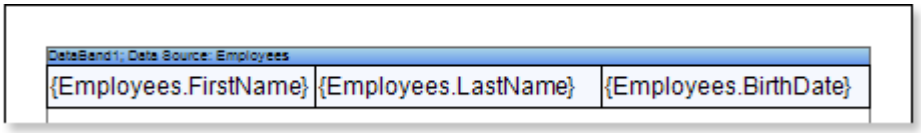
1. Put the **DataBand** page on the report template.



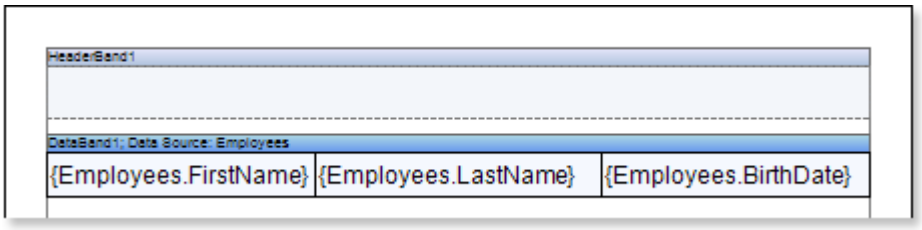
2. Edit **DataBand**:
   2.1. Align the **DataBand** by height;
   2.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
   2.3. Change the **DataBand** background;
   2.4. Enable **Borders** for the **DataBand**, if required;
   2.5. Change the border color.
3. Specify the data source in the **DataBand** using the **Data Source** property:



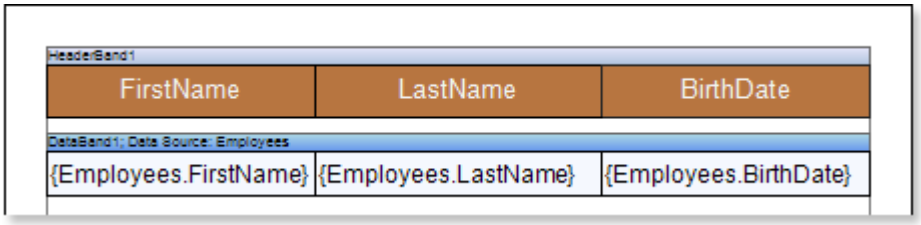
4. Put text components with expressions on **DataBands**. Where expression is a reference to the data field. For example, put two text components with the following expressions:**{Employees.FirstName}, {Employees.LastName}** and **{Employees.BirthDate}**;
5. Edit **Text** and **TextBox** component:
   5.1. Drag and drop the text component in **DataBands**;
   5.2. Change parameters of the text font: size, type, color;
   5.3. Align the text component by width and height;
   5.4. Change the background of the text component;
   5.5. Align text in the text component;
   5.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   5.7. Enable **Borders** for the text component, if required.
   5.8. Change the border color.



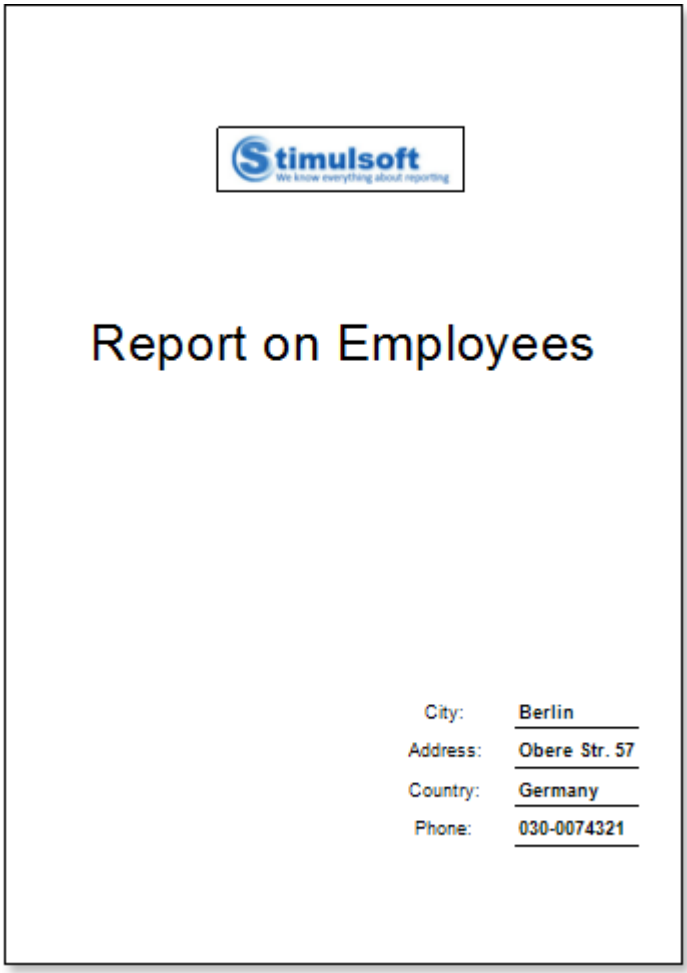
6. Add other bands to the report template, for example, the **HeaderBand**;
7. Edit this bands:
   7.1. Align it by height;
   7.2. Change values of properties, if required;
   7.3. Change the background of bands;
   7.4. Enable **Borders**, if required;
   7.5. Set the border color.

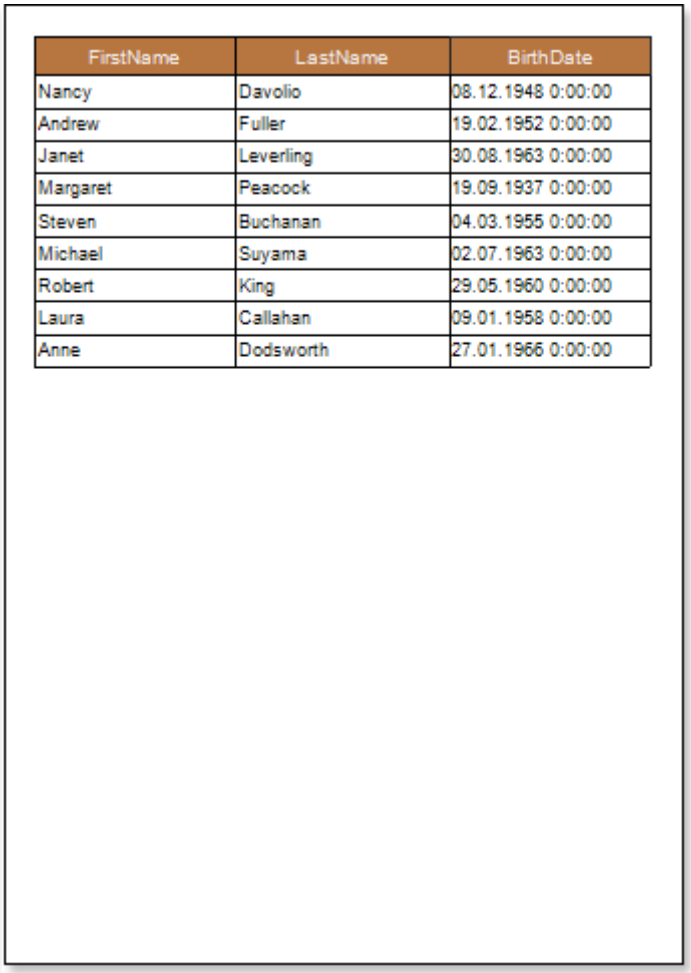


8. Put text components with expressions in the the band. The expression in the text component is a header in the **HeaderBand**.
9. Edit text and text component:
   9.1. Drag and drop the text component in the band;
   9.2. Change font options: size, type, color;
   9.3. Align text component by height and width;
   9.4. Change the background of the text component;
   9.5. Align text in the text component;
   9.6. Change values of text component properties, if required;
   9.7. Enable **Borders** of the text component, if required;
   9.8. Set the border color.



9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database.

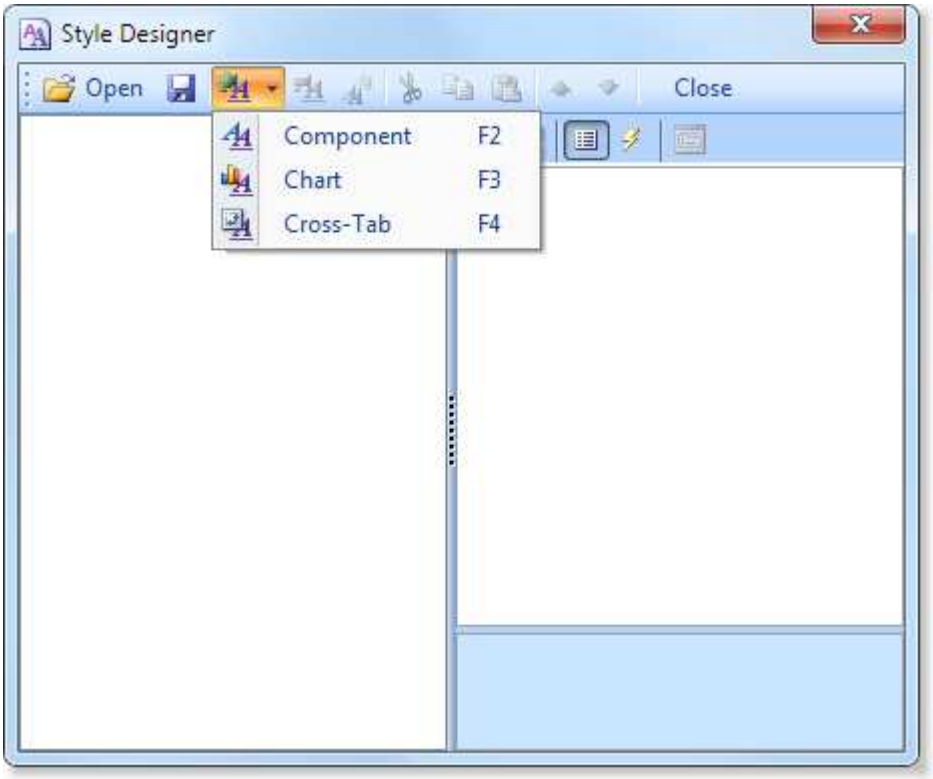




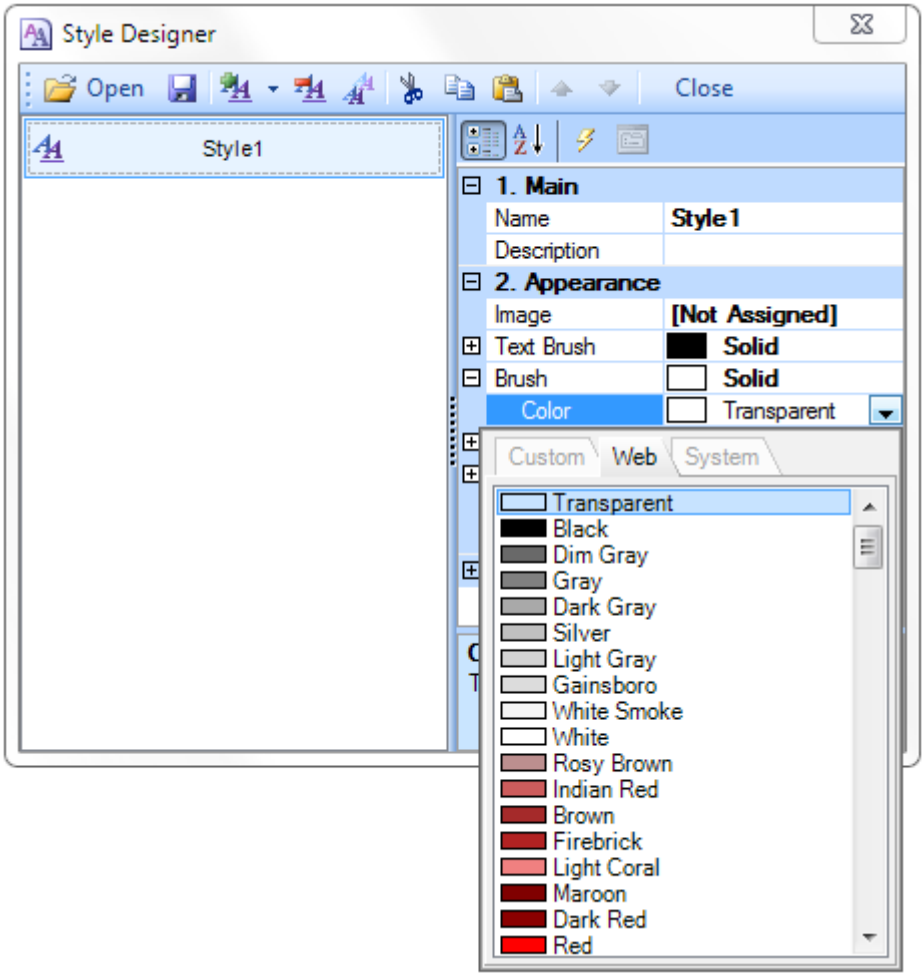
FirstName	LastName	BirthDate
Nancy	Davolio	08.12.1948 0:00:00
Andrew	Fuller	19.02.1952 0:00:00
Janet	Leverling	30.08.1963 0:00:00
Margaret	Peacock	19.09.1937 0:00:00
Steven	Buchanan	04.03.1955 0:00:00
Michael	Suyama	02.07.1963 0:00:00
Robert	King	29.05.1960 0:00:00
Laura	Callahan	09.01.1958 0:00:00
Anne	Dodsworth	27.01.1966 0:00:00

Adding Styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

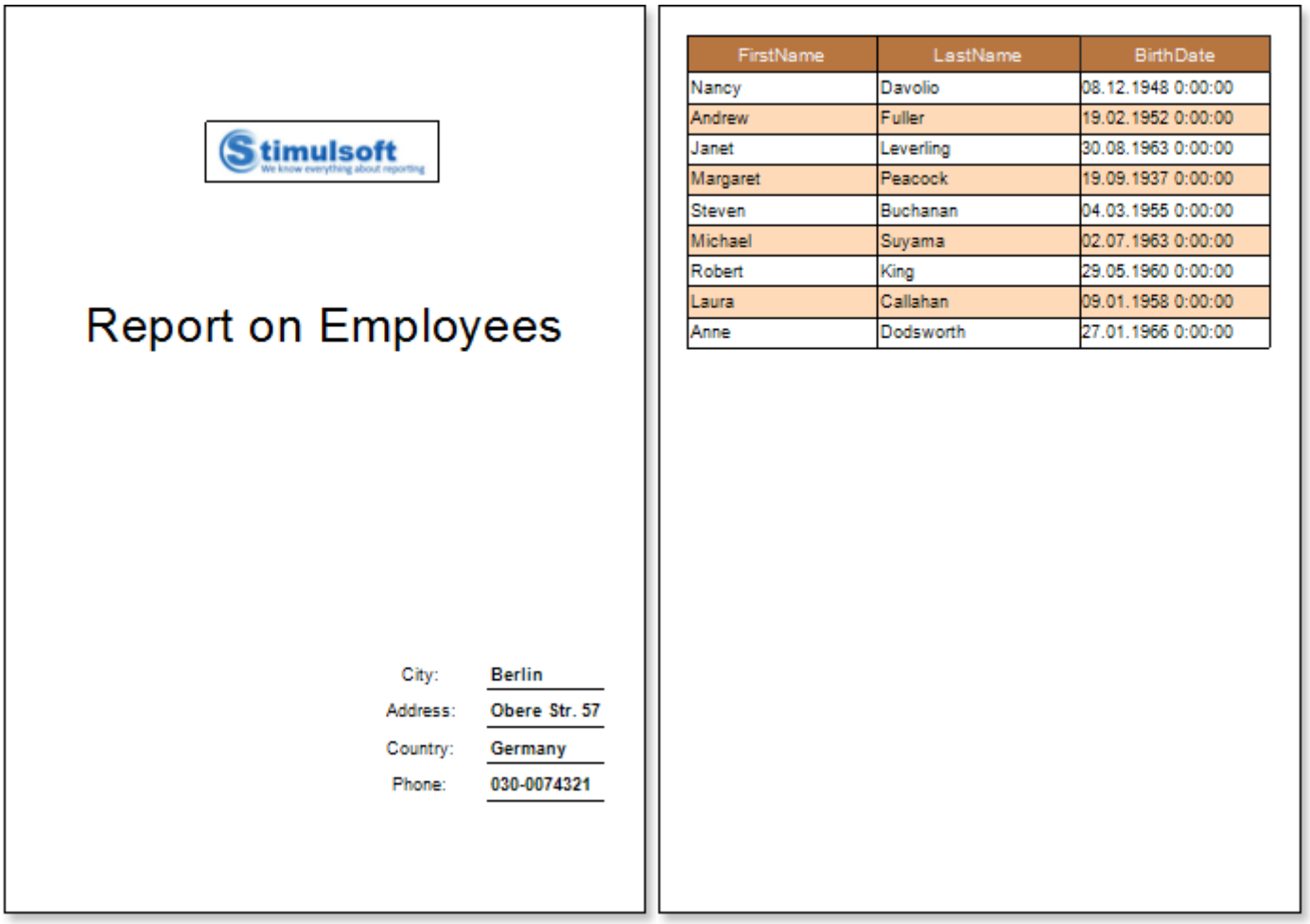


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.

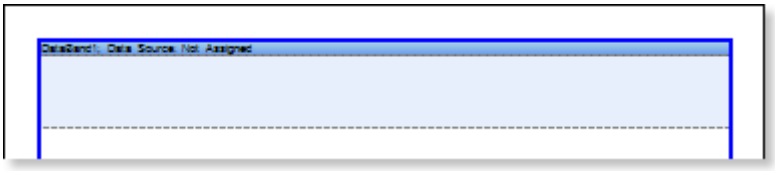


Report with Segmented Pages

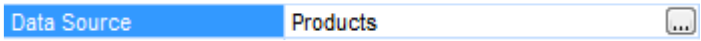
If data in a report should be placed on a single page by width or height, and a page size is small, you can add the required number of segments by width and/or height. In this case, one segment is a whole page and summary page consists of several segments across by width or height. In order to design a report with segmented pages, follow the steps below:

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

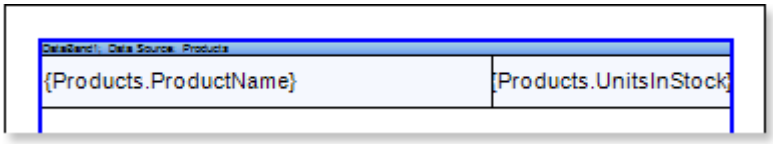
3. Define the number of segments by height and/or width. For example, set the **Segment per Height** property to **2**, i.e. the number of segments by height is **2**.
4. Put the **DataBand** on a segment of the report template.



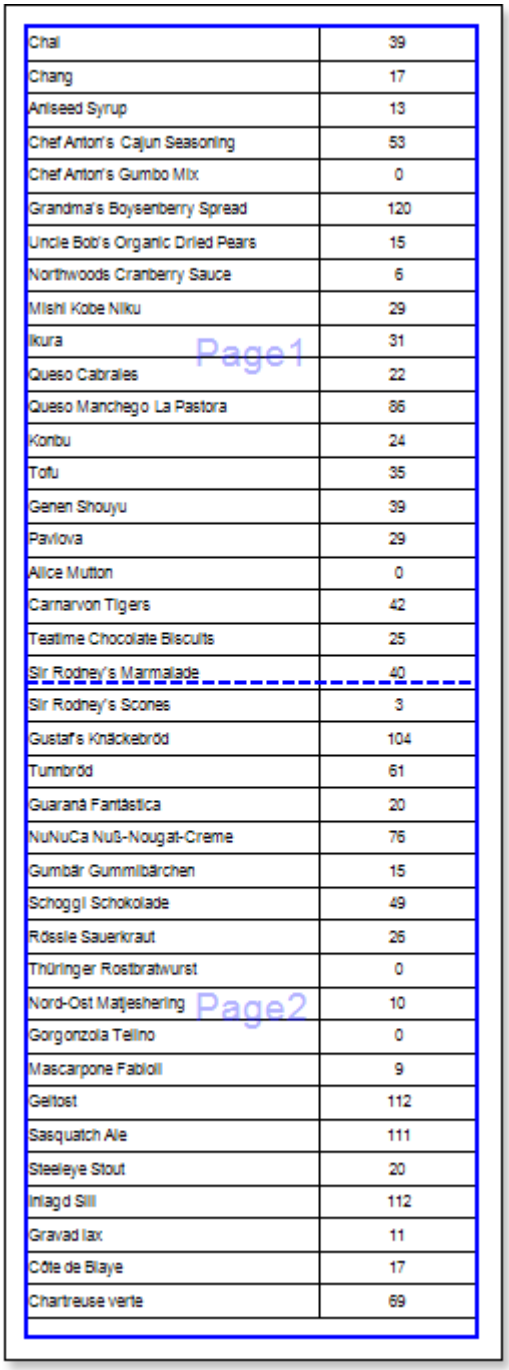
5. Edit **DataBand**:
   5.1. Align the **DataBand** by height;
   5.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
   5.3. Change the **DataBand** background;
   5.4. Enable **Borders** for the **DataBand**, if required;
   5.5. Change the border color.
6. Specify the data source in the **DataBand** using the **Data Source** property:



7. Put text components with expressions on **DataBands**. Where expression is a reference to the data field. For example, put two text components with the following expressions: **{Products.ProductName}** and **{Products.UnitsInStock}**;
8. Edit **Text** and **TextBox** component:
   8.1. Drag and drop the text component in **DataBands**;
   8.2. Change parameters of the text font: size, type, color;
   8.3. Align the text component by width and height;
   8.4. Change the background of the text component;
   8.5. Align text in the text component;
   8.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   8.7. Enable **Borders** for the text component, if required.
   8.8. Change the border color.

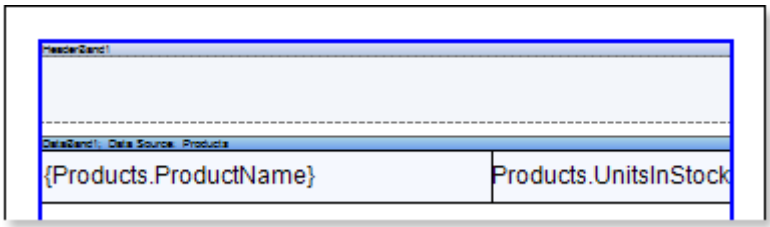


9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database.



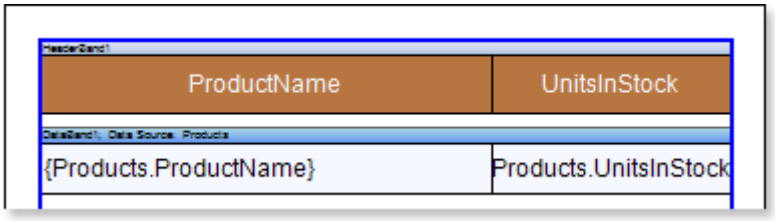
Chai	39
Chang	17
Aniseed Syrup	13
Chef Anton's Cajun Seasoning	53
Chef Anton's Gumbo Mix	0
Grandma's Boysenberry Spread	120
Uncle Bob's Organic Dried Pears	15
Northwoods Cranberry Sauce	6
Mishi Kobe Niku	29
Ikura	31
Queso Cabrales	22
Queso Manchego La Pastora	86
Konbu	24
Tofu	35
Genen Shouyu	39
Pavlova	29
Alice Mutton	0
Carnarvon Tigers	42
Teatime Chocolate Biscuits	25
Sir Rodney's Marmalade	40
Sir Rodney's Scones	3
Gustaf's Knäckebröd	104
Tunnbröd	61
Guaraná Fantástica	20
NuNuCa Nuß-Nougat-Creme	76
Gumbär Gummibärchen	15
Schoggi Schokolade	49
Rössle Sauerkraut	26
Thüringer Rostbratwurst	0
Nord-Ost Matjesherring	10
Gorgonzola Telino	0
Mascarpone Fabioli	9
Geitost	112
Sasquatch Ale	111
Steeleye Stout	20
Inlagd Sill	112
Gravad lax	11
Côte de Blaye	17
Chartreuse verte	69

10. Add other bands to the report template, for example, the **HeaderBand**;
11. Edit this bands:
    11.1. Align it by height;
    11.2. Change values of properties, if required;
    11.3. Change the background of bands;
    11.4. Enable **Borders**, if required;
    11.5. Set the border color.



12. Put text components with expressions in the the band. The expression in the text component is a header in the **HeaderBand**.

13. Edit text and text component:
    13.1. Drag and drop the text component in the band;
    13.2. Change font options: size, type, color;
    13.3. Align text component by height and width;
    13.4. Change the background of the text component;
    13.5. Align text in the text component;
    13.6. Change values of text component properties, if required;
    13.7. Enable **Borders** of the text component, if required;
    13.8. Set the border color.



14. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.

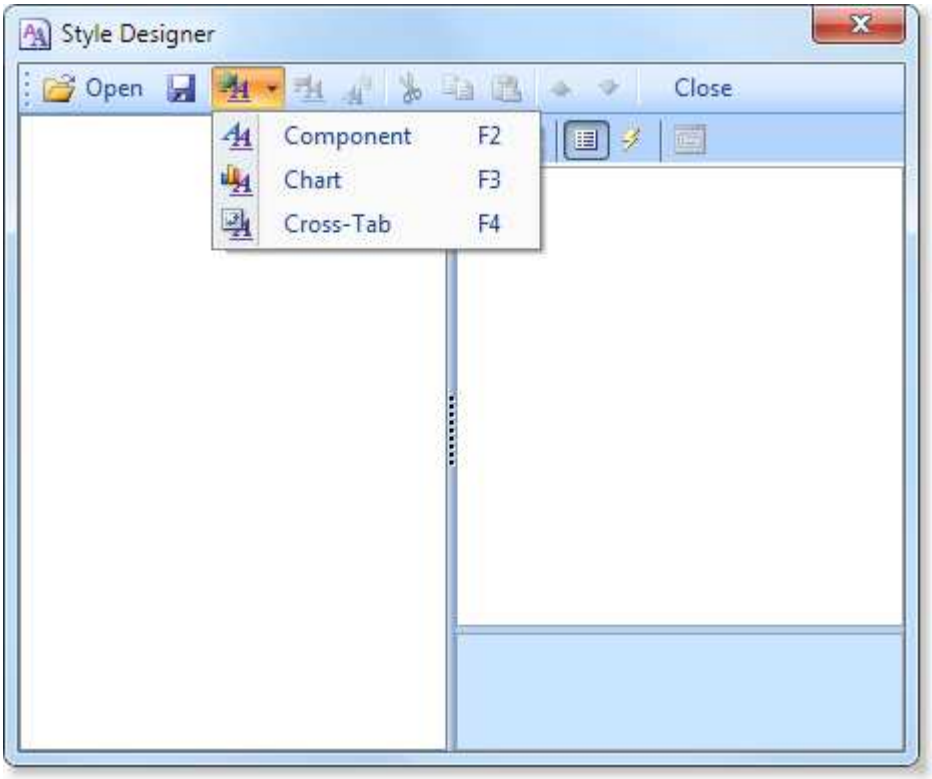
ProductName	UnitsInStock
Chai	39
Chang	17
Aniseed Syrup	13
Chef Anton's Cajun Seasoning	53
Chef Anton's Gumbo Mix	0
Grandma's Boysenberry Spread	120
Uncle Bob's Organic Dried Pears	15
Northwoods Cranberry Sauce	6
Mishi Kobe Niku	29
Ikura	31
Queso Cabrales	22
Queso Manchego La Pastora	86
Konbu	24
Tofu	35
Genen Shouyu	39
Pavlova	29
Alice Mutton	0
Carnarvon Tigers	42
Teatime Chocolate Biscuits	25
Sir Rodney's Marmalade	40
Sir Rodney's Scones	3
Gustaf's Knäckebröd	104
Tunnbröd	61
Guaraná Fantástica	20
NuNuCa Nuß-Nougat-Creme	76
Gumbär Gummibärchen	15
Schoggi Schokolade	49
Rössle Sauerkraut	26
Thüringer Rostbratwurst	0
Nord-Ost Matjeshering	10
Gorgonzola Telino	0
Mascarpone Fabioli	9
Geitost	112
Sasquatch Ale	111
Steeleye Stout	20
Inlagd Sill	112
Gravad lax	11
Côte de Blaye	17

Page1

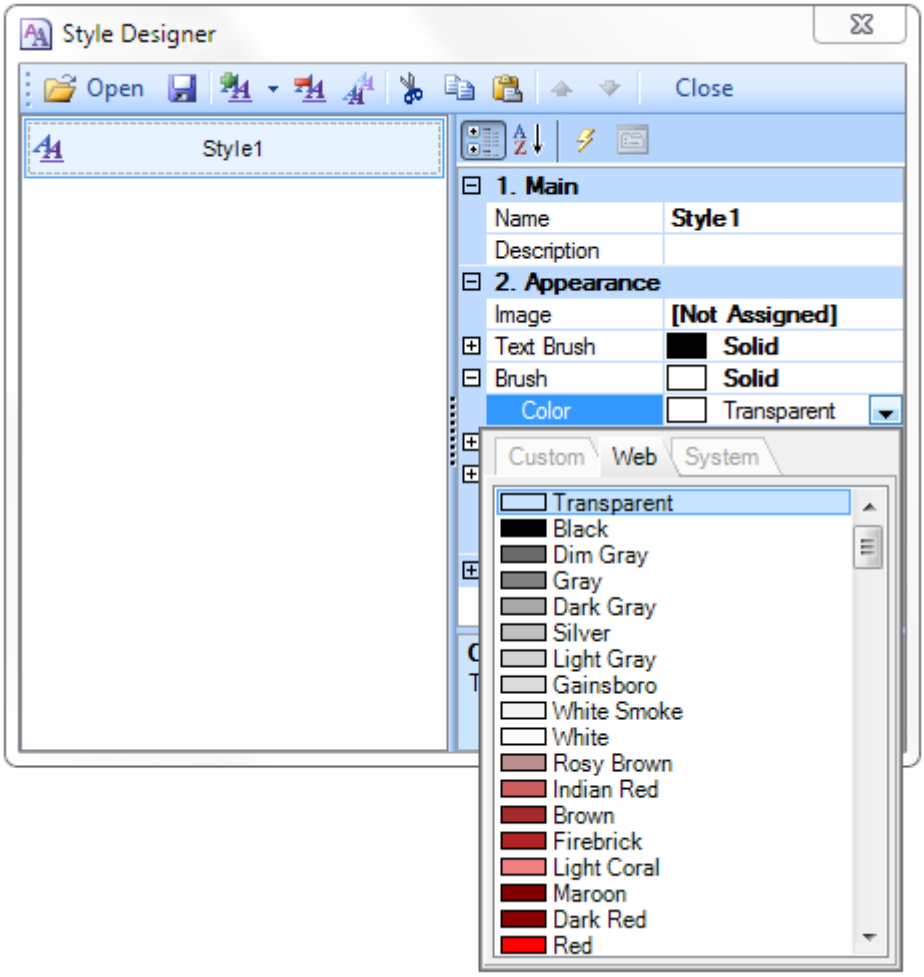
Page2

Adding Styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

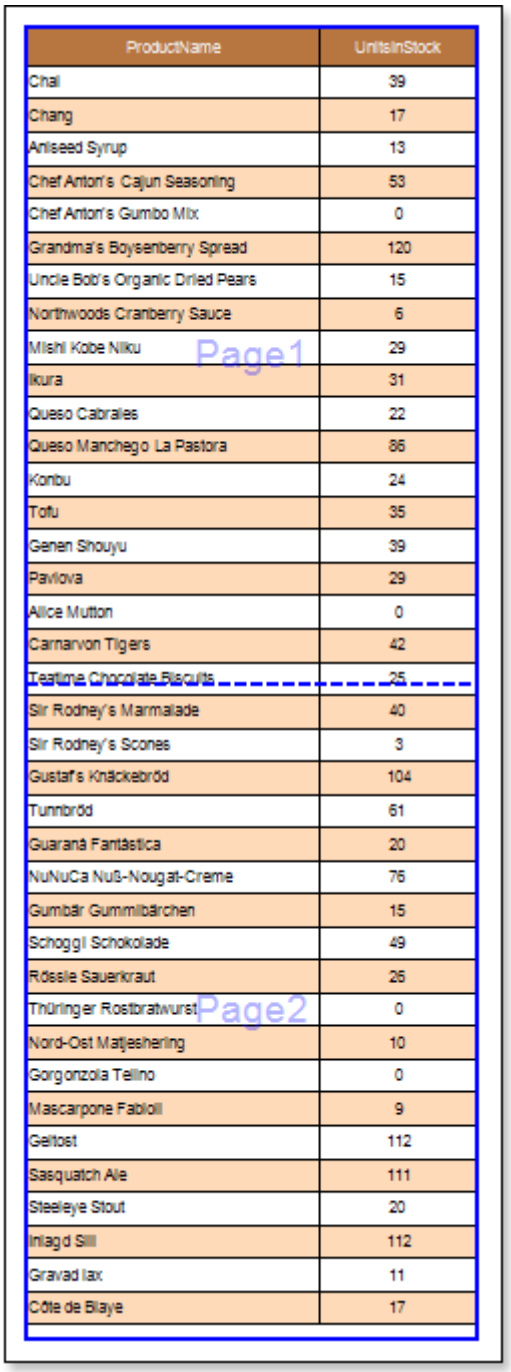


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.



ProductName	UnitsInStock
Chai	39
Chang	17
Aniseed Syrup	13
Chef Anton's Cajun Seasoning	53
Chef Anton's Gumbo Mix	0
Grandma's Boysenberry Spread	120
Uncle Bob's Organic Dried Pears	15
Northwoods Cranberry Sauce	6
Mishi Kobe Niku	29
Ikura	31
Queso Cabrales	22
Queso Manchego La Pastora	86
Konbu	24
Tofu	35
Genen Shouyu	39
Pavlova	29
Alice Mutton	0
Carnarvon Tigers	42
Teatime Chocolate Biscuits	25
Sir Rodney's Marmalade	40
Sir Rodney's Scones	3
Gustaf's Knäckebröd	104
Tunnbröd	61
Guaraná Fantástica	20
NuNuCa Nuß-Nougat-Creme	76
Gumbär Gummibärchen	15
Schoggi Schokolade	49
Rössle Sauerkraut	26
Thüringer Rostbratwurst	0
Nord-Ost Matjeshering	10
Gorgonzola Telino	0
Mascarpone Fabioli	9
Geitost	112
Sasquatch Ale	111
Steeleye Stout	20
Inlagd Sill	112
Gravad lax	11
Côte de Blaye	17

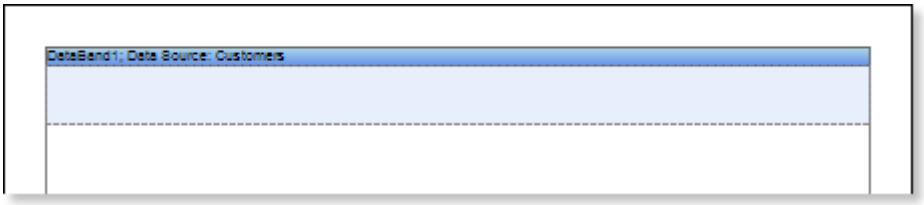
Report with Primitives on Page

For better understanding this step-by-step instruction, please watch the video file.

Primitives are: **Horizontal Line**, **Vertical Line**, **Rectangle** and **Rounded Rectangle**. Besides, you may use the **Shape** component. When placing a primitive on a page, the primitive will be rendered as a page item. In order to design a report with primitives on a page, follow the steps below:

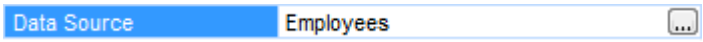
1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Put the **DataBand** on a page of a report template.

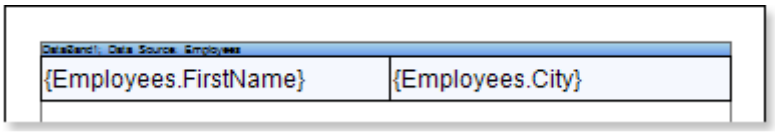


4. Edit **DataBand**:
    4.1. Align the **DataBand** by height;
    4.2. Change values of band properties. For example, set the **Can Shrink** property to **true**, if you wish the data band to be broken;
    4.3. Change the **DataBand** background;
    4.4. Enable **Borders** for the **DataBand**, if required;
    4.5. Change the border color.

5. Define the data source for the **DataBand** using the **Data Source** property:



6. Put text components with expressions on the **DataBand**. Where expression is a reference to the data field. For example, put two text components with expressions: **{Employees.FirstName}** and **{Employees.City}**;
7. Edit **Text** and **TextBox** component:
    7.1. Drag and drop the text component in the **DataBand**;
    7.2. Change parameters of the text font: size, type, color;
    7.3. Align the text component by width and height;
    7.4. Change the background of the text component;
    7.5. Align text in the text component;
    7.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
    7.7. Enable **Borders** for the text component, if required.
    7.8. Change the border color.

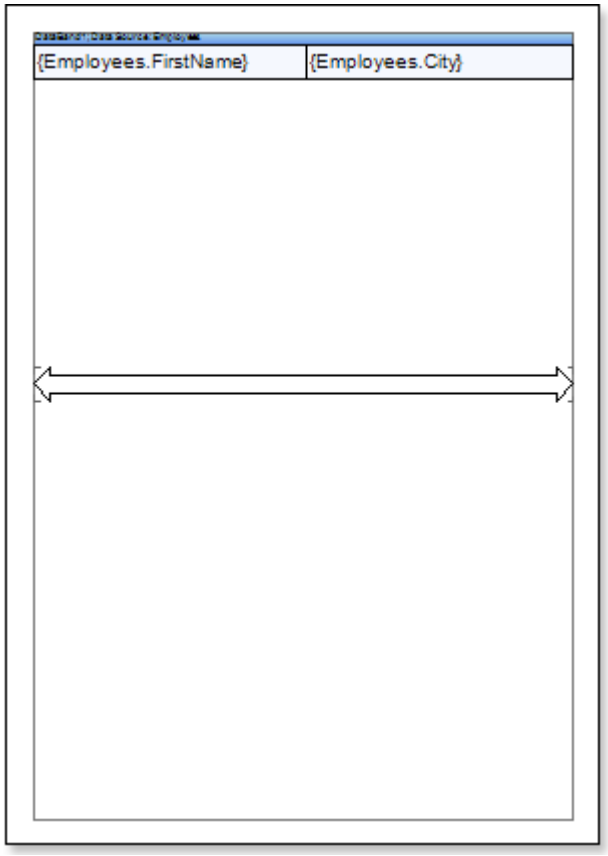


8. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.

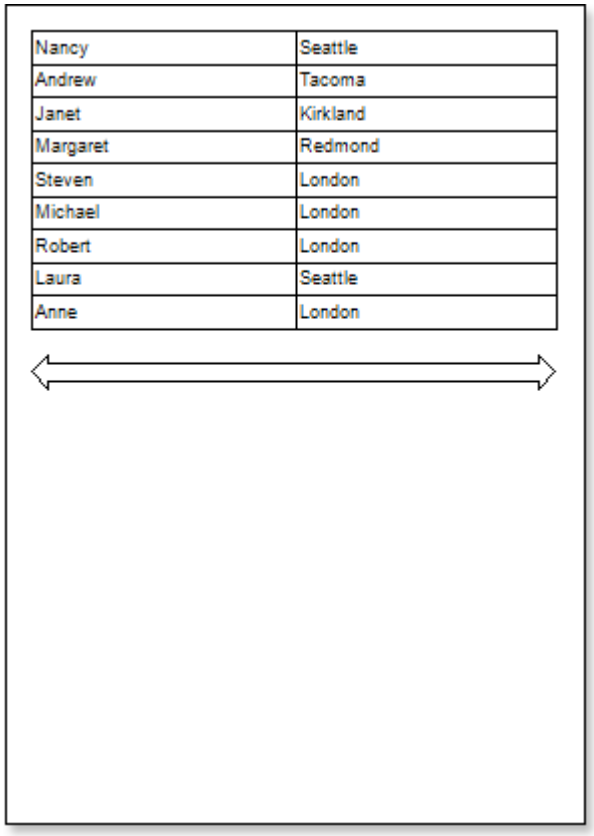
Nancy	Seattle
Andrew	Tacoma
Janet	Kirkland
Margaret	Redmond
Steven	London
Michael	London
Robert	London
Laura	Seattle
Anne	London

9. Go back to the report template.
10. Add the **Shape** component to a report template and edit it:
    10.1. Drag and drop the **Shape** component on the page;
    10.2. Change the type of a shape using the **Shape Type** property. Set the **Shape Type** property to **Complex Arrow**;
    10.3. Stretch the **Shape** component horizontally and vertically;
    10.4. Change the value of other properties. For example, set the **Grow to Height** property to **true**.

The picture below shows a report template with the **Shape** component placed on the report page:



11. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.

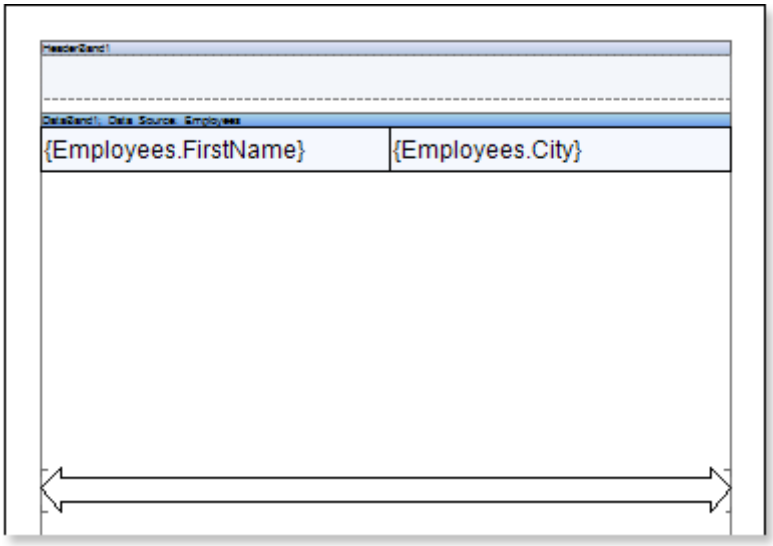


Nancy	Seattle
Andrew	Tacoma
Janet	Kirkland
Margaret	Redmond
Steven	London
Michael	London
Robert	London
Laura	Seattle
Anne	London

12. Go back to the report template.
13. If needed, add other bands to the report template, for example, **HeaderBand**;
14. Edit this bands:
    14.1. Align it by height;
    14.2. Change values of properties, if required;

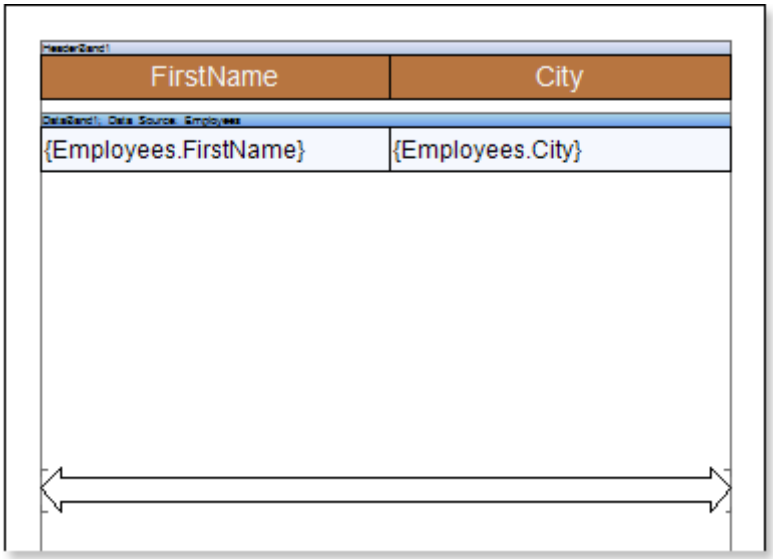
14.3. Change the background color of the band;
14.4. Enable **Borders**, if required;
14.5. Set the border color.

The picture below shows a report template with a **HeaderBand**:

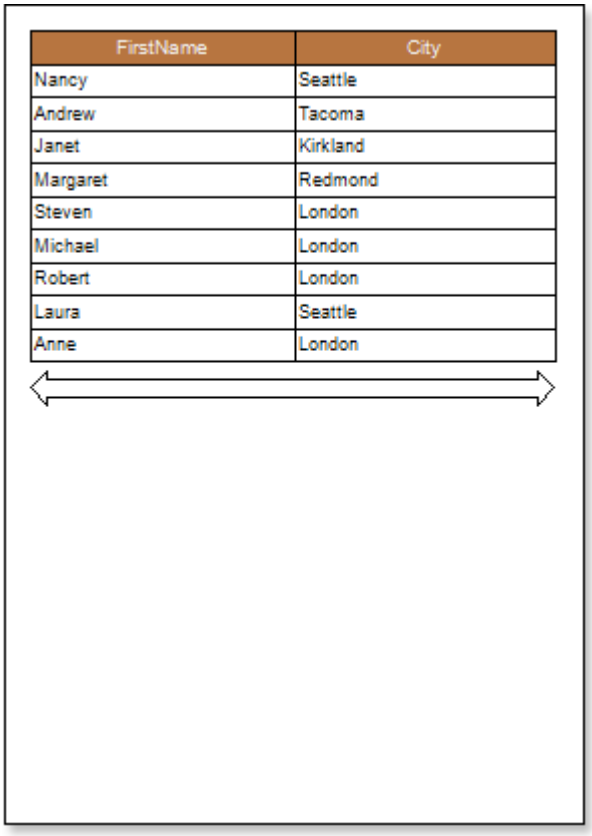


15. Put text components with expressions in the this band. The expression in the text component is a header in the **HeaderBand**.

16. Edit text and text components:
    16.1. Drag and drop the text component in the band;
    16.2. Change font options: size, type, color;
    16.3. Align text component by height and width;
    16.4. Change the background of the text component;
    16.5. Align text in the text component;
    16.6. Change values of text component properties, if required;
    16.7. Enable **Borders** of the text component, if required;
    16.8. Set the border color.



17. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database.



FirstName	City
Nancy	Seattle
Andrew	Tacoma
Janet	Kirkland
Margaret	Redmond
Steven	London
Michael	London
Robert	London
Laura	Seattle
Anne	London

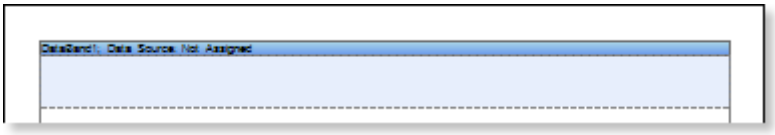
Report with Primitives in Band

For better understanding this step-by-step instruction, please watch the video file.

Primitives are: **Horizontal Line**, **Vertical Line**, **Rectangle** and **Rounded Rectangle**. Besides, you may use the **Shape** component. When placing a primitive on a band, the primitive will be rendered on a page as many times as the band will be printed. In order to design a report with primitives on a band, follow the steps below:

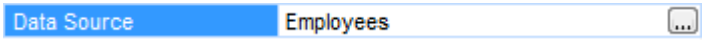
1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Put the **DataBand** on a page of a report template.

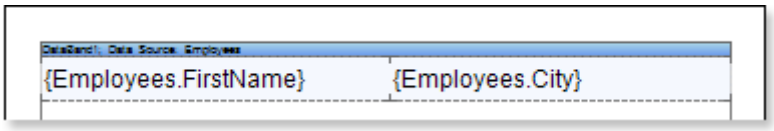


4. Edit **DataBand**:
    4.1. Align the **DataBand** by height;
    4.2. Change values of band properties. For example, set the **Can Shrink** property to **true**, if you wish the data band to be broken;
    4.3. Change the **DataBand** background;
    4.4. Enable **Borders** for the **DataBand**, if required;
    4.5. Change the border color.

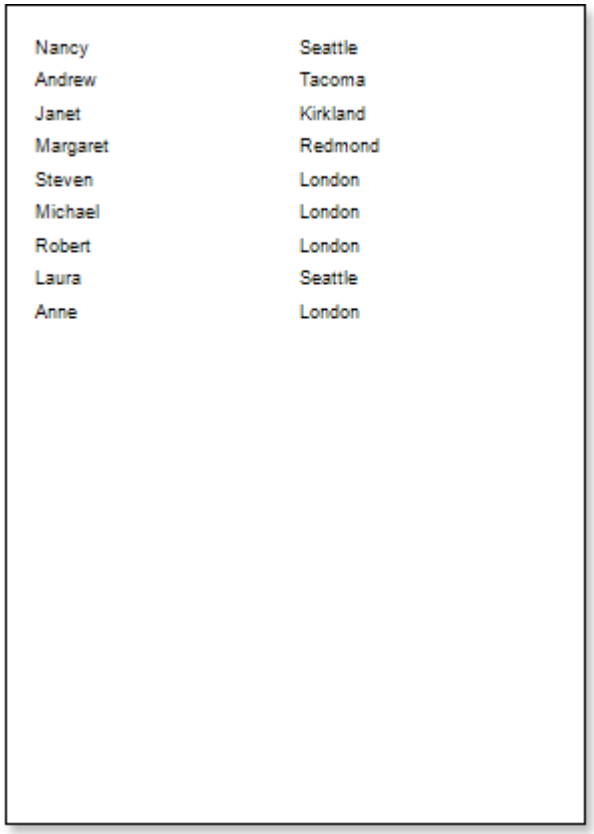
5. Define the data source for the **DataBand** using the **Data Source** property:



6. Put text components with expressions on the **DataBand**. Where expression is a reference to the data field. For example, put two text components with expressions: **{Employees.FirstName}** and **{Employees.City}**;
7. Edit **Text** and **TextBox** component:
   7.1. Drag and drop the text component in the **DataBand**;
   7.2. Change parameters of the text font: size, type, color;
   7.3. Align the text component by width and height;
   7.4. Change the background of the text component;
   7.5. Align text in the text component;
   7.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   7.7. Enable **Borders** for the text component, if required.
   7.8. Change the border color.

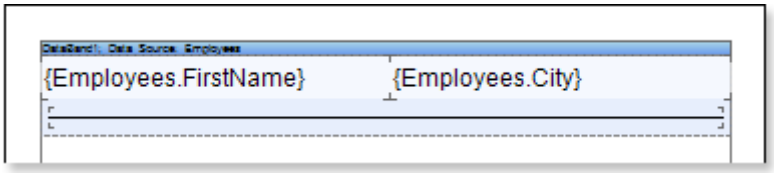


8. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.

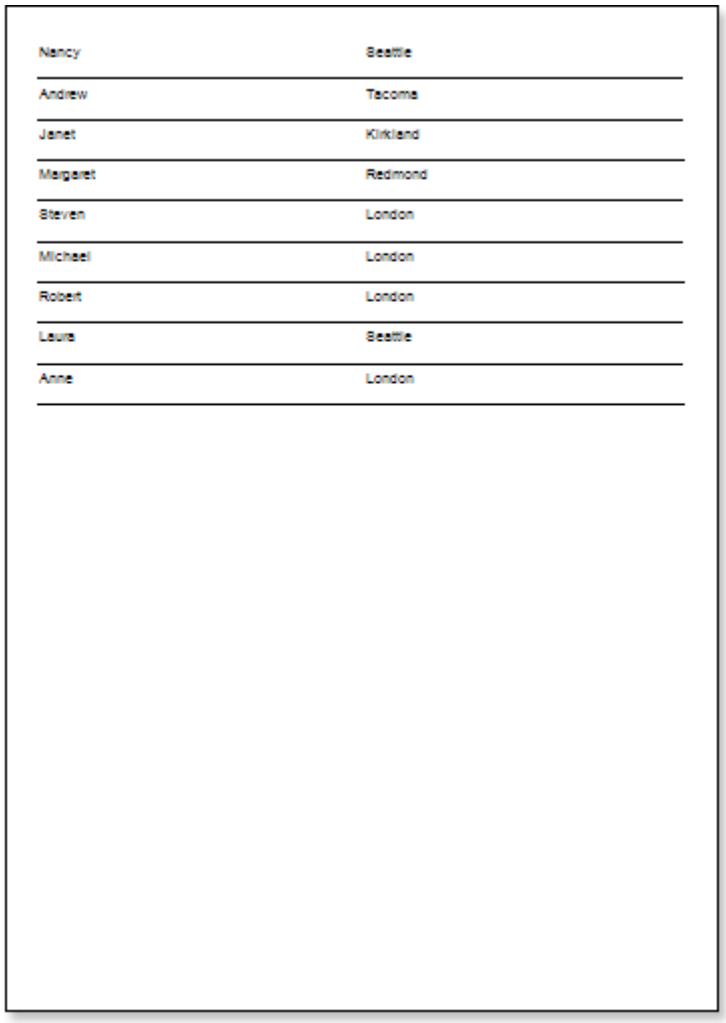


9. Go back to the report template.
10. Add the **Shape** component to a report template in the **DataBand** and edit it:
   10.1. Drag and drop the **Shape** component on the page;
   10.2. Change the type of a shape using the **Shape Type** property. Set the **Shape Type** property to **Complex Arrow**;
   10.3. Stretch the **Shape** component horizontally and vertically;
   10.4. Change the value of other properties. For example, set the **Grow to Height** property to **true**.

The picture below shows a report template with the **Shape** component placed on the report page:



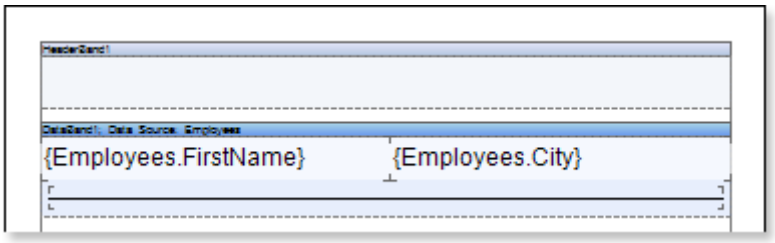
11. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.



Nancy	Seattle
Andrew	Tacoma
Janet	Kirkland
Margaret	Redmond
Steven	London
Michael	London
Robert	London
Laura	Seattle
Anne	London

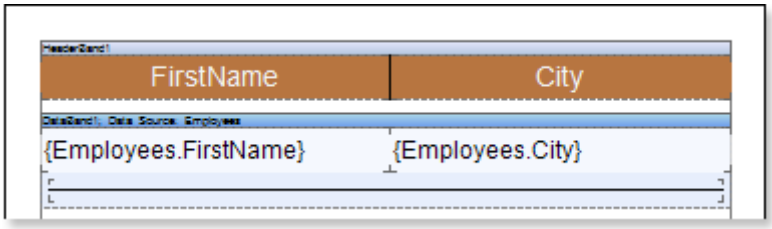
12. Go back to the report template.
13. If needed, add other bands to the report template, for example, **HeaderBand**;
14. Edit this bands:
    14.1. Align it by height;
    14.2. Change values of properties, if required;
    14.3. Change the background color of the band;
    14.4. Enable **Borders**, if required;
    14.5. Set the border color.

The picture below shows a report template with a **HeaderBand**:

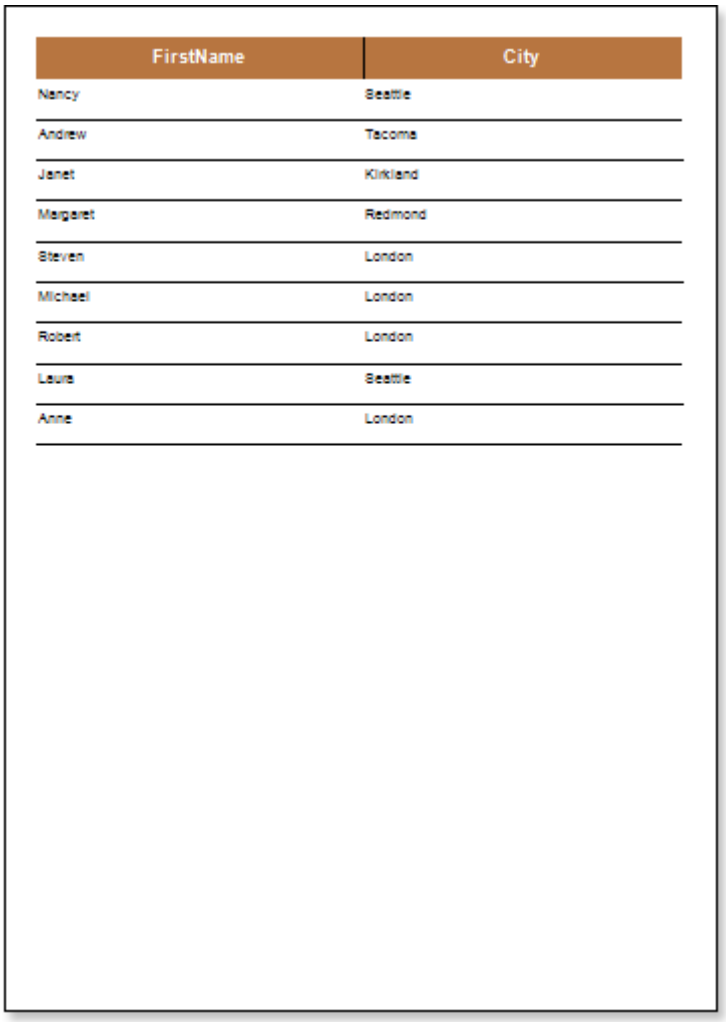


15. Put text components with expressions in the this band. The expression in the text component is a header in the **HeaderBand**.

16. Edit text and text components:
    16.1. Drag and drop the text component in the band;
    16.2. Change font options: size, type, color;
    16.3. Align text component by height and width;
    16.4. Change the background of the text component;
    16.5. Align text in the text component;
    16.6. Change values of text component properties, if required;
    16.7. Enable **Borders** of the text component, if required;
    16.8. Set the border color.



17. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database.

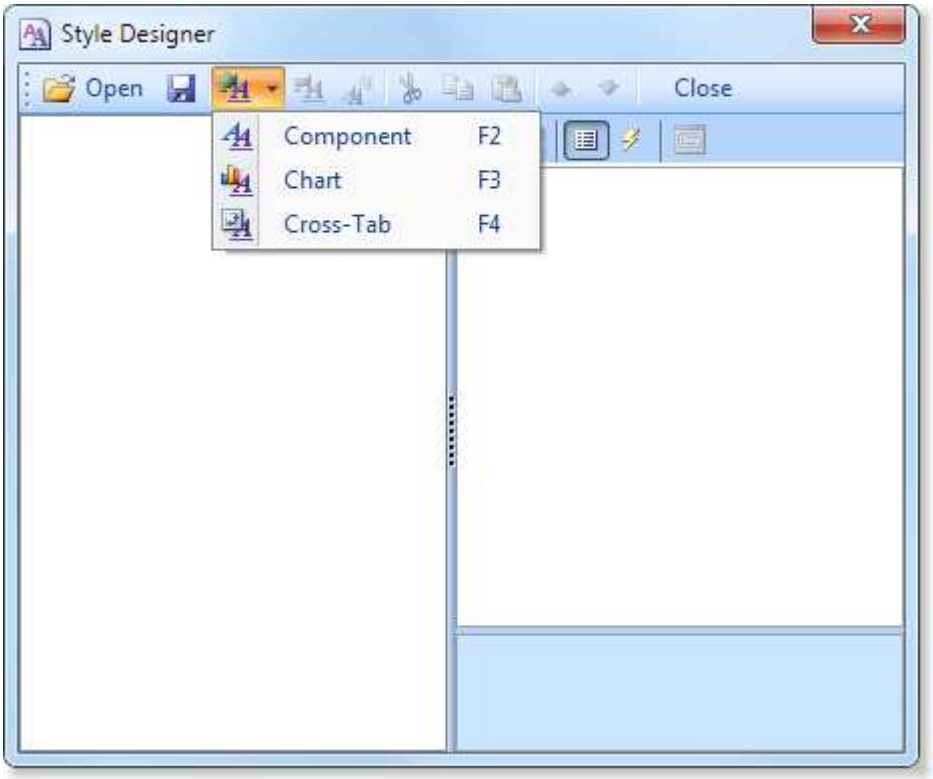


FirstName	City
Nancy	Seattle
Andrew	Tacoma
Janet	Kirkland
Margaret	Redmond
Steven	London
Michael	London
Robert	London
Laura	Seattle
Anne	London

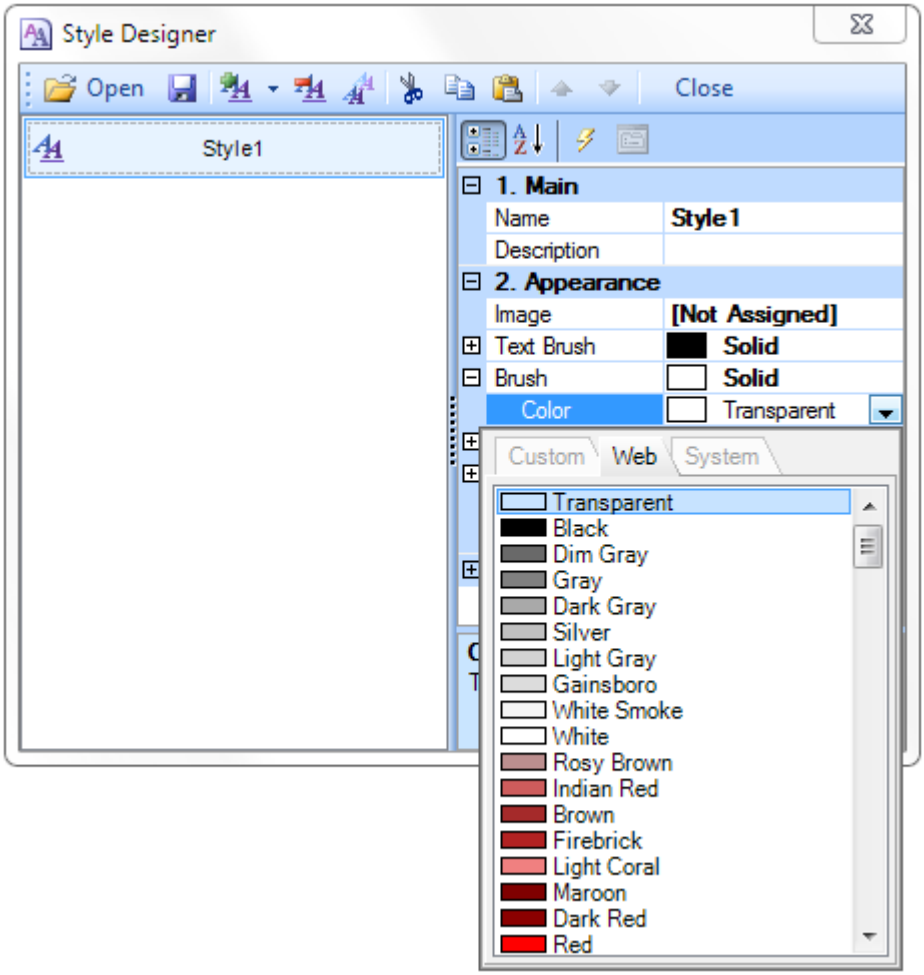
Adding Styles

1. Go back to the report template;
2. Select **DataBand**;

3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

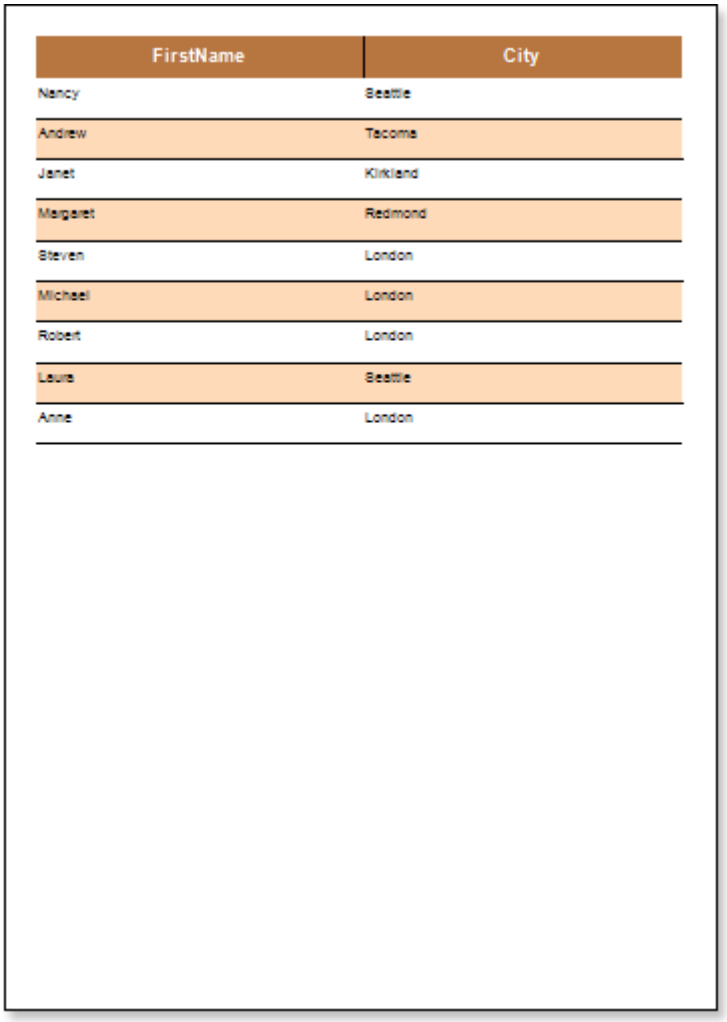


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.



FirstName	City
Nancy	Seattle
Andrew	Tacoma
Janet	Kirkland
Margaret	Redmond
Steven	London
Michael	London
Robert	London
Laura	Seattle
Anne	London

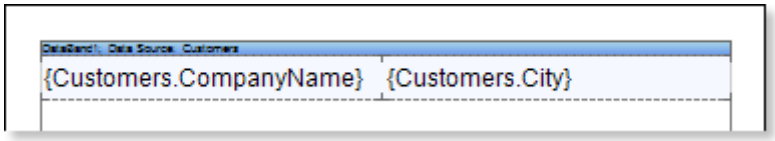
Report with Cross-Primitives

For better understanding this step-by-step instruction, please watch the video file and another one.

Cross-primitives include: **Vertical Line**, **Rectangle** and **Rounded Rectangle**. The start and end points of cross-primitives can be placed on different components of a report. In order to design a report with cross-primitives, follow the steps below:

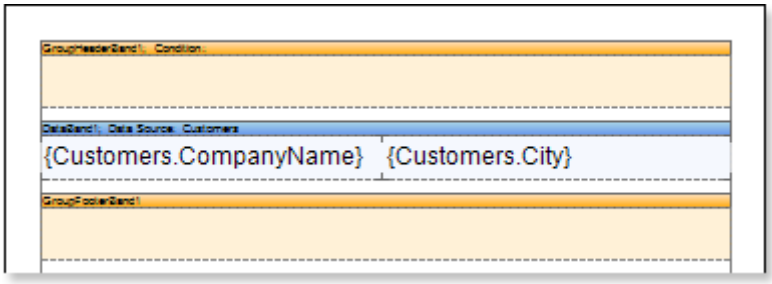
1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Create a report or load previously saved one. For our example we take a Simple List Report report, described in **Simple List Report** article.



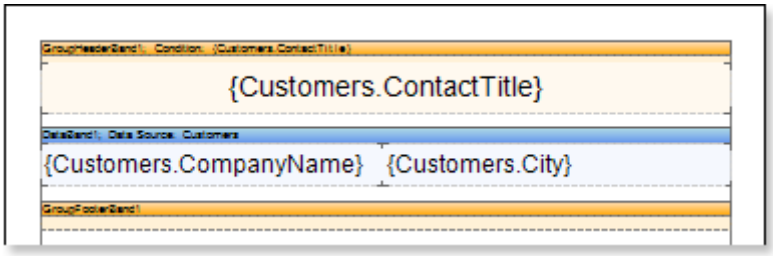
4. Add **GroupHeaderBand** and **GroupFooterBand** to a report template. The **GroupHeaderBand** should be placed above the **DataBand** to which it applies. The **GroupFooterBand** should be placed below the **DataBand**. And it is

meant exactly the **DataBand**, that is associated with the **GroupHeaderBand**. Each **GroupFooterBand**, refers to a certain **GroupHeaderBand**. The **GroupFooterBand** will not be output without the **GroupHeaderBand**.

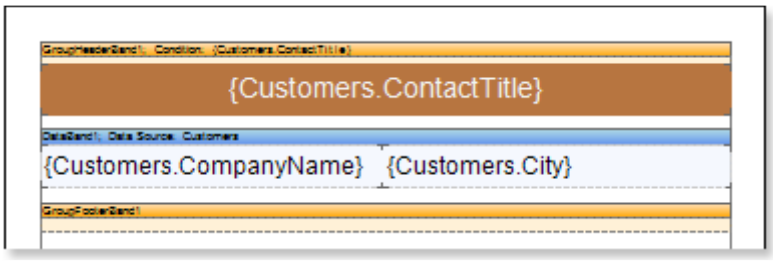


5. Edit the **GroupHeaderBand** and the **GroupFooterBand**:
   5.1. Align them by height;
   5.2. Change the values of the required properties. For example, set the **KeepGroupHeaderTogether** property for the **GroupHeaderBand** to **true**, if you want the **GroupHeaderBand** be kept with the group. And set the **KeepGroupFooterTogether** property for the **GroupFooterBand** to **true**, if you want this band be kept with the group;
   5.3. Set the background color for the **GroupHeaderBand**;
   5.4. If necessary, set the **Borders** for the **DataBand**;

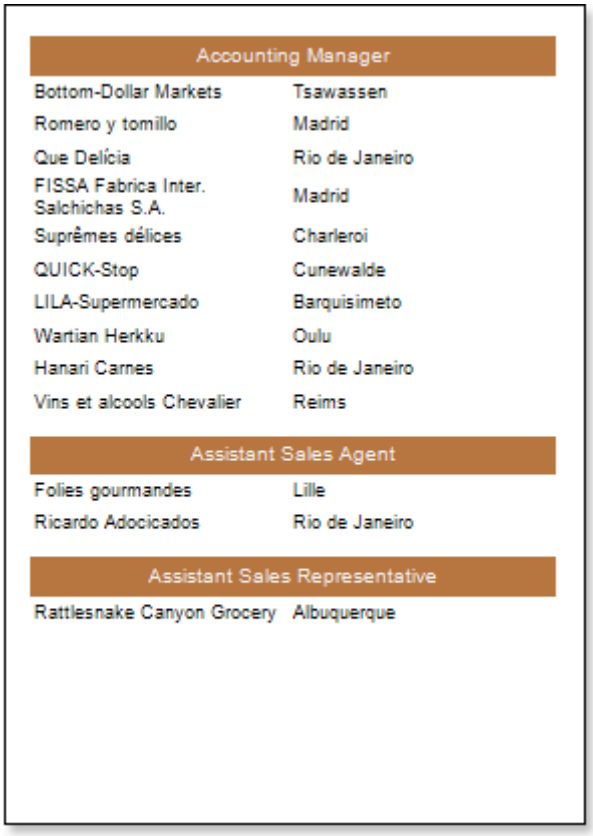
6. Set the condition of data grouping in the report using the **Condition** property of the **GroupHeaderBand**. Condition for the grouping can be set by specifying an expression or by selecting a column from a data source. In this example, we specify the **{Customers.ContactTitle}** expression of the grouping condition, so, when rendering the report, a list of companies will be grouped by the **ContactTitle** column data.
7. Put a text component in the **GroupHeaderBand** with the following expression: **{Customers.ContactTitle}**. So when rendering the report, as a group header, the entries from the **ContactTitle** data column will be output. Put a text component in the **GroupFooterBand** with the following expression: **{Count ()}**. The **{Count ()}** function will count the number of entries in each group.



8. Edit expressions, and text components:
   8.1. Drag and drop text components in the **GroupHeaderBand** and **GroupFooterBand**;
   8.2. Set the font settings: size, style, color;
   8.3. Align text components by height and width;
   8.4. Set background color of text components;
   8.5. Set the expression in the text components;
   8.6. Set the value of the required properties;
   8.7. Set **Borders** of text components, if required;
   8.8. Set the border color.

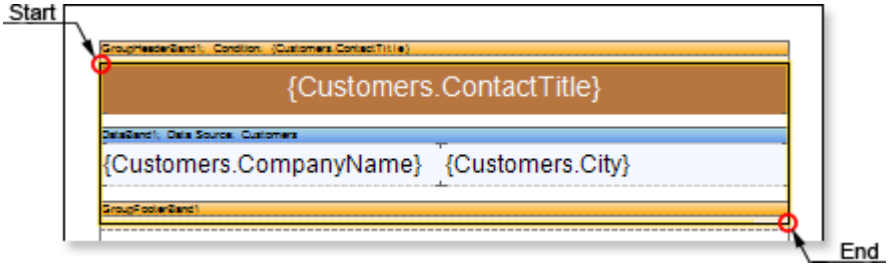


9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering a report all references to data fields will be changed on data from specified fields.

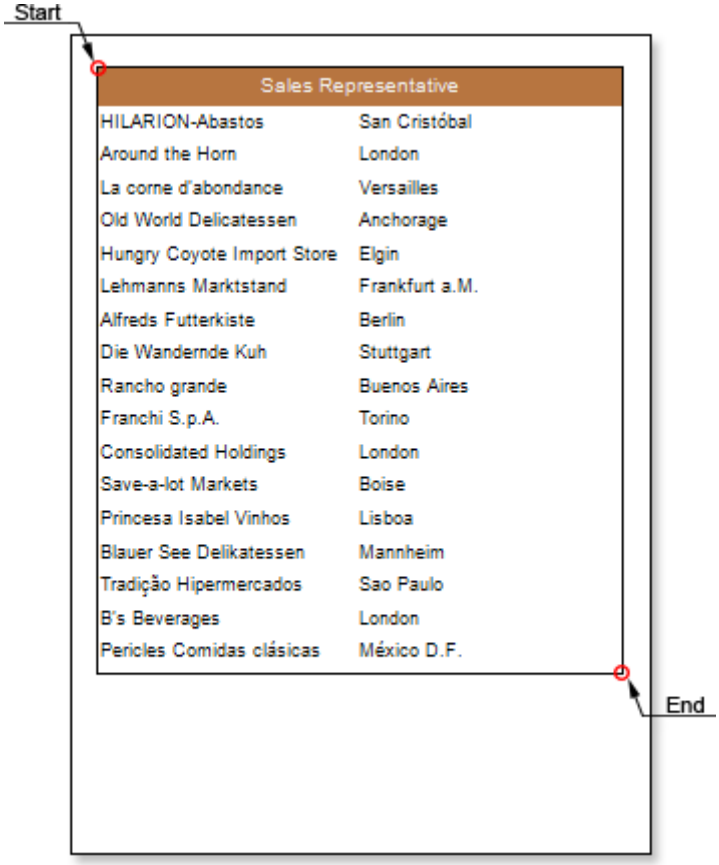


Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delícia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	Reims
Assistant Sales Agent	
Folies gourmandes	Lille
Ricardo Adocicados	Rio de Janeiro
Assistant Sales Representative	
Rattlesnake Canyon Grocery	Albuquerque

10. Go back to the report template;
11. Add the **Rectangle** cross-primitive to the report template. Starting points of the rectangle will lie in the **GroupHeaderBand**, and the end point will lie in the **GroupFooterBand**.

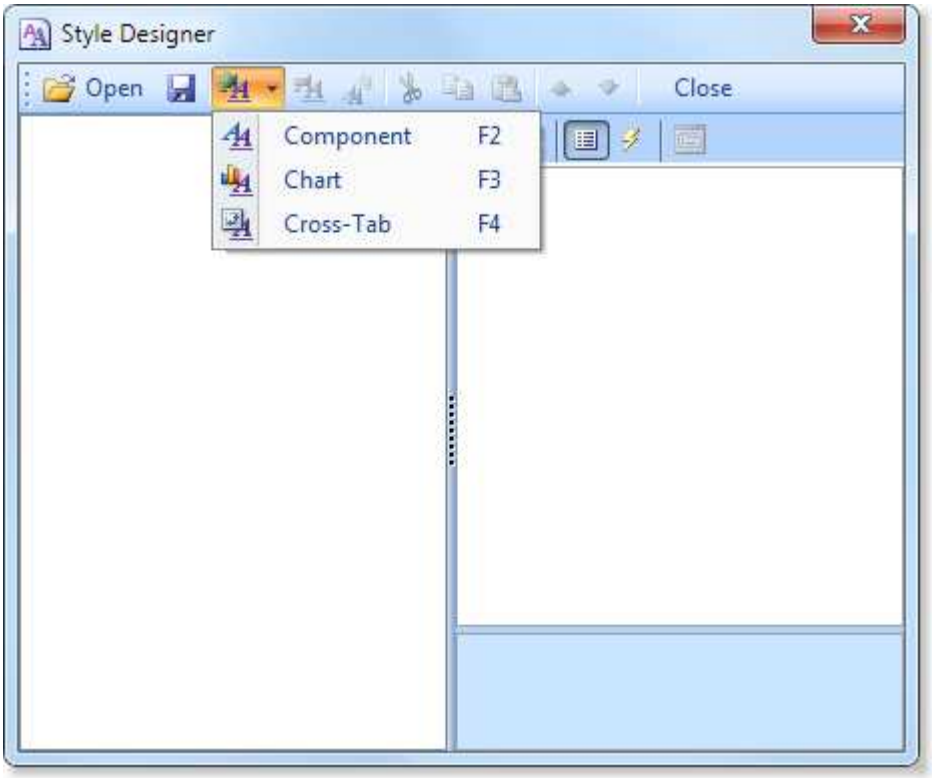


12. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering a report all references to data fields will be changed on data from specified fields. The picture below shows a rendered report page with grouping and the rendered **Rectangle** cross-primitive:

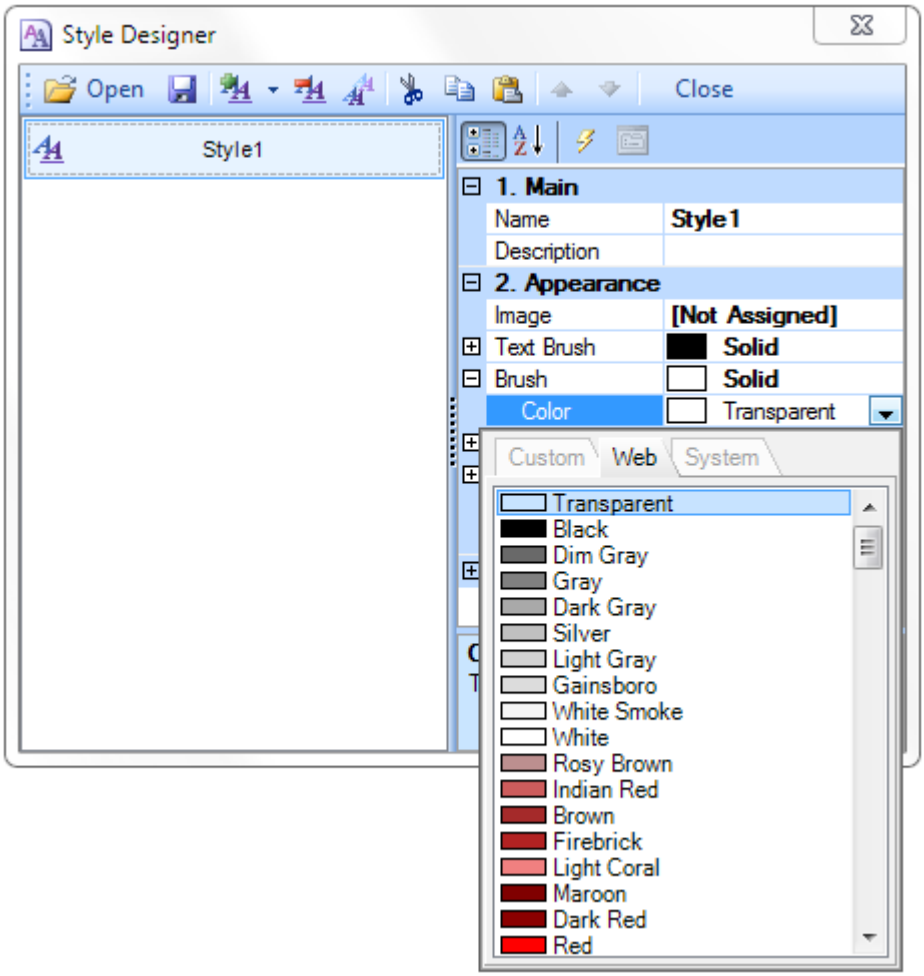


Adding Styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:



Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.

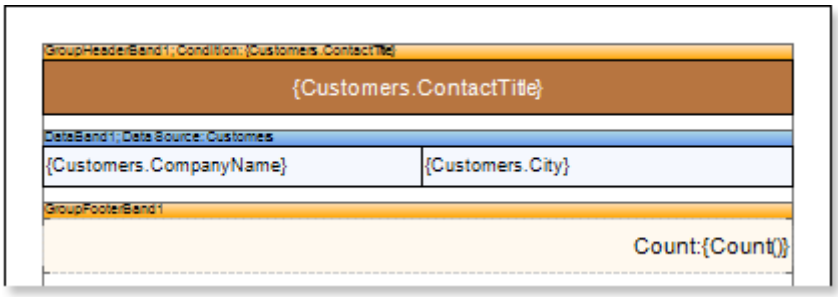
Sales Representative	
HILARION-Abastos	San Cristóbal
Around the Horn	London
La corne d'abondance	Versailles
Old World Delicatessen	Anchorage
Hungry Coyote Import Store	Elgin
Lehmanns Marktstand	Frankfurt a.M.
Alfreds Futterkiste	Berlin
Die Wandernde Kuh	Stuttgart
Rancho grande	Buenos Aires
Franchi S.p.A.	Torino
Consolidated Holdings	London
Save-a-lot Markets	Boise
Princesa Isabel Vinhos	Lisboa
Blauer See Delikatessen	Mannheim
Tradição Hipermercados	Sao Paulo
B's Beverages	London
Pericles Comidas clásicas	México D.F.

Drill-Down Report

A Drill-Down report is an interactive report in what blocks can collapse/expand its content by clicking on the block title. Follow the steps below to create a report with dynamic folding in the preview window:

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Design a report or load already created one. For example, take a group report, which was reviewed in the "**Report with Grouping**". The picture below shows a report template with groups:



4. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering a report all references to data fields will be changed on data from specified fields.

Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delicia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	Reims

Count:10

Assistant Sales Agent	
Folies gourmandes	Lille
Ricardo Adocicados	Rio de Janeiro

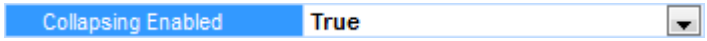
Count:2

Assistant Sales Representative	
Rattlesnake Canyon Grocery	Albuquerque

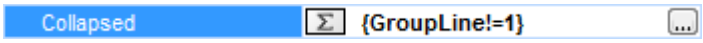
Count:1

Marketing Assistant	
Queen Cozinha	Sao Paulo
Familia Arquibaldo	Sao Paulo
Morgenstern Gesundkost	Leipzig
Mère Paillarde	Montréal

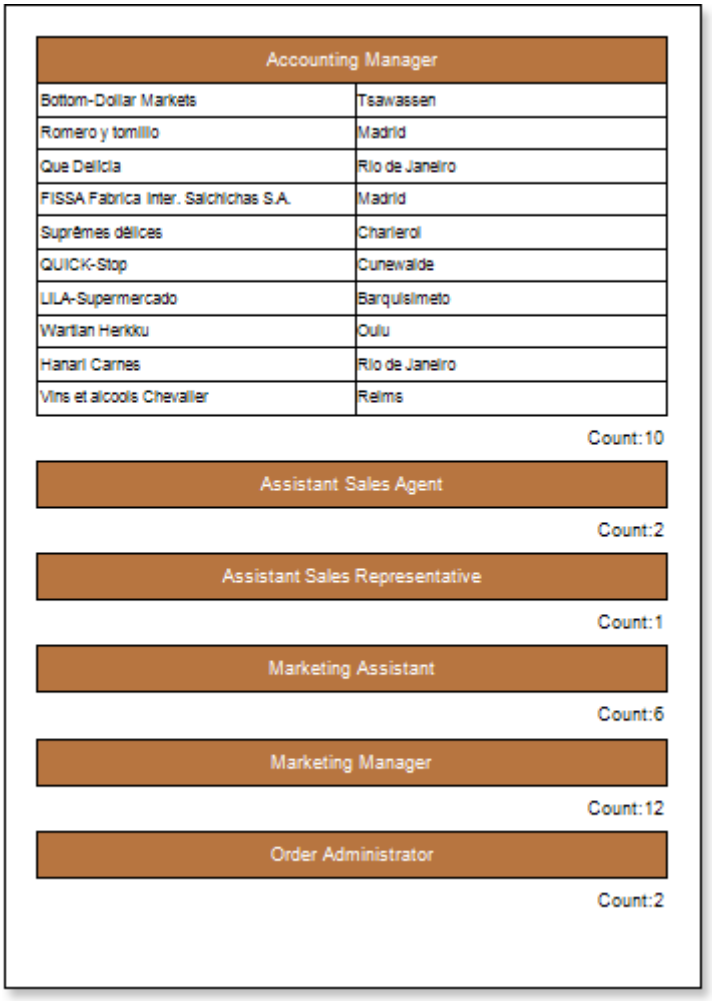
5. Go back to the report template.
6. Select the **GroupHeaderBand.**
7. Set the **Interaction.Collapsing Enabled** property to **true.**



8. Change the value of the **Interaction.Collapsed** property. In our case, set the **Interaction.Collapsed** property to **{GroupLine! = 1}**. So, when rendering a report all the groups except the first one will be collapsed.



9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering a report all references to data fields will be changed on data from specified fields.



Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delicia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	Reims

Count:10

Assistant Sales Agent

Count:2

Assistant Sales Representative

Count:1

Marketing Assistant

Count:6

Marketing Manager

Count:12

Order Administrator

Count:2

To expand or collapse a group you should click on the **GroupHeaderBand** in the rendered report. If it is necessary for the group be collapsed together with the group summary, the **Interaction.CollapseGroupFooter** property should be set to **true**. The picture below shows the report page rendered with the collapsed report:

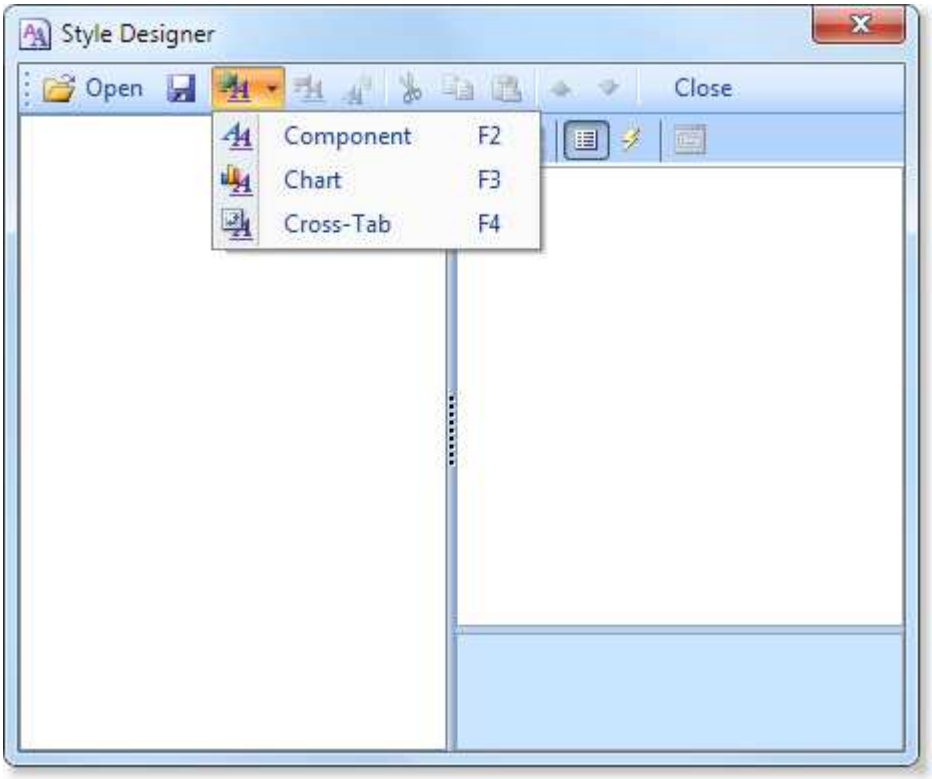
Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delicia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	Reims

Count: 10

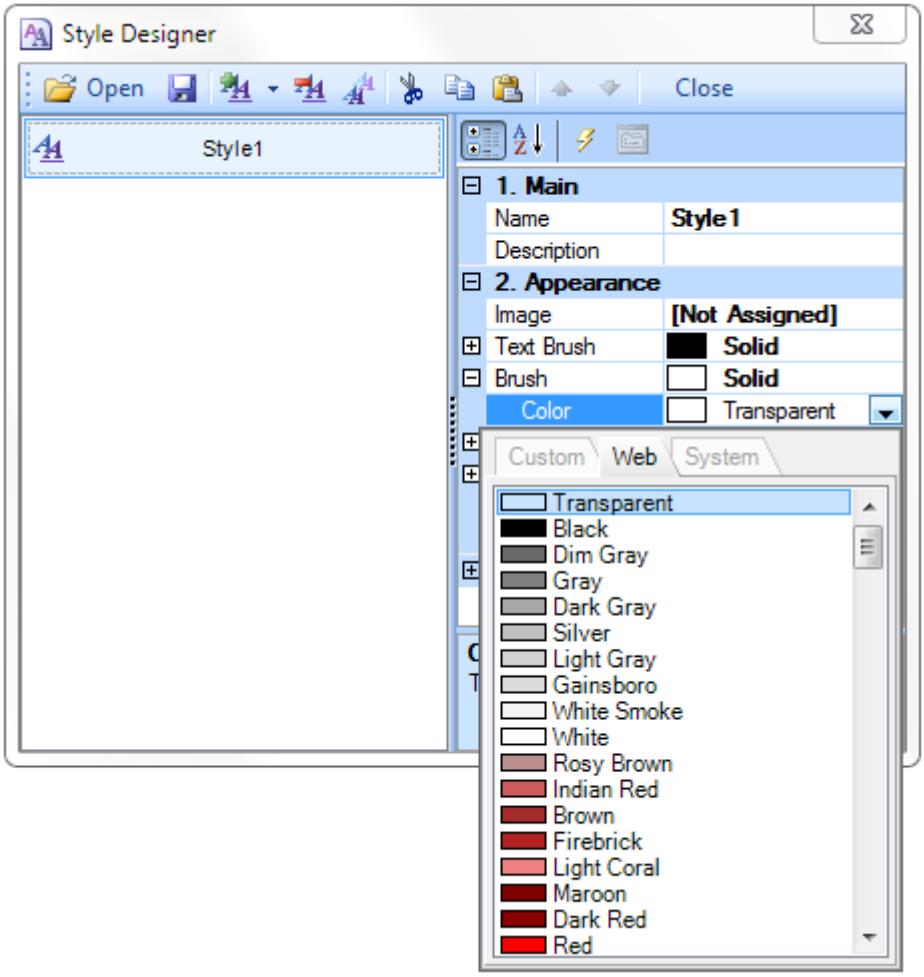
Assistant Sales Agent
Assistant Sales Representative
Marketing Assistant
Marketing Manager
Order Administrator
Owner
Owner/Marketing Assistant
Sales Agent
Sales Associate
Sales Manager

Adding Styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

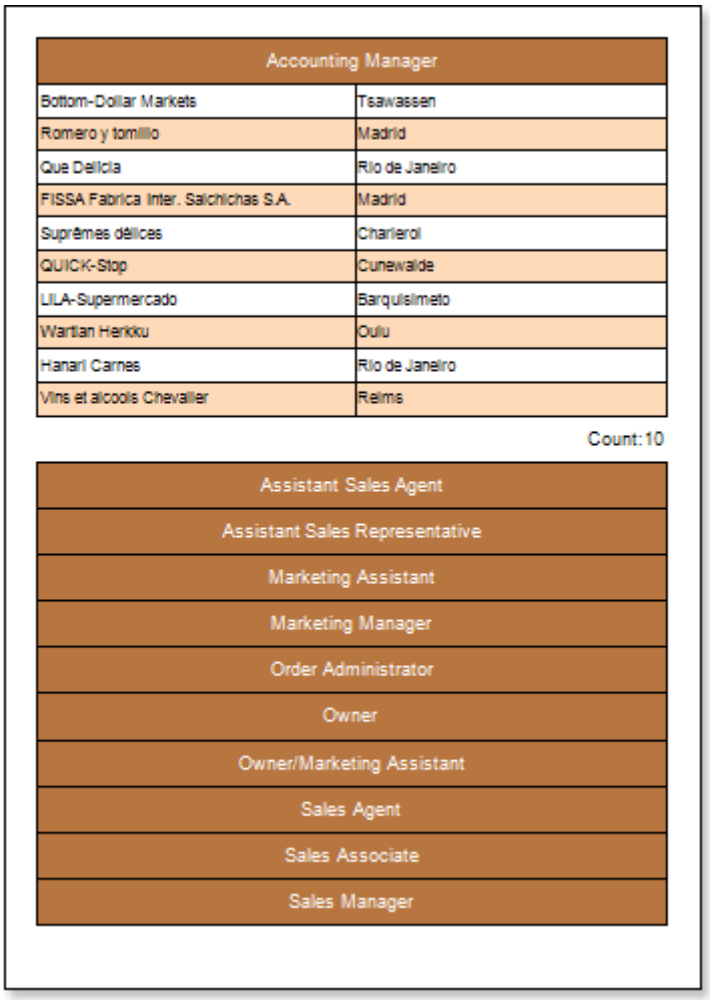


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.



Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delicia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	Reims

Count:10

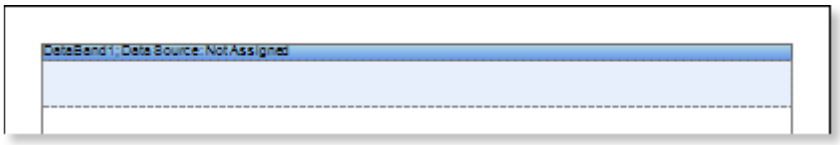
Assistant Sales Agent
Assistant Sales Representative
Marketing Assistant
Marketing Manager
Order Administrator
Owner
Owner/Marketing Assistant
Sales Agent
Sales Associate
Sales Manager

Report with Dynamic Data Sorting in Preview

When designing a report, data used in a report are not always sorted in the order that is needed. In this case, the sorting can be done by means of the report generator. One way to sort the data is dynamic sorting. A report with dynamic data sorting in the preview window is an interactive report in which changing of dynamic data sorting is done by clicking the component, which dynamic sorting is enabled. Follow the steps below in order to render a report with dynamic data sorting in the preview window:

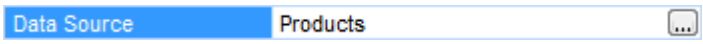
1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Put a **DataBand** on a page of a report template.



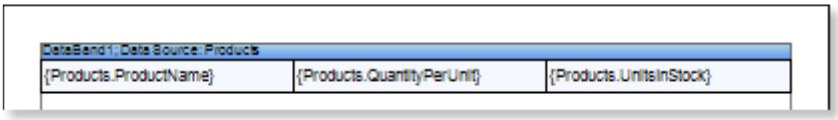
4. Edit **DataBand**:
   4.1. Align the **DataBand** by height;
   4.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
   4.3. Change the **DataBand** background;
   4.4. Enable **Borders** for the **DataBand**, if required;
   4.5. Change the border color.

5. Set the data source for the **DataBand** using the **Data Source** property:

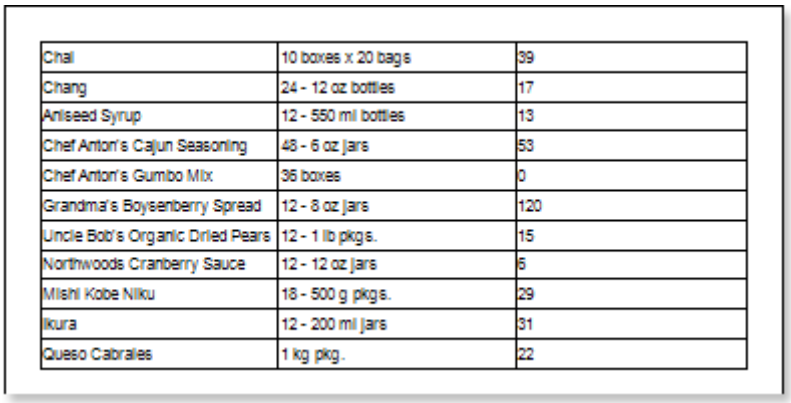


6. Put text components with expressions in the **DataBand**. Where expression is a reference to the data field. For example, put three text components with expressions: **{Products.ProductName}**, **{Products.QuantityPerUnit}**, and **{Products.UnitsInStock}**;

7. Edit **Text** and **TextBox** component:
   7.1. Drag and drop the text component in the **DataBand**;
   7.2. Change parameters of the text font: size, type, color;
   7.3. Align the text component by width and height;
   7.4. Change the background of the text component;
   7.5. Align text in the text component;
   7.6. Change the value of properties of the text component. For example, set the **Word Wrap** property to **true**, if you need a text to be wrapped;
   7.7. Enable **Borders** for the text component, if required.
   7.8. Change the border color.

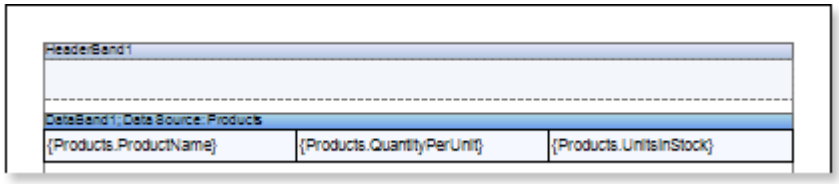


8. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database.

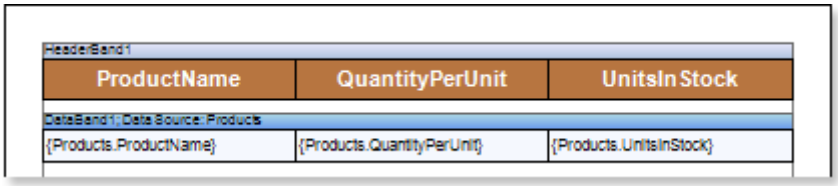


Chai	10 boxes x 20 bags	39
Chang	24 - 12 oz bottles	17
Aniseed Syrup	12 - 550 ml bottles	13
Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
Chef Anton's Gumbo Mix	36 boxes	0
Grandma's Boysenberry Spread	12 - 8 oz jars	120
Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
Northwoods Cranberry Sauce	12 - 12 oz jars	6
Mishi Kobe Niku	18 - 500 g pkgs.	29
Ikura	12 - 200 ml jars	31
Queso Cabrales	1 kg pkg.	22

9.Go back to the report template;
10. If needed, add other bands to the report template, for example, **ReportTitleBand** and **ReportSummaryBand**;
11. Edit these bands:
   11.1. Align them by height;
   11.2. Change values of properties, if required;
   11.3. Change the background of bands;
   11.4. Enable **Borders**, if required;
   11.5. Set the border color.



12. Put text components with expressions in the these bands. The expression in the text component is a title in the **ReportTitleBand**, and a summary in the **ReportSummaryBand**.
13. Edit text and text components:
    13.1. Drag and drop the text component in the band;
    13.2. Change font options: size, type, color;
    13.3. Align text component by height and width;
    13.4. Change the background of the text component;
    13.5. Align text in the text component;
    13.6. Change values of text component properties, if required;
    13.7. Enable **Borders** of the text component, if required;
    13.8. Set the border color.



14. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database.

ProductName	QuantityPerUnit	UnitsInStock
Chai	10 boxes x 20 bags	39
Chang	24 - 12 oz bottles	17
Aniseed Syrup	12 - 550 ml bottles	13
Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
Chef Anton's Gumbo Mix	36 boxes	0
Grandma's Boysenberry Spread	12 - 8 oz jars	120
Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
Northwoods Cranberry Sauce	12 - 12 oz jars	6
Mishi Kobe Niku	18 - 500 g pkgs.	29
Ikura	12 - 200 ml jars	31
Queso Cabrales	1 kg pkg.	22
Queso Manchego La Pastora	10 - 500 g pkgs.	86
Konbu	2 kg box	24
Tofu	40 - 100 g pkgs.	35
Genen Shouyu	24 - 250 ml bottles	39
Pavlova	32 - 500 g boxes	29
Alice Mutton	20 - 1 kg tins	0
Carnarvon Tigers	16 kg pkg.	42
Teatime Chocolate Biscuits	10 boxes x 12 pieces	25
Sir Rodney's Marmalade	30 gift boxes	40
Sir Rodney's Scones	24 pkgs. x 4 pieces	3
Gustaf's Knäckebröd	24 - 500 g pkgs.	104
Tunnbröd	12 - 250 g pkgs.	61
Guaraná Fantástica	12 - 355 ml cans	20
NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	76
Gumbär Gummibärchen	100 - 250 g bags	15
Schoggi Schokolade	100 - 100 g pieces	49
Rössle Sauerkraut	25 - 825 g cans	26
Thüringer Rostbratwurst	50 bags x 30 sausgs.	0
Nord-Ost Matjesherling	10 - 200 g glasses	10
Gorgonzola Telino	12 - 100 g pkgs	0
Mascarpone Fabioli	24 - 200 g pkgs.	9
Geitost	500 g	112

15. Go back to the report template;
16. Select a text component or any other component, on what one clicks and in the rendered report sorting will be done. In this case, select the **TextBox4** component in the **HeaderBand** with the **ProductName** text;
17. Change the value of the **Interaction.Sorting Column** property. The value of this property will be a column of the data source by what sorting will be be done. Set the **Interaction.Sorting Column** property to **DataBand1.ProductName**;
18. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database.

ProductName	QuantityPerUnit	UnitsInStock
Chai	10 boxes x 20 bags	39
Chang	24 - 12 oz bottles	17
Aniseed Syrup	12 - 550 ml bottles	13
Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
Chef Anton's Gumbo Mix	36 boxes	0
Grandma's Boysenberry Spread	12 - 8 oz jars	120
Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
Northwoods Cranberry Sauce	12 - 12 oz jars	6
Mishi Kobe Niku	18 - 500 g pkgs.	29
Ikura	12 - 200 ml jars	31
Queso Cabrales	1 kg pkg.	22
Queso Manchego La Pastora	10 - 500 g pkgs.	86
Konbu	2 kg box	24
Tofu	40 - 100 g pkgs.	35
Genen Shouyu	24 - 250 ml bottles	39
Pavlova	32 - 500 g boxes	29
Alice Mutton	20 - 1 kg tins	0
Carnarvon Tigers	16 kg pkg.	42
Teatime Chocolate Biscuits	10 boxes x 12 pieces	25
Sir Rodney's Marmalade	30 gift boxes	40
Sir Rodney's Scones	24 pkgs. x 4 pieces	3
Gustaf's Knäckebröd	24 - 500 g pkgs.	104
Tunnbröd	12 - 250 g pkgs.	61
Guaraná Fantástica	12 - 355 ml cans	20
NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	76
Gumbär Gummibärchen	100 - 250 g bags	15
Schoggi Schokolade	100 - 100 g pieces	49
Rössle Sauerkraut	25 - 825 g cans	26
Thüringer Rostbratwurst	50 bags x 30 sausgs.	0
Nord-Ost Matjesherling	10 - 200 g glasses	10
Gorgonzola Telino	12 - 100 g pkgs	0
Mascarpone Fabioli	24 - 200 g pkgs.	9
Geitost	500 g	112

19. To enable sorting of data by the specified data column, you should click a report component which the **Interaction.Sorting Column** property was set earlier. In our example, you should click the **TextBox4**. After clicking the text component, data will be sorted in **Ascending** direction. To change the sorting direction from **Ascending** to **Descending**, you need to click the text component again, i.e. each time after clicking the text component sorting direction will be changed. The picture below shows the first page of the report rendered with different sorting directions:

Ascending

ProductName	QuantityPerUnit	UnitsInStock
Alice Mutton	20 - 1 kg tins	0
Aniseed Syrup	12 - 550 ml bottles	13
Boston Crab Meat	24 - 4 oz tins	123
Camembert Pierrot	15 - 300 g rounds	19
Carnarvon Tigers	16 kg pkg.	42
Chai	10 boxes x 20 bags	39
Chang	24 - 12 oz bottles	17
Chartreuse verte	750 cc per bottle	69
Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
Chef Anton's Gumbo Mix	36 boxes	0
Chocolade	10 pkgs.	15
Côte de Blaye	12 - 75 cl bottles	17
Escargots de Bourgogne	24 pieces	62
Filo Mix	16 - 2 kg boxes	38
Flotemysost	10 - 500 g pkgs.	26
Geitost	500 g	112
Genen Shouyu	24 - 250 ml bottles	39
Gnocchi di nonna Alice	24 - 250 g pkgs.	21
Gorgonzola Telino	12 - 100 g pkgs	0
Grandma's Boysenberry Spread	12 - 8 oz jars	120
Gravad lax	12 - 500 g pkgs.	11
Guaraná Fantástica	12 - 355 ml cans	20
Gudbrandsdalsost	10 kg pkg.	26
Gula Malacca	20 - 2 kg bags	27
Gumbär Gummibärchen	100 - 250 g bags	15
Gustaf's Knäckebröd	24 - 500 g pkgs.	104
Ikura	12 - 200 ml jars	31
Inlagd Sill	24 - 250 g jars	112
Ipoh Coffee	16 - 500 g tins	17
Jack's New England Clam Chowder	12 - 12 oz cans	85
Konbu	2 kg box	24
Lakkalikööri	500 ml	57
Laughing Lumberjack Lager	24 - 12 oz bottles	52

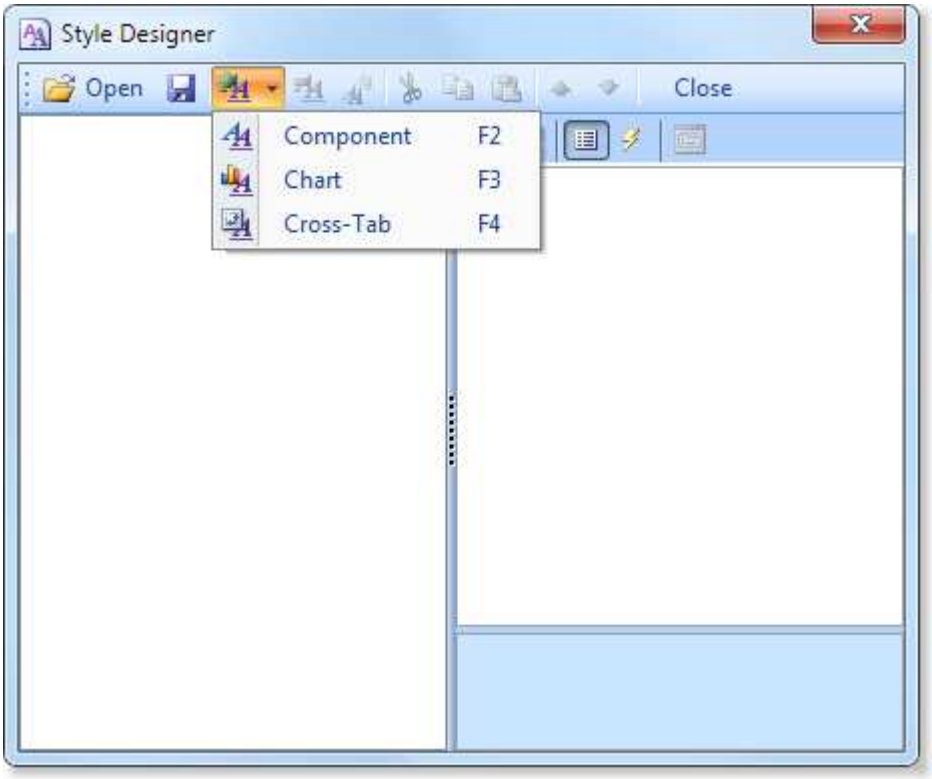
Descending

ProductName	QuantityPerUnit	UnitsInStock
Zaanse koeken	10 - 4 oz boxes	36
Wimmers gute Semmelknödel	20 bags x 4 pieces	22
Vegie-spread	15 - 625 g jars	24
Valkoinen suklaa	12 - 100 g bars	65
Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
Tunnbröd	12 - 250 g pkgs.	61
Tourtière	16 pies	21
Tofu	40 - 100 g pkgs.	35
Thüringer Rostbratwurst	50 bags x 30 sausgs.	0
Teatime Chocolate Biscuits	10 boxes x 12 pieces	25
Tarte au sucre	48 pies	17
Steeleye Stout	24 - 12 oz bottles	20
Spegesild	4 - 450 g glasses	95
Sirop d'érable	24 - 500 ml bottles	113
Sir Rodney's Scones	24 pkgs. x 4 pieces	3
Sir Rodney's Marmalade	30 gift boxes	40
Singaporean Hokkien Fried Mee	32 - 1 kg pkgs.	26
Scottish Longbreads	10 boxes x 8 pieces	6
Schoggi Schokolade	100 - 100 g pieces	49
Sasquatch Ale	24 - 12 oz bottles	111
Rössle Sauerkraut	25 - 825 g cans	26
Rogede sild	1k pkg.	5
Röd Kaviar	24 - 150 g jars	101
Rhönbräu Klosterbier	24 - 0.5 l bottles	125
Ravioli Angelo	24 - 250 g pkgs.	36
Raclette Courdavault	5 kg pkg.	79
Queso Manchego La Pastora	10 - 500 g pkgs.	86
Queso Cabrales	1 kg pkg.	22
Perth Pasties	48 pieces	0
Pavlova	32 - 500 g boxes	29
Pâté chinois	24 boxes x 2 pies	115
Outback Lager	24 - 355 ml bottles	15
Original Frankfurter grüne Soße	12 boxes	32

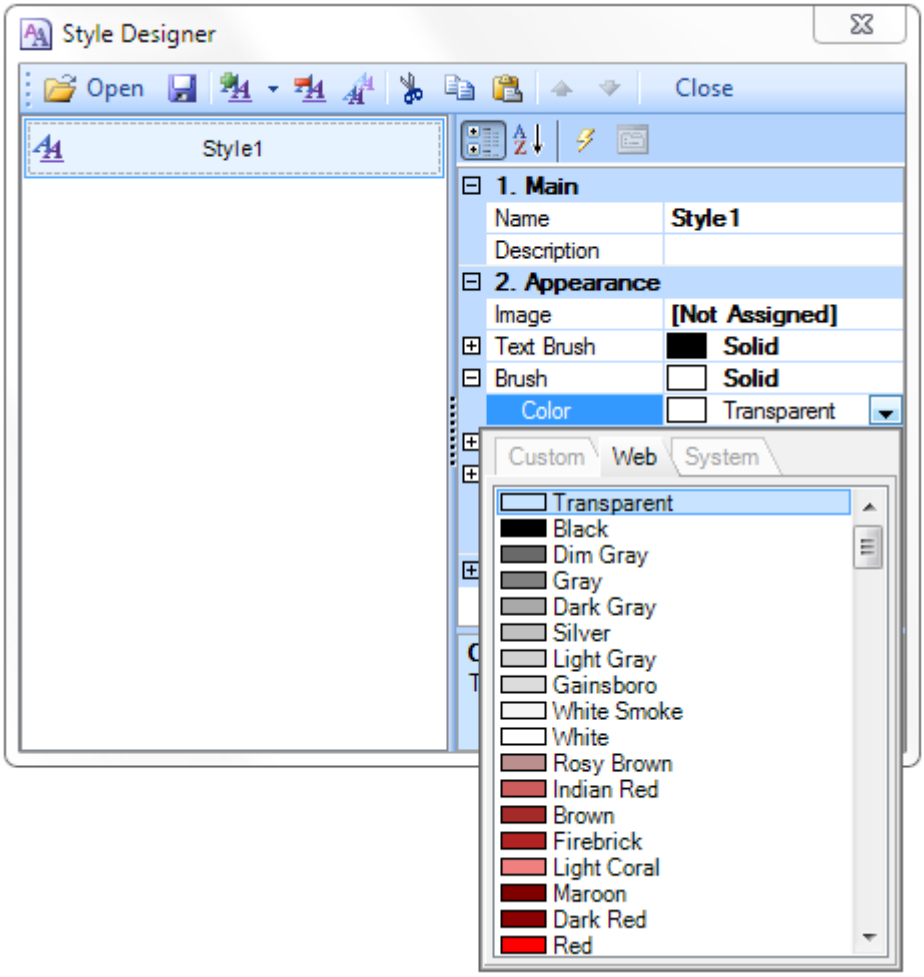
Sorting direction displays the "arrow" icon.

Adding Styles

1. Go back to the report template;
2. Select **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:



Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.

Ascending

ProductName	QuantityPerUnit	UnitsInStock
Alice Mutton	20 - 1 kg tins	0
Aniseed Syrup	12 - 550 ml bottles	13
Boston Crab Meat	24 - 4 oz tins	123
Camembert Pierrot	15 - 300 g rounds	19
Carnarvon Tigers	16 kg pkg.	42
Chai	10 boxes x 20 bags	39
Chang	24 - 12 oz bottles	17
Chartreuse verte	750 cc per bottle	69
Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
Chef Anton's Gumbo Mix	36 boxes	0
Chocolade	10 pkgs.	15
Côte de Blaye	12 - 75 cl bottles	17
Escargots de Bourgogne	24 pieces	62
Filo Mix	16 - 2 kg boxes	38
Flotemysost	10 - 500 g pkgs.	26
Geitost	500 g	112
Genen Shouyu	24 - 250 ml bottles	39
Gnocchi di nonna Alice	24 - 250 g pkgs.	21
Gorgonzola Telino	12 - 100 g pkgs	0
Grandma's Boysenberry Spread	12 - 8 oz jars	120
Gravad lax	12 - 500 g pkgs.	11
Guaraná Fantástica	12 - 355 ml cans	20
Gudbrandsdalsost	10 kg pkg.	26
Gula Malacca	20 - 2 kg bags	27
Gumbär Gummibärchen	100 - 250 g bags	15
Gustaf's Knäckebröd	24 - 500 g pkgs.	104
Ikura	12 - 200 ml jars	31
Inlagd Sill	24 - 250 g jars	112
Ipoh Coffee	16 - 500 g tins	17
Jack's New England Clam Chowder	12 - 12 oz cans	85
Konbu	2 kg box	24
Lakkalikööri	500 ml	57
Laughing Lumberjack Lager	24 - 12 oz bottles	52

Descending

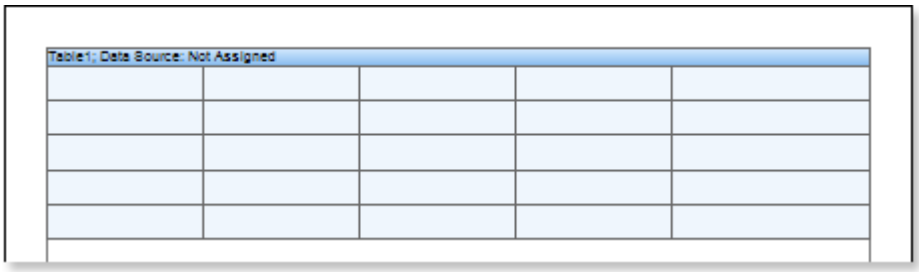
ProductName	QuantityPerUnit	UnitsInStock
Zaanse koeken	10 - 4 oz boxes	36
Wimmers gute Semmelknödel	20 bags x 4 pieces	22
Vegie-spread	15 - 625 g jars	24
Valkoinen suklaa	12 - 100 g bars	65
Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
Tunnbröd	12 - 250 g pkgs.	61
Tourtière	16 pies	21
Tofu	40 - 100 g pkgs.	35
Thüringer Rostbratwurst	50 bags x 30 sausgs.	0
Teatime Chocolate Biscuits	10 boxes x 12 pieces	25
Tarte au sucre	48 pies	17
Steeleye Stout	24 - 12 oz bottles	20
Spegesild	4 - 450 g glasses	95
Sirop d'érable	24 - 500 ml bottles	113
Sir Rodney's Scones	24 pkgs. x 4 pieces	3
Sir Rodney's Marmalade	30 gift boxes	40
Singaporean Hokkien Fried Mee	32 - 1 kg pkgs.	26
Scottish Longbreads	10 boxes x 8 pieces	6
Schoggi Schokolade	100 - 100 g pieces	49
Sasquatch Ale	24 - 12 oz bottles	111
Rössle Sauerkraut	25 - 825 g cans	26
Rogede sild	1k pkg.	5
Röd Kaviar	24 - 150 g jars	101
Rhönbräu Klosterbier	24 - 0.5 l bottles	125
Ravioli Angelo	24 - 250 g pkgs.	36
Raclette Courdavault	5 kg pkg.	79
Queso Manchego La Pastora	10 - 500 g pkgs.	86
Queso Cabrales	1 kg pkg.	22
Perth Pasties	48 pieces	0
Pavlova	32 - 500 g boxes	29
Pâté chinois	24 boxes x 2 pies	115
Outback Lager	24 - 355 ml bottles	15
Original Frankfurter grüne Soße	12 boxes	32

Report with Table Component

Do the following steps to design a report with the **Table** component:

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Put a **Table** component on a page of a report template.



4. Edit the **Table** component:

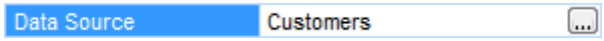
4.1. Set the amount of columns and rows using, for example, the **RowCount** and **ColumnCount** properties. Set these properties to 5 and 3 respectively;

4.2. Set the number of headers and footers in the table using, for example, the **HeaderRowsCount** and **FooterRowsCount** properties. Set the **HeaderRowsCount** property to **1**;

4.3. Align the **Table** component by height;

4.4. Change values of the component. for example, set the **CanBreak** property to **true**, if it is required for the **Table** component be broken;

5. Set the data source of the **Table** component using the **Data Source** property:



6. Put some text and expressions in the table cells. For example, cells of the first and third rows will contain only text, that will be a data header. Cells of the second and fourth rows will contain expressions, references to data source;

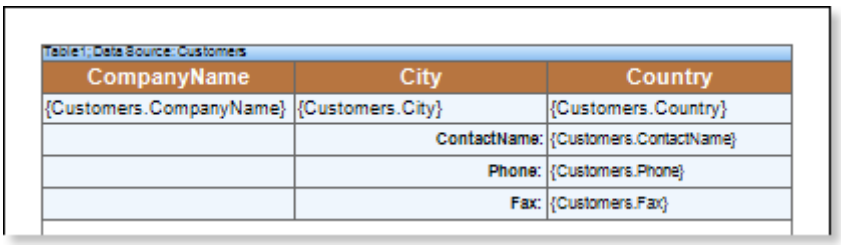
7. Edit text and cells:

7.1. Set font parameters of text: size, style, color;

7.2. Set color of table cells;

7.3. Align text in cells;

7.4. Change values of cells. For example, set the **WordWrap** property to **true**, if it is necessary for the text to be wrapped.

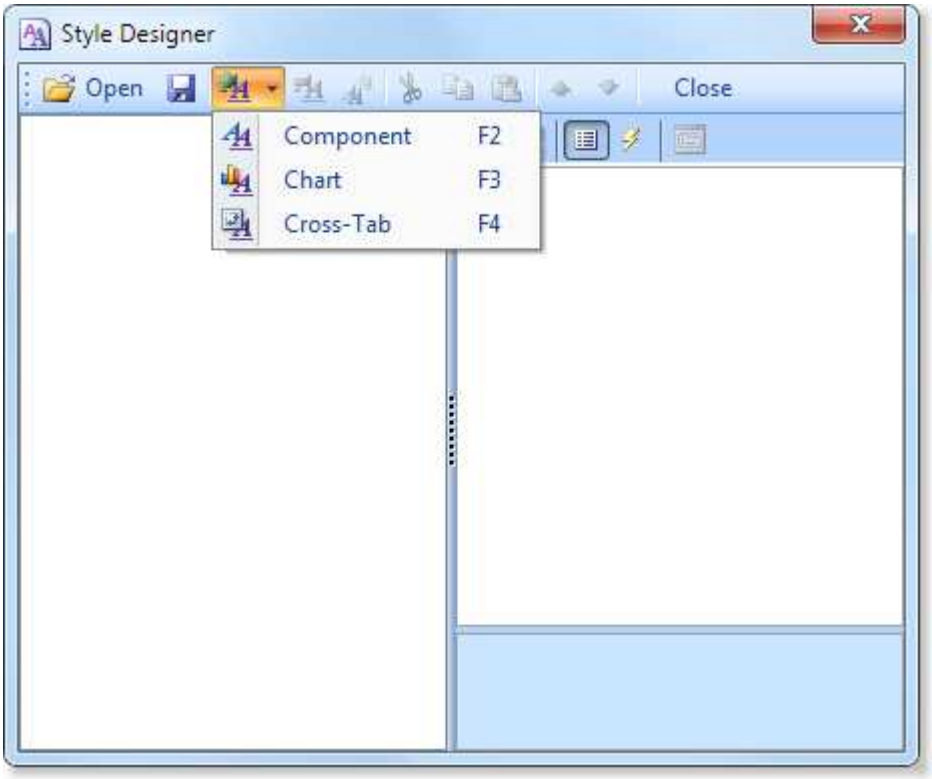


8. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **Table** in the rendered report will be the same as the amount of data rows in the database.

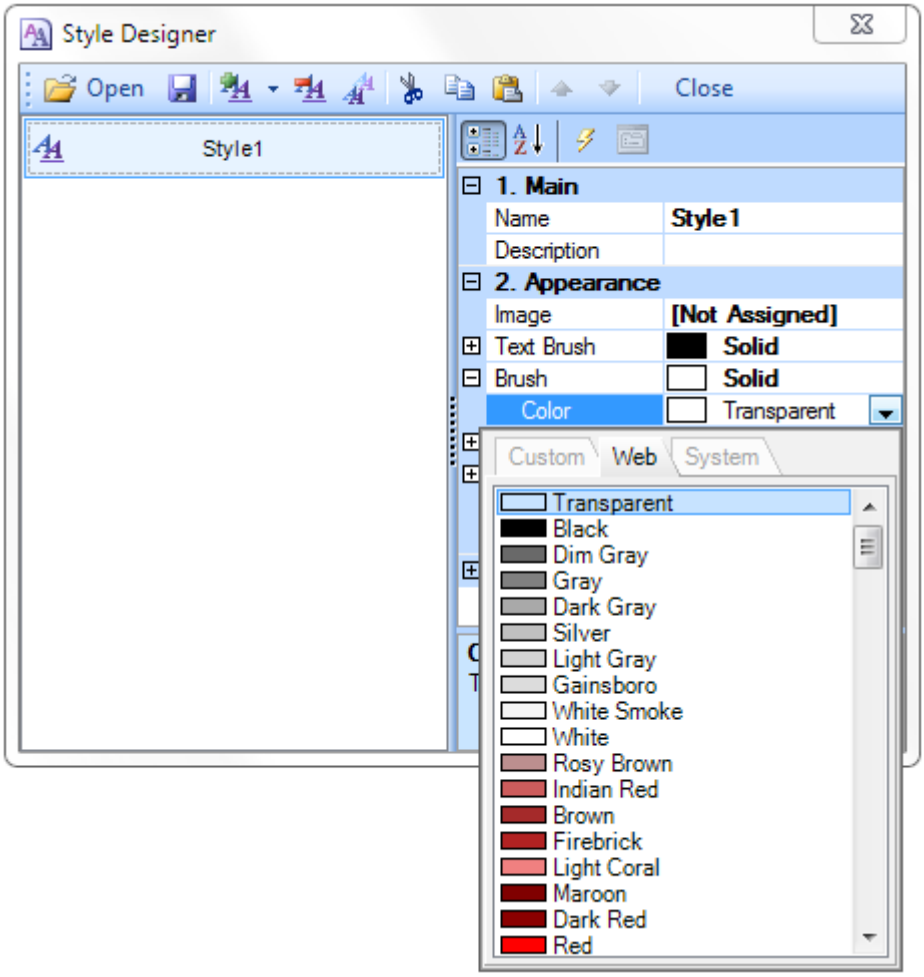
CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
	ContactName:	Maria Anders
	Phone:	030-0074321
	Fax:	030-0076545
Ana Trujillo Emparedados y helados	México D.F.	Mexico
	ContactName:	Ana Trujillo
	Phone:	(5) 555-4729
	Fax:	(5) 555-3745
Antonio Moreno Taquería	México D.F.	Mexico
	ContactName:	Antonio Moreno
	Phone:	(5) 555-3932
	Fax:	
Around the Horn	London	UK
	ContactName:	Thomas Hardy
	Phone:	(171) 555-7788
	Fax:	(171) 555-6750
Berglunds snabbköp	Luleå	Sweden
	ContactName:	Christina Berglund
	Phone:	0921-12 34 65
	Fax:	0921-12 34 67
Blauer See Delikatessen	Mannheim	Germany
	ContactName:	Hanna Moos
	Phone:	0621-08460
	Fax:	0621-08924
Blondesddsl père et fils	Strasbourg	France
	ContactName:	Frédérique Citeaux
	Phone:	88.60.15.31
	Fax:	88.60.15.32
Bólido Comidas preparadas	Madrid	Spain
	ContactName:	Martín Sommer
	Phone:	(91) 555 22 82
	Fax:	(91) 555 91 99

Adding Styles

1. Go back to the report template;
2. Select the **Table** component;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

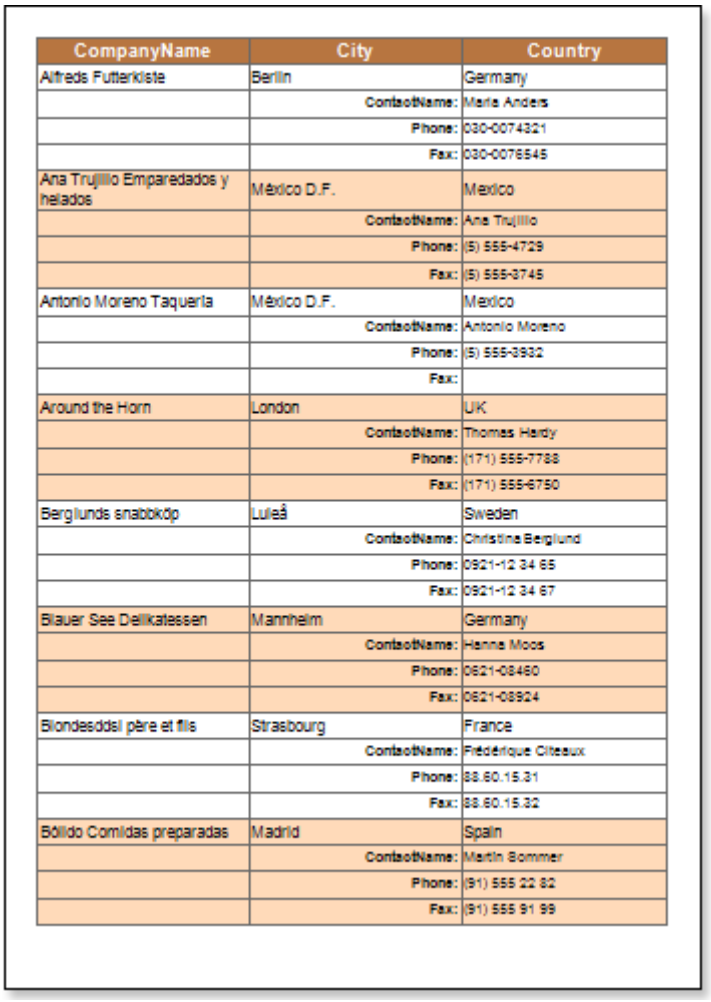


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.



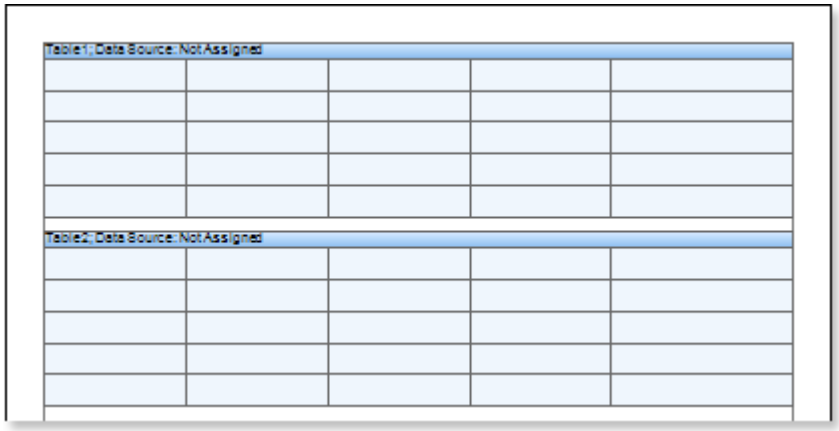
CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
	ContactName:	Maria Anders
	Phone:	030-0074321
	Fax:	030-0076545
Ana Trujillo Emparedados y helados	México D.F.	Mexico
	ContactName:	Ana Trujillo
	Phone:	(5) 555-4729
	Fax:	(5) 555-3745
Antonio Moreno Taquería	México D.F.	Mexico
	ContactName:	Antonio Moreno
	Phone:	(5) 555-3932
	Fax:	
Around the Horn	London	UK
	ContactName:	Thomas Hardy
	Phone:	(171) 555-7788
	Fax:	(171) 555-6750
Berglunds snabbköp	Luleå	Sweden
	ContactName:	Christina Berglund
	Phone:	0921-12 34 65
	Fax:	0921-12 34 67
Blauer See Delikatessen	Mannheim	Germany
	ContactName:	Hanna Moos
	Phone:	0621-08460
	Fax:	0621-08924
Blondesddsl père et fils	Strasbourg	France
	ContactName:	Frédérique Citeaux
	Phone:	88.60.15.31
	Fax:	88.60.15.32
Bólido Comidas preparadas	Madrid	Spain
	ContactName:	Martín Sommer
	Phone:	(91) 555 22 82
	Fax:	(91) 555 91 99

Master-Detail Report with Table

Do the following steps to design a **Master-Detail** report with the **Table** component:

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

3. Create **Relation** between data sources. If the relation will not be created and/or the **Relation** property of the **Detail** data source will not be filled, then, for **Master** entry, all **Detail** entries will be output.
4. Put two **Table** components on a page of a report template.



5. Edit **Table** components:

5.1. Change the number of rows and columns in the **Table** component. For example, using the **RowCount** and **ColumnCount** properties. Set the **RowCount** and **ColumnCount** properties of the **Table1** component to **3** and **1** respectively. And for the **Table2** component - values of **3** and **3**;

5.2. Set the number of headers and footers in the table using, for example, the **HeaderRowsCount** and **FooterRowsCount** properties. Set the **FooterRowsCount** property of the **Table1** to **1**. Set the **HeaderRowsCount** and **FooterRowsCount** property of the **Table2** to **1** and **1** respectively;

5.3. Align the **Table** component by height;

5.4. Set the height of rows in the table. To do this, select the **Table** component and, dragging the horizontal border line, edit the row height. In addition, if you want to change the row height, leaving the height of the **Table** component unchanged, it is necessary to hold down the **Ctrl** button before editing the row height;

5.5. Change columns width in the table. To do this, select the **Table** component, and change width by dragging the vertical border of a column;

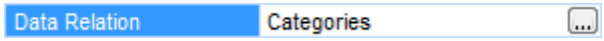
5.6. Change values of properties. For example, set the **Print if Detail Empty** property of the **Table** component, which is the **Master** component in the **Master-Detail** report, to **true**, if you want the **Master** entries be printed in any case, even if the **Detail** entries are not available. Set the **CanShrink** property of the **Table** component, which is the **Detail** component in the **Master-Detail** report to **true**, if you want this component be shrunk;

5.7. Set color of table cells;

5.8. Set **Borders** of cells of the **Table** component, if necessary;

6. Specify data sources for the **Table** components, as well as set the **Master** component. In our case, the **Master** component is the **Table1**. This means that in the **Data Setup** window of the **Table2** component on the tab of the **Master Component**, specify **Table1** as the **Master** component;

7. Fill in the DataRelation property of the **Table2** component, which is the **Detail** entry in this report:



8. Set expressions in table cells. Where an expression is a reference to a data source. For example: the **Table1** component, which is the **Master** component, set the following expressions for the first and second rows: **{Categories.CategoryName}** and **{Categories.Description}**, respectively. The third row of the **Table1** is a total row, and in this case, it is blank. The first row of the **Table2** is the header row of data, so the expression in cells of the first row will be the data header. In the cells of the second row we specify references to data sources. The third row in the **Table2** is the total row, so the expression in this line will be a total. Set the Count function for the third row;

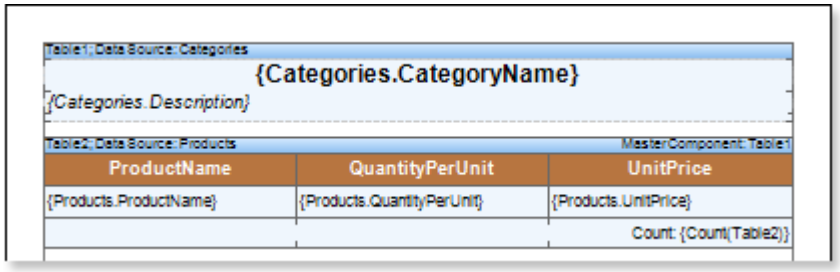
9. Edit text boxes and cells:

9.1. Set the font options: size, style, color;

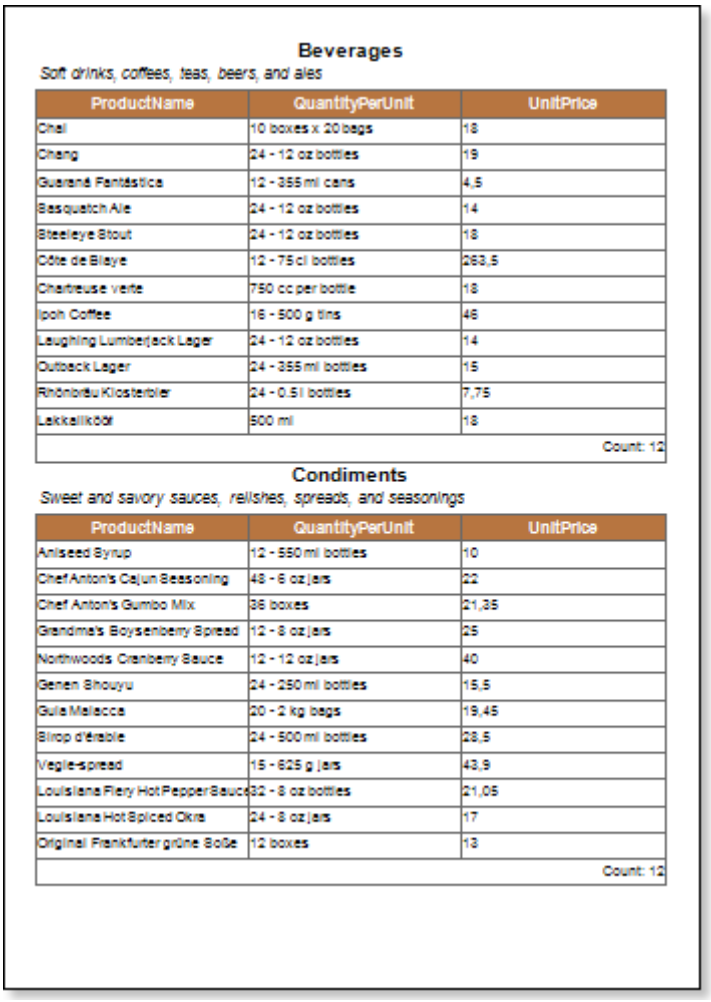
9.2. Set the background color of cells;

9.3. Align the text in cells;

9.4. Set the value of properties of cells. For example, set the **Word Wrap** property to **true**, if you want the text be wrapped;



10. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields.



Beverages

Soft drinks, coffees, teas, beers, and ales

ProductName	QuantityPerUnit	UnitPrice
Chai	10 boxes x 20 bags	18
Chang	24 - 12 oz bottles	19
Guaraná Fantástica	12 - 355 ml cans	4,5
Sasquatch Ale	24 - 12 oz bottles	14
Steeleye Stout	24 - 12 oz bottles	18
Côte de Blaye	12 - 75 cl bottles	263,5
Chartreuse verte	750 cc per bottle	18
Ipoh Coffee	16 - 500 g tins	46
Laughing Lumberjack Lager	24 - 12 oz bottles	14
Outback Lager	24 - 355 ml bottles	15
Rhönbräu Klosterbier	24 - 0.5 l bottles	7,75
Lakkalikööri	500 ml	18
		Count: 12

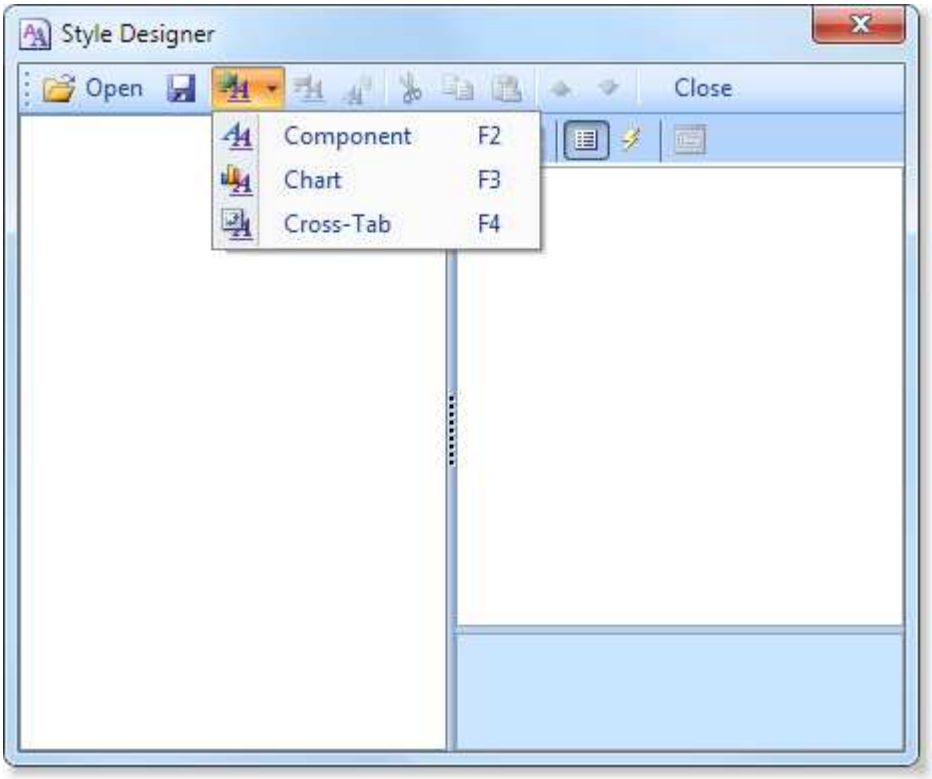
Condiments

Sweet and savory sauces, relishes, spreads, and seasonings

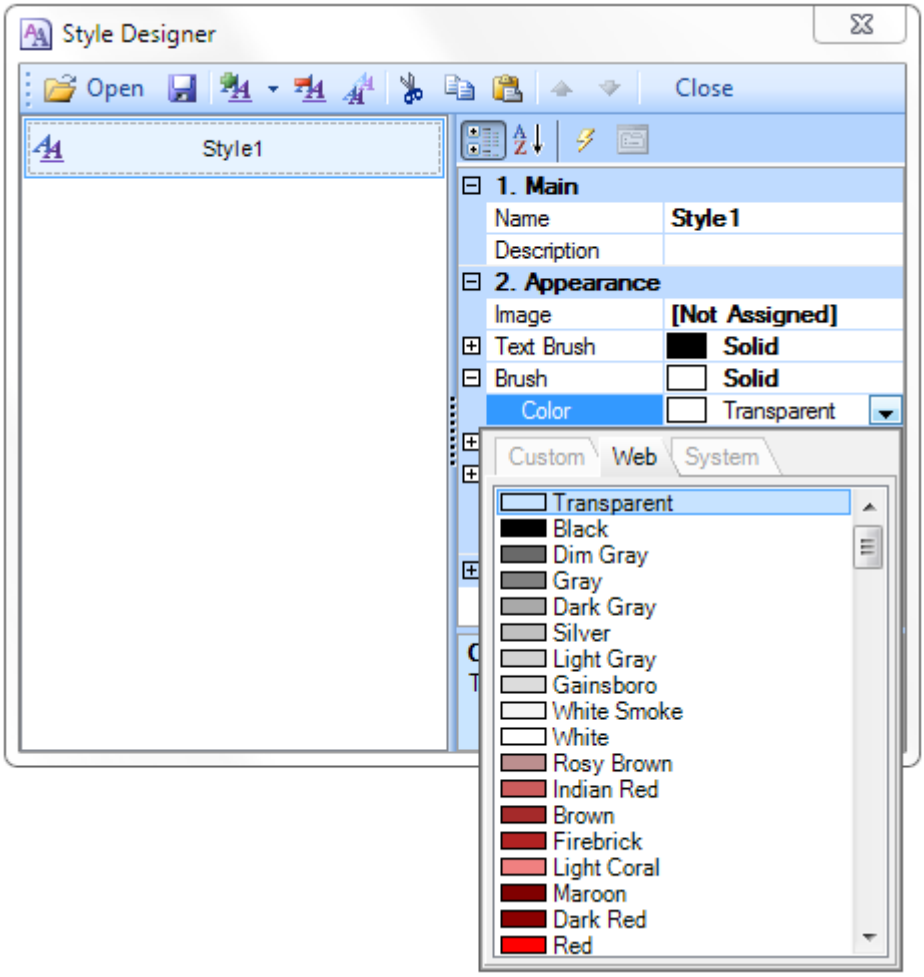
ProductName	QuantityPerUnit	UnitPrice
Aniseed Syrup	12 - 550 ml bottles	10
Chef Anton's Cajun Seasoning	48 - 6 oz jars	22
Chef Anton's Gumbo Mix	36 boxes	21,35
Grandma's Boysenberry Spread	12 - 8 oz jars	25
Northwoods Cranberry Sauce	12 - 12 oz jars	40
Genen Shouyu	24 - 250 ml bottles	15,5
Gula Malacca	20 - 2 kg bags	19,45
Sirop d'érable	24 - 500 ml bottles	28,5
Vegie-spread	15 - 625 g jars	43,9
Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21,05
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Original Frankfurter grüne Soße	12 boxes	13
		Count: 12

Adding Styles

1. Go back to the report template;
2. Select the **Table** component. In this case the **Table2** component;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:

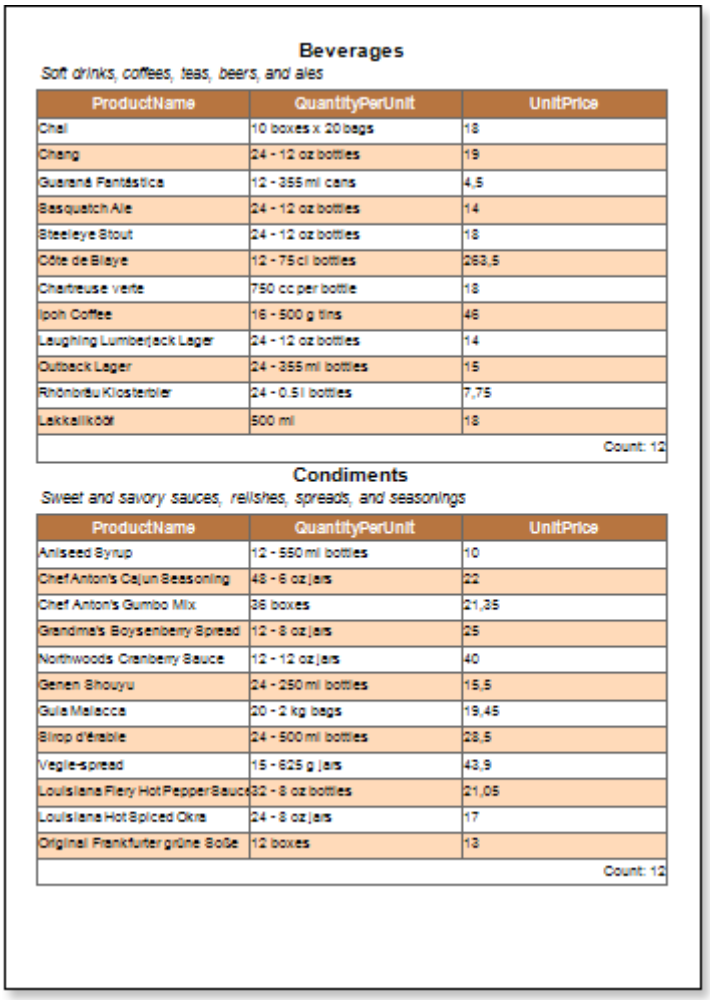


Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.



Beverages

Soft drinks, coffees, teas, beers, and ales

ProductName	QuantityPerUnit	UnitPrice
Chai	10 boxes x 20 bags	18
Chang	24 - 12 oz bottles	19
Guaraná Fantástica	12 - 355 ml cans	4,5
Sasquatch Ale	24 - 12 oz bottles	14
Steeleye Stout	24 - 12 oz bottles	18
Côte de Blaye	12 - 75 cl bottles	263,5
Chartreuse verte	750 cc per bottle	18
Ipoh Coffee	16 - 500 g tins	46
Laughing Lumberjack Lager	24 - 12 oz bottles	14
Outback Lager	24 - 355 ml bottles	15
Rhönbräu Klosterbier	24 - 0.5 l bottles	7,75
Lakkalikööri	500 ml	18
		Count: 12

Condiments

Sweet and savory sauces, relishes, spreads, and seasonings

ProductName	QuantityPerUnit	UnitPrice
Aniseed Syrup	12 - 550 ml bottles	10
Chef Anton's Cajun Seasoning	48 - 6 oz jars	22
Chef Anton's Gumbo Mix	36 boxes	21,35
Grandma's Boysenberry Spread	12 - 8 oz jars	25
Northwoods Cranberry Sauce	12 - 12 oz jars	40
Genen Shouyu	24 - 250 ml bottles	15,5
Gula Malacca	20 - 2 kg bags	19,45
Sirop d'érable	24 - 500 ml bottles	28,5
Vegie-spread	15 - 625 g jars	43,9
Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21,05
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Original Frankfurter grüne Soße	12 boxes	13
		Count: 12

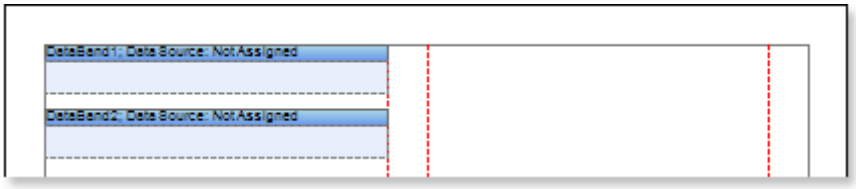
Anchors in Report

A report with anchors is a report in what there is a page of contents and links (called anchors) to other pages in the report. Follow the steps below in order to design a report with the anchors.

Creating a page of contents

1. Run the designer;
2. Connect the data:
    2.1. Create a **New Connection**;
    2.2. Create a **New Data Source**;

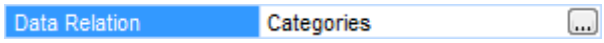
3. Create **Relation** between data sources. If the relation will not be created and/or the **Relation** property of the **Detail** data source will not be filled, then, for **Master** entry, all **Detail** entries will be output;
4. Change the number of columns on a page. For example, set the **Columns** property to **2**, and the **ColumnGaps** property to **1**;
5. Put two **DataBands** on a page of the report template



6. Edit **DataBand1** and **DataBand2**:
   6.1. Align them by height;
   6.2. Change values of required properties. For example, if to set the **PrintIfDetailEmpty** property of the **DataBand1** that is the **Master** component in the **Master-Detail** report to **true**, if it is necessary all **Master** entries be printed in any case, even if **Detail** entries not present. And set the **CanShrink** property of the **DataBand2** that is the **Detail** component in the **Master-Detail** report to **true**, if it is necessary to shrink this band;
   6.3. Change the background color of the **DataBands**;
   6.4. Enable **Borders** of the band, if required;

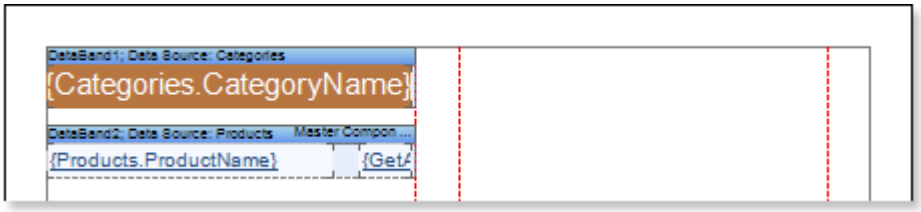
7. Specify the data sources for **DataBands**, as well as assign the **Master** component. In this case, the **Master** component is the upper **DataBand1**, and hence in the **DataSetup** window the lower **DataBand2** on the **Wizard** tab in the **Master Component** should indicate **DataBand1** as a **Master** component. Indicate the data sources for **DataBands** using the **Data Source** property:

8. Fill the **DataRelation** property of the **DataBand2**, which is the **Detail** component:



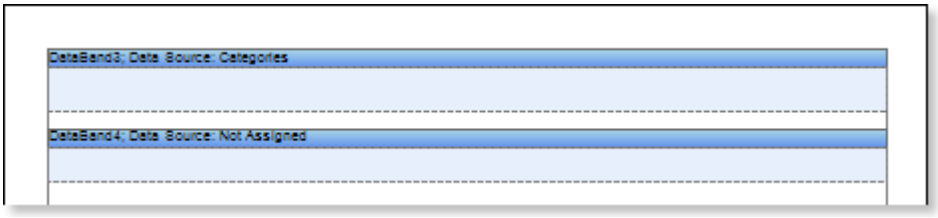
9. Put text components with expressions on **DataBands**. For example: on the **DataBand1**, which is the **Master** component, we put the text component with the following expression: **{Categories.CategoryName}**, and on the **DataBand2**, which is the **Detail** component we put two text components with expressions: **{Products.ProductName}** and **{GetAnchorPageNumber (sender.TagValue)}**;

10. Edit texts and text components of **DataBands**:
   10.1. Drag and drop a text component in the **DataBand**;
   10.2. Set the font settings: size, style, color;
   10.3. Align the text component by height and width;
   10.4. Set the background color of the text component;
   10.5. Align the text in the component;
   10.6. Change the values of the required properties. For example set **WordWrap** property to **true**, if you want the text be wrapped;
   10.7. If necessary, set **Borders** for the text component;
   10.8. Set the border color.
   10.9. Change the value of the **Hyperlink** property for the text component with the **{Products.ProductName}** expression. In this case, set the **Hyperlink** property to the **#{Products.ProductName}** value;
   10.10 Change the value of the **Hyperlink** and **Tag** properties for the text component with the **{GetAnchorPageNumber(sender.TagValue)}**. The **Hyperlink** property should be set to **#{Products.ProductName}**, and the **Tag** property to **{Products.ProductName}**.



Creating a master list

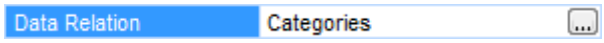
11. Create a second page in the report template;
12. Put two **DataBands** on the page of the report template.



13. Edit **DataBand3** and **DataBand4**:
    13.1. Align the **DataBand** by height;
    13.2. Change the values of the required properties. For example set the **Print if Detail Empty** property of the **DataBand3**, which is the **Master** component in the Master-Detail report to **true**, if you want the Master records be printed in any case, even if the **Detail** entries are not present. Set the **CanShrink** property of the **DataBand4**, which is the **Detail** component in the Master-Detail report to **true**, if it is necessary for this band be shrunk;
    13.3. Set background color of the **DataBand**;
    13.4. If it is necessary, set **Borders** for the **DataBand**;

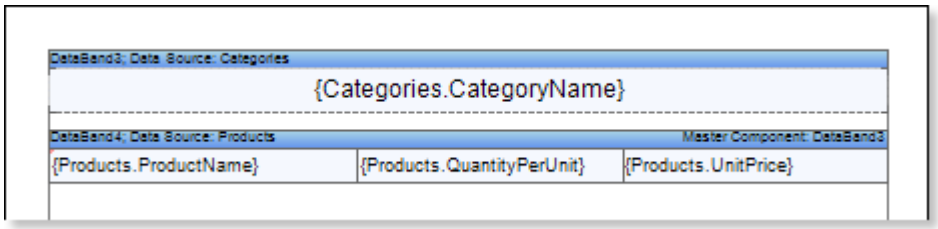
14. Specify the data sources for DataBands, as well as assign the **Master** component. In this case, the **Master** component is the upper **DataBand3**, and hence in the **DataSetup** window the lower **DataBand4** on the **Wizard** tab in the **Master Component** should indicate **DataBand3** as a **Master** component. Indicate the data sources for **DataBands** using the **Data Source** property:

15. Fill the **DataRelation** property of the **DataBand4**, which is the **Detail** component:



16. Put text components with expressions on **DataBands**. For example: on the **DataBand3**, which is the **Master** component, we put the text component with the following expression: **{Categories.CategoryName}**, and on the **DataBand4**, which is the **Detail** component we put two text components with expressions: **{Products.ProductName}, {Products.QuantityPerUnit}**, and **{Products.UnitPrice}**;

17. Edit texts and text components of **DataBands**:
    17.1. Drag and drop a text component in the **DataBand**;
    17.2. Set the font settings: size, style, color;
    17.3. Align the text component by height and width;
    17.4. Set the background color of the text component;
    17.5. Align the text in the component;
    17.6. Change the values of the required properties. For example set **WordWrap** property to **true**, if you want the text be wrapped;
    17.7. If necessary, set **Borders** for the text component;
    17.8. Set the border color.



18. Select the **DataBand**, which is the **Master** data source. In our case, this is the **DataBand3**:
    18.1. Set the **Interaction.Bookmark** property of the **DataBand3** to **{Categories.CategoryName}**;

19. Select the **DataBand**, which is the Detail data source. In our case, this is the **DataBand4**:
    19.1. Set the **Interaction.Bookmark** property to **{Products.ProductName}**;
    19.2. Subscribe to the event. Set the **RenderingEvent** to **{AddAnchor (Products.ProductName);}**;

Report rendering

20. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering a report all references to data fields will be changed on data from specified fields.

Beverages

Chai 3
Chang 3
Guaraná Fantástica 3
Sasquatch Ale 3
Steeleye Stout 3
Côte de Blaye 3
Chartreuse verte 3
Ipoh Coffee 3
Laughing Lumberjack Lager 3
Outback Lager 3
Rhönbräu Klosterbier 3
Lakkalikööri 3

Condiments

Aniseed Syrup 3
Chef Anton's Cajun Seasoning 3
Chef Anton's Gumbo Mix 3
Grandma's Boysenberry Spread 3
Northwoods Cranberry Sauce 3
Genen Shouyu 3
Gula Malacca 3
Sirop d'érable 3
Vegie-spread 3
Louisiana Fiery Hot Pepper Sau 3
Louisiana Hot Spiced Okra 3
Original Frankfurter grüne Soße 3

Confections

Pavlova 3
Teatime Chocolate Biscuits 3
Sir Rodney's Marmalade 3
Sir Rodney's Scones 3
NuNuCa Nuß-Nougat-Creme 3
Gumbär Gummibärchen 3
Schoggi Schokolade 4
Zaanse koeken 4
Chocolade 4
Maxilaku 4
Valkoinen suklaa 4
Tarte au sucre 4
Scottish Longbreads 4

Dairy Products

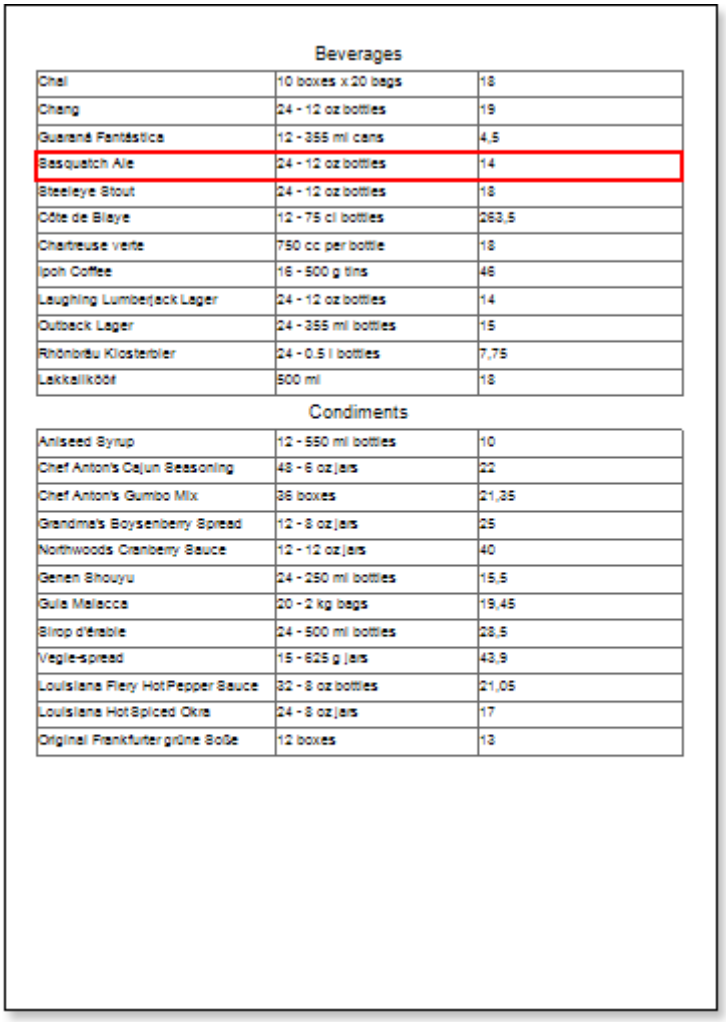
Queso Cabrales 4
Queso Manchego La Pastora 4
Gorgonzola Telino 4
Mascarpone Fabioli 4
Geitost 4
Raclette Courdavault 4
Camembert Pierrot 4
Gudbrandsdalsost 4
Flotemysost 4
Mozzarella di Giovanni 4

Grains/Cereals

Gustaf's Knäckebröd 4
Tunnbröd 4
Singaporean Hokkien Fried Mee 4
Filo Mix 4
Gnocchi di nonna Alice 4
Ravioli Angelo 4
Wimmers gute Semmelknödel 4

Meat/Poultry

Mishi Kobe Niku 5
Alice Mutton 5
Thüringer Rostbratwurst 5
Perth Pasties 5
Tourtière 5
Pâté chinois 5



Beverages

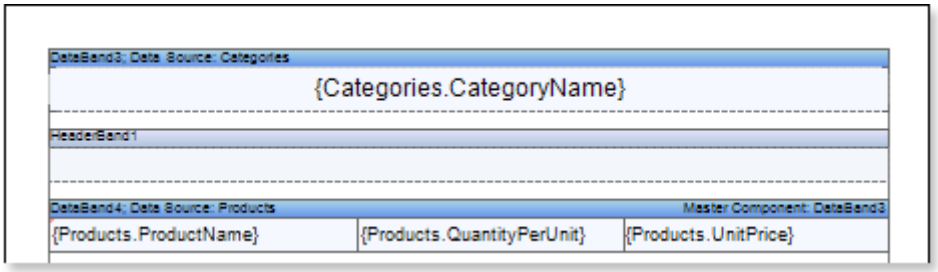
Chai	10 boxes x 20 bags	18
Chang	24 - 12 oz bottles	19
Guaraná Fantástica	12 - 355 ml cans	4,5
Sasquatch Ale	24 - 12 oz bottles	14
Steeleye Stout	24 - 12 oz bottles	18
Côte de Blaye	12 - 75 cl bottles	263,5
Chartreuse verte	750 cc per bottle	18
Ipoh Coffee	16 - 500 g tins	46
Laughing Lumberjack Lager	24 - 12 oz bottles	14
Outback Lager	24 - 355 ml bottles	15
Rhönbräu Klosterbier	24 - 0.5 l bottles	7,75
Lakkalikööri	500 ml	18

Condiments

Aniseed Syrup	12 - 550 ml bottles	10
Chef Anton's Cajun Seasoning	48 - 6 oz jars	22
Chef Anton's Gumbo Mix	36 boxes	21,35
Grandma's Boysenberry Spread	12 - 8 oz jars	25
Northwoods Cranberry Sauce	12 - 12 oz jars	40
Genen Shouyu	24 - 250 ml bottles	15,5
Gula Malacca	20 - 2 kg bags	19,45
Sirop d'érable	24 - 500 ml bottles	28,5
Vegie-spread	15 - 625 g jars	43,9
Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21,05
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Original Frankfurter grüne Soße	12 boxes	13

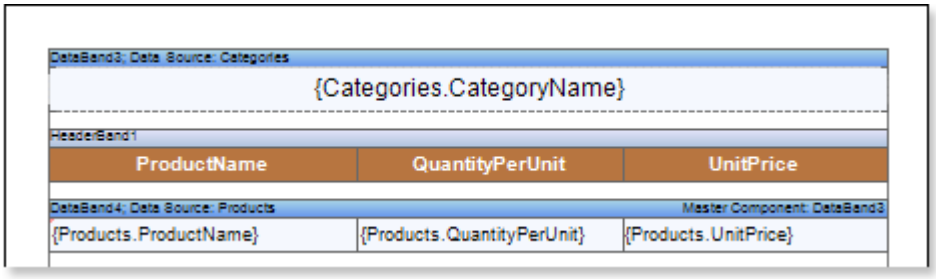
In the rendered report, when clicking an entry in the table of contents the transition to this entry in the report will be done.

21. Go back to the report template;
22. If needed, add other bands to the report template, for example, **HeaderBand**;
23. Edit this band:
    23.1. Align it by height;
    23.2. Change values of properties, if required;
    23.3. Change the background of the band;
    23.4. Set **Borders**, if required;
    23.5. Set the border color.



24. Put text components with expressions in this band. The expression in the text component is a header in the **HeaderBand**.
25. Edit text and text components:
    25.1. Drag and drop the text component in the band;
    25.2. Change font options: size, type, color;
    25.3. Align text component by height and width;
    25.4. Change the background of the text component;

25.5. Align text in the text component;
25.6. Change values of text component properties, if required;
25.7. Enable **Borders** of the text component, if required;
25.8. Set the border color.



26. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report will be the same as the amount of data rows in the database.

Beverages

Chai	3
Chang	3
Guaraná Fantástica	3
Sasquatch Ale	3
Steeleye Stout	3
Côte de Blaye	3
Chartreuse verte	3
Ipoh Coffee	3
Laughing Lumberjack Lager	3
Outback Lager	3
Rhönbräu Klosterbier	3
Lakkalikööri	3

Condiments

Aniseed Syrup	3
Chef Anton's Cajun Seasoning	3
Chef Anton's Gumbo Mix	3
Grandma's Boysenberry Spread	3
Northwoods Cranberry Sauce	3
Genen Shouyu	3
Gula Malacca	3
Sirop d'érable	3
Vegie-spread	3
Louisiana Fiery Hot Pepper Sau	3
Louisiana Hot Spiced Okra	3
Original Frankfurter grüne Soße	3

Confections

Pavlova	3
Teatime Chocolate Biscuits	3
Sir Rodney's Marmalade	3
Sir Rodney's Scones	4
NuNuCa Nuß-Nougat-Creme	4
Gumbär Gummibärchen	4
Schoggi Schokolade	4
Zaanse koeken	4
Chocolade	4
Maxilaku	4
Valkoinen suklaa	4
Tarte au sucre	4
Scottish Longbreads	4

Dairy Products

Queso Cabrales	4
Queso Manchego La Pastora	4
Gorgonzola Telino	4
Mascarpone Fabioli	4
Geitost	4
Raclette Courdavault	4
Camembert Pierrot	4
Gudbrandsdalsost	4
Flotemysost	4
Mozzarella di Giovanni	4

Grains/Cereals

Gustaf's Knäckebröd	4
Tunnbröd	4
Singaporean Hokkien Fried Mee	4
Filo Mix	4
Gnocchi di nonna Alice	5
Ravioli Angelo	5
Wimmers gute Semmelknödel	5

Meat/Poultry

Mishi Kobe Niku	5
Alice Mutton	5
Thüringer Rostbratwurst	5
Perth Pasties	5
Tourtière	5
Pâté chinois	5

Beverages

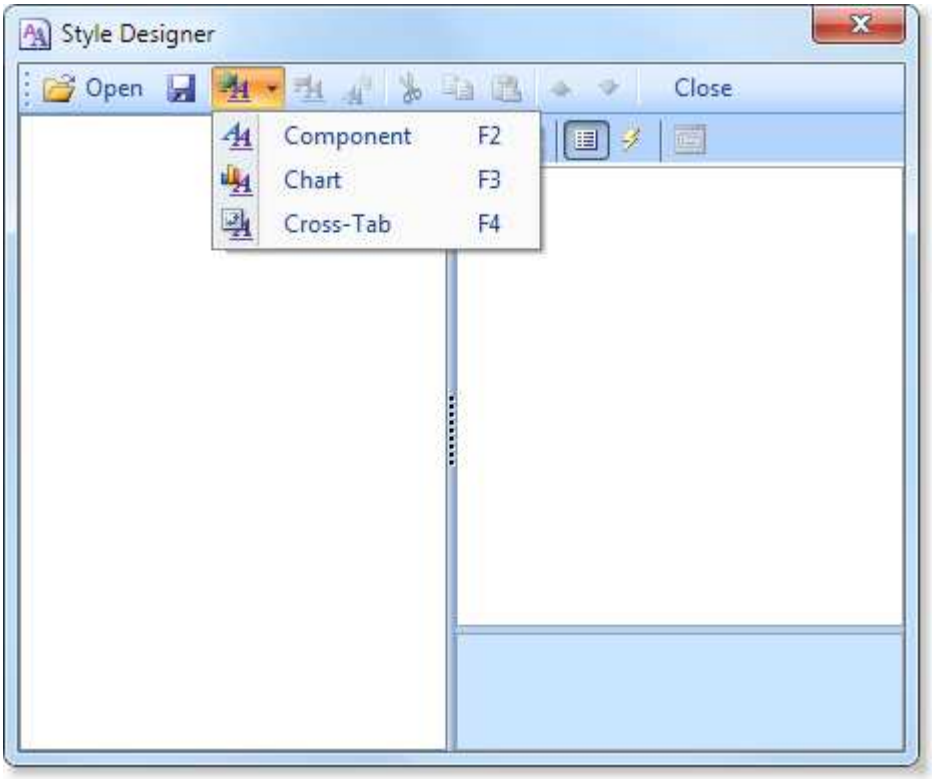
ProductName	QuantityPerUnit	UnitPrice
Chai	10 boxes x 20 bags	18
Chang	24 - 12 oz bottles	19
Guaraná Fantástica	12 - 355 ml cans	4,5
Sasquatch Ale	24 - 12 oz bottles	14
Steeleye Stout	24 - 12 oz bottles	18
Côte de Blaye	12 - 75 cl bottles	263,5
Chartreuse verte	750 cc per bottle	18
Ipoh Coffee	16 - 500 g tins	46
Laughing Lumberjack Lager	24 - 12 oz bottles	14
Outback Lager	24 - 355 ml bottles	15
Rhönbräu Klosterbier	24 - 0.5 l bottles	7,75
Lakkalikööri	500 ml	18

Condiments

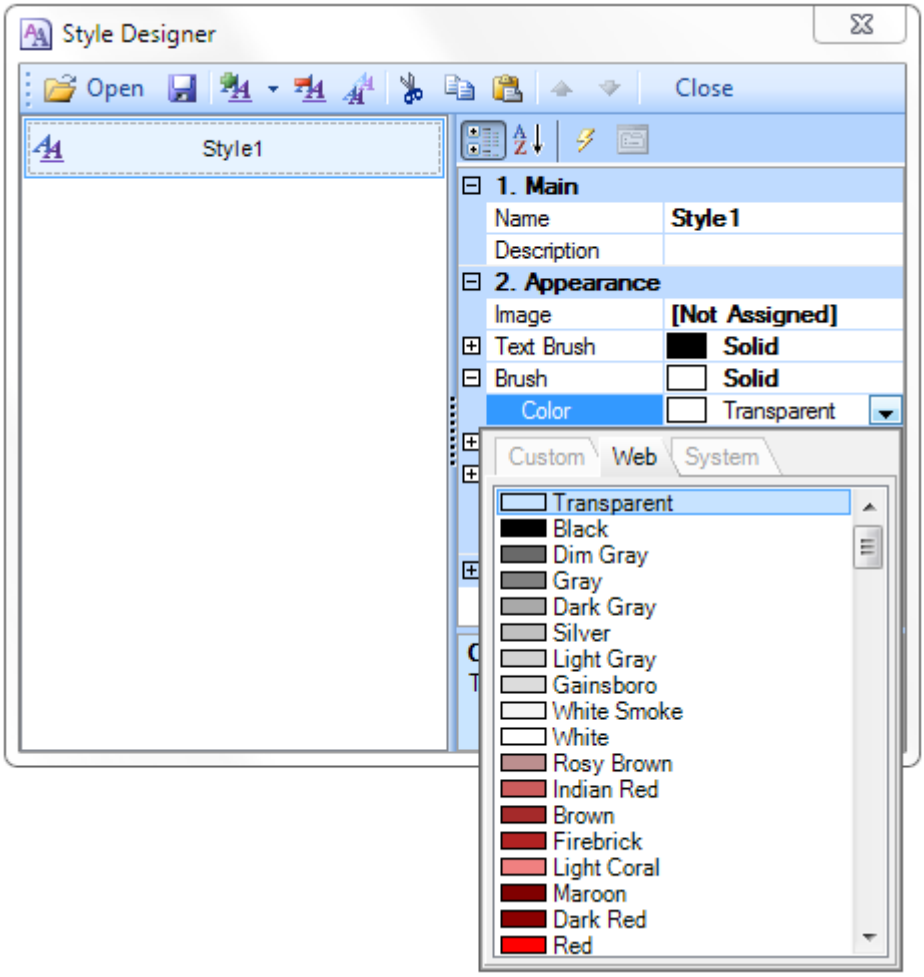
ProductName	QuantityPerUnit	UnitPrice
Aniseed Syrup	12 - 550 ml bottles	10
Chef Anton's Cajun Seasoning	48 - 6 oz jars	22
Chef Anton's Gumbo Mix	36 boxes	21,35
Grandma's Boysenberry Spread	12 - 8 oz jars	25
Northwoods Cranberry Sauce	12 - 12 oz jars	40
Genen Shouyu	24 - 250 ml bottles	15,5
Gula Malacca	20 - 2 kg bags	19,45
Sirop d'érable	24 - 500 ml bottles	28,5
Vegie-spread	15 - 625 g jars	43,9
Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21,05
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Original Frankfurter grüne Soße	12 boxes	13

Adding Styles

1. Go back to the report template;
2. Select the **DataBand**. In our case, select the **DataBand4**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**:



Click the **Add Style** button to start creating a style. Select **Component** from the drop down list. Set the **Brush.Color** property to change the background color of a row. The picture below shows a sample of the **Style Designer** with the list of values of the **Brush.Color** property:



Click **Close**. Then a new value in the list of **Even style** and **Odd style** properties (a style of a list of odd and even rows) will appear.

4. To render the report, click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item.

Beverages

Chai	3
Chang	3
Guaraná Fantástica	3
Sasquatch Ale	3
Steeleye Stout	3
Côte de Blaye	3
Chartreuse verte	3
Ipoh Coffee	3
Laughing Lumberjack Lager	3
Outback Lager	3
Rhönbräu Klosterbier	3
Lakkalikööri	3

Condiments

Aniseed Syrup	3
Chef Anton's Cajun Seasoning	3
Chef Anton's Gumbo Mix	3
Grandma's Boysenberry Spread	3
Northwoods Cranberry Sauce	3
Genen Shouyu	3
Gula Malacca	3
Sirop d'érable	3
Vegie-spread	3
Louisiana Fiery Hot Pepper Sau	3
Louisiana Hot Spiced Okra	3
Original Frankfurter grüne Soße	3

Confections

Pavlova	3
Teatime Chocolate Biscuits	3
Sir Rodney's Marmalade	3
Sir Rodney's Scones	4
NuNuCa Nuß-Nougat-Creme	4
Gumbär Gummibärchen	4
Schoggi Schokolade	4
Zaanse koeken	4
Chocolade	4
Maxilaku	4
Valkoinen suklaa	4
Tarte au sucre	4
Scottish Longbreads	4

Dairy Products

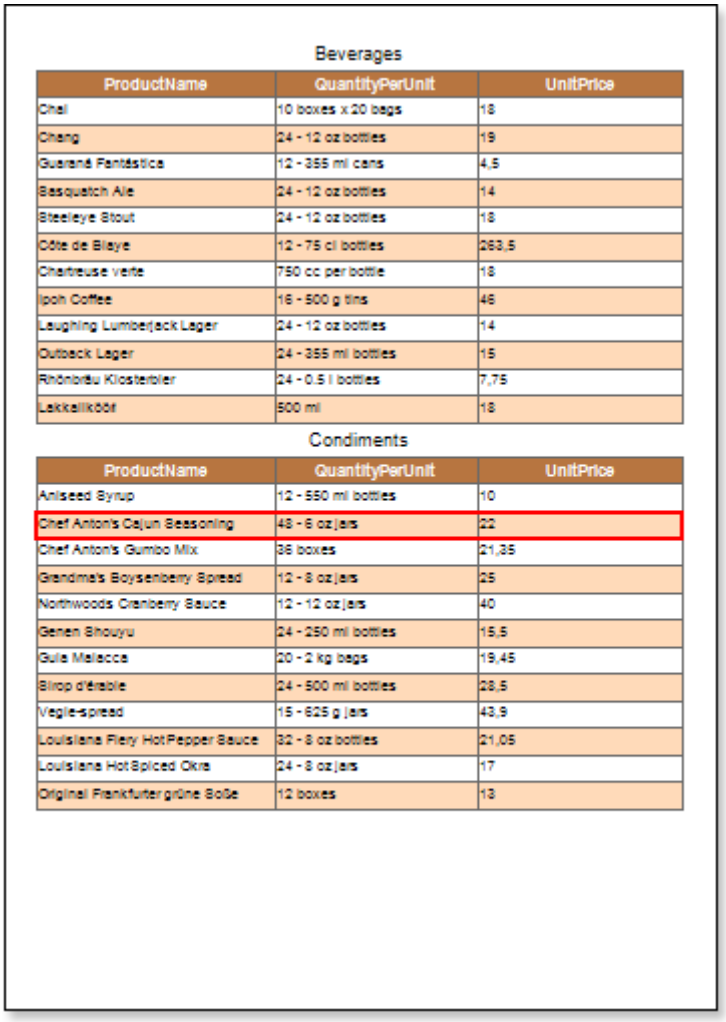
Queso Cabrales	4
Queso Manchego La Pastora	4
Gorgonzola Telino	4
Mascarpone Fabioli	4
Geitost	4
Raclette Courdavault	4
Camembert Pierrot	4
Gudbrandsdalsost	4
Flotemysost	4
Mozzarella di Giovanni	4

Grains/Cereals

Gustaf's Knäckebröd	4
Tunnbröd	4
Singaporean Hokkien Fried Mee	4
Filo Mix	4
Gnocchi di nonna Alice	5
Ravioli Angelo	5
Wimmers gute Semmelknödel	5

Meat/Poultry

Mishi Kobe Niku	5
Alice Mutton	5
Thüringer Rostbratwurst	5
Perth Pasties	5
Tourtière	5
Pâté chinois	5



Beverages

ProductName	QuantityPerUnit	UnitPrice
Chai	10 boxes x 20 bags	18
Chang	24 - 12 oz bottles	19
Guaraná Fantástica	12 - 355 ml cans	4,5
Sasquatch Ale	24 - 12 oz bottles	14
Steeleye Stout	24 - 12 oz bottles	18
Côte de Blaye	12 - 75 cl bottles	263,5
Chartreuse verte	750 cc per bottle	18
Ipoh Coffee	16 - 500 g tins	46
Laughing Lumberjack Lager	24 - 12 oz bottles	14
Outback Lager	24 - 355 ml bottles	15
Rhönbräu Klosterbier	24 - 0.5 l bottles	7,75
Lakkalikööri	500 ml	18

Condiments

ProductName	QuantityPerUnit	UnitPrice
Aniseed Syrup	12 - 550 ml bottles	10
Chef Anton's Cajun Seasoning	48 - 6 oz jars	22
Chef Anton's Gumbo Mix	36 boxes	21,35
Grandma's Boysenberry Spread	12 - 8 oz jars	25
Northwoods Cranberry Sauce	12 - 12 oz jars	40
Genen Shouyu	24 - 250 ml bottles	15,5
Gula Malacca	20 - 2 kg bags	19,45
Sirop d'érable	24 - 500 ml bottles	28,5
Vegie-spread	15 - 625 g jars	43,9
Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21,05
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Original Frankfurter grüne Soße	12 boxes	13

Report Designer

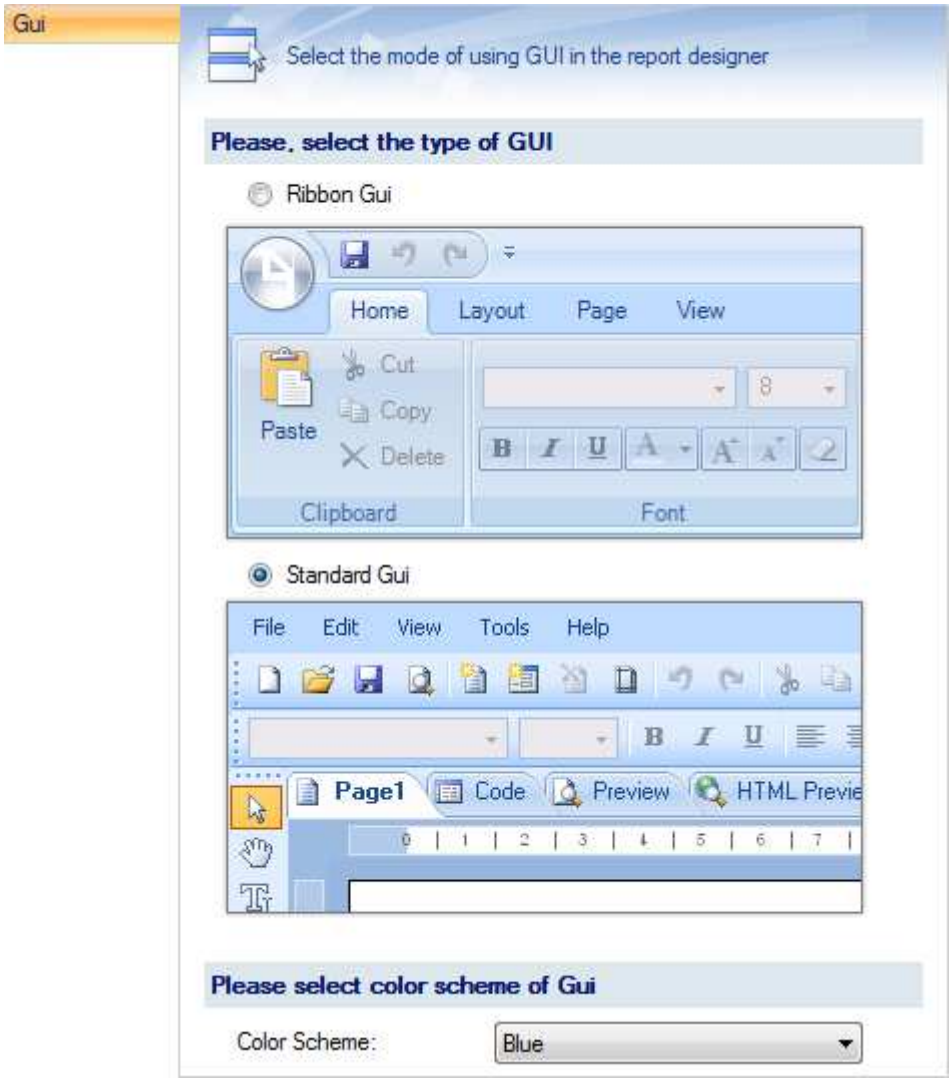
In this topic the basic knowledge of the report designer are represented. Basic elements, keyboard shortcuts and settings from code are described.

Standard UI and Ribbon UI

Different report designers support different UI's. The Standard mode uses toolbars as in Microsoft Office 2003. The Ribbon mode uses "ribbon tabs". Compatibility of designers and UI types is shown in the table below:

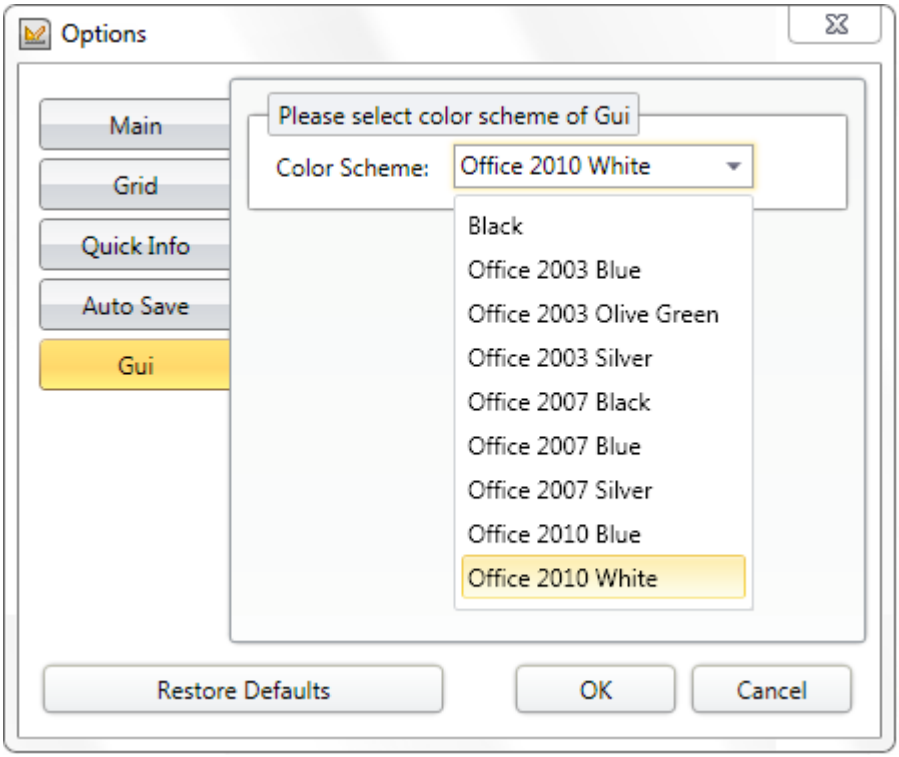
UI type	Reports.NET	Reports.Wpf	Reports.Web	Reports.Silverlight
Standard (Microsoft Office 2003)	+	+		
Ribbon (Microsoft Office 2007)	+	+	+	
Ribbon (Microsoft Office 2010)		+		+

For **Stimulsoft Reports.Net** the UI type is offered to choose when first running the report designer. Also the UI type can be chosen in the **GUI** tab of the **Options** window of the report designer. The picture below shows the **GUI** tab of the **Options** window of the report designer:



As seen on the picture, it is possible to choose either **Standard** or **Ribbon** UI. The color scheme of UI can also be changed. The following types of color themes are available: **Blue**, **Silver**, **Black**. The full path for invoking the **Options** window in the **Ribbon** UI: **Main Menu** -> **Options** button -> **Options** window. The full path for invoking the **Options** window in the **Standard** UI: **Tools** menu -> **Options** -> **Options** window. If you need to change the UI type before running the designer, then hold on the "**Ctrl**" button when loading the designer. The **Select the type of GUI** window will be invoked. The picture below shows **Select the type of GUI** window:

When **Stimulsoft Reports.Wpf** is run first the user will not see the **Select the type of GUI**. The designer will be loaded with the default **Ribbon 2010** UI **White** theme. To change the type of UI, it is necessary to call the **Options** window in the **Gui** tab and make changes. The picture below shows the **Gui** tab of the **Options** window with the color themes drop down list:



As seen on the picture, in **Stimulsoft Reports.Wpf**, it is impossible to separately change the UI type and the color theme. A type of UI corresponds to a definite color theme. The full path for invoking the **Options** window in **Ribbon 2010** UI: **Main Menu** - **Options** - **Options** window. The full path for invoking the **Options** window in **Ribbon 2007** UI: **Main Menu** - **Options** button - **Options** window. The full path for invoking the **Options** window in **Standard** UI: **View** - **Options** - **Options** window.

Stimulsoft Report.Web supports **Ribbon 2007** UI **Blue** theme.

Stimulsoft Report.Silverlight supports **Ribbon 2010** UI **Blue** theme.

Standard UI

Standard Interface - a type of interface used in **Microsoft word 2003** and consists of a main menu and the set of toolbars.

Toolbar

Toolbar - is an element of a graphical user interface, intended for placing on it a few controls. Usually consists of a horizontal or vertical rectangle, which are grouped frequently used items.

Standard Toolbar

The basic panel of the reports designer. The main element to control a report in the reports designer are placed on this panel. Besides, controls of basic parameters of the reports designer are placed on this panel.

1 Create a new report in the reports designer.
2 Open recently saved report.
3 Save a report.
4 Run the preview of a report in the window.
5 Add a new page to a report.
6 Add a new dialog form to the report.
7 Delete the current page or a dialog form in the report. If there is only one page of a report or a dialog form then this menu item is not active..
8 Call the Page Setup window of page settings. If the dialog form is the current then this menu item is not active.
9 Undo the recent action.
10 Redo the recent action.

Cut the selected components from the current page to the Clipboard.

Copy the selected components on the current page to the Clipboard.

Paste components from the Clipboard on the current page of a report.

Delete selected components on the current page.

Select all components on the current page.

Control showing a grid on a page.

Control automatic alignment of components by the grid.

Show/hide headers of bands.

Show/hide the order of placing components on a page.

Control Quick Info.

Show/hide rulers on a page.

Control zoom of a page in the reports designer.

Close the reports designer.

Borders Toolbar

The **Borders** panel is used to control borders of components.

1 Set borders from all sides of a component.
2 Remove borders from all sides of a component.
3 Set the border from the top side of a component.
4 Set the border from the left side of a component.

5 Set the border from the right side of a component.
6 Set the border from the bottom side of a component.
7 Set the shadow of a component.
8 Background color of a component.
9 Border color of a component.
10 The type of the border line of a component.

Designer Toolbar

The **Design** toolbar is used to place components.

1 Align selected components to the grid of a page.
2 Align all selected components to their common left margin.
3 Align horizontally all selected components to their common center.
4 Align all selected components to their common right margin.
5 Align all selected components to their common top margin.
6 Align vertically all selected components to their common center.
7 Align all selected components to their common bottom margin.
8 Make the same size of components as the size of the first selected component.
9 Make the same width of components as the size of the first selected component.
10 Make the same height of components as the size of the first selected component.

Make horizontal spacing of selected components equal by their width.

Make vertical spacing of selected components equal by their height.

Center all selected components horizontally.

Center all selected components vertically.

Bring selected components to Front.

Send selected components to Back.

Move selected components on one level forward.

Move selected components on one level backward.

Control the Lock property.

Control the Link property.

Formatting Toolbar

The **Formatting** toolbar is used to output text with specified font type, color etc..

1. Select the font type of the selected components on the current page.
2. Select font size of the selected text components on the current page.
3. Set the Bold font style.
4. Set the Italic font style.
5. Set the Underlined font style.
6. Align left the content of a component.
7. Align center horizontally the content of a component.
8. Align right the content of a component.
9. Justify the content of a text component.
10. Align top the content of a component.

Align center vertically the content of a component.

Align bottom the content of a component.

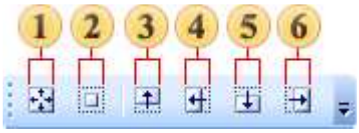
The angle of the text rotation. This command can be applied only to the text component.

Select font color for selected components on the current page.

Open the Conditions Editor window for selected components.

Dock Style Toolbar

The **Dock style** toolbar is used to dock selected components.



1. Docks selected components to all edges.
2. Docks selected components to the left edge.
3. Docks selected components to the right edge.
4. Removes docking of selected components.
5. Docks selected components to the top edge.
6. Docks selected components to the bottom edge.

Styles Toolbar

The **Styles** toolbar is used to control styles in a report.

1. Opens the styles editor window.
2. A list of styles available in a report.

Text Format Toolbar

The **Text format editor** is used to quick text formatting.

1 Runs the text format editor.
2 Sets the text format to **Standard**.
3 Sets the text format to **Numerical**.
4 Sets the text format to **Currency**.
5 Sets the text format to **Date**.
6 Sets the text format to **Time**.
7 Sets the text format to **Percent**.
8 Sets the text format to **Boolean**.
9 Sets the text format to **Custom**.

Service Toolbar

The **Service** toolbar is used to call service functions.

1 Runs the **Data Store**.
2 Runs the **Page Manager**.
3 Runs the **Service Configurator**.

Status Bar in Standard UI

Status bar is placed under the designer window. The picture below shows a status bar of the **Standard** UI:

The bar contains 3 sections:
1 Units. This field shows current units in a report. It is possible to change them.
2 The field shows the currently selected component.
3 Shows cursor coordinates on a page of a report template. (X:0,0 ; Y:0,0) coordinates corresponds to the top left corner of a page of a report template.

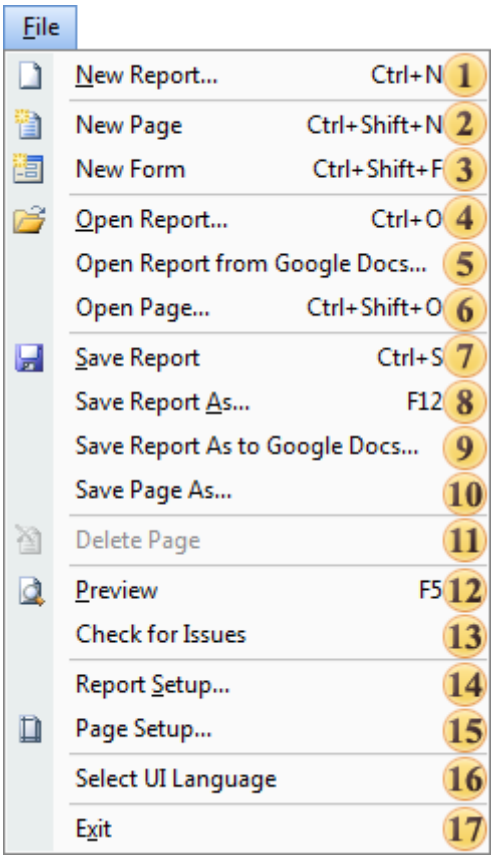
Main Menu

The main menu of the Designer contains 5 submenus: **File, Edit, View, Tools, Help**. Each submenu contains common commands.

The **File** submenu contains the global commands for working with a report. The **Edit** submenu contains commands to work with Clipboard and stack of operation. The **View** submenu contains parameters of showing report in the Reports Designer. The **Tools** submenu contains commands of calling the service functions. The **Help** submenu contains information of the Reports Designer and web links on report generator resources in Web.

File Menu

The **File** submenu contains the global commands for working with a report.



1 Create a new report in the report designer.
2 Add a new page to the report.
3 Add a new dialog page to the report.
4 Open the recently saved report.
5 Открывает ранее сохраненный отчет из сервиса **Google Docs**.
5 Open the recently saved page of a report. The page will be added to the report after the last page of a report.
6 Save a report. If the report was not saved earlier then the **Save as...** menu item will be called.
7 Save a report to the specified folder.
8 Save the current page to the specified folder.
9 Сохраняет отчет с использованием сервиса **Google Docs** в указанную коллекцию.
9 Delete the current page or a dialog form in the report. If there is only one page of a report or a dialog form then this menu item is not active.

10 Run the preview of a report in the window.

Call the Report Setup window of report options.

Call the Page Setup window of page settings. If the dialog form is the current then this menu item is not active.

Вызывает **Инспектор отчета (Report Checker)**.

Select UI localization language.

Close the Reports Designer.

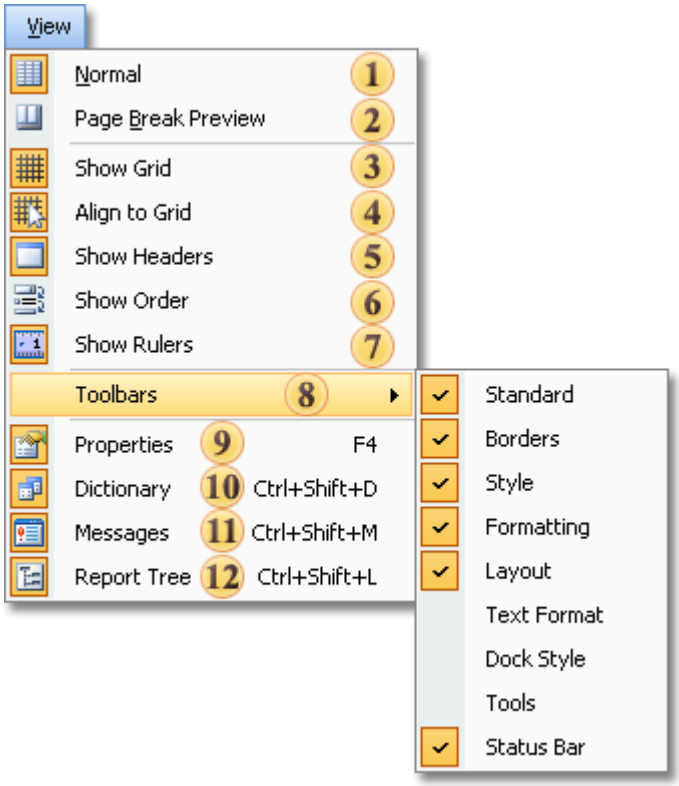
Edit Menu

The **Edit** submenu contains commands to work with Clipboard and stack of operation.

1 Undo the recent action.
2 Redo the recent action.
3 Cut the selected components from the current page to the Clipboard.
4 Copy the selected components on the current page to the Clipboard.
5 Paste components from the Clipboard on the current page of a report.
6 Delete selected components on the current page.
7 Select all components on the current page.
8 Clear the content of all selected components.

View Menu

This menu contains controls of showing report pages in the reports designer, controls for panels, and controls for the toolbars of the reports designer.



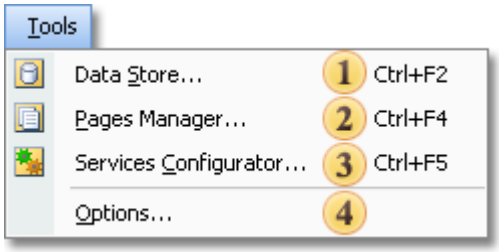
1 Standard mode of showing a page.
2 Page break preview (used only for segmented pages).
3 Control showing grid on a page.
4 Control automatic alignment of components by the grid.
5 Show/hide headers of bands.
6 Show/hide the order of placing components on a page.
7 Show/hide rulers on a page.
8 Control visibility of toolbars.
9 Control visibility of the **Properties** panel.
10 Control visibility of the **Dictionary** panel.

Control visibility of the **Messages** panel.

Control visibility of the **Report Tree** panel.

Tools Menu

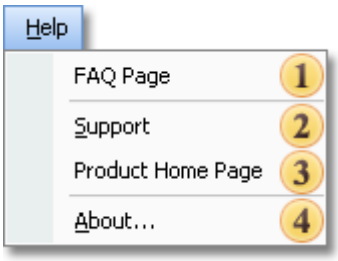
The **Service** submenu contains items to call service functions.



1 Runs the **Data Store**.
2 Runs the **Pages Manager**.
3 Runs the **Services Configurator**.

Help Menu

The **Help** submenu contains information of the Reports Designer and web links on report generator resources in Web.



1. Runs in the Web browser the FAQ Page at Stimulsoft official site.
2. Runs in the Web browser the technical support web page.
3. Runs the product web page in the Web browser.
4. Shows the About box of the report designer.

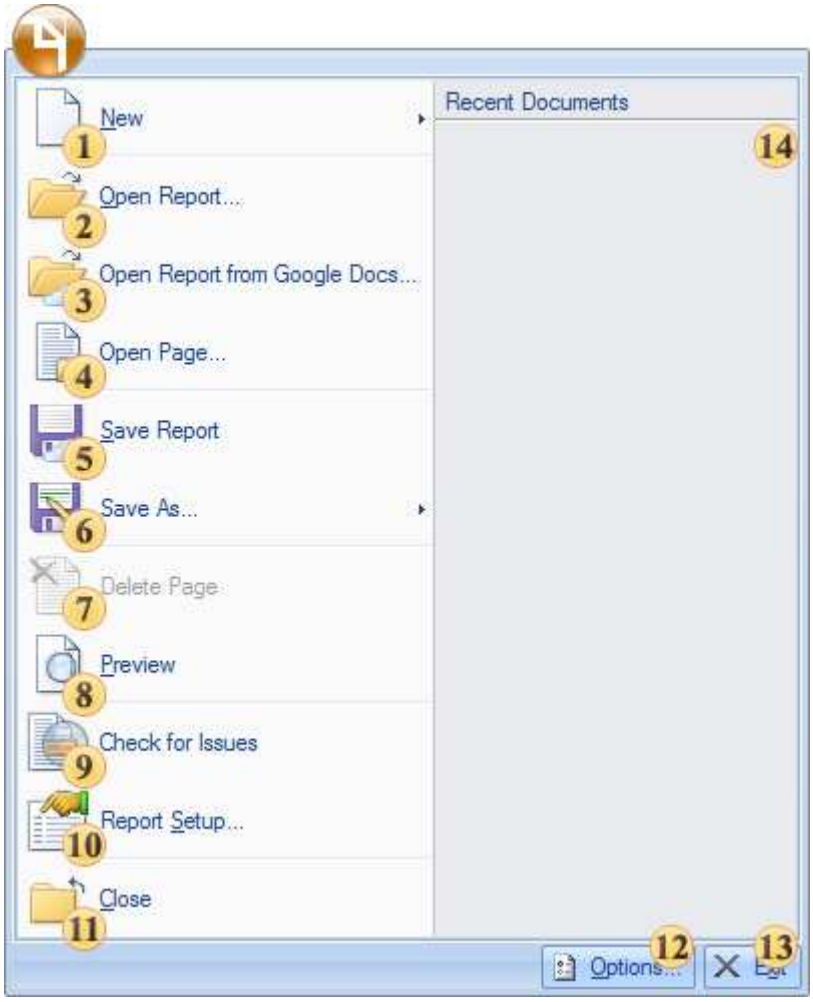
Ribbon UI Mode 2007

Ribbon interface 2007, based on the tabs (similar to **Microsoft Office 2007**). The tabs are grouped instruments, thereby reducing the number of toolbars. Consider this type of interface in more detail.

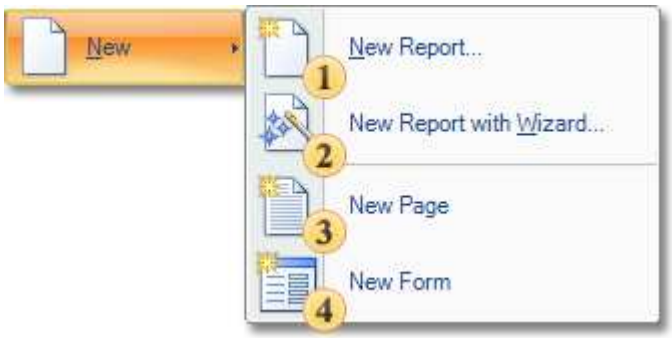
Application Menu

The basis of the **Ribbon UI** mode is the **Ribbon** panel that is shown in the picture below:

The main element of the **Ribbon UI** is the **Application Button** and menu that is called by clicking on this button. This is a main menu of the application. The is represented by basic commands used to work with reports in the reports designer. Elements of the application menu are shown on the picture below.



1 Create a new report or a new control in a report. It is possible to create a new report, a new page, a new form. The submenu of the New menu is shown in the picture below:



The first item of this submenu 1 creates an empty report in the report designer. The second item 2 creates a new report using the wizard. The third item 3 creates a new page. And the fourth item 4 creates a new dialog form.

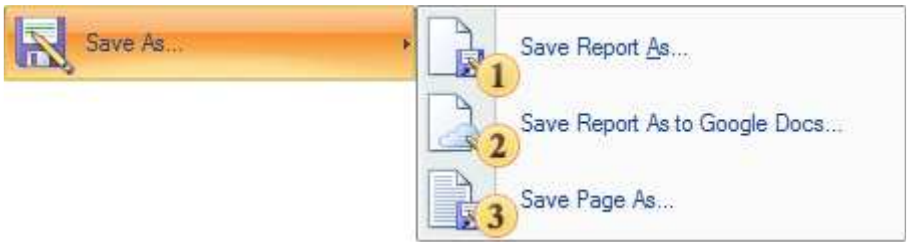
2 Open recently saved report.

3 Открытие ранее сохраненного отчета из docs.google.com.

4 Open the recently saved page of a report. The page will be added to the report after the last page of a report.

5 Save a report. If the report was not saved earlier then the **Save as...** menu item will be called.

6 Save a report to the specified folder. This item contains submenu:



The first item saves a report to the specified folder. The first item saves a page of a report to the specified folder. Второй - сохранит отчет с использованием сервиса Google Docs в указанной коллекции. Третий - сохранит страницу в указанном пользователем месте.

7 Delete the current page or a dialog form in the report. If there is only one page of a report or a dialog form then this menu item is not active.

8 Run the preview of a report in the window.

9 Вызывает **Инспектор отчета (Report Checker)**.

10 Call the Report Setup window of report options.

Close a report that is opened in the reports designer.

Call the report designer setup window.

Close the report designer.

This panel shows a list of recently opened reports. If to select one report in the list then it will be opened in the designer.

Ribbon Tabs

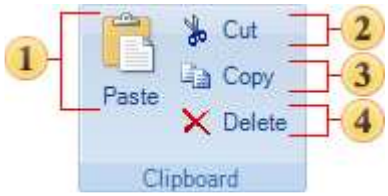
Tab is a part of the interface on which the toolbar. At the designer shows four tabs: **Home**, **Page**, **Layout**, **View**. Consider these tabs and the main instruments are located on them more.

Home Tab

This is a basic tab of the report designer. Main commands of setting report components are placed on this tab.

Clipboard Group

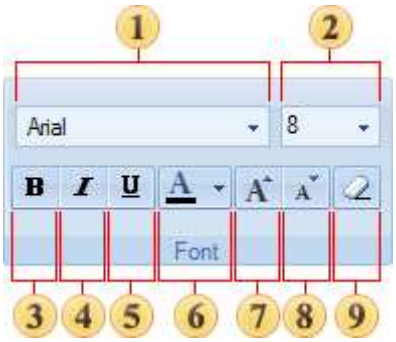
This group allows working with the Clipboard of the report designer.



1 Paste components from the Clipboard on the current page of a report.
2 Cut the selected components from the current page to the Clipboard.
3 Copy the selected components on the current page to the Clipboard.
4 Delete selected components on the current page.

Font Group

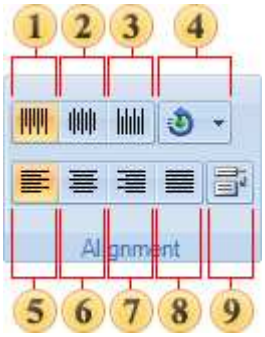
This group is used to output text with specified font type, color etc.



1. Select the font type of the text components on the current page.
2. Select font size of the text components on the current page.
3. Set the Bold font style.
4. Set the Italic font style.
5. Set the Underlined font style.
6. Set the font color of the text components on the current page.
7. Increase the font size.
8. Decrease the font size.
9. Delete the content of all selected text components.

Alignment Group

The group is used to align the content of components horizontally and vertically. Also it is possible to set the angle of the text rotation and control the **WordWrap** property.

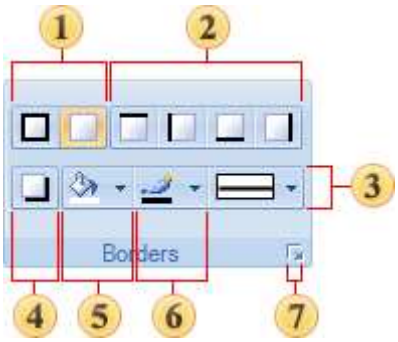


All commands are applied to selected components on the current page.

1. Align top the content of a component.
2. Align center vertically the content of a component.
3. Align bottom the content of a component.
4. The angle of the text rotation. This command can be applied only to the text component.
5. Align left the content of a component.
6. Align center horizontally the content of a component.
7. Align right the content of a component.
8. Justify the content of a text component.
9. Used for the WordWrap property of the text component.

Borders Group

This group contains the commands to setup border components.

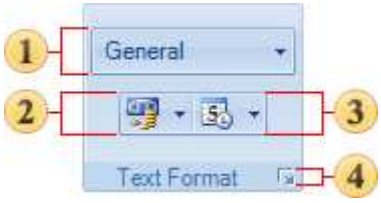


All commands can be applied to selected components on the current page.

1. Set or remove borders from all sides of a component.
2. Set or remove borders from each side of a component.
3. The type of the border line.
4. Set the shadow of a component.
5. Background color of a component.
6. Border color of a component.
7. Call the form of changing border parameters.

Text Format Group

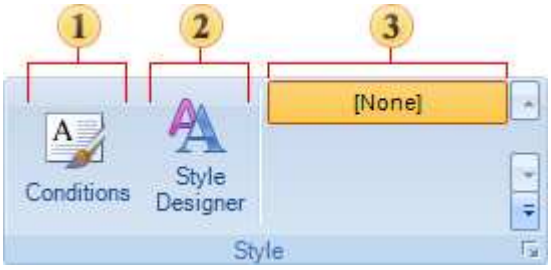
The group to control text formatting.



1. Select text format.
2. Select symbols of currency.
3. Set one of the predefined formats to output a text.
4. Call a form of formats editing.

Styles Group

This group is used to control styles and conditions which are used to automatically design components in a report.



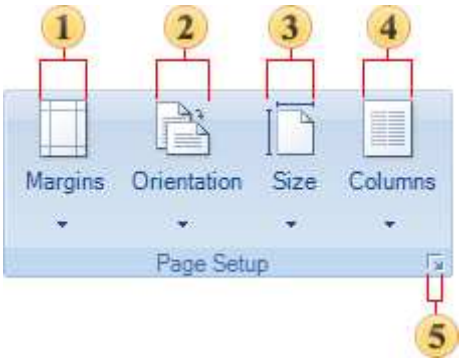
1. Open a window of the Conditions Editor for selected components.
2. Open a window of Styles Editor.
3. A gallery of styles in a report.

Page Tab

This tab is used to control page parameters.

Page Setup Group

This group contains elements to control basic parameters of a page. These are page margins, orientation, page size, columns.

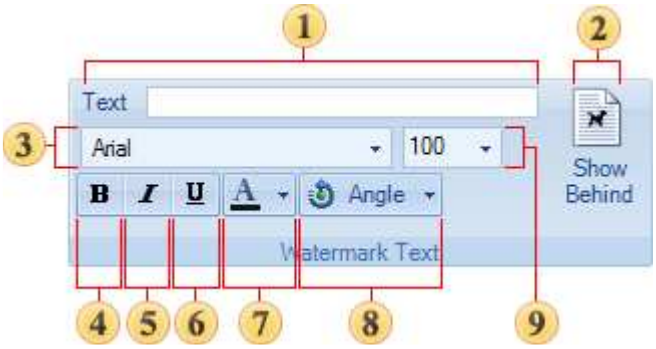


Changes can be applied to the current page in the Reports Designer.

1. Select sizes of page margins.
2. Select Portrait or Landscape orientation of a page.
3. Select page size.
4. Select number of columns on a page.
5. Call the Page Setup dialog window.

Watermark Text Group

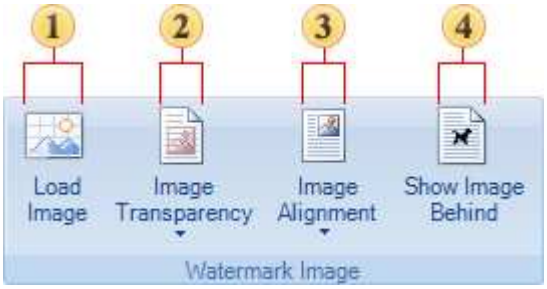
This group is used to control watermark text output.



1. Watermark text.
2. Show watermark text behind the components of a page.
3. Select a font type to output a text of the watermark.
4. Set the Bold font style of the watermark text.
5. Set the Italic font style of the watermark text.
6. Set the Underlined font style of the watermark text.
7. Watermark text color.
8. Set angle rotation of the watermark text.
9. Font size of the watermark text.

Watermark Image Group

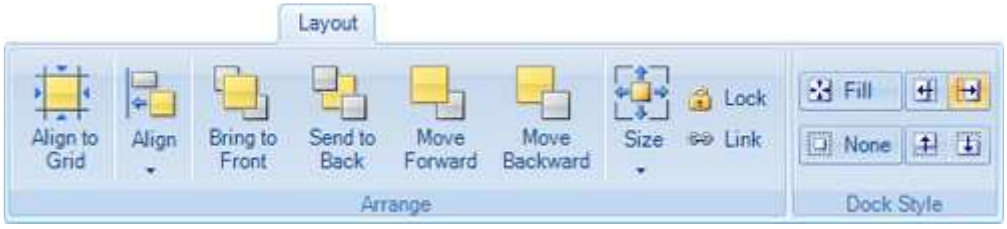
This group is used to control the watermark image.



1. Load an image for the watermark.
2. Set transparency of the watermark image.
3. Align the watermark image.
4. Show the watermark image behind page components on page.

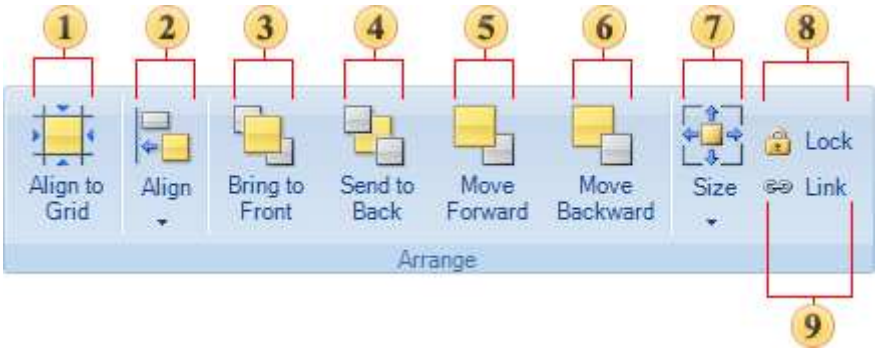
Layout Tab

This tab is used to control placing different components on a page and also to specify styles of docking components.



Arrange Group

The group contains a lot of commands to change position of components on a page. The picture below shows this group.



1. Align all selected components to the page grid.
2. Align selected components. This element contains submenu and short description in this topic below.
3. Bring selected components to Front.
4. Send selected components to Back.
5. Move selected components on one level forward.
6. Move selected components on one level backward.
7. Choose the size of selected components. It contains submenu and is described in this topic below.
8. Control the Lock property.
9. Control the Link property.

The description of the Align button, specified with number 2 on the picture above.

1. Align all selected components to their common left margin.
2. Align horizontally all selected components to their common center.
3. Align all selected components to their common right margin.
4. Align all selected components to their common top margin.
5. Align vertically all selected components to their common center.
6. Align all selected components to their common bottom margin.
7. Make horizontal spacing of selected components equal by their width.

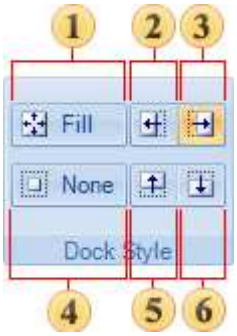
8 Make vertical spacing of selected components equal by their height.
9 Center all selected components horizontally.
10 Center all selected components vertically.

The description of the Size button, specified with number 7 on the topmost picture.

1 Make the same size of components as the size of the first selected component.
2 Make the same width of components as the size of the first selected component.
3 Make the same height of components as the size of the first selected component.

Dock Style Group

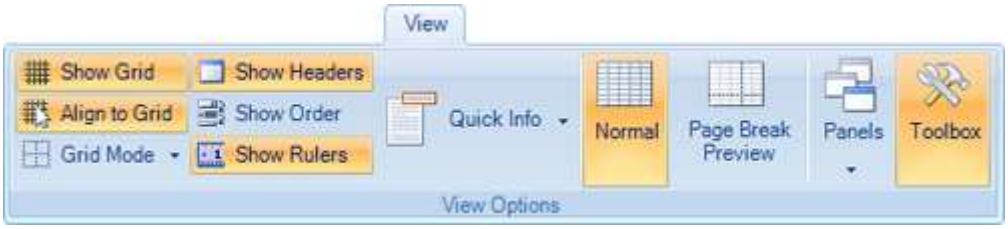
This group contains elements to set the dock style of selected components.



1 Dock selected components to all edges.
2 Dock selected components to the left edge.
3 Dock selected components to the right edge.
4 Removes dock style of selected components.
5 Dock selected components to the top edge.
6 Dock selected components to the bottom edge.

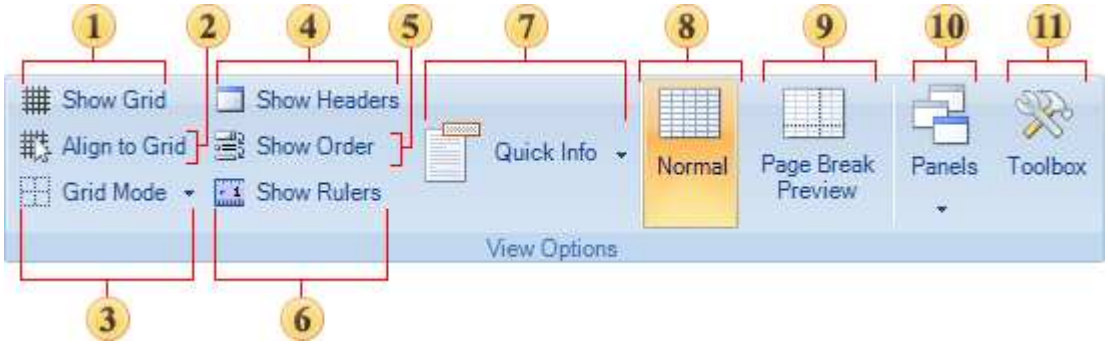
View Tab

This tab is used to control different options of working with report designer.



View Options Group

This group contains settings for different parameters of showing a report and working with the reports designer.



1 Control showing grid on a page.
2 Control automatic alignment of components by the grid.
3 Select grid type.
4 Show/hide headers of bands.
5 Show/hide the order of placing components on a page.
6 Show/hide rulers on a page.
7 Control Quick Info.
8 Standard mode of showing a page.
9 Page break preview (used only for segmented pages).
10 Control visible designer panels.

Show/hide Toolbox.

2007 Status Bar

Status bar in 2007 **Ribbon** interface is located at the bottom of the designer. This is a panel on what controls are placed. The picture below shows a status bar Ribbon Interface:

This panel contains 4 fields:

1 Units. This field displays the units used in a report, and also allows you to change them.
2 This field displays the currently selected components. In this case, select a page "**Page 1**".
3 This field displays the coordinates of the cursor on the page template report. Moreover, the origin (X: 0,0 and Y: 0,0) coincides with the upper left corner of border crossing at the report template.
4 Control zoom of a report template page.

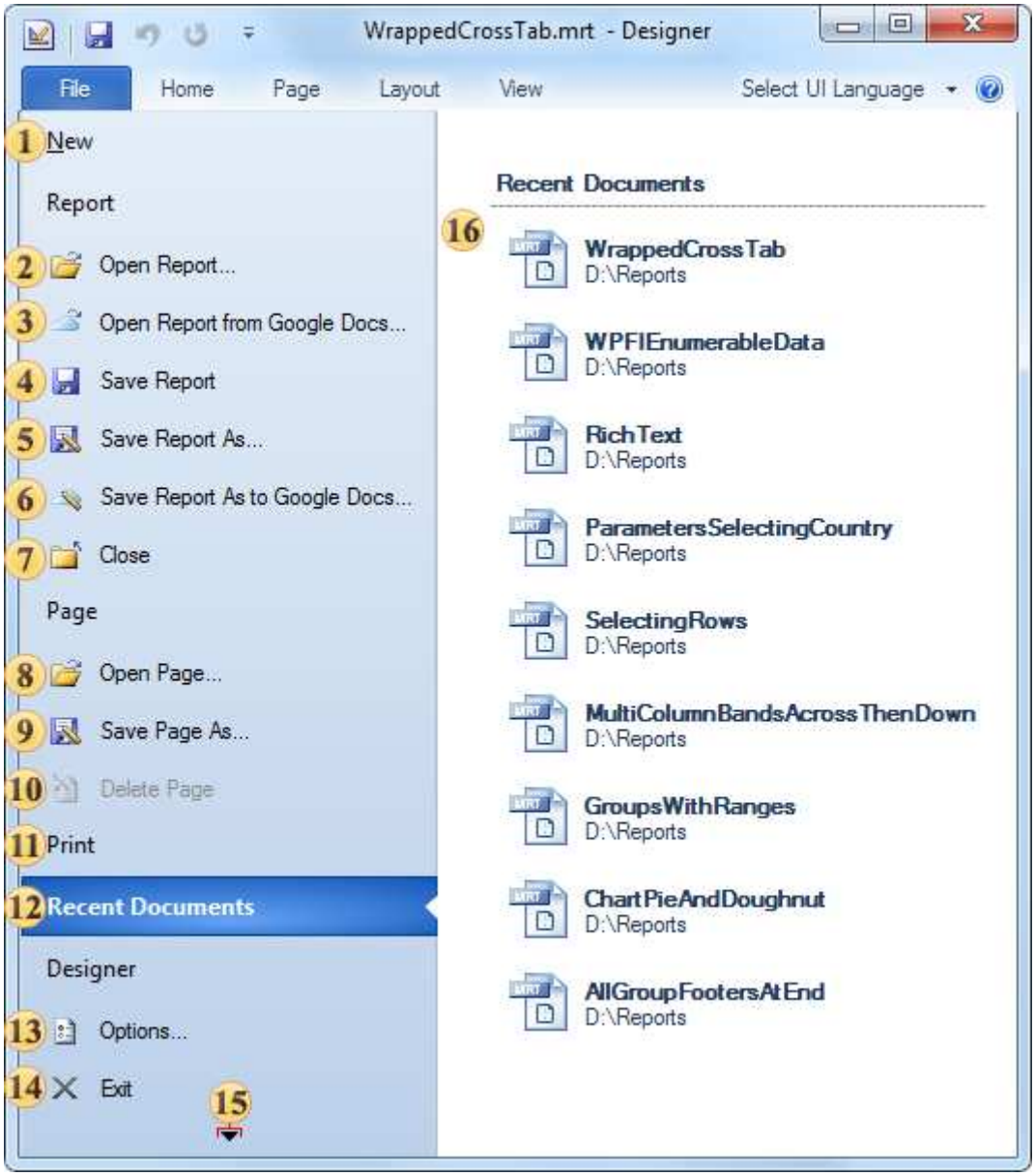
Ribbon UI 2010

Ribbon interface 2010, based on the tabs (similar to **Microsoft Office 2010**). The tabs are grouped instruments, thereby reducing the number of toolbars. Consider this type of interface in more detail.

Main Menu

The basic **Ribbon** 2010 interface is a **Ribbon** 2010 panel. The picture below shows how it looks like:

The main element of the **Ribbon** panel is the **Application Button** and the menu that is called by pressing this button. This is a main menu of the report designer. Basic commands for work with reports in the report designer are represented in the menu. The picture below shows a menu of the application and its items.



The **Ribbon** menu of the 2010 interface is split on 6 group: **New** group, **Report** group, **Page** group, **Print** group, **Recent Documents** group and **Designer** group. The first group is represented by one of the main menu items:

1 The **New** menu item contains submenu where a list of new report components is available for creation is shown.

The second group from 2 to 7 is a set of items to report control:

2 The **Open Report** menu item. When calling this menu item, a dialog for opening a report will appear.

3 С помощью данного пункта можно открыть ранее сохраненный отчет на сервисе Google Docs.

4 The **Save Report** menu item saves changes in a report. If a report was not changed previously, then the **Save Report As** menu item will be called automatically.

5 The **Save Report As** menu item. When calling this menu item, a dialog for saving a report will appear.

6 С помощью данного пункта меню можно сохранить отчет на сервисе Google Docs в определенной коллекции.

7 The **Close** menu item closes a report.

The third group from 8 to 10 is a set of items to control report pages:

6 The **Open Page** menu item. When calling this menu item, a dialog for opening a page will appear.

7 The **Save Page As** menu item. When calling this menu item, a dialog for selecting place for saving a current page.

8 The **Delete Page** menu item. When calling this menu item, a dialog for deleting a page or form will appear.

The next menu items from to represents separate group:

9 The **Print** menu item contains submenu where a list of commands for report printing is shown.

10 The **Recent Documents** menu item contains submenu where recently opened reports are shown.

The last group from to represents a group of items to control designer:

Пункт меню **Печать (Print)** содержит подменю, в котором отображается список команд, связанных с печатью отчета.

The **Recent Documents** menu item contains submenu where recently opened reports are shown.

Последняя группа с по представляет группу элементов управления дизайнером:

The **Options** menu item calls a window for designer parameters settings.

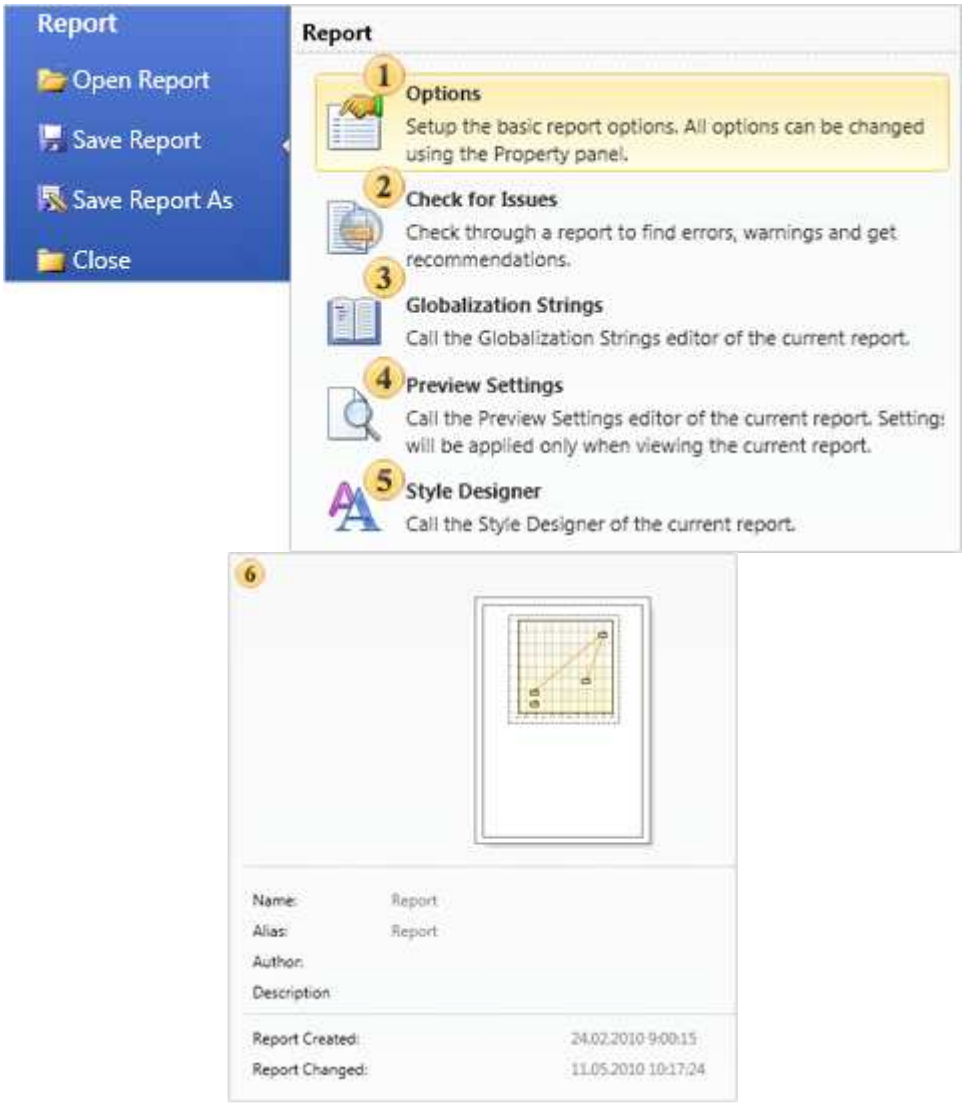
The **Exit** button closes a report designer.

Кнопка **меню групп** содержит выпадающий список, в котором перечислены группы из главного меню. С помощью данной кнопки можно быстро осуществлять навигацию по группам в главном меню дизайнера отчетов.

The panel shows a submenu of selected menu item or selected group.

Report Group

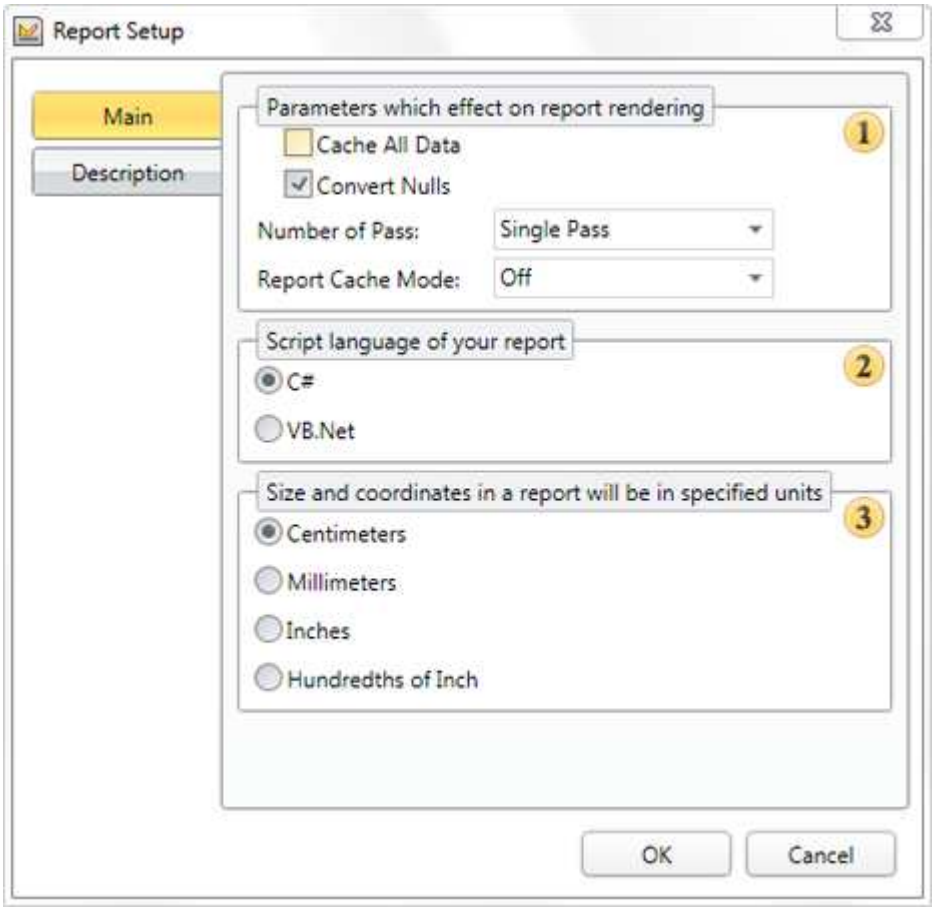
The **Report** group is a set of commands to control a report. The picture below shows the submenu of the **Report** group:



1 The **Options** submenu item calls the **Report Setup** window, in what it is possible to change the basic report options.
2 The **Check for Issues** submenu item calls the **Report Checker** window in what it is possible to get information about errors, warnings, messages regarding the report.
3 The **Globalization Strings** submenu item calls the **Globalization Editor**, using what, it is possible to localize a report to other cultures.
4 The **Preview Settings** submenu item calls the **Page Settings** window of the preview.
5 The **Style Designer** submenu item calls the style editor where it is possible to edit report styles.
6 The **Preview Panel** shows a reduced copy of a report template, and also shows an information about a report.

Report Setup Window

If to select the **Options** item in the submenu of the **Report** group, then the **Report Setup** window is invoked that allows you to identify the basic information and report parameters. The picture below shows the **Report Setup** window:



As can be seen from the picture above, the editor of the report parameters contains two tabs: **Main** and **Description**. The **Main** tab is represented by three groups, which define the most important parameters of the report:

1. In this group, basic parameters that affect the designing of the report are defined.
2. This group defines a scripting language of a report. You may switch between C # and VB.NET.
3. In this group you may select units of the report.

The **Description** tab defines information of report parameters. The picture below shows the **Description** tab:

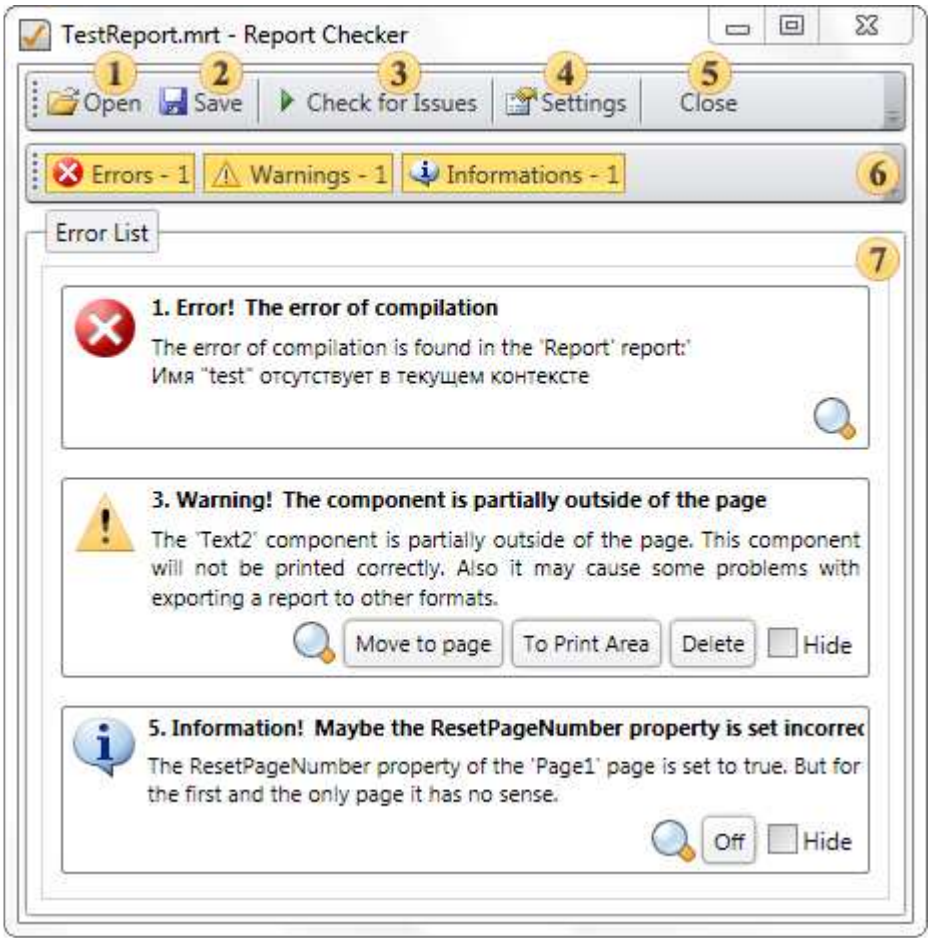
As can be seen from the picture above, the **Description** tab is represented by three groups:

1. A group of names. In this group the **Name**) and **Alias** of a report are specified, as well as the **Author's** name of the report.
2. A group of the report description. In this group the report description is defined.

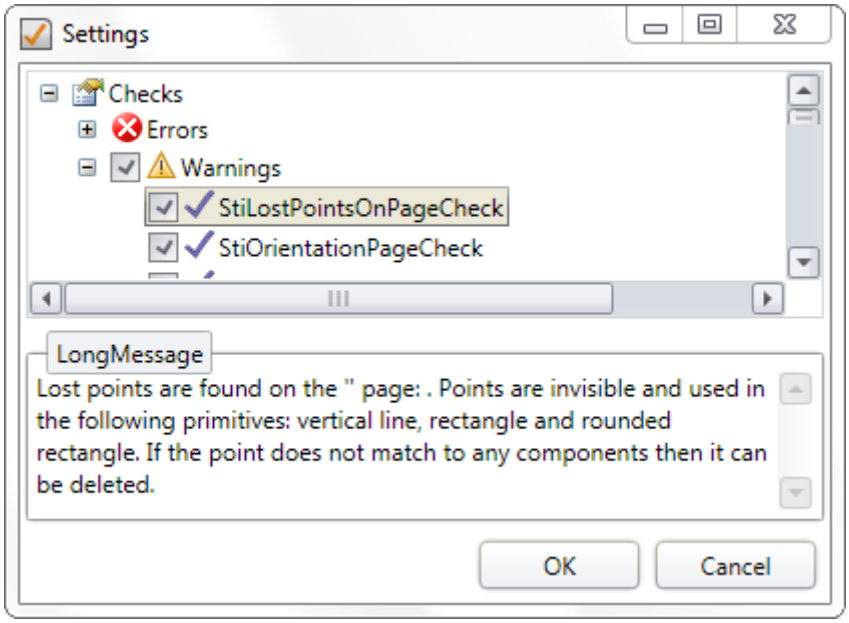
3 This group is not available for editing and displays temporary information: when the report was created **(Report Created)** and the date of last modification of the report **Report Changed**.

Report Checker Window

In order to check the report for errors you should use the **Report Checker**. The **Report Checker** will analyze the report, resulting in an error message, comments, or inaccuracies found in this report. The picture below shows the **Report Checker**:



1 The **Open** button. Clicking this button, the user will see a dialog box to select a previously saved report and loading it to the **Report Checker**.
2 The **Save** button saves changes in the report, that was opened in the **Report Checker**.
3 The **Check for Issues** button starts the process of checking the report.
4 The **Settings** button opens the window of settings of the **Report Checker**. The picture below shows the **Settings** window:



In this window, you can mark messages and warnings you want notifications to be displayed.

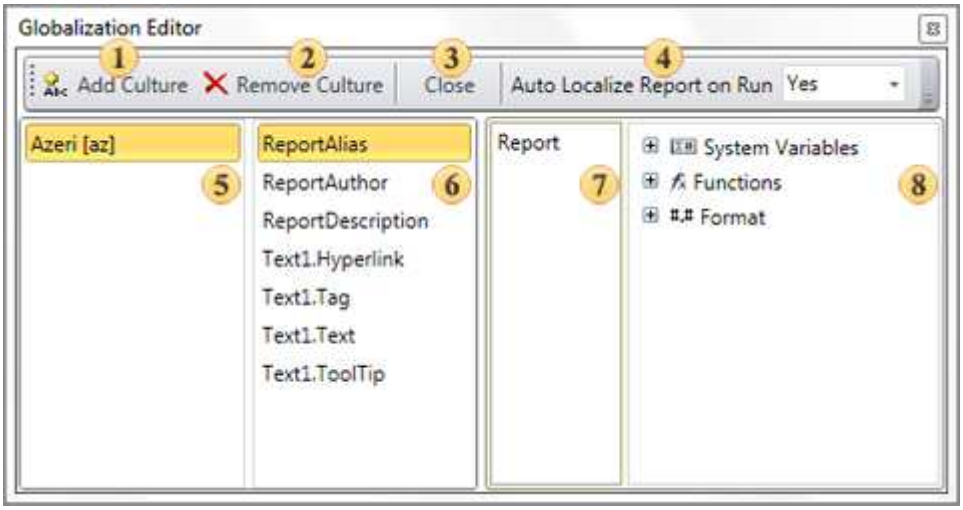
5 The **Close** button closes the window of the **Report Checker**.

6 The panel for showing messages.

7 The panel for showing descriptions of **Errors**, **Warnings**, **Information**.

Globalization Editor

The **Globalization Editor** is in the report designer. This editor allows you to locate the report in several languages. If the value of the **AutoLocalizeReportonRun** property is set to **Yes**, then, when rendering a report, the report generator checks the culture of the installed operating system. Following the culture of the installed system, the reporting tool will check the presence of this culture in the list of cultures used in the report. And if t it found identical culture, it will substitute all the expressions in the report. As a result, the report will be localized in the culture of the operating system, according to certain parameters in the **Globalization Editor**. If the culture of your operating system is not in the list of cultures used in the report, then, when rendering a report, it will not be localized. If the **AutoLocalizeReportonRun** property is set to **No**, then auto localization will not start. The picture below shows the **Globalization Editor** window:



1 The **Add Culture** button calls a list of cultures.

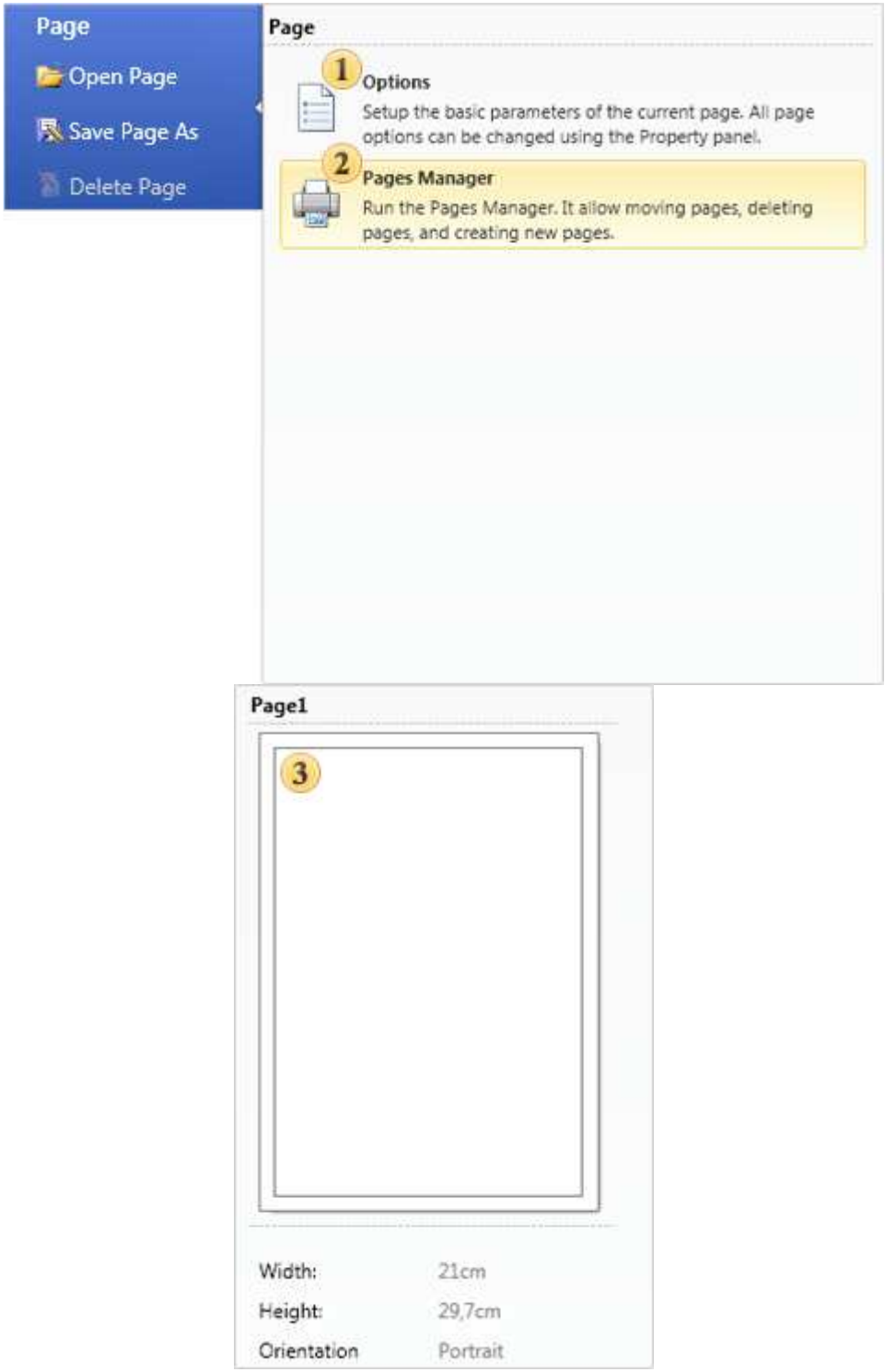
2 The **Remove Culture** button removes the selected culture.

3 The **Close** button closes the editor window

4 The **AutoLocalizeReportonRun** property is used to enable/disable automatic localization when starting report rendering. If the **AutoLocalizeReportonRun** property is set to **Yes**, automatic localization will be done. If the **AutoLocalizeReportonRun** property is set to **No**, automatic localization will not start.
5 A list of cultures used in this report.
6 A list of components for localization.
7 Translation of the contents of the selected component.
8 A list of system variables, functions, formats of the report designer.

Page Group

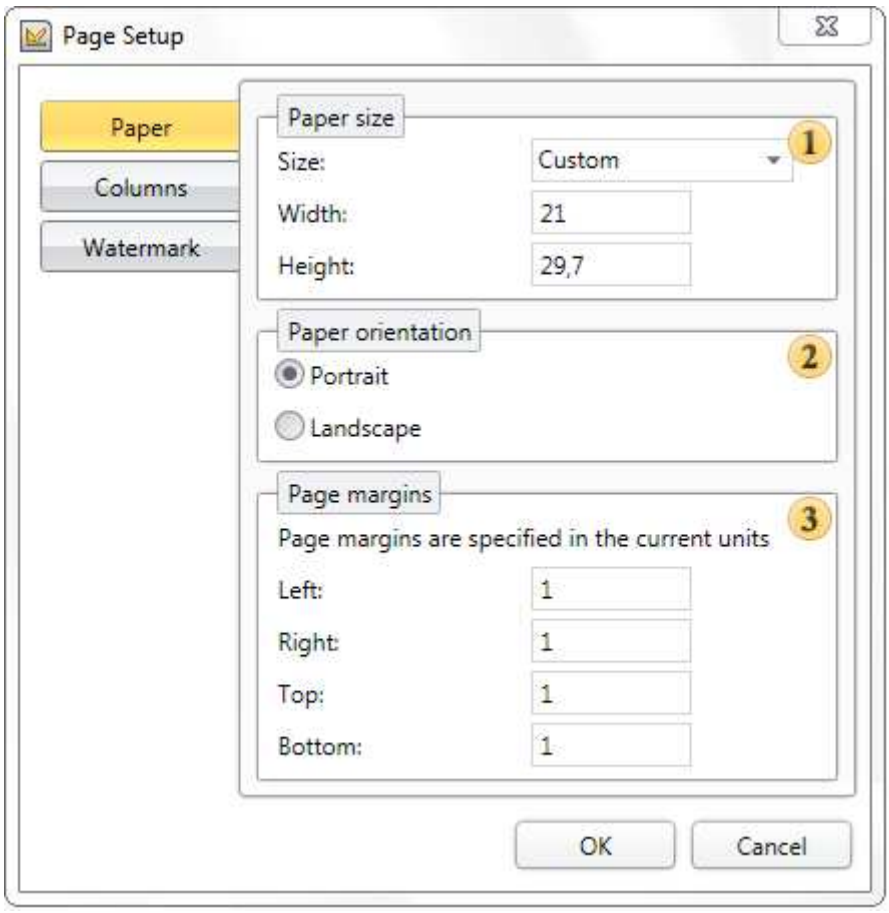
The **Page** group contains commands to control the current page. The picture below shows the **Page** group submenu:



1 The **Options** submenu item calls the **Page Setup** window, where one may specify basic options of a page.
2 The **Pages Manager** submenu item calls the **Pages Manager** window where one may control report pages (create, remove, and move them).
3 The **Preview Panel** shows a scaled-down preview of a selected page of a report template and contains information about page: page **Width**, **Height**, and **Orientation**.

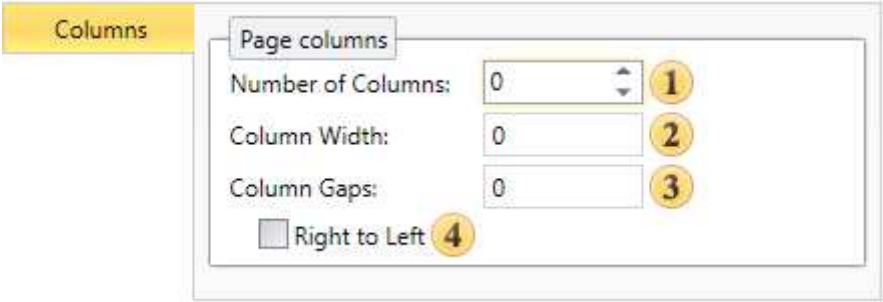
Page Setup Window

If to select the **Options** item in the submenu of the **Page** group, then the **Page Setup** window is invoked that allows you to identify the basic information and report parameters. The picture below shows the **Page Setup** window



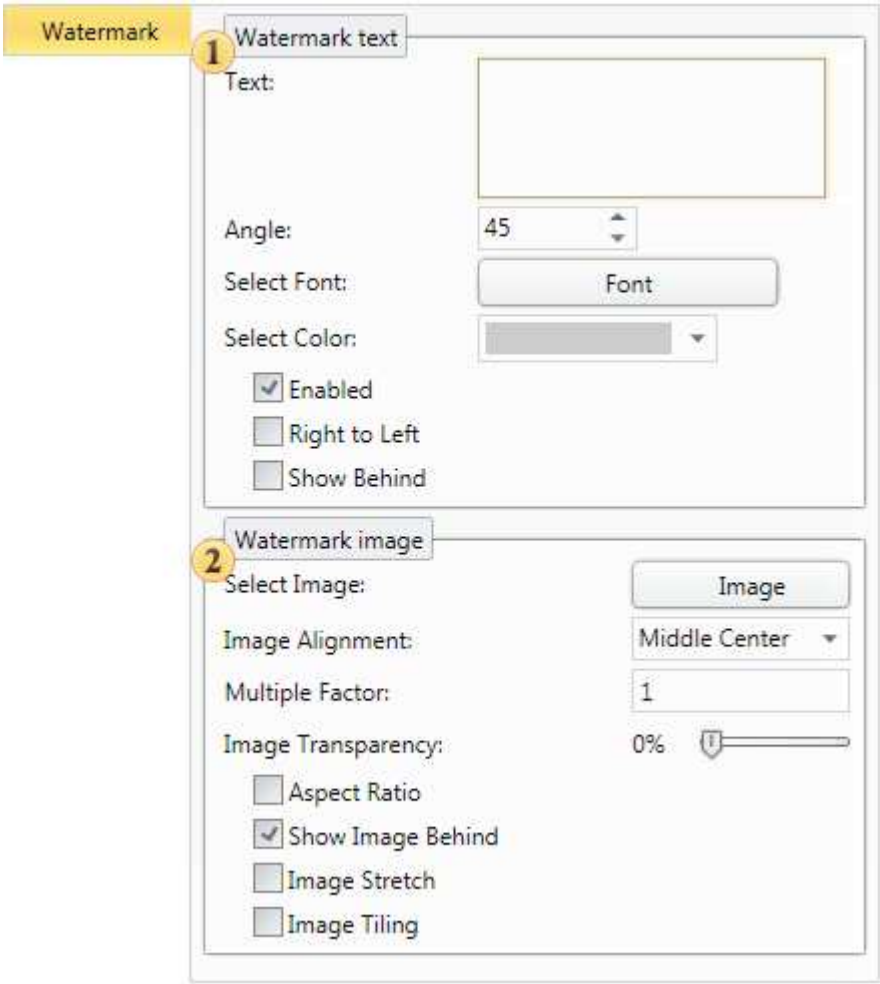
As can be seen from the picture above, the editor of the page parameters contains three tabs: **Paper**, **Columns**, **Watermark**. The **Paper** tab consists of three groups, which define the basic page settings:
1 The **Size** group. This group contains three parameters: page **Size**, **Width**, and **Height**.
2 The **Orientation** group. In this group page orientation options are defined: **Portrait** or **Landscape**.
3 The **Borders** group. In this group the following parameters of **Margins** are defined: **Left**, **Right**, **Top**, and **Bottom** padding.

The **Columns** tab defines the parameters of columns on a page. The picture below shows the **Columns** tab of the **Page Setup** editor:



As can be seen on the picture above the **Columns** tab contains a group of parameters:
1 **Number of Columns** on the page. Any positive number will be the number of columns. A value of 0 is equal to one column on a page.
2 **Column Width**. The value of this parameter will be the width of the column. If the value is equal to 0, the column width will be determined automatically: page width divided by the number of columns. This value is indicated in units of the report.
3 **Column Gaps** is the distance between two columns. The value is indicated in units of the report.
4 **Right to left** enables/disables the mode of arrangement of columns on the page from right to left. By default, columns are arranged from left to right.

The **Watermark** tab defines the watermark options on the page. The picture below shows the **Watermark** tab of the **Page Setup** editor:

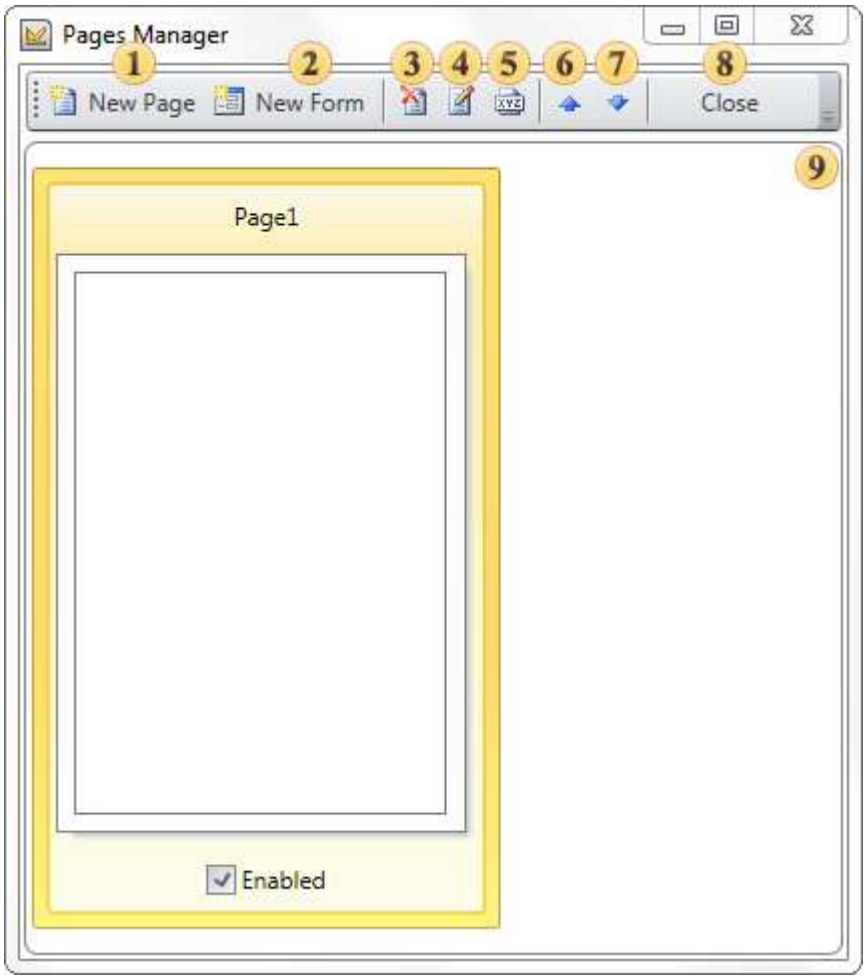


As can be seen from the picture above, the **Watermark** tab contains two groups of parameters:
1 **Watermark text group**. In this group there are options of a watermark that relate to its text part: text (**Text**) is a typed text of the watermark; rotation angle of the text (**Angle**), the font type (**Select Font**) text; text color (**Select Color**); **Enabled** enables/disables the watermark, right to left (**Right to left**) mode of the text layout, display behind (**Show Behind**) is the location of the text in the background.

2 **Watermark image** group. In this group there are options of the watermark, such as: **Select Image** that opens the image loader; **Image Alignment** aligns images according to the selected value; **Multiple Factor** sets zoom of the displayed image relative to loaded one; **Image Transparency; Aspect Radio** retains the proportions of the image; **Show Image Behind** shows the image behind the text; **Image Stretch** stretches the image on the report page; **Image Tiling**, if the picture is less than a page, it can be tiled.

Pages Manager

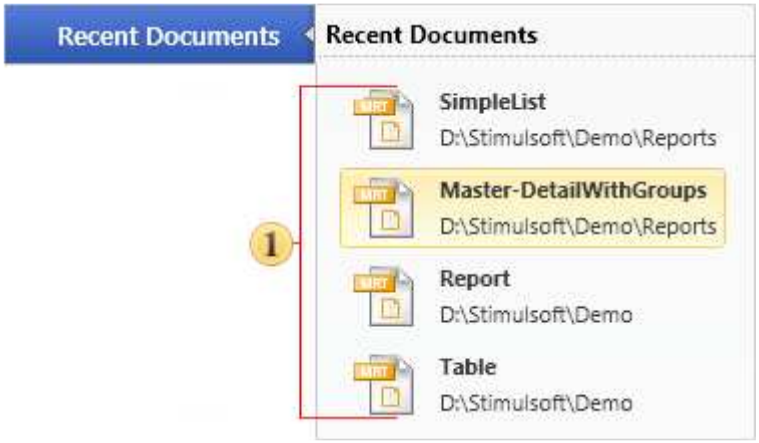
The **Pages Manager** is used to control pages of a report. With this manager it is possible to create, delete pages and forms of a report. The picture below shows the window of a **Pages Manager**:



1 The **New Page** button creates a new page in a report.
2 The **New Form** button creates a new form in a report.
3 The **Delete Page** button deletes selected page or a form.
4 The **Page Setup** button calls the **Page Setup** window.
5 The **Rename** button calls the **Rename** window, where one may change a name of a page or form.
6 The **Up** button is used to move cursor on top or left.
7 The **Down** button is used to move cursor to bottom or right.
8 The **Close** button closes pages manager.
9 View panel. Shows all pages and report forms.

Recent Documents Group

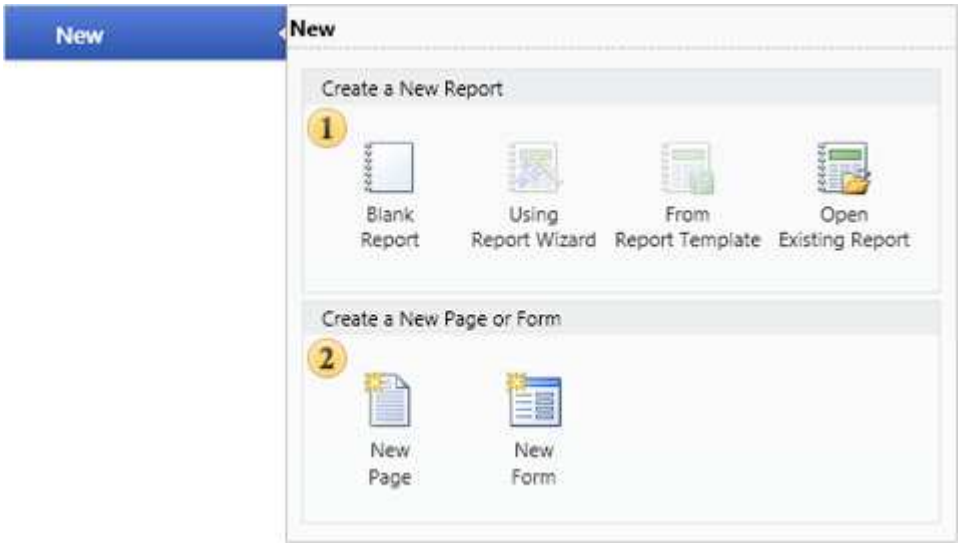
The **Recent Documents** group contains a list of recently loaded documents in the designer. The picture below shows the submenu of the **Recent Documents** group:



1 The list of recently opened documents.

New Group

The **New** group contains a submenu with options of creating reports or forms.



As seen from the picture above the **New** group submenu contains two fields:
1 In this field the way of creating reports are shown. There are **Blank Report**, **Using Report Wizard**, **From Report Template**, **Open Existing Report**.
2 In this field the components available for creating are displayed. These are **New Page**, **New Form**.

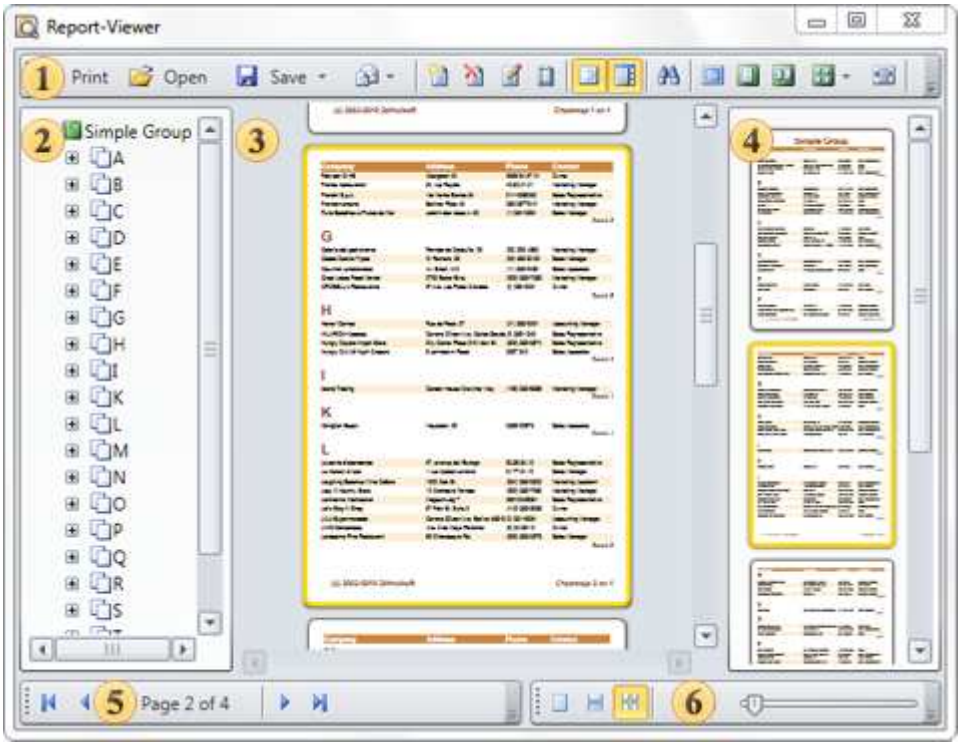
Print Group

The **Print** group contains a submenu where tools for report printing are present. The picture below shows this submenu:

1 **Quick Print**. Print selected report by default without calling the dialog window.
2 **Print**. Print with calling the printing dialog window. You should define the **Printer**, **Print Range**, **Copies**.
3 **Preview** calls a **Viewer**.

Viewer

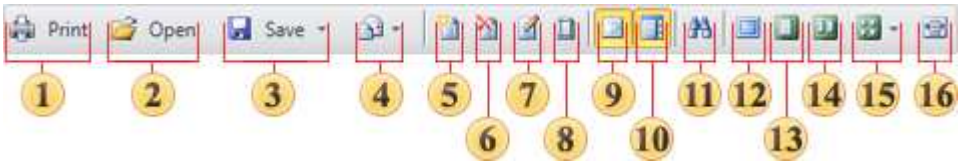
On the picture below the basic elements of the report viewer are shown.



1 The basic commands to control the report are represented on the toolbar.
2 Tree of bookmarks of the output report. Using these bookmarks it is possible to jump by structure elements of a report.
3 The output report.
4 The report thumbnails panel. Decreased copies of a report are shown on this panel. The panel is used to quickly navigate throughout of a report.
5 The toolbar to scroll up or down in reports pages.
6 The toolbar to select the mode of report showing.
7 The toolbar to increase or decrease report zoom.

Toolbar panel

Main controls are placed on this toolbar. The picture below shows the structure of the toolbar:



1 Run report printing. After activation of this command the printing dialog with parameters of printing will be displayed.
2 Open previously saved report. Any rendered report can be saved to .mdc or .mdz format for further preview.
3 Save the rendered report to other file formats.
4 Send the render report via EMail. The report will be converted to one of the file formats.
5 Add a new empty page to the rendered report.
6 Delete the current page of a report.
7 Open the reports designer and show the current page for editing.
8 Opens the window of changing basic parameters of the rendered report.
9 Show/hide the tree of bookmarks. If there are no bookmarks in the rendered report then the viewer will automatically hide the tree of bookmarks. If there are bookmarks in a report, then the viewer will automatically show the tree of bookmarks.
10 Show/hide the reports thumbnails.

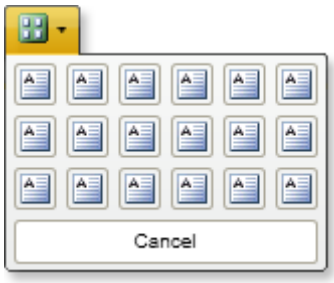
Enable the search panel.

Run the full screen mode of report showing.

Change zoom of the report to display only one full page. More than one page by the width can be output.

Change zoom of the report to display two pages on the screen.

Change zoom of the report according to horizontal and vertical sizes of pages.



Change zoom of the report to fit the page width to the screen width.

Run matrix mode of the report showing.

Navigation

On the picture below the toolbar that is used for report navigation is shown.

1 Set the first page of a report as the current page.
2 Set the previous page of a report as the current one.
3 Show the number of the current page and the number of pages in a report. If click on it then it is possible to indicate the number of a page that should be the current.

4 Set the next page of a report as the current one.
5 Set the last page of a report as the current page.

Showing pages

The viewer for WPF supports three modes of viewing pages:
1. **Single Page**
2. **Continuous**
3. **Multiple Pages**

Three buttons are used to control the page view mode. They are placed at the bottom of the viewer in the right corner.

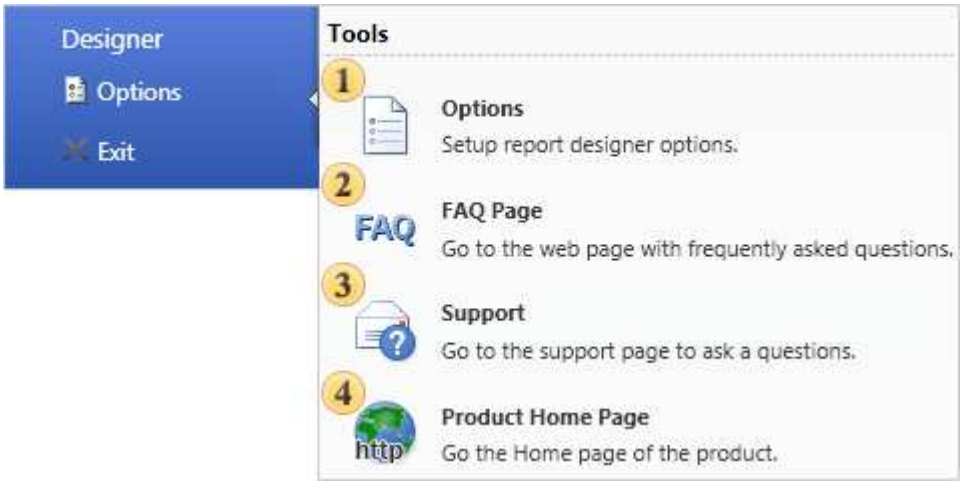
Search panel

The search panel is used to search some text in the report. On the main toolbar this option can be enabled by clicking the binocular icon. All controls for search are placed on a single panel.

1 Close the search panel.
2 The text that should be found.
3 The button to find previous phrase.
4 The button to find next phrase.
5 Find specific search results using the match case option.
6 Find specific search results using the whole word option.

Designer

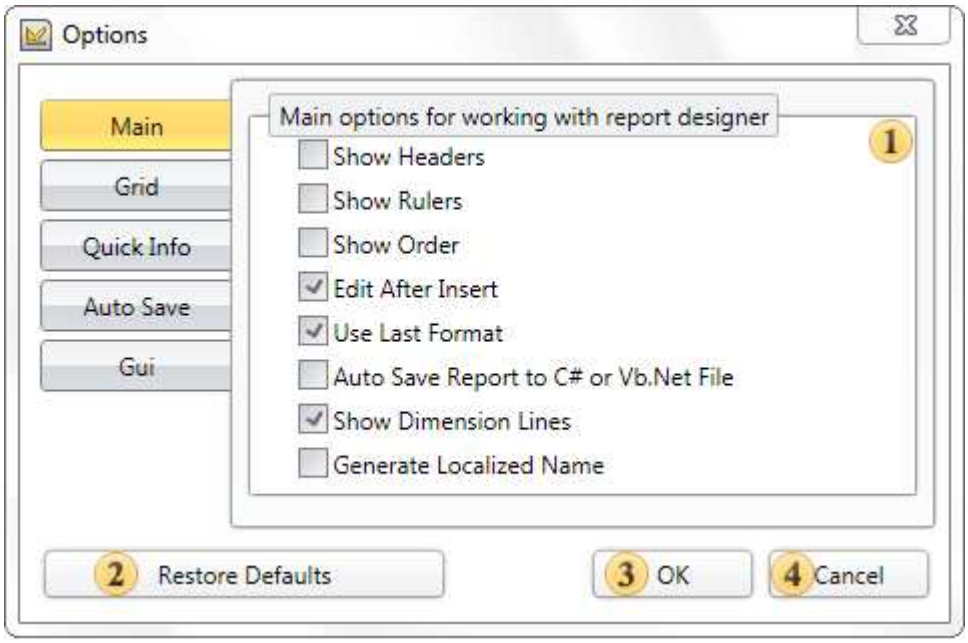
The **Designer** group is relevant to the report designer. The picture below shows the submenu of the **Designer** group:



1 The **Options** submenu item calls the **Options** window for setting parameters of the designer.
2 The **FAQ** submenu item redirects to a Web page of Stimulsoft site with Frequently Asked Questions.
3 The **Support** submenu item redirects to a form for sending a request to the technical support team of Stimulsoft.
4 The **Product Home Page** submenu item redirects to the product home page: www.stimulsoft.com

Options

If to select the **Options** item in the submenu of the **Designer** group, then the **Options** window is invoked where you can do basic settings and information settings of a report. The picture below the **Options** dialog box:

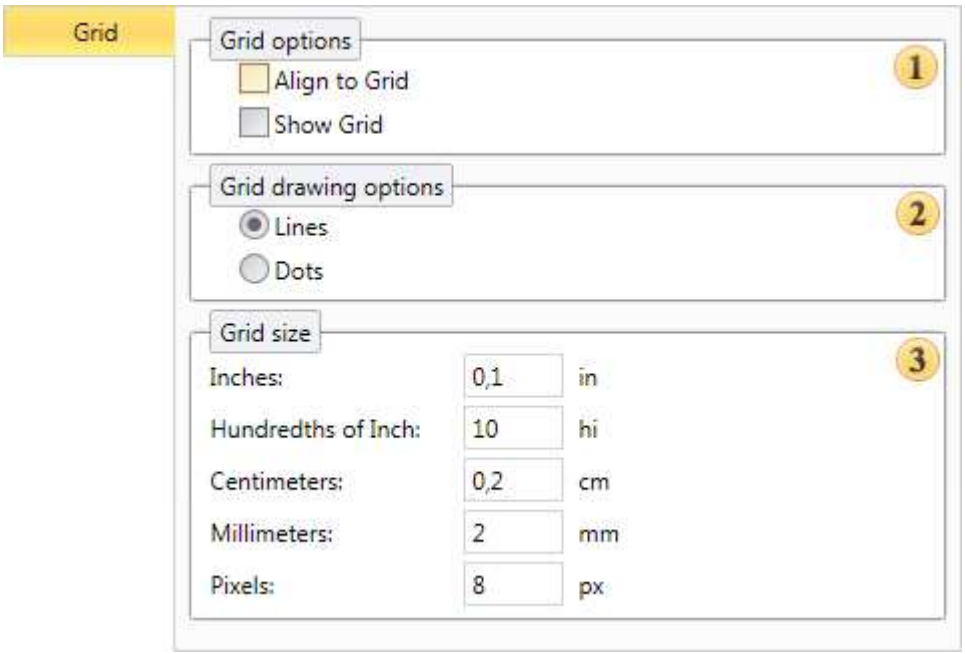


As can be seen from the picture above, the **Options** dialog box contains five tabs: **Main**, **Grid**, **Quick Info**, **Auto Save**, **Gui**. The **Main** tab is represented by one 1, which has the basic options of a designer such as:

1. The Show Headers option enables/disables displaying headers of components of the report.
2. The **Show Rulers** option enables/disables displaying rulers.
3. The **Show Order** option enables/disables displaying the order number of the report component.
4. The **Edit After Insert** option enables/disables invoking the editor after creating a component in the report.
5. The **Use Last Format** option enables/disables using the latest format of a component.
6. The **Auto Save Report to C # or VB.NET File** option enables/disables auto-saving of a report as a source file. This source file will be saved together with a report in the .mrt file.
7. The **Show Dimension Lines** option enables/disables the dimension lines.

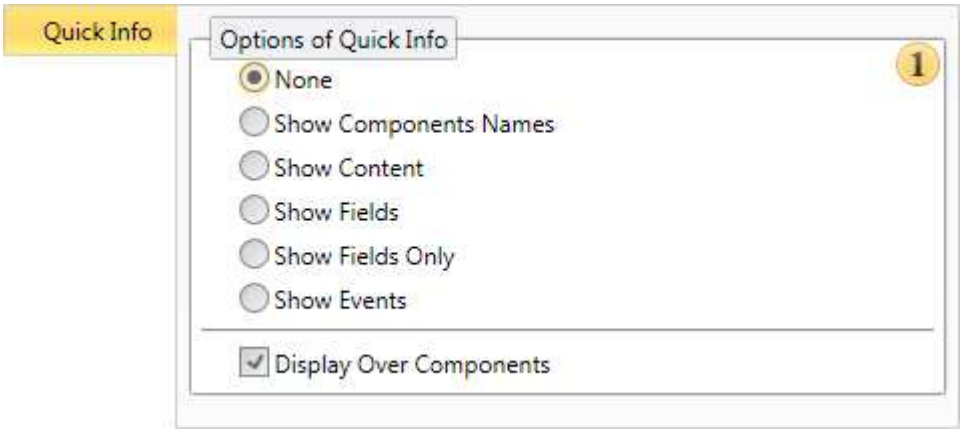
8. The **Generate Localized Name** option enables/disables the mode of creating a component with localized names. If this option is disabled, then the components are created with names in English. If included, then the component name will be localized according to the selected language.

The **Grid** tab defines the parameters of displaying a grid.



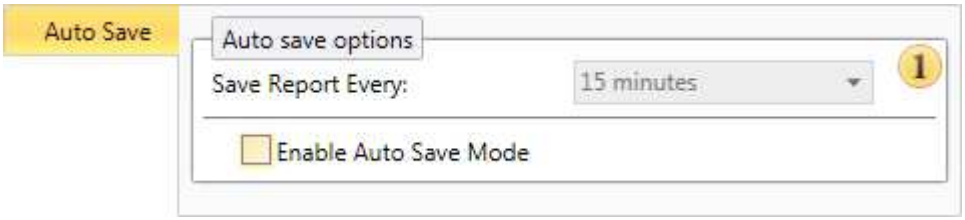
As can be seen from the picture above, this tab consists by three groups:
1 The **Grid Options** group includes such parameters as: **Align to Grid** snaps a report component to grid; **Show Grid** enables/disables the grid.
2 The **Grid drawing options** group. The grid style can be applied in this group: **Lines** or **Dots**;
3 The **Grid size** group. This group sets the grid spacing in different units: **Inches**, **Hundredths of Inch**, **Centimeters**, **Millimeters**, **Pixels**.

Parameters of quick info messages are defined in the **Quick Info** tab.



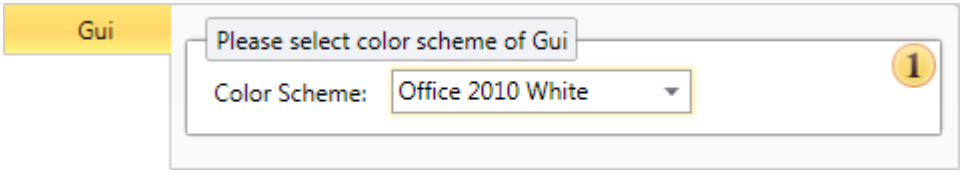
As can be seen from the picture above, this tab contains a single group 1, which defines the following parameters:
1. The Options of Quick Info option specifies what information you want to display.
2. The Display Over Components option enables/disables the mode of displaying the quick info in the foreground, i.e. over all components.

The **Auto Save** tab contains the parameters responsible for the report auto-saving.



This tab contains a single group 1, which contains the following options:
1. The **Save Report Every** option determines the time interval after which an auto-save event occurs.
2. The **Enable Auto Save Mode** option enables/disables the auto-save mode of the report.

Interface options of the designer are defined on the **Gui** tab.



This tab contains a single group 1 and one **Color Scheme** parameter, which is required to change the type of interface and/or color theme.

Tabs

Tab is a part of the interface on which the toolbar. At the designer shows four tabs: **Home**, **Page**, **Layout**, **View**. Consider these tabs and the main instruments are located on them more.

Home Tab

This is a basic tab of the reports designer. Main commands of setting report components are placed on this tab.

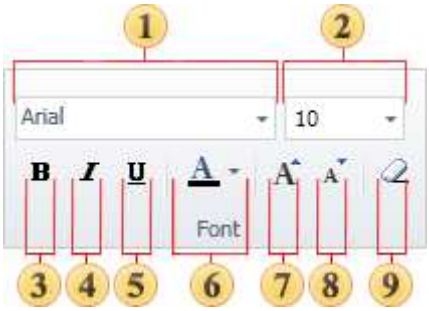
Clipboard Group

This group allows working with the Clipboard of the report designer.

1. Paste components from the Clipboard on the current page of a report.
2. Cut the selected components from the current page to the Clipboard.
3. Copy the selected components on the current page to the Clipboard.
4. Delete selected components on the current page.

Font Group

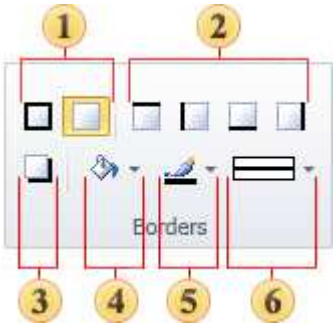
This group is used to output text with specified font type, color etc.



1. Select the font type of the text components on the current page.
2. Select font size of the text components on the current page.
3. Set the Bold font style.
4. Set the Italic font style.
5. Set the Underlined font style.
6. Set the font color of the text components on the current page.
7. Increase the font size.
8. Decrease the font style.
9. Delete the content of all selected text components.

Borders Group

This group contains the commands to setup border components.



All commands can be applied to selected components on the current page.

1. Set or remove borders from all sides of a component.
2. Set or remove borders from each side of a component.
3. Border color of a component.
4. Set the shadow of a component.
5. Background color of a component.
6. The type of the border line.

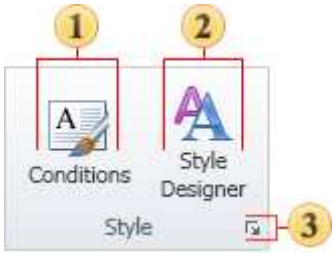
Text Format Group

The group to control text formatting.

1. Select text format.
2. Select symbols of currency.
3. Set one of the predefined formats to output a text.
4. Call a form of formats editing.

Style Group

This group is used to control styles and conditions which are used to automatically design components in a report.



1. Open a window of the Conditions Editor for selected components.
2. Open a window of Styles Editor.
3. Call a form of styles editing.

Alignment Group

The group is used to align the content of components horizontally and vertically. Also it is possible to set the angle of the text rotation and control the **WordWrap** property.

All commands are applied to selected components on the current page.

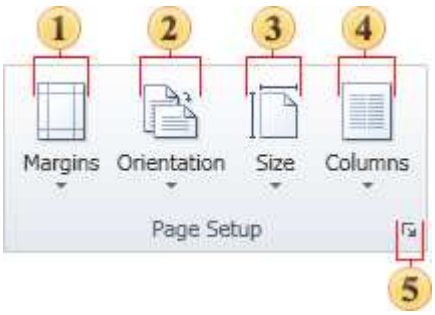
1. Align top the content of a component.
2. Align center vertically the content of a component.
3. Align bottom the content of a component.
4. The angle of the text rotation. This command can be applied only to the text component.
5. Align left the content of a component.
6. Align center horizontally the content of a component.
7. Align right the content of a component.
8. Justify the content of a text component.
9. Used for the WordWrap property of the text component.

Page Tab

This tab is used to control page parameters.

Page Setup Group

This group contains elements to control basic parameters of a page. These are page margins, orientation, page size, columns.

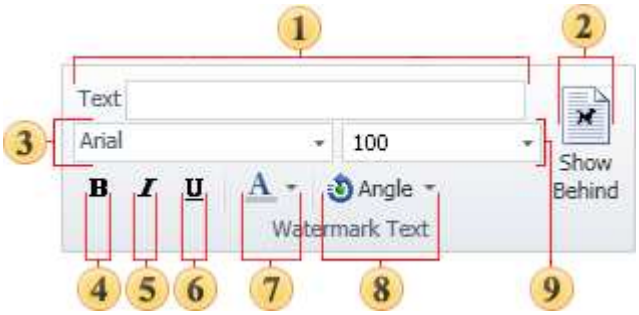


Changes can be applied to the current page in the Reports Designer.

1. Select sizes of page margins.
2. Select Portrait or Landscape orientation of a page.
3. Select page size.
4. Select number of columns on a page.
5. Call the Page Setup dialog window.

Watermark Text Group

This group is used to control watermark text output.



1. Watermark text.
2. Show watermark text behind the components of a page.
3. Select a font type to output a text of the watermark.
4. Set the Bold font style of the watermark text.
5. Set the Italic font style of the watermark text.
6. Set the Underlined font style of the watermark text.
7. Watermark text color.
8. Set angle rotation of the watermark text.
9. Font size of the watermark text.

Watermark Image Group

This group is used to control the watermark image.

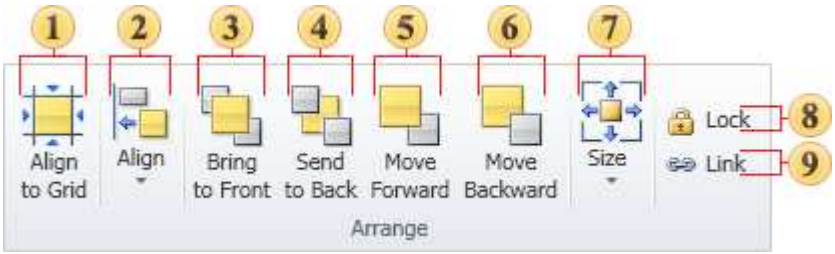
1. Load an image for the watermark.
2. Set transparency of the watermark image.
3. Align the watermark image.
4. Show the watermark image behind page components on page.

Layout Tab

This tab is used to control placing different components on a page and also to specify styles of docking components.

Arrange Group

The group contains a lot of commands to change position of components on a page. The picture below shows this group.



1. Align all selected components to the page grid.
2. Align selected components. This element contains submenu and short description in this topic below.
3. Bring selected components to Front.
4. Send selected components to Back.
5. Move selected components on one level forward.
6. Move selected components on one level backward.
7. Choose the size of selected components. It contains submenu and is described in this topic below.
8. Control the Lock property.
9. Control the Link property.

The description of the **Align** button, specified with number 2 on the picture above.

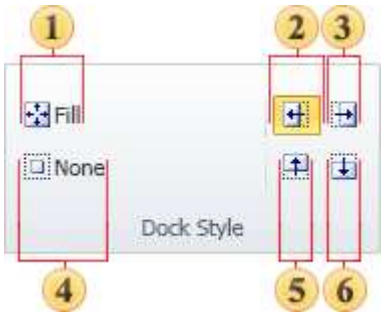
1. Align all selected components to their common left margin.
2. Align horizontally all selected components to their common center.
3. Align all selected components to their common right margin.
4. Align all selected components to their common top margin.
5. Align vertically all selected components to their common center.
6. Align all selected components to their common bottom margin.
7. Make horizontal spacing of selected components equal by their width.
8. Make vertical spacing of selected components equal by their height.
9. Center all selected components horizontally.
10. Center all selected components vertically.

The description of the **Size** button, specified with number 7 on the topmost picture.

1. Make the same size of components as the size of the first selected component.
2. Make the same width of components as the size of the first selected component.
3. Make the same height of components as the size of the first selected component.

Dock Style Group

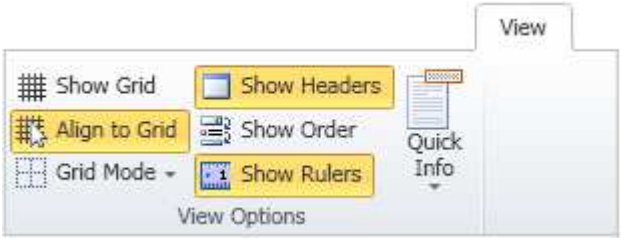
This group contains elements to set the dock style of selected components.



1. Dock selected components to all edges.
2. Dock selected components to the left edge.
3. Dock selected components to the right edge.
4. Removes dock style of selected components.
5. Dock selected components to the top edge.
6. Dock selected components to the bottom edge.

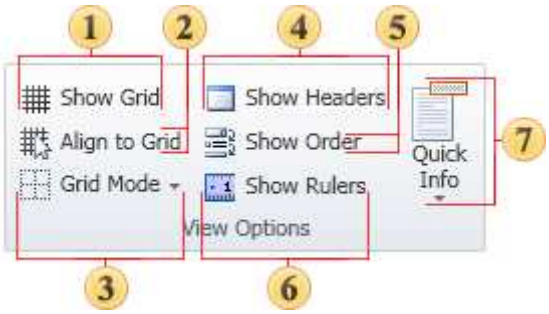
View Tab

This tab is used to control different options of working with report designer.



View Options Group

This group contains settings for different parameters of showing a report and working with the reports designer.



1. Control showing grid on a page.
2. Control automatic alignment of components by the grid.
3. Select grid type.
4. Show/hide headers of bands.
5. Show/hide the order of placing components on a page.
6. Show/hide rulers on a page.
7. Control Quick Info.

2010 Status Bar

Status bar in 2010 **Ribbon** interface is located at the bottom of the designer. This is a panel on what controls are placed. The picture below shows a status bar Ribbon Interface:

This panel contains 5 fields:

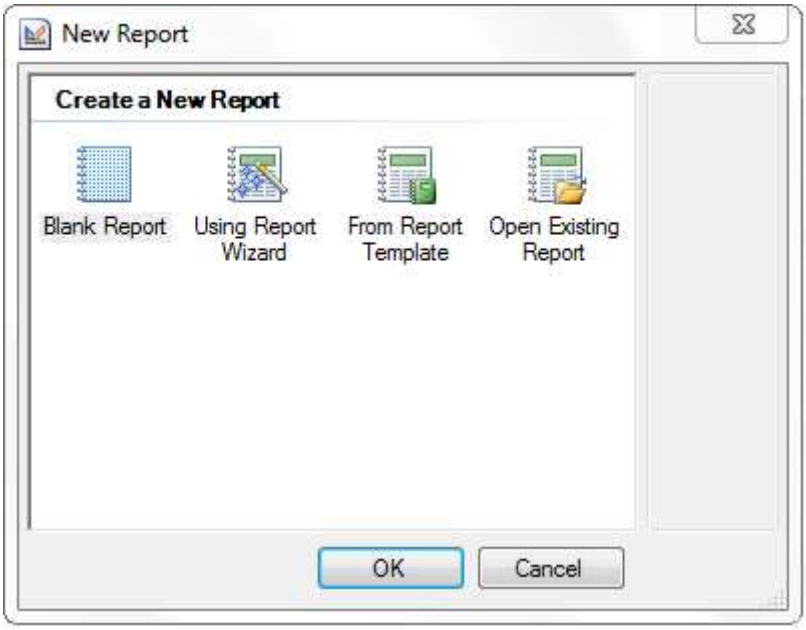
1. Units. This field displays the units used in a report, and also allows you to change them.
2. This field displays the currently selected components. In this case, a page "**Page 1**"is selected.
3. This field displays the coordinates of the cursor on a page of a report template.
4. Show a messages window. Messages are information about errors of compilation or errors which occur while report rendering.
5. Control zoom of a report template page.

Creating Reports in Designer

A report in the designer can be created using the tools for creating reports and report components. Also, you can create a report using Report Wizards: Standard Report, Master-Detail Report and Label Report.

Introduction

When creating a new report in the **New Report** dialog you should choose a way to create a report. The picture below shows the **New Report** dialog:



As can be seen from the picture above, there are four ways of creating a report: **Blank Report**, **Using Report Wizard**, **From Report Template**, and **Open Existing Report**.

1. **Blank Report**. A report is created manually, i.e., a blank page is loaded. A user adds report components without using the wizard.

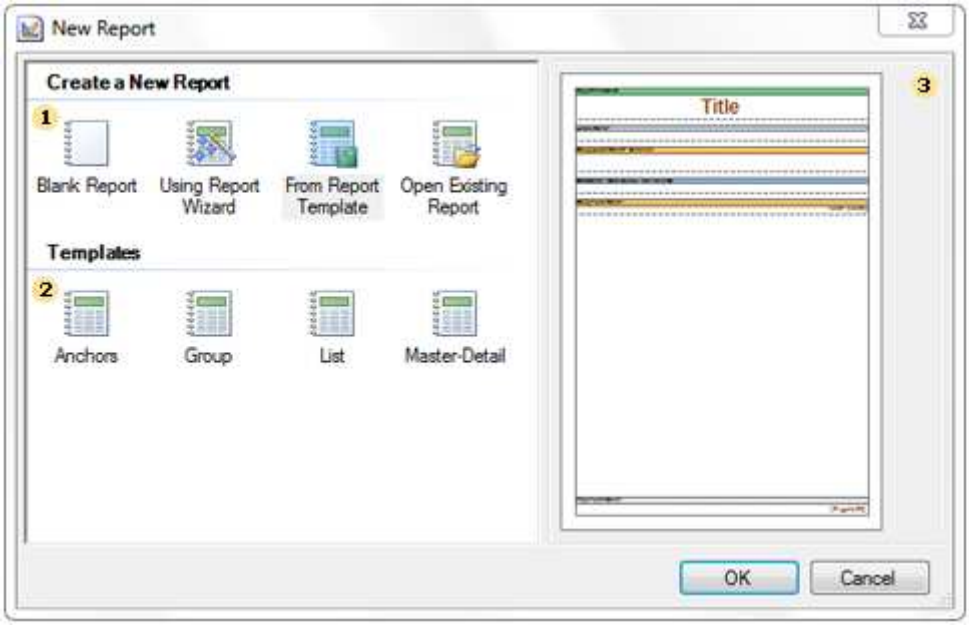
2. **Using Report Wizard**. A report is created using the wizard. A user selects a wizard for step-by-step designing a report.

3. **From Report Template**. A user chooses a report template among available ones.

4. **Open Existing Report**. A user can open a previously saved report.

Report Templates

When creating a new report it is possible to use the report template. Select **File | New Report | Create a New Report | From Report Template**. A report template is a report that is not rendered. The picture below shows a **New Report** window where you can see a list of available report templates:

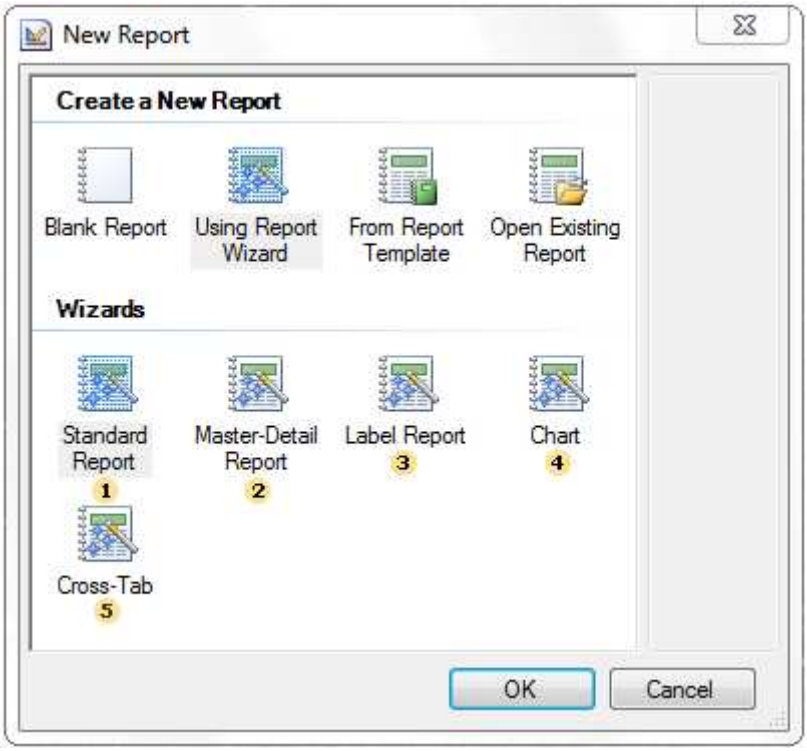


1 A field for selecting the way of creating a report. In our case we need to select the **From Report Template** item.
2 A field where one can see a list of report templates.
3 A window shows a preview of a report template.

Report templates, are stored in **Templates** folder of the root directory of the report designer. If this folder is not created or there are no reports in this folder, then the field marked 2 on a picture above will be empty. It is possible to add their own report templates to the **Templates** folder.

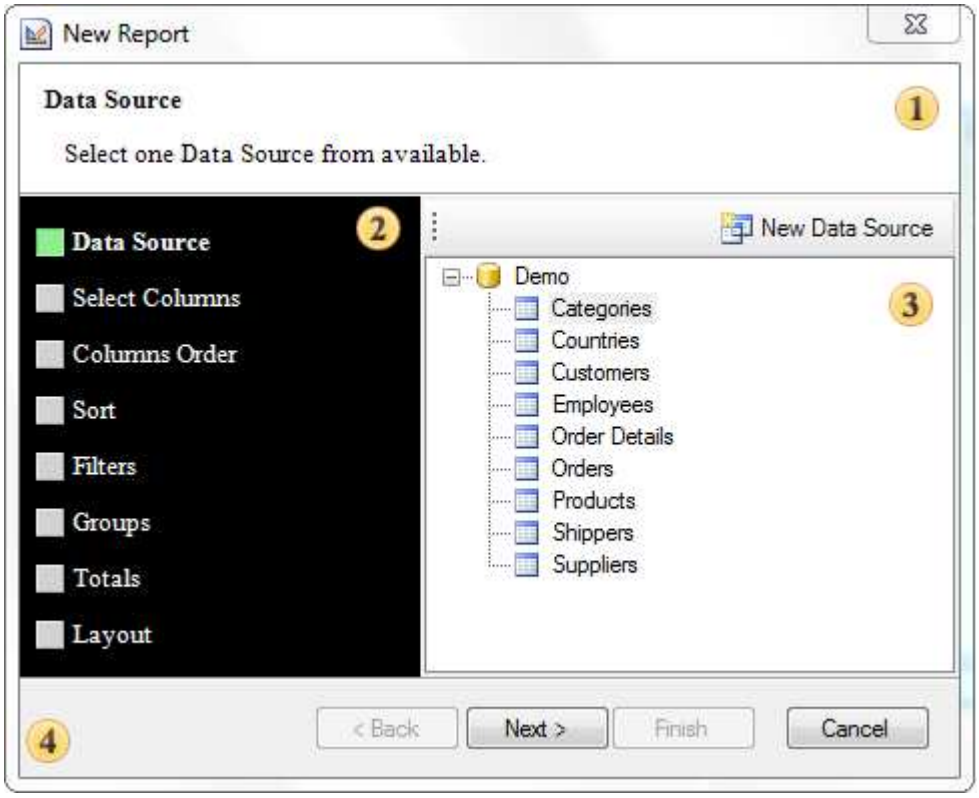
Report Wizards

Select the **Using Report Wizard** item in the **New Report** window to call the **Report Wizard**. Then it is necessary to select the type of a report wizard. The picture below shows the **New Report** window to start creating a report using reports wizard:



1 The **Standard Report** wizard is used to create reports as a list.
2 The **Master-Detail Report** wizard is used to create a **Master-Detail** reports.
3 The **Label Report** wizard is used to create Label reports.
4 The **Chart** wizard is used to create reports with charts.
5 The **Cross-Tab** wizard is used to create Cross-Tab reports.

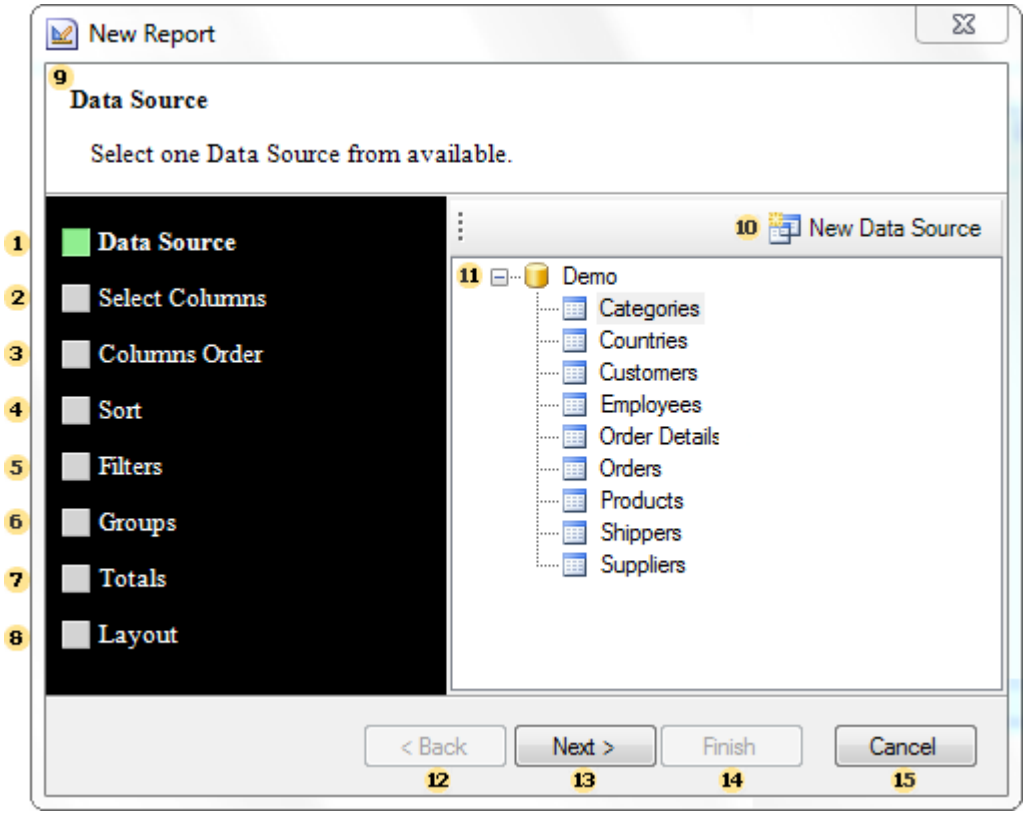
Any **Report Wizard** has the following panels: **Description Panel**, **Steps Panel**, **Selection Parameters Panel**, **Control Panel**. The picture below shows the **Standard Report** wizard:



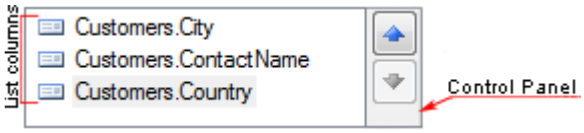
1 The **Description Panel**. This panel shows description of each steps to be done.
2 The **Steps Panel**. Shows steps of creating reports using a report wizard.
3 The **Selection Parameters Panel**. This panel shows report parameters. On each step of report creation its own options are available.
4 The **Control Panel**. Contains buttons to control the **Report Wizard**.

Standard Report Wizard

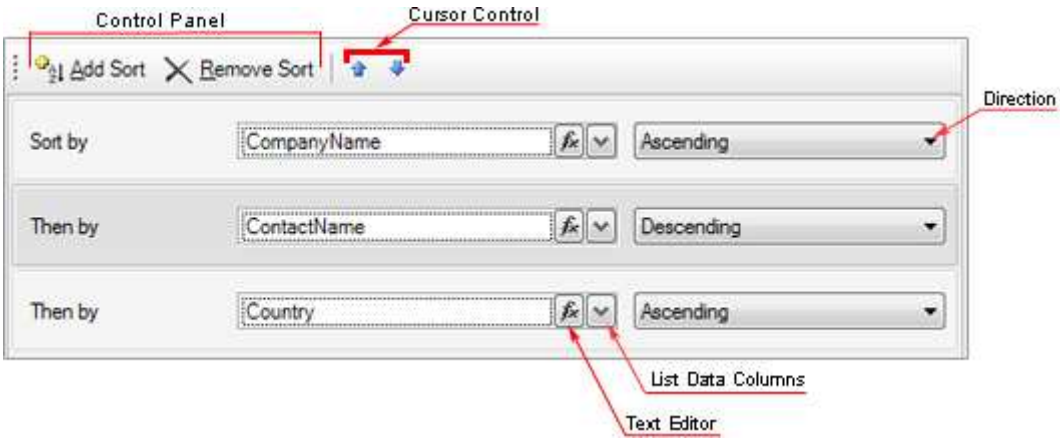
When creating a report using the **Standard Report** wizard, this report will contain one **DataBand** or one data **Table** (depends on what is selected). The picture below shows a window of the **Standard Report** wizard:



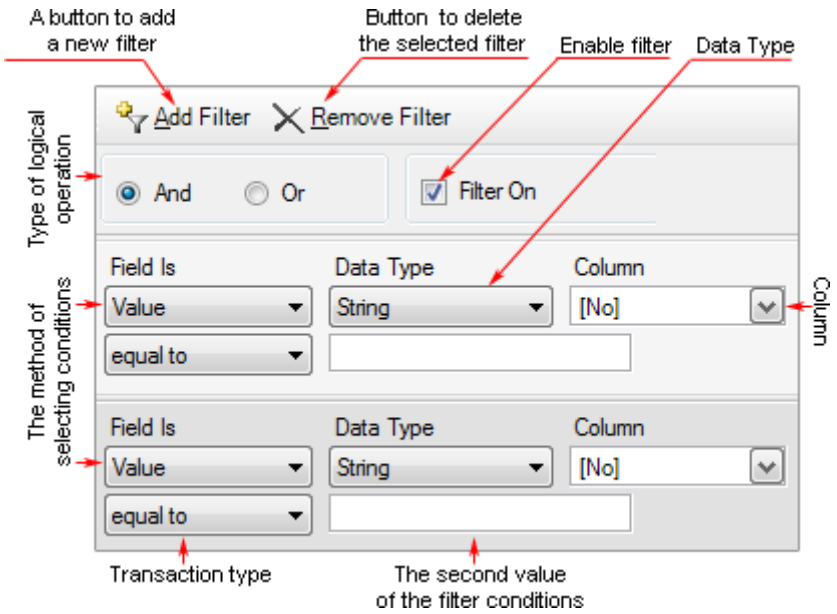
1 **Data Source**. On this step the data source is defined. This step is obligatory.
2 **Select Columns**. On this step columns of a data source are selected. This step is obligatory.
3 **Columns Order**. This step defines position of columns in the **DataBand**. Data columns selected in the second stage will be shown as a list on the **Selection Parameters Panel**. The top-down order of columns shown in the panel corresponds to their left-to-right position in a report. It is possible to change the position of data columns by dragging them or by clicking the buttons on the control panel of this step. The picture below shows the order of columns on the **Selection Parameters Panel**:



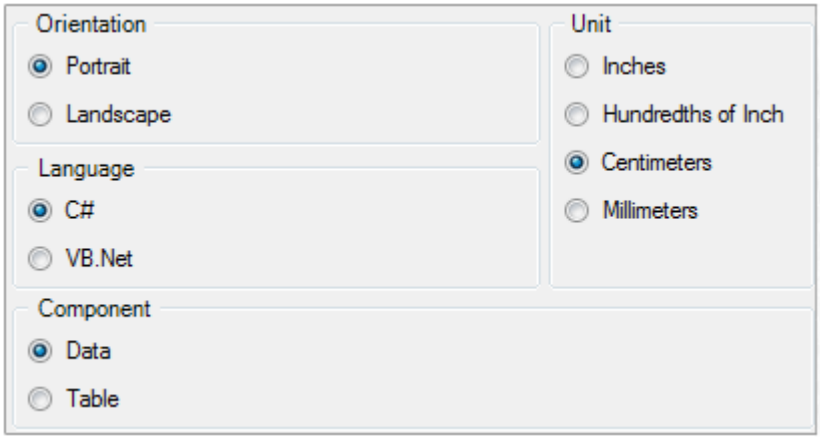
4 **Sort**. On this step, it is possible to specify elements and sorting direction. The picture below shows a sample of the **Selection Parameters Panel** of sorting:



5 **Filters**. On this step, it is possible to set the conditions of filtration. The picture below shows a sample of selection filtration parameters:



6 **Groups**. This step defines the condition of grouping. It is necessary to select a data column by what conditions of grouping will be created.
7 **Totals**. On this step, it is possible to select a function for calculating totals by any data source column. For each data column its own function of aggregation can be set.
8 **Layout**. On this step, the basic report options are set. Among them are: page **Orientation**, script **Language**, a **Component** that will be used for report rendering (DataBand or Table), report **Units**. The picture below shows a sample of the **Selection Parameters Panel** layout:



9 The **Description Panel**. Shows description for the current step.
10 The **New Data Source** button is used to create a new data source.

The **Selection Parameters Panel** shows options, actions, settings available on this step.

- buttons are used to control the **Control Panel** wizard:

The **Back** button is used to go to the previous step.

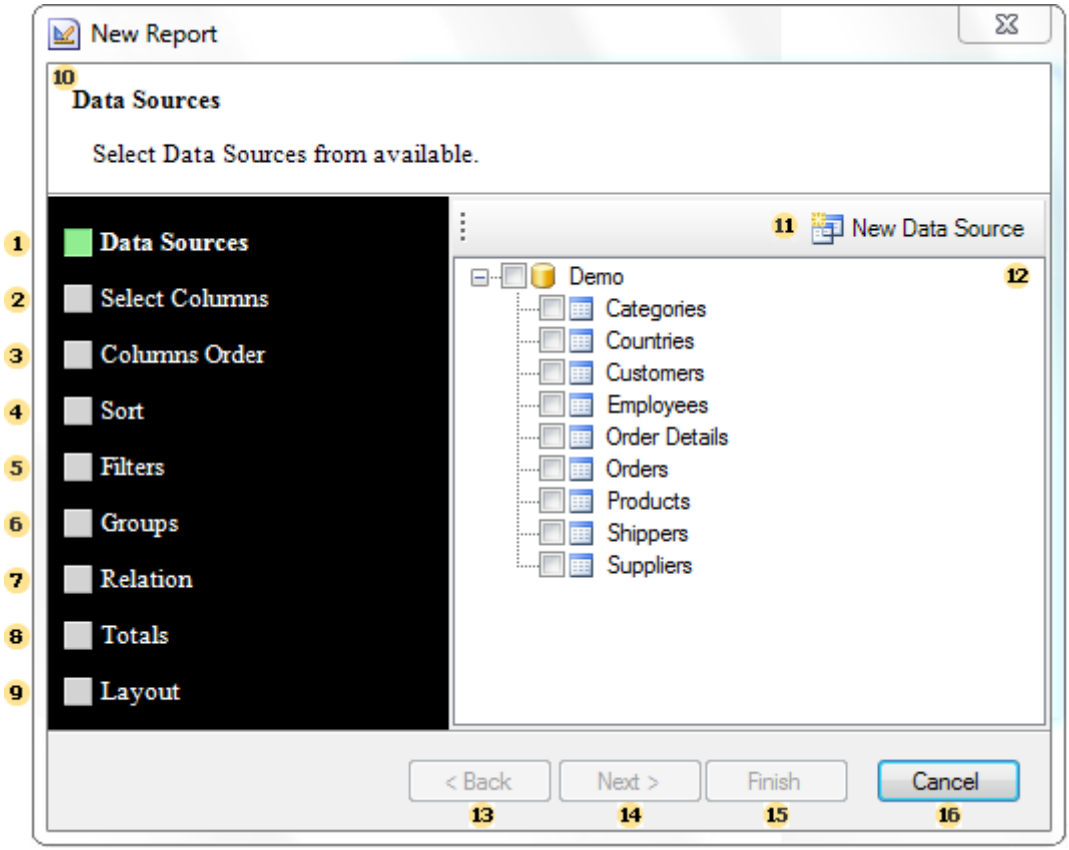
The **Next** button is used to go to the next step.

The **Finish** button is used to finalize work with the report wizard.

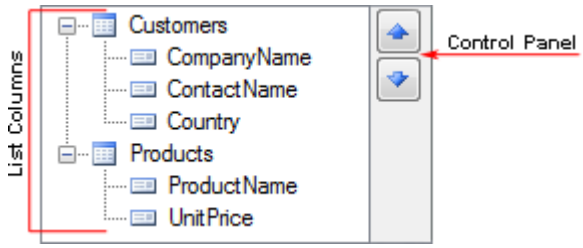
The **Cancel** button cancels the report creation with the report wizard.

Master-Detail Report Wizard

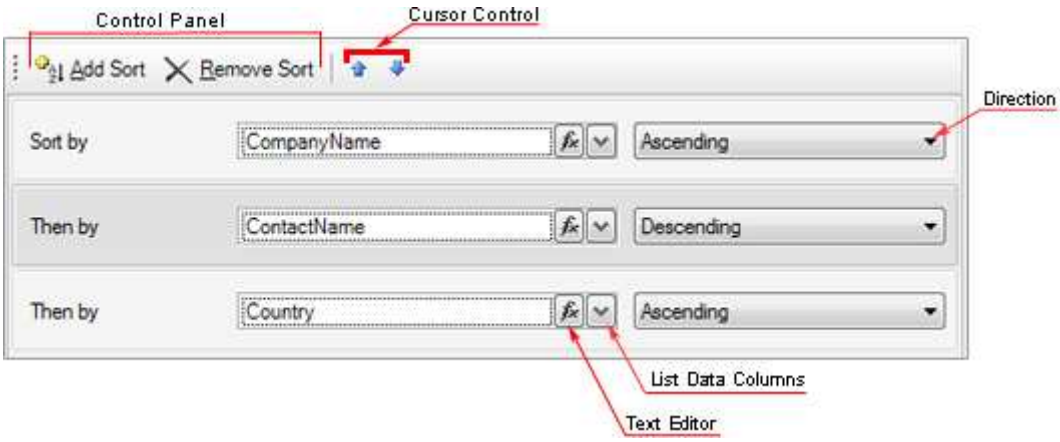
The **Master-Detail** report can be created using the **Master-Detail Report** report wizard. The picture below shows a window of the **Master-Detail Report** wizard:



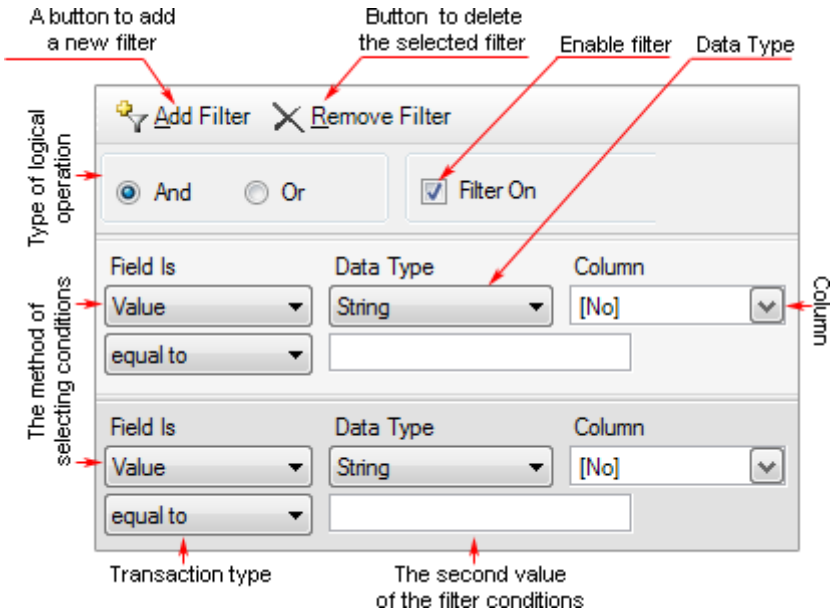
1 **Data Source**. On this step the data source is defined. This step is obligatory. For creating the **Master-Detail Report**, the report template should have no less than one **Master** band and one **Detail** band.
2 **Select Columns**. On this step columns of a data source are selected. This step is obligatory.
3 **Columns Order**. This step defines the order of columns. Data columns selected in the second stage will be shown as a list on the **Selection Parameters Panel**. The top-down order of columns shown in the panel corresponds to their left-to-right position in a report. It is possible to change the position of data columns by dragging them or by clicking the buttons on the control panel of this step. The picture below shows the order of columns on the **Selection Parameters Panel**:



4 **Sort**. On this step, it is possible to specify elements and sorting direction. The picture below shows a sample of the **Selection Parameters Panel** of sorting:



5 **Filters**. On this step, it is possible to set the conditions of filtration. The picture below shows a sample of selection filtration parameters:

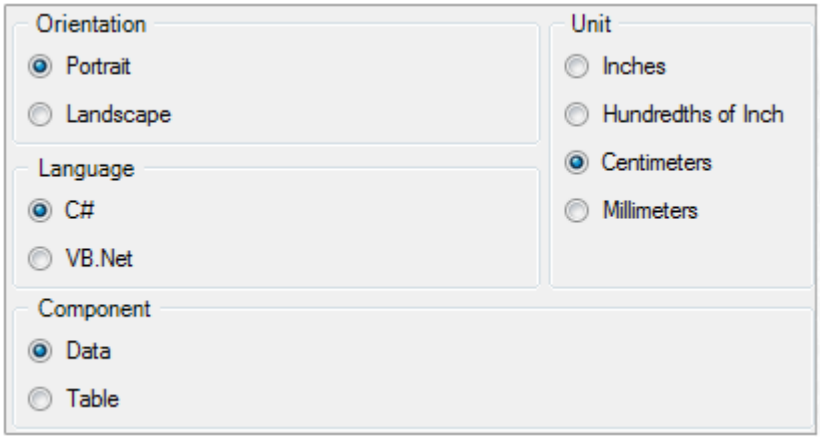


6 **Groups**. This step defines the condition of grouping. It is necessary to select a data column by what conditions of grouping will be created.

7 **Relation**. defines the relation between **Master** and **Detail** bands. The relation is used for selecting detail data only for the specified **Master** band row. If a relation will not be specified then all **Details** data rows will be output for each row of the **Master** band. Selection is done between relations which are created between **Master** and **Detail** data sources, and where a **Detail** data source is a detail data source. More than one relation can be. So it is necessary to select the correct relation.

8 **Totals**. On this step, it is possible to select a function for calculating totals by any data source column. For each data column its own function of aggregation can be set.

9 **Layout**. On this step, the basic report options are set. Among them are: page **Orientation**, script **Language**, a **Component** that will be used for report rendering (DataBand or Table), report **Units**. The picture below shows a sample of the **Selection Parameters Panel** layout:



10 The **Description Panel**. Shows description for the current step.

The **New Data Source** button is used to create a new data source.

The **Selection Parameters Panel** shows options, actions, settings available on this step.

- buttons are used to control the **Control Panel** wizard:

The **Back** button is used to go to the previous step.

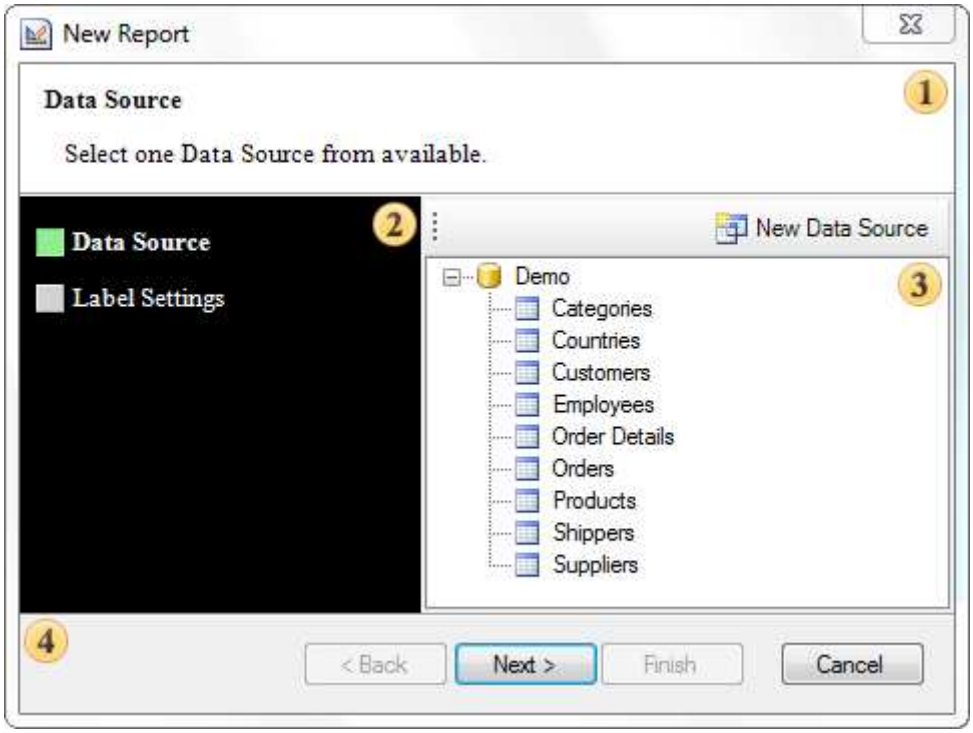
The **Next** button is used to go to the next step.

The **Finish** button is used to finalize work with the report wizard.

The **Cancel** button cancels the report creation with the report wizard.

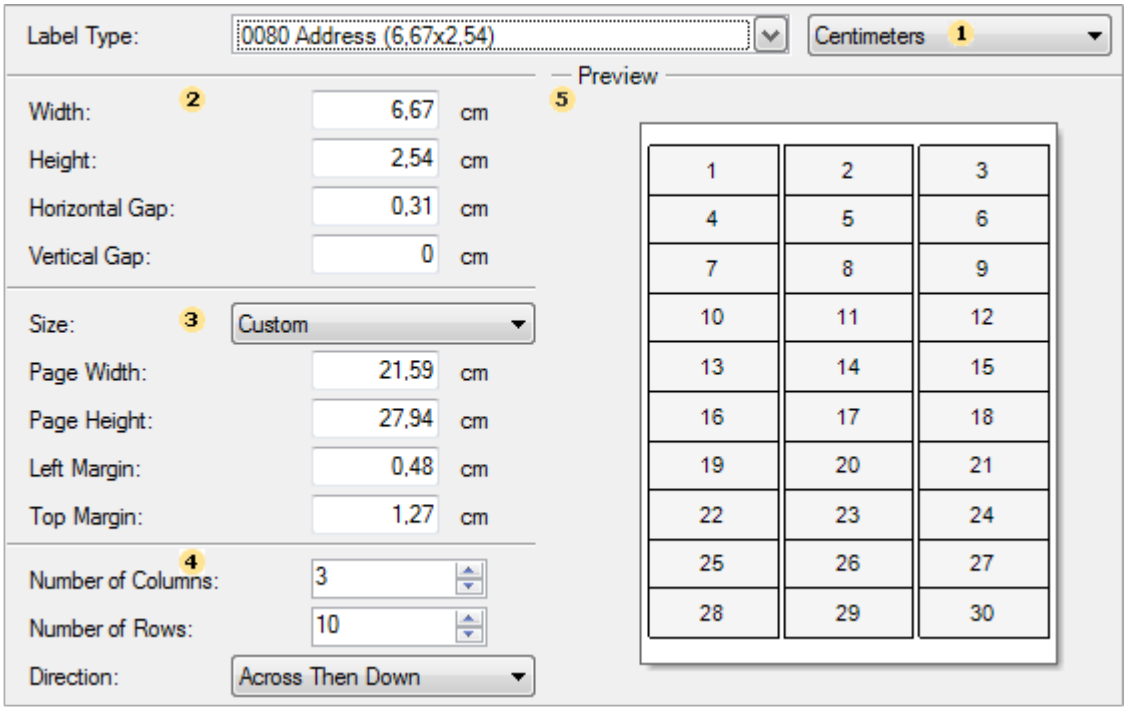
Label Report Wizard

The **Label Report** wizard is used to create reports which have labels. The picture below shows a window of the **Label Report** wizard:



1 The **Description Panel**. Shows description for the current step.
2 The **Steps Panel** shows step of report creation.
3 The **Selection Parameters Panel** shows options, actions, settings available on this step.
4 The **Control Panel**. Contains buttons to control the **Report Wizard**.

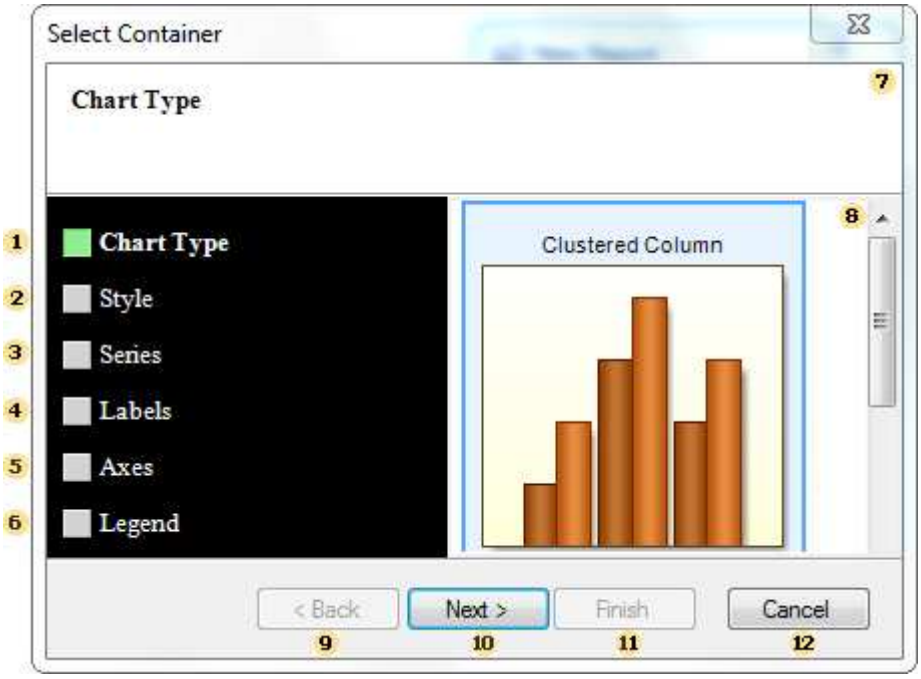
A **Label Report** is created in two steps. The **Data Source** is defined on the first step, **Label Settings** are defined on the second step. The picture below shows the **Selection Parameters Panel** on the second step of the **Label Settings:**



1 The **Type Panel** is used to select the **Label Type** and units.
2 The **Size Label Panel** is used to change the label size.
3 The **Size Pages Panel** is used to select the page size or manually set width and height and margins of a page.
4 The **Configuration Label Panel** is used to set a number of rows, columns and direction of labels.
5 The **Preview Panel** is used to preview how labels are placed on a page.

Chart Wizard

The **Chart** wizard is used to create reports with charts. The picture below shows a window of the **Chart** wizard:

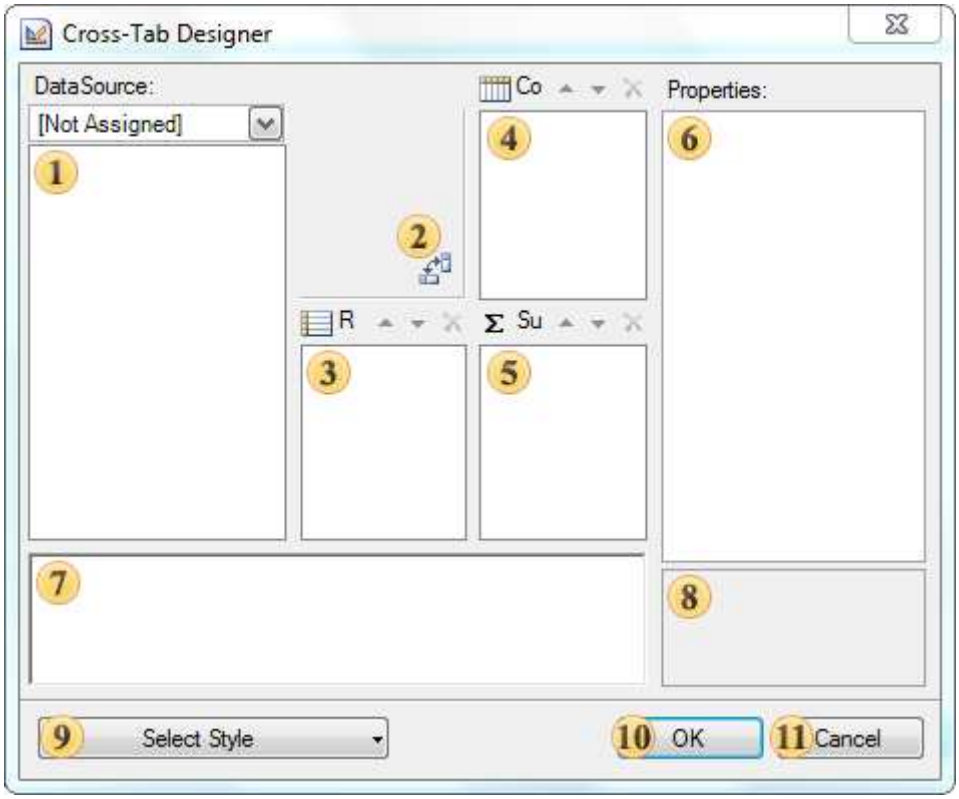


1 **Chart Type**. Select the chart type.
2 **Style**. Select the chart style from multiple templates.
3 **Series**. Add series using the series editor. Also, it is possible to specify the column of values and arguments for the data source.
4 **Labels**. The following parameters are defined on this step: series position, **Value Type** of series, **Text before/after** the series, and an **Angle** of inclination.
5 **Axes**. This step is available only if selected chart type is in **Axes Area**. The following options are set on this step: axis **Title** and its **Alignment**, **Ticks** length and their **Visibility**, **Grid Lines** and its **Interlaced**, **Labels** and their **Visible** property. Also, a chart can be shown vertically or horizontally. The Reverse property for X or Y axis should be applied for this.
6 **Legend**. On this step legend parameters and charts such as **Title**, legend **Alignment** horizontally and vertically, **Direction** of rows in legend, **Visible** and **Size** of a marker, **Spacing**, **Visible** of the legend.
7 The **Description Panel**. Shows description for the current step.
8 The **Selection Parameters Panel** shows options, actions, settings available on this step.

9 - buttons are used to control the **Control Panel** wizard:
9 The **Back** button is used to go to the previous step.
10 The **Next** button is used to go to the next step.
The **Finish** button is used to finalize work with the report wizard.
The **Cancel** button cancels the report creation with the report wizard.

Cross-Tab Wizard

The **Cross-Tab** wizard is used to create reports with cross-tab. The picture below shows the window of the **Cross-Tab** wizard:



1 **Data Source Panel**. In the **Data Source** field it is necessary to select the data source. Then data source columns will be shown on the panel of the data source.
2 The **Swap Rows/Columns** button is used to change data between columns, which are placed on the **Rows** and **Columns** panels.
3 The **Rows** panel shows data source columns, which are rows of a cross table.
4 The **Columns** panels shows data source columns, which are columns of a cross table.
5 The **Summary** shows data source columns, which are the key column and row in the cross table. Key column and row generate summary cell.
6 The **Properties** panel shows a table of properties of selected column of the data source.
7 The **Preview Panel** is used to preview the template of a cross table.
8 The **Property description panel** shows a description of the selected property.
9 The **Select Style** button is used to select style of the cross table appearance.
10 The **Ok** button. Click on the button to finish work with the wizard and either a cross table will be added on a page or cross table options of an existed table will be applied.

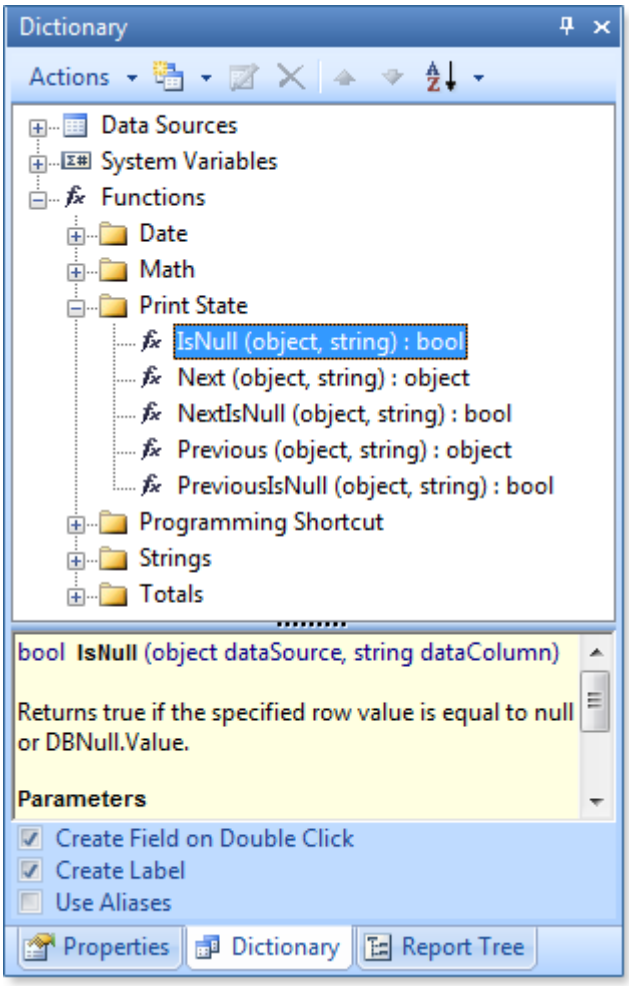
The **Cancel** button cancels the report creation with the report wizard.

Panels

In this section the basic panels of the designer, such as: **Dictionary**, **Messages**, **Report Tree**, **Properties** will be reviewed. These panels contain different properties, functions and commands to control various components of a report, information fields arranged for notification and hints for a user. These panels can be shown or hidden. In the **Ribbon** UI showing or hiding panels can be done using the **Panels** button on the **View** tab. In the **Standard** UI showing or hiding panels can be done in the**View** tab of the **Main Menu**.

Dictionary

The **Dictionary** panel shows the connected data available data sources, system variables and functions. Besides, a connection and connecting new data sources can be done in the **Dictionary**. The picture below shows an example of the **Dictionary**:

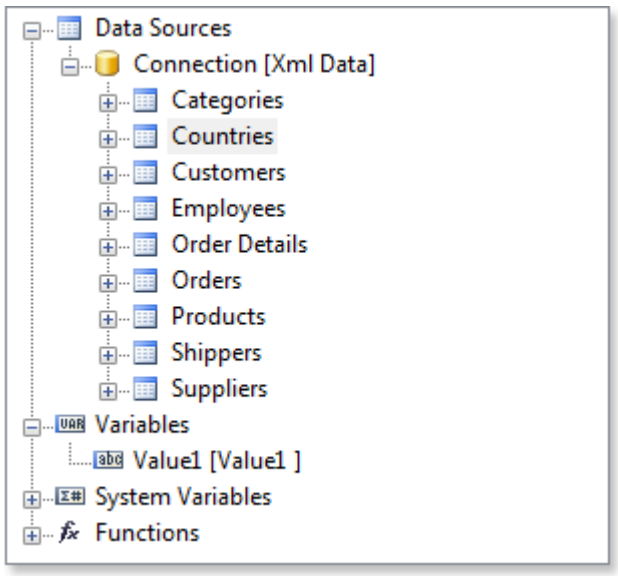


The **Dictionary** panel includes: **Dictionary ToolBar**, **Data Tree**, **Description Panel**, **Dictionary Setting Panel**.

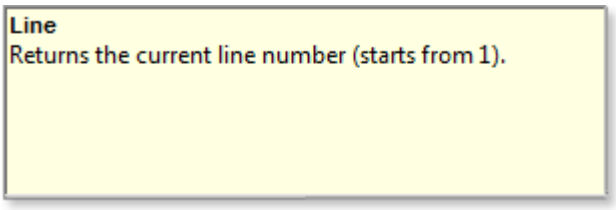
1. The **Dictionary ToolBar** is a set of tools and commands to work with the **Dictionary**. The picture below shows the **Dictionary ToolBar**:

1 The **Actions** menu is a set of commands to work with vocabulary such as: creating, opening, saving a dictionary, adding, importing and exporting a dictionary to an XML schema, the synchronization of data presented in the dictionary with the data registered in a data store.
2 The **New Item** menu contains commands for creating new elements: new connection, new data source, new connection, new variable, business objects.
3 The **Edit** button is used to edit created elements. Pressing the button runs the editor of the element.
4 The **Delete** button deletes created elements.
5 The Up and Down buttons move the selected item in the hierarchy of a dictionary within a single level of a tree.
6 The **Sort Items** menu provides the opportunity to choose one of two directions of Sorting: **Ascending** from A to Z, **Descending** from Z to A. And also to enable or disable the **Auto Sort** mode.

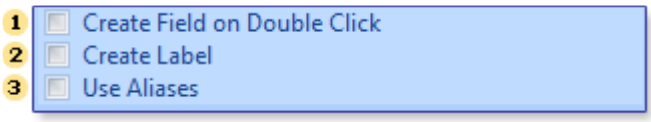
2. The **Data Tree** represents a list of all data of a dictionary, which are displayed in a tree. The picture below shows an example of the **Data Tree**:



3. The **Description Panel** displays a short description of the selected system variable or function. The picture below shows an example of the **Description Panel** with the description of the **Line** system variable:



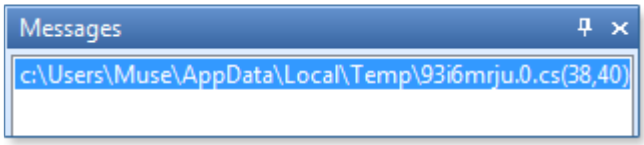
4. The **Dictionary Setting Panel** is a panel with three options to optimize the work with the dictionary and its contents.



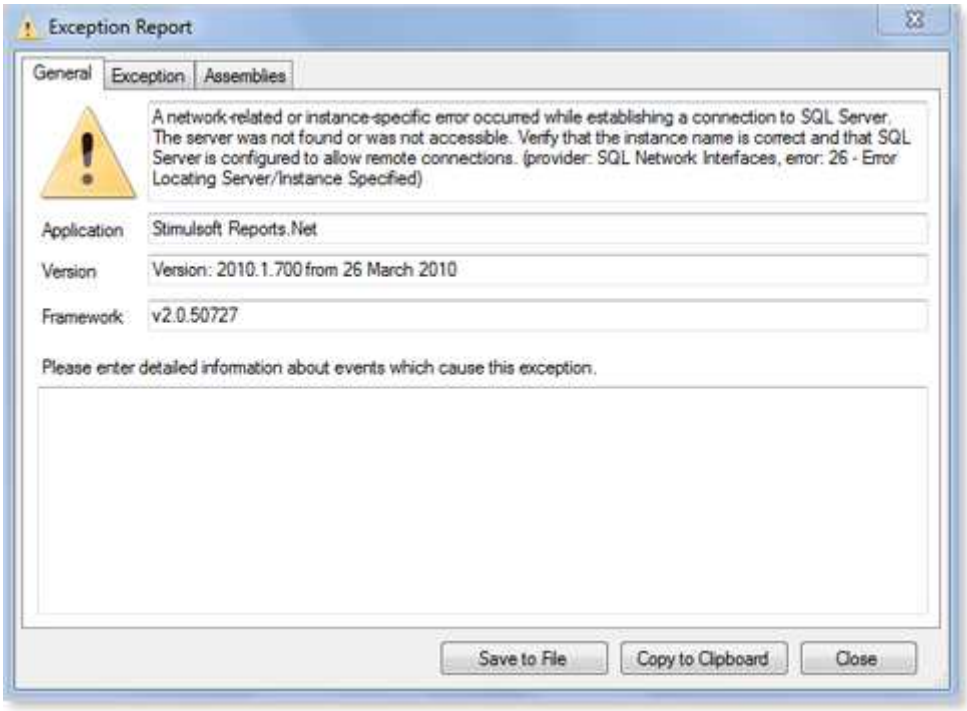
1 The **Create Field on Double Click** option provides the ability to create a field on the **DataBand**. Fields are created on the band, that has selected data source by which the fields the double-click in the dictionary is done.
2 The **Create Label** option attaches the column data header when it is dragged on the report template.
3 The **Use Aliases** option provides an opportunity to show **Aliases** of components instead of the **Name** in the report template.

Messages

The **Messages** panel is located at the bottom of the designer (above the status bar) and may be rolled out, and can remain to be deployed. This panel is required to notify a user about errors when compiling or when rendering a report. If, for example, when compiling, the syntax error was detected, the user will be sent to the Message. The picture below shows an example of the syntax error message:

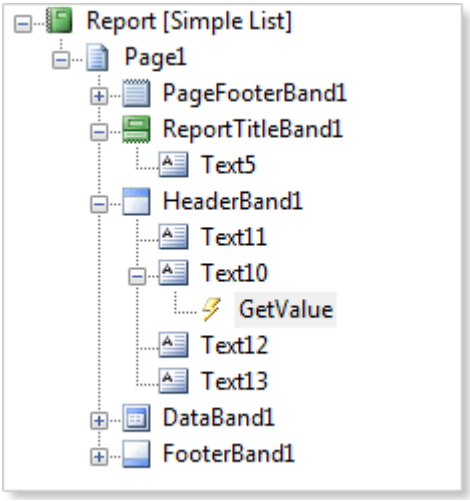


If, for example, when rendering a report a critical error occurs, the user will be notified with the message in a separate window. The picture below shows an example of a critical error message when rendering a report:



Report Tree

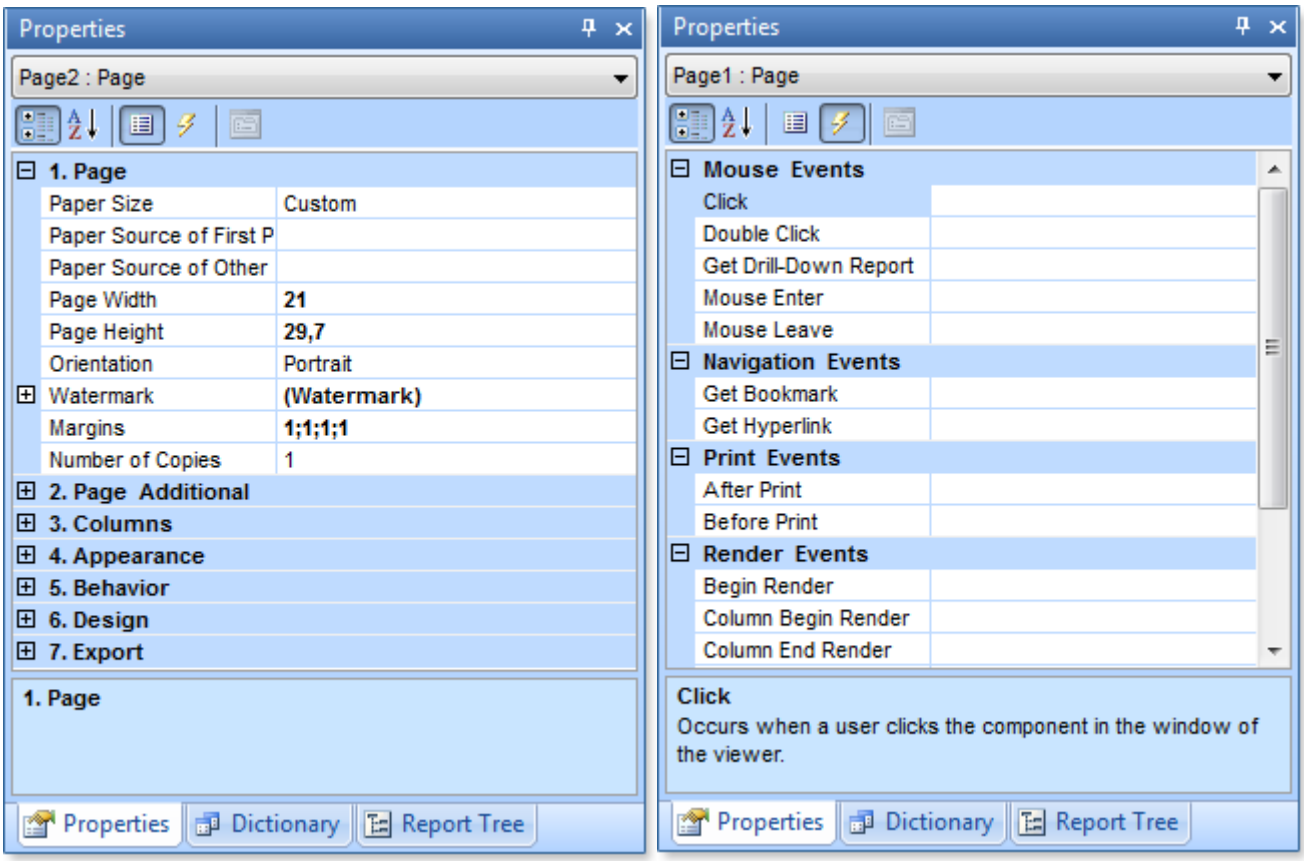
The **Report Tree** panel shows the hierarchy of the report, i.e. represents all the components of the report in the form of a tree. In addition, if an event handler is added to the component, it will also be displayed in the hierarchy of the report. The picture below shows an example of the **Report Tree** panel:



As can be seen on the picture above, hierarchy is represented on the principle of "nesting", and an event handler is added for the **GetValue** event of the **Text10** component. The **Report Tree** panel provides the ability to visually identify the submission of a "component to a component".

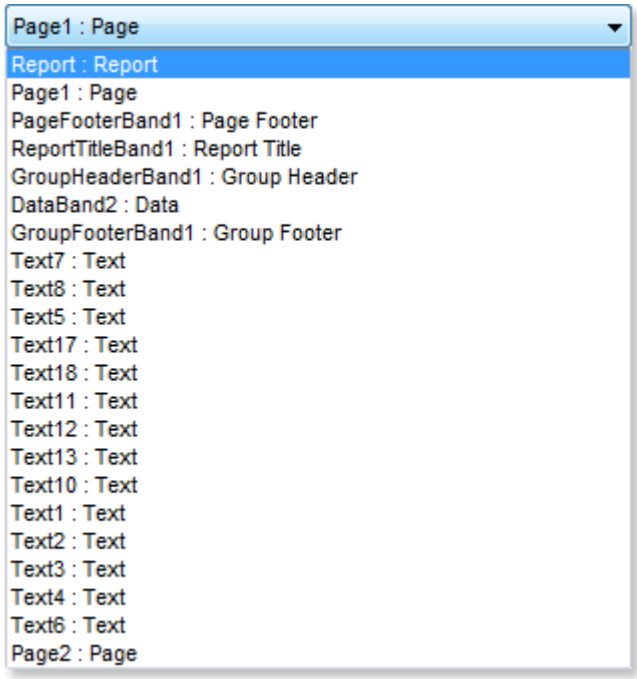
Properties

The **Properties** panel shows all the properties of the selected component, and also its events. The picture below shows the **Properties** panel, displaying the properties of a component (left) and an event of a component (right):



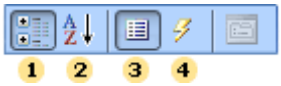
The **Properties** panel includes: **Drop-Down List of Components**, **Properties ToolBar**, **Properties** or **Events Table**, **Description Panel**.

1. The **Drop-Down List of Components** displays a list of all the components of a report. The picture below shows an example of the **Drop-Down List of Components**:



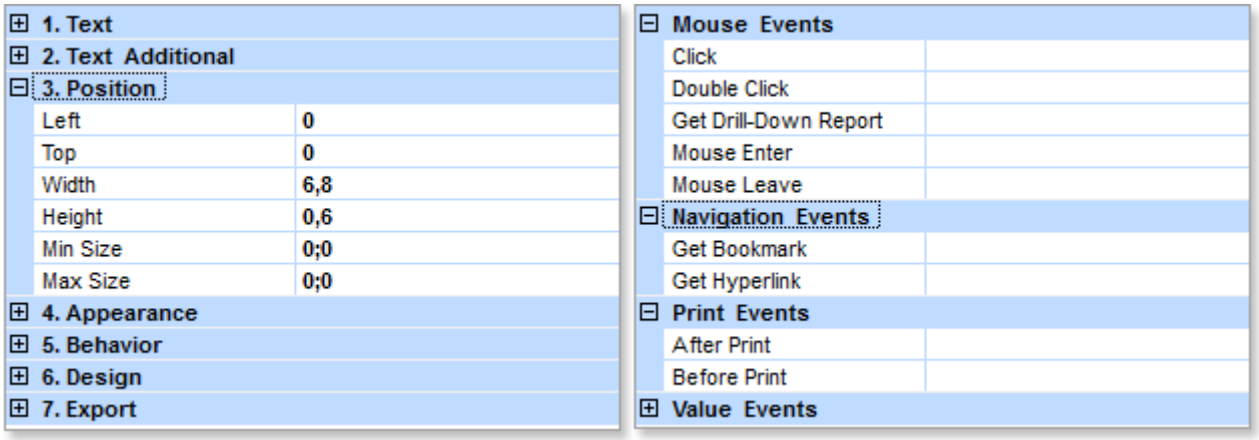
As can be seen on the picture above, the list is presented in two columns. The first column displays the **Name** of a component and the second one shows its type. For example, the string "**Text7: Text", "Text7**" is a **Name**, "**Text**" this is a type. If to select a component in this list then, on the Properties panel, properties and events of the selected component will be shown.

2. The **Properties ToolBar** is designed to control the **Properties** panel. The picture below shows the **Properties ToolBar**:

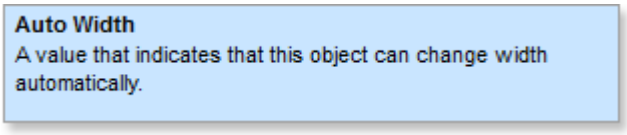


1. The **Categorized** sorting button is used to show a list of properties or events sorted by category.
2. The **Alphabetical** sorting button is used to show a list of properties or events are sorted alphabetically from A to Z.
3. The button for enabling the **Properties Tab**.
4. The button for enabling the **Events Tab**.

3. The **Properties** or **Events Tab** is table with two columns. The first column shows names of properties or events. The second column shows values of these properties or events. The number of rows depends on the number of properties or events, because one property or event takes a single row. The picture below presents a table of properties (left) and a table of event (right):



4. The **Description Panel** displays a short description of the selected property or event. The description can be hidden by right-clicking the **Properties** panel and clearing the check of the **Show Description** command in the context menu of the **Properties** panel. The picture below presents the **Description Panel**:



The **Localize Property Grid** present in the context menu of the **Properties** panel and can be called by right-clicking on this panel. If this command is enabled (checked), then the translation of the **Properties** panel will be implemented. If this command is disabled (not checked), then the names of properties, events, values and description of the **Properties** panel will be in English.

Keyboard Shortcuts in Designer

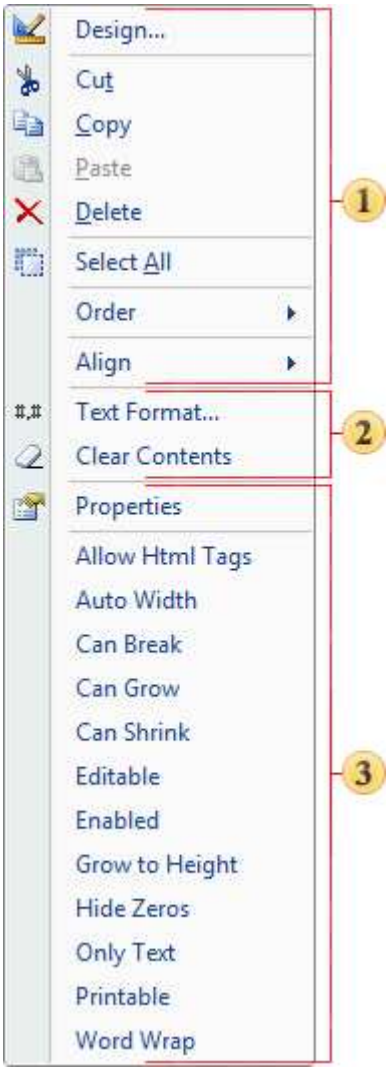
The reports designer supports many keyboard shortcuts. Using them can speed up the effectiveness of work in creating reports. Some keyboard shortcuts are available both in the Ribbon mode and in the Standard mode. Some of them are available only in the Standard mode (in the Ribbon mode they are duplicated by context commands).

Ctrl+B	Makes letters **bold** for the selected text component
Ctrl+I	Makes letters *Italic* for the selected text component
Ctrl+U	Makes letters Underlined for the selected text component
Ctrl+"+"	Increase the font size for the selected component
Ctrl+"-"	Decrease the font size for the selected component
Ctrl+L	Align selection or paragraph to the left
Ctrl+E	Align selection or paragraph to the centre
Ctrl+R	Align selection or paragraph to the right
Ctrl+J	Justify selection
Ctrl+Shift+D	Activate the **"Dictionary"** panel
Ctrl+Shift+M	Activate the **"Messages"** panel
Ctrl+Shift+L	Activate the **"Report Tree"** panel
F4 **Shift+Enter**	Activate the **"Properties"** panel
Ctrl+C **Ctrl+Insert**	Copy the selected text or object
Delete **Ctrl+Delete**	Delete the selected component
Ctrl+V **Shift+Insert**	Paste the text or object from the Clipboard
Ctrl+X **Shift+Delete**	Cut the selected text or object
Ctrl+A	Select All
Ctrl+Z	Undo
Ctrl+Y	Redo
Ctrl+F2	Show **"Data Store"**
Ctrl+F4	Show **"Page Manager"**
Ctrl+F5	Show **"Services Configurator"**
Ctrl+N	Create a new report
Ctrl+Shift+N	Add a page to the report
Ctrl+Shift+F	Add a form to the report
Ctrl+O	Load a report from the file
Ctrl+Shift+O	Load a page from the file
Ctrl+S	Save a report
Ctrl+F12	Save a report as...
F5	Report preview
Ctrl+Enter	Call the designer be default for the elected component
Enter	Call the text editor for the selected component

F1	Select the **“Select”** tool
F2	Select the **“Hand”** tool
F3	Select the **“Text Editor”** tool
F6	Select the **“Copy Style”** tool
Cursor keys	Move selection
Shift+Cursor keys	Resize selected component (one step = grid size)
Shift+Alt+Cursor keys	Resize selected component (one step = half grid size)
Ctrl+Cursor keys	Move selected component (one step = grid size)
Ctrl+Alt+Cursor keys	Move selected component (one step = half grid size)
Ctrl+Drag mouse	Copy selected components
Alt+Drag mouse	Ignore **“Align to Grid”** when moving and resizing

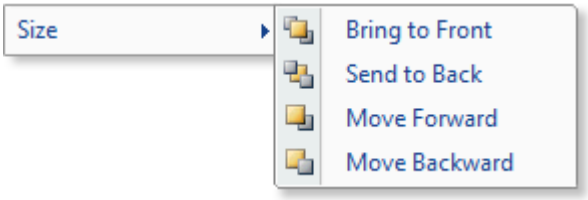
Context Menu

Context Menu is a menu in a graphical user interface that appears upon user interaction (a right mouse click). A context menu offers a set of choices that are available in the current state of the component. The picture below shows a context menu of the text component:

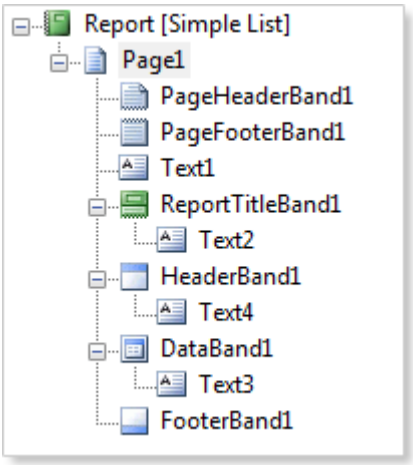


The context menu consists of three groups:

1 General commands. These are static commands, which can be applied to any component of the designer:
1) The **Design...** command invokes the editor of a selected component. For example, if it is a text component then the **Text Editor** will be called.
2) The **Cut** command cuts the selected component to the clipboard.
3) The **Copy** command copies the selected component to the clipboard.
4) The **Paste** command pastes from the copied or cut component from the clipboard.
5) The **Delete** command deletes the selected component.
6) When selecting two or more components, the **Size** command appears in the context menu. This command contains submenu in what it is possible to define the size parameters for all selected components.

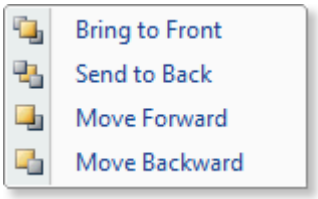


If you select all the components on the page, i.e. choose the **Select all** command in the context menu, or press the **Ctrl+A** key combination, then the prototype of the size for all components will be the size of the component that is placed on a higher level and higher than other components in the report tree in that level. It is possible to see the report tree on the **Report Tree** panel. The picture below shows an example of a report tree.

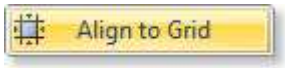


As seen on the picture above, the **PageHeaderBand** is located higher, so, in this case, its size is a sample of sizes for the remaining components. If the **PageHeaderBand** is absent, then, as a sample of sizes, the **PageFooterBand** will be taken. If the **PageFooterBand** is absent, then, as a sample of sizes, the **Text1** will be taken.

7) The **Select All** command selects all components on the current page.
8) The **Order** command invokes the submenu, in what it is possible to define the position of the selected component. The picture below shows the **Order** submenu:



9) The **Align** command invokes the submenu, in what it is possible to select the **Align to Grid** command. The picture below shows the **Align** submenu:



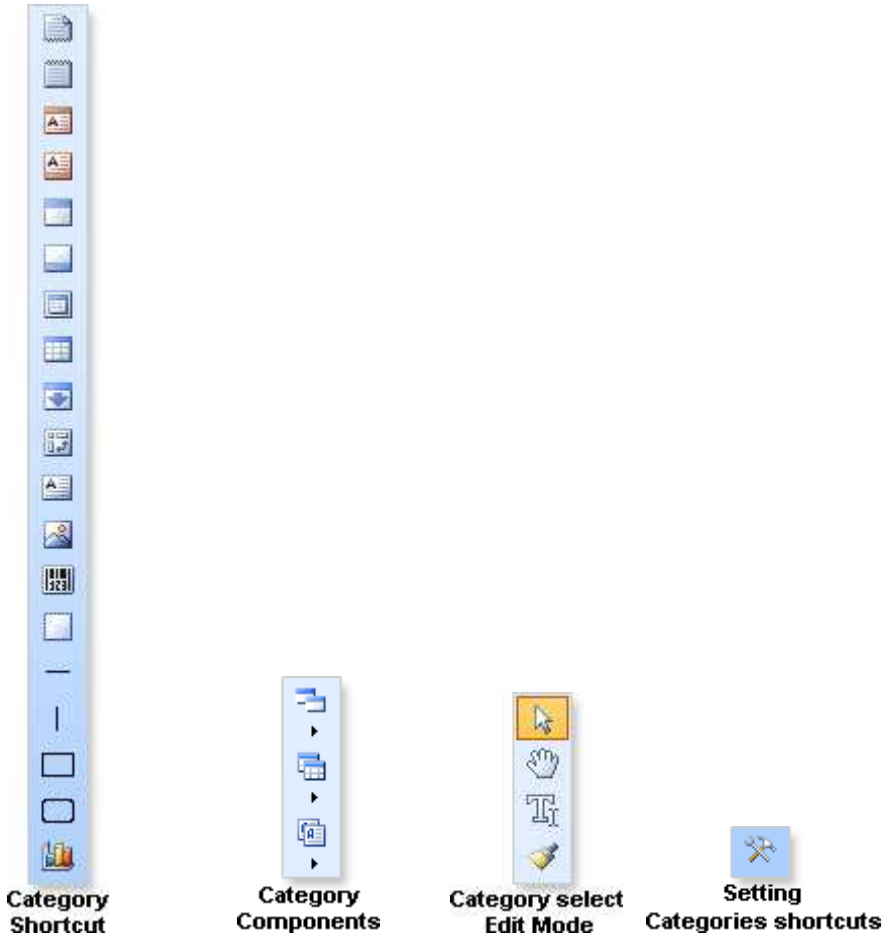
2 Commands which are specific for the component.
3 It is possible to enable/disable different properties, without closing the context menu of the selected component.

Previewing Reports in Designer

Report Designer allows previewing a report before printing, exporting, sending via e-mail or any other action, in order to identify possible errors. Clicking the **Preview** or **HTML Preview** tabs it is possible to preview a report. You can also preview the report in the separate window by using the F5 shortcut key or selecting **Preview** from the main menu.

Toolbox

The **ToolBox** panel contains the main tools for creating reports. All items on this panel are divided into the following categories: **Select Edit Mode Category**, **Components Category**, **Shortcut Category**, **Shortcut Category Settings**. The picture below shows the **ToolBox**:



The Toolbox is located on the left side in the designer window and looks like a vertical panel. If necessary, the Toolbox bar can be shown or hidden. In Ribbon UI hiding or showing the **Toolbox** can be done by pressing the **Toolbox** button, which is located in the **View** tab. If the button is pressed, the **Toolbox** is shown. In Standard UI hiding or showing the **Toolbox** can be done by right-clicking and calling the context menu of the Toolbox or the Main Menu.

Shortcut Panel

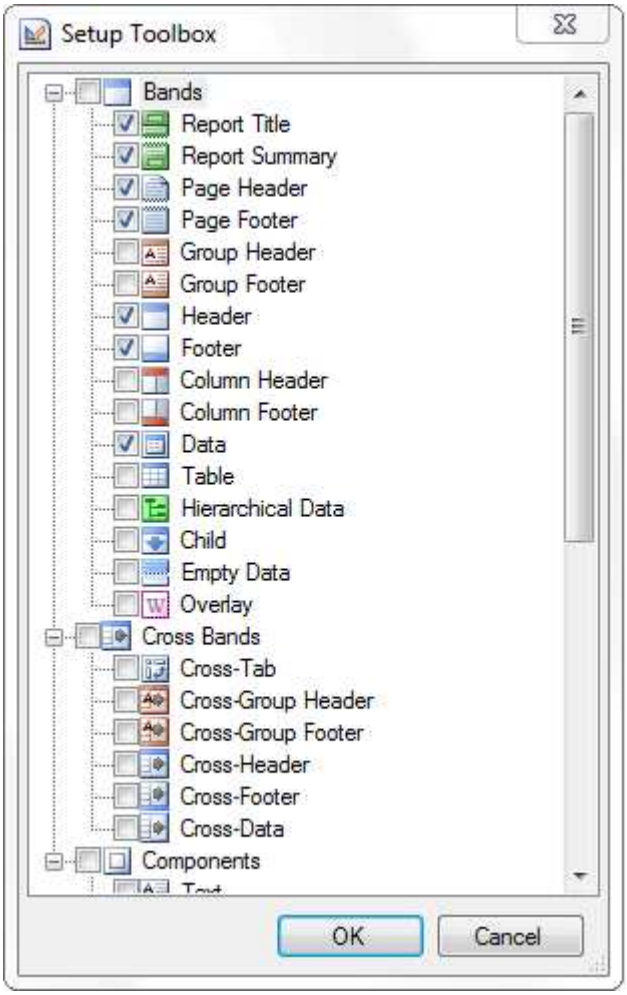
The shortcut category is one of the basic panels of the **Toolbox** and is designed for quick selection of the component when creating a report template. There are some ways to add components on a page:
1. Drag and Drop. To do this, put the cursor on the component, left-click and drag the component on the page of a report template.
2. Select the required component, and then draw it on a report page.
To draw several components of one type, hold down the **Shift** key, select the components from the shortcut category. Once the component is selected, the **Shift** key can be released. Now you can draw components unlimited number of times. The picture below shows an example of the shortcut category:

Category
Shortcut

In addition, the **Toolbox** has the **Setup Toolbox** button.

Setting
Categories shortcuts

This button invokes the **Setup Toolbox** window, which is necessary to check the elements that will appear on the toolbox panel in the shortcut category. The picture shows an example of the **Setup Toolbox** window:



Components which are marked with the "check" in the **Setup Toolbox** window will appear on tools in the shortcut category. Accordingly, the components that are not marked will not be displayed.

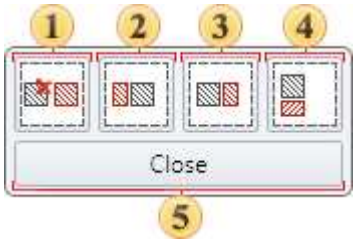
Components Placement Wizard

When you drag components from the dictionary, toolbar, or any other container into bands of a report template, and margins of a component are beyond the margins of a band, then the **Components Placement Wizard** will be invoked. With this wizard you can define the parameters of the location of the component in the band. The picture below shows the structure of the **Components Placement Wizard**:

1. Move a component to the right side of a free space, stretching the component by the height of the free space.
2. Move a component to the left side of a free space, stretching the component by the height of the free space.
3. Close the **Components Placement Wizard**.

Drag and Drop Wizard

When dragging a text component with an expression from the Dictionary and overlapping on another text component, the **Drag and Drop Wizard** will be invoked. Using this wizard it is possible to choose the way of merging the contents of the components. The picture below show how the **Drag and Drop Wizard** looks like:



1 Substitute an expression in the text component which is placed in the report template, for an expression of the text component which is being dragged;
2 Place an expression from the dragged component before an expression of the text component which is placed in the report template;
3 Place an expression from the dragged component after an expression of the text component which is placed in the report template;
4 Place an expression from the dragged component one row below an expression of the text component which is placed in the report template;
5 Close the wizard.