Table of Contents

Part IV Dashboards   862

Part V Data   1010

1 Introduction

Welcome to the online version of the documentation Stimulsoft Reports. This part of documentation describes basic approaches in work with GUI. Here you can find the answers on all your questions, find out about main abilities, master the skills of working with the report designer, viewer and other utilities. Also you will learn how to use the components for designing reports, formatting, filtering, and creating styles for reports.

❯ Welcome to **Stimulsoft**:
- *i* Technical Support
- *i* Trial License Limitations
- *i* Information in Web and References
- *i* Evaluate Demo Version
- *i* Features - Stimulsoft Reports Product Line
- *i* Features - Stimulsoft Reports.Fx Product Line

❯ Report Designer:
- *i* Ribbon UI 2013
- *i* Creating Reports in Designer
- *i* Keyboard Shortcuts
- *i* Previewing Reports
- *i* More...

❯ Learn more about the basic tools and reporting components:
- *i* Data Band
- *i* Expressions
- *i* Groups
- *i* Creating Lists
- *i* Appearance
- *i* More...

❯ Reports Viewer
- *i* Report Viewer Structure
- *i* Basic Toolbar of Report Viewer
- *i* Dot-Matrix viewer
- *i* Special Viewing Options in Web
- *i* More...

❯ Read the basics of creating reports:
- *i* Simple List Report
- *i* Master-Detail Report
- *i* Report with Chart
- *i* Report with Cross-Tab
- *i* Anchors in Report
- *i* Invoice Report
- *i* More...

❯ Right to Left Mode:
- *i* Text Component
- *i* Columns on Page
- *i* Cross Table Component
- *i* Columns in Data Band
- *i* More...

❯ Work with Data:

❯ Reports:

The second part of the manual contains the description of non-visual parts of products Stimulsoft.

1.1   Activation
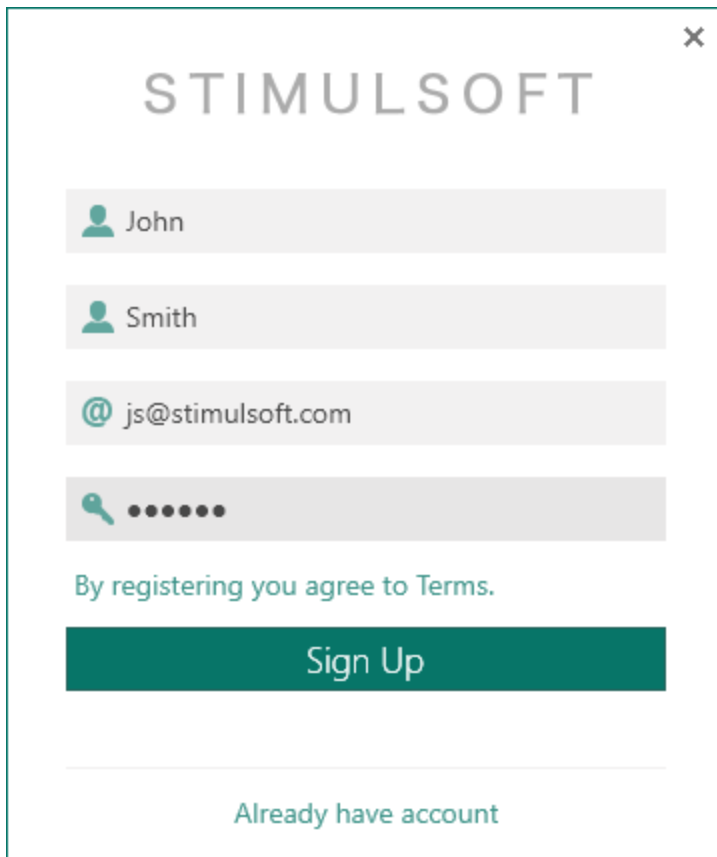
Trial version
The free trial version of Stimulsoft Reports is a full-featured version and has no time limit for evaluation. The only restriction is the **Trial** watermark that is printed on every report page. The trial version can be used with or without a registered user account. Registration of the account is free.
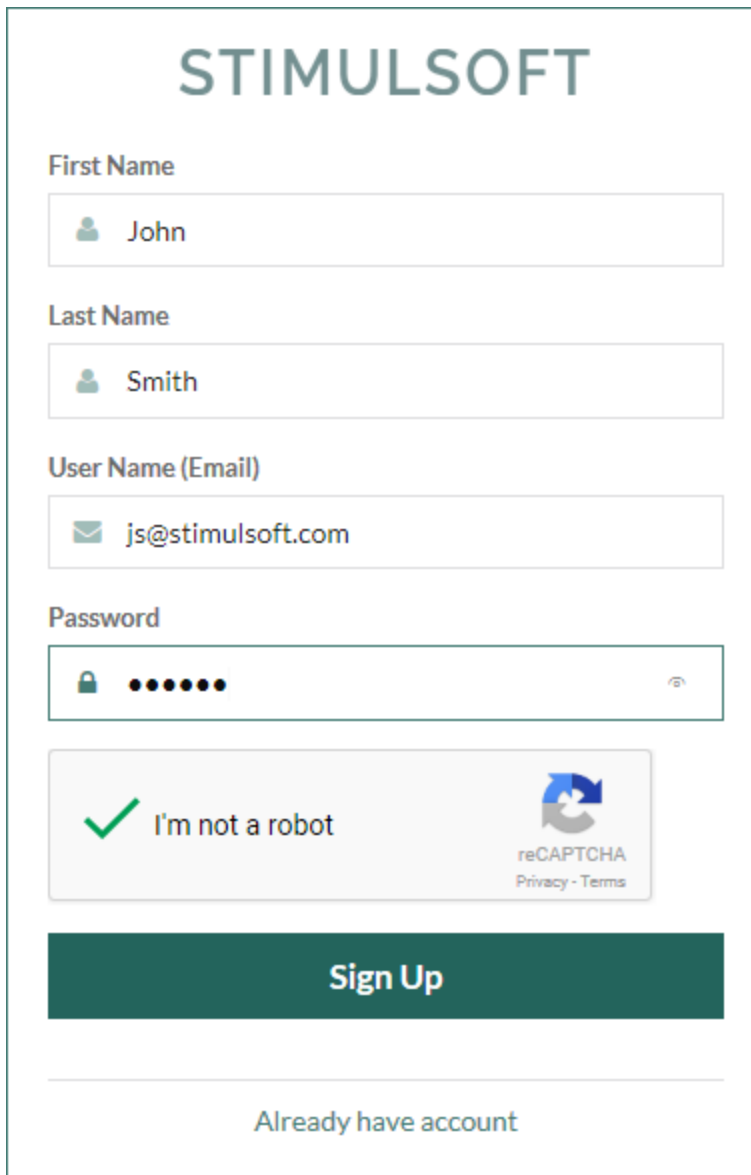
User account registration
Registration of the account is absolutely free and can be performed in the following ways:

❯ In the report designer, in the **Login** window, select Create account for free. Then fill in the first name, last name, e-mail address, password. Please note that by registering a free account you accept and agree to the terms of the license agreement.

❯ Go to the Stimulsoft website. Click the **Sign Up** button. Then fill in the first name, second name, e-mail address, and password. Please note that by registering a free account you accept and agree to the terms of the license agreement.

Licensed version

Use an account for authentication in standalone report designers. If after authentication, the **Trial** watermark is output on report pages, you probably do not have a license for this product. In this case, you can do the following:

❯ In the report designer, in the account menu, select **Subscription**. You will see the list of products that you are using and the current status of the license. Select the product, license and click on the price. You will be redirected to the purchase page. Select the payment method and follow the steps to finish the purchase.

❯ Login to your account, click the **Purchase** license button. You will be redirected to the

purchase page of the product.

Activating the application
If you embed Stimulsoft products into your application, then the activation of products can be done in the following ways:

❯ Activation using code. In this case, in your user account you should copy the license code and embed it in the code of your application.
❯ Activation using the license key file. In this case, you need to download the license file, place it in the Designer folder of the application and it will be automatically read by the designer.

In case of any other situations or if you have any questions, please contact us.

❯ **Phone**:
+1-314-266-3473
+44-20-3287-5081
+372-712-2227
+375-29-2125706
+7-499-404-2610

❯ **Email**:
info@stimulsoft.com
sales@stimulsoft.com
support@stimulsoft.com

❯ **Telegram**:
+375-29-7175260

❯ **Skype**:
stimulsoft

1.2 Technical Support

Registered users and users who are evaluating the software may get technical support.

For technical questions, use the Email address: support@stimulsoft.com
For licensing, subscription, payment questions, use Email address:
sales@stimulsoft.com
For other questions, use Email address: info@stimulsoft.com

If you have problems with our products, you may contact us through our feedback
form at http://www.stimulsoft.com/support.aspx
It is possible to send questions from the standard UI of the report designer. To do this,
select the Help menu -> Support.
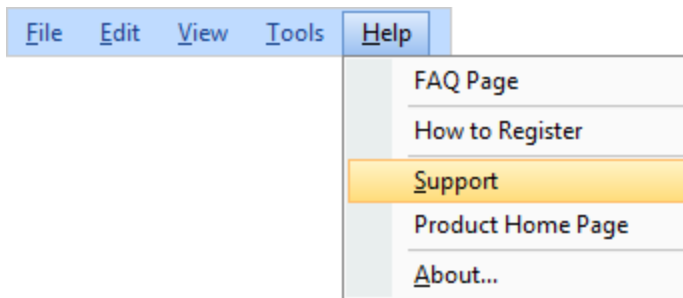
If you are a registered user and you contact us for technical support, use the same Email address you used when you purchased our product. Otherwise, it will be difficult to identify you as a registered user. This can slow down our response. Please let us know when your Email address changes.

To solve your problem quickly, we need the following information:

> Product name and its version;
> A detailed description of the problem and how to reproduce it;
> Your operating system (98, ME, 2000, XP, Vista, Window 7 etc.), its version, and the localization of established service packs;
> Version of Microsoft .NET Framework or other development environment and installed service packs;
> A name of your development environment and its version;
> Additional information that can help us solve the problem.

1.3   Web Links and Online Resources

This section describes how to get information about the latest news and announcements of software products, as well as information about known issues and questions that users are interested in.

> The official website of the company is available at https://stimulsoft.com
  i Description of products can be found at https://stimulsoft.com/en/products
  i Download the latest version of the product by the link https://stimulsoft.com/en/downloads
  i You can read the latest news of the company at https://stimulsoft.com/en/blog/news

> In addition, you can download packages of products Stimulsoft from other resources:

  i Reports.Web, Reports.Net, Reports.Wpf, Reports.UWP, Reports.Web for .NET Core from NuGet at https://www.nuget.org/profiles/Stimulsoft
  i Reports.Java from Maven at http://central.maven.org/maven2/com/stimulsoft

i Reports.JS from npm at https://www.npmjs.com/package/stimulsoft-reports-js

❯ You can evaluate our reporting tools online:
- *i* .NET, JS, Java reporting - https://demo.stimulsoft.com
- *i* Silverlight reporting - https://sl.stimulsoft.com
- *i* Silverlight (client/server) reporting - https://websl.stimulsoft.com
- *i* Flex reporting - https://flex.stimulsoft.com

❯ To create, store and then deploy reports in your applications, use the cloud service of Stimulsoft https://cloud.stimulsoft.com

❯ A huge number of video lessons are available:
- *i* on our YouTube cannel https://www.youtube.com/user/StimulsoftVideos
- *i* on our website https://www.stimulsoft.com/en/videos

❯ You can use samples for various platforms:
- *i* on GitHub at https://github.com/stimulsoft
- *i* on our website at https://www.stimulsoft.com/en/samples

❯ Find us is social networks and messengers:

https://twitter.com/stimulsoft

https://www.linkedin.com/company/stimulsoft

https://www.facebook.com/Stimulsoft

https://www.stimulsoft.com/en/rss

https://t.me/stimulsoft

+375-29-2125706

WhatsApp - Stimulsoft

Skype - Stimulsoft

Also visit our Forum to communicate with other users of Stimulsoft Reports - http://forum.stimulsoft.com/index.php

Here you can read and discuss various topics related to tools for creating reports. For more information about the product in other Internet resources, please use the search engines.

2   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.

2.1   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:

Data Source	Customers	...

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:
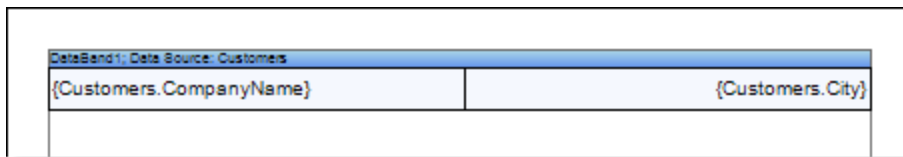
DataBand1: Data Source: Customers	
{Customers.CompanyName}	{Customers.City}

8. Click the **Preview** button or invoke the **Viewer**, 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:**

ReportTitleBand1	
DataBand1: Data Source: Customers	
{Customers.CompanyName}	{Customers.City}
ReportSummaryBand1	

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:
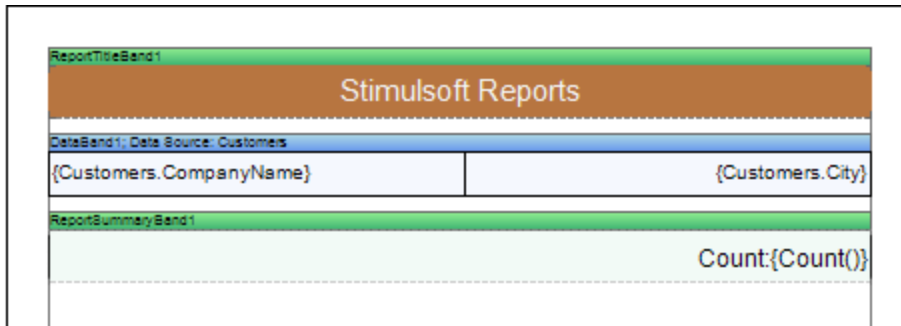
14. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies 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**, 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 Taqueria	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

2.2   **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**:

8. Put text components with expressions on **DataBands**. Where expression is a reference to the data field. For example, put a text component with the expression **{Customers.CompanyName}** 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-detail report template.

10. Click the **Preview** button or invoke the **Viewer**, 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:

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**:
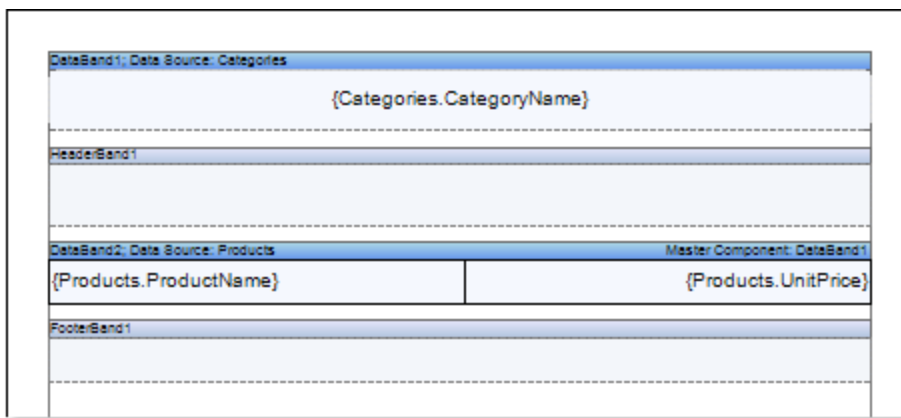
14. Put text components with expressions in the these bands. The expression in the text component is a header in the **HeaderBand**, and a footer in the **FooterBand**.
15. Edit text and text components:
     15.1. Drag and drop the text component in the band;
     15.2. Change font options: size, type, 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**, 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:

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**, clicking the **Preview** menu item. The picture below shows a sample of a rendered master-detail report with alternative color of rows:

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.

2.3   **Report with Groups**

For better understanding this step-by-step instruction, please watch the video file.
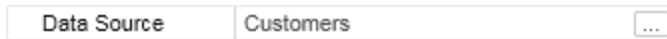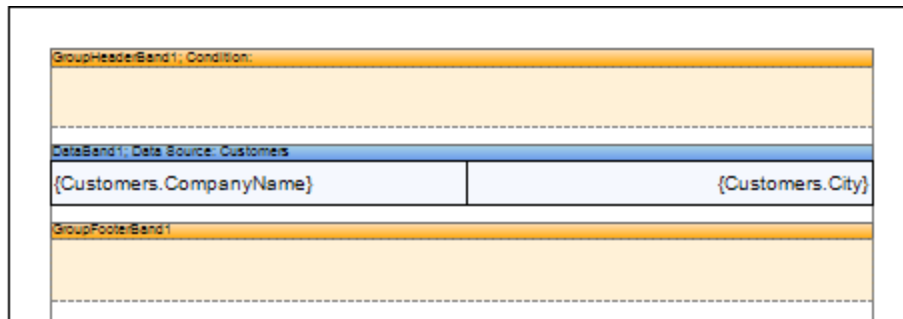
Do the following steps to create a report with grouping:

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".

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 expression **{Customers.ContactTitle}** into this text component. Put a text component in the **GroupFooterBand** and put the expression **{Count()}** 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:**
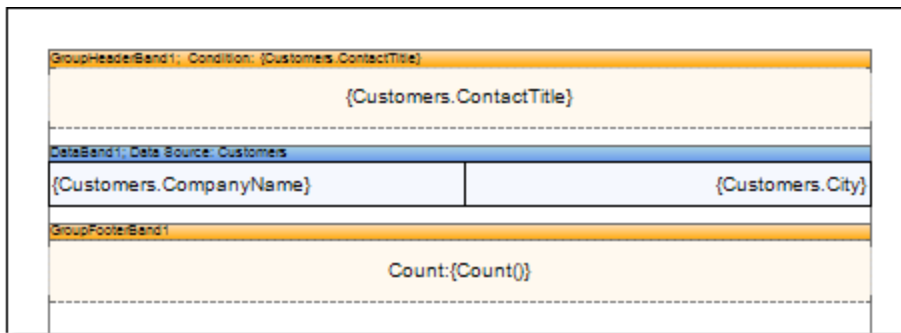
8. Edit expressions and text components:

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:
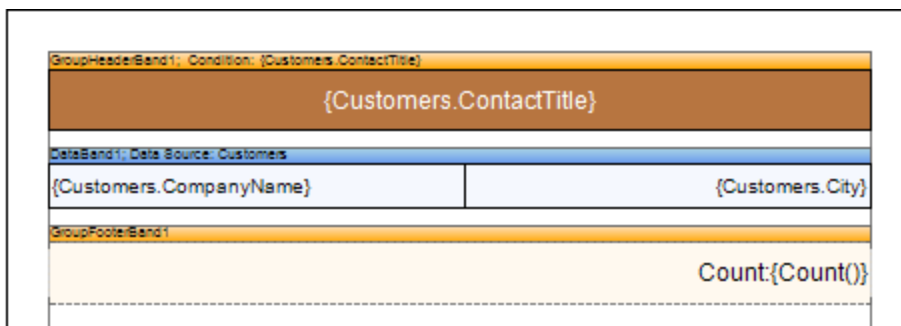
9. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies 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:
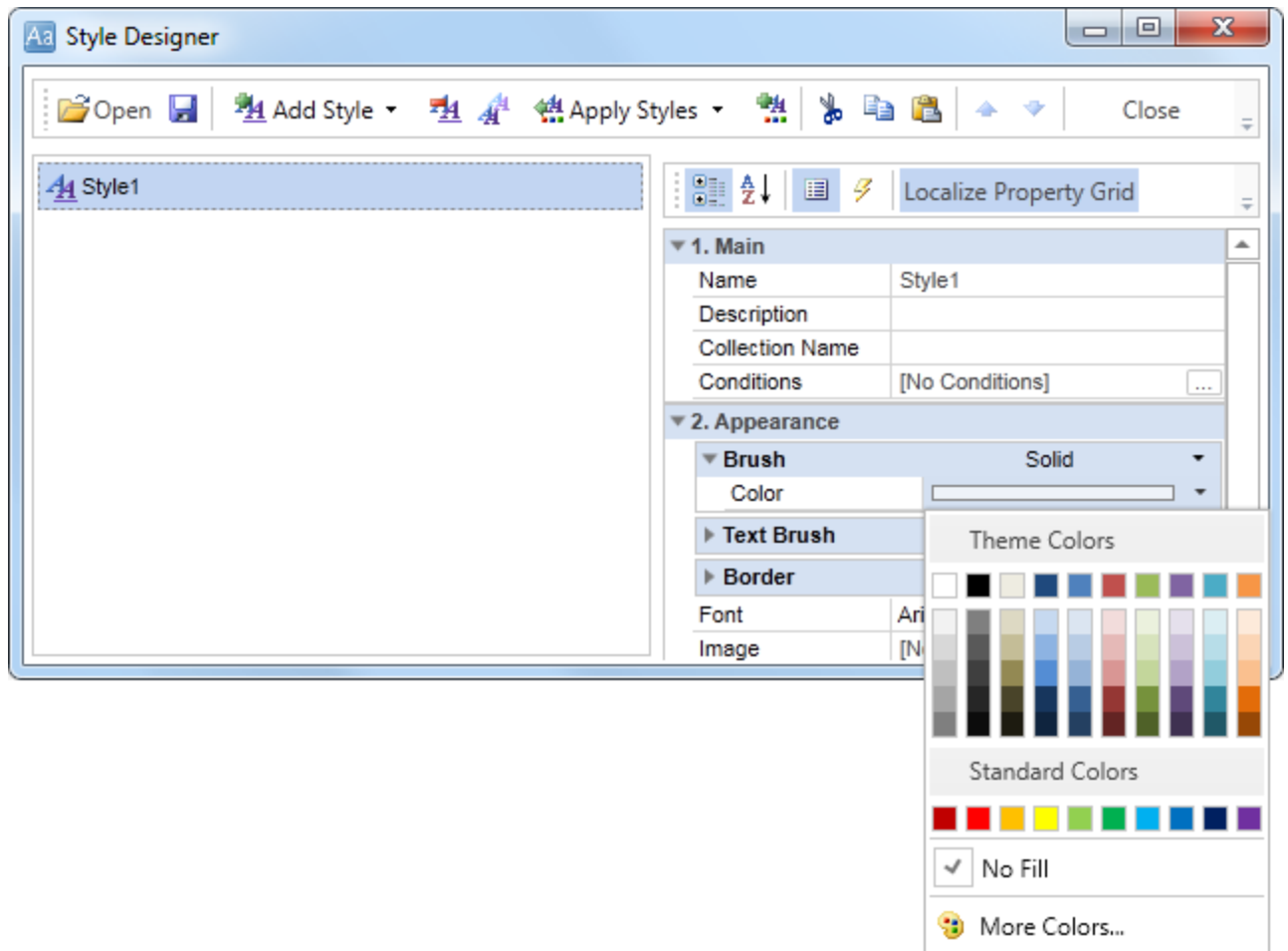
Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delícia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	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**, 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

2.4 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:
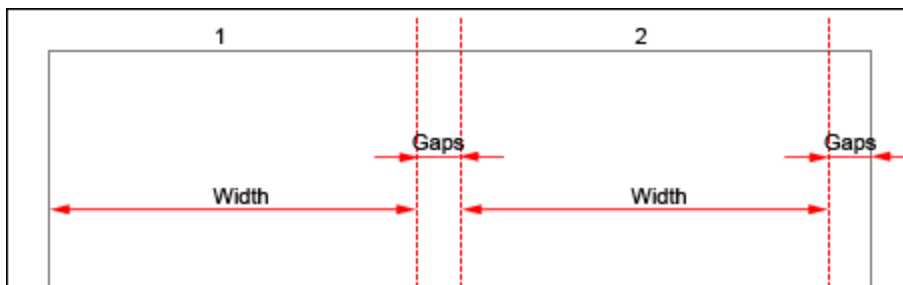
4. Put **DataBand** on a page.
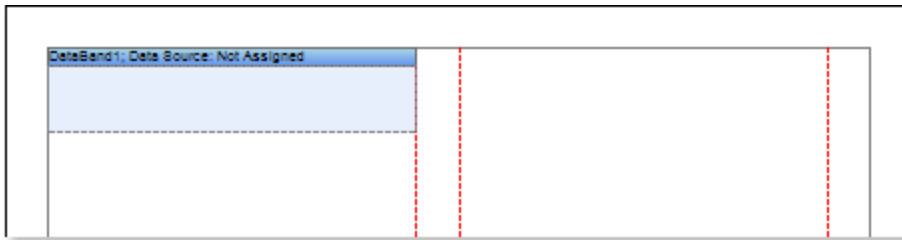
5. Edit **DataBand**:
    5.1. Align the **DataBand** by height;
    5.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
    5.3. Change the **DataBand** background;
    5.4. Enable **Borders** for the **DataBand**, if 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}**.
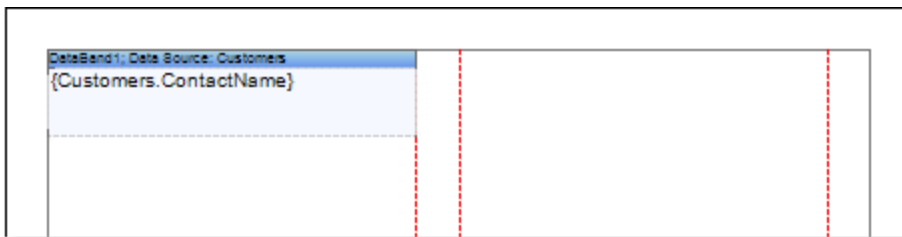
8. Edit expressions and text components:
    8.1. Drag and drop the text component in **DataBand**;
    8.2. Change parameters of the text font: size, type, color;
    8.3. Align the text component by width and height;
    8.4. Change the background of the text component;
    8.5. Align text in the text 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**, 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**;
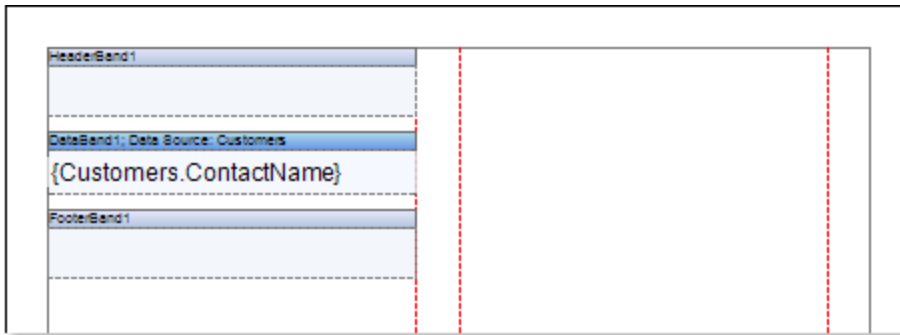
12. Edit these bands:
    12.1. Align them by height;
    12.2. Change values of properties, if required;
    12.3. Change the background of bands;
    12.4. Enable **Borders**, if required;
    12.5. Set the border color.

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**.
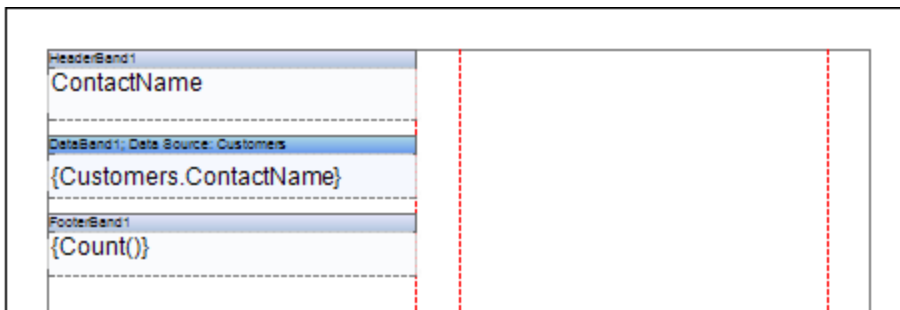
14. Edit text and text components:
    14.1. Drag and drop the text component in the band;
    14.2. Change font options: size, type, color;
    14.3. Align text component by height and width;
    14.4. Change the background of the text component;
    14.5. Align text in the text component;
    14.6. 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**, 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**, 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:

2.5 Report with Columns in Data Band

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.
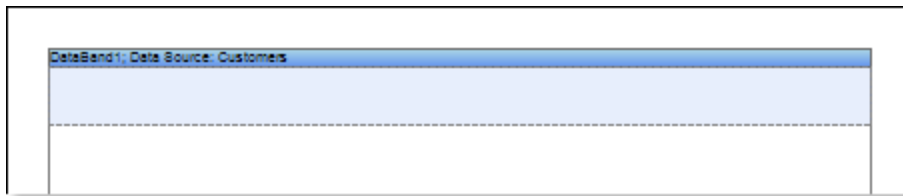
4. Define the data source for the **DataBand** using, for example, the **Data Source** property**:**

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**:
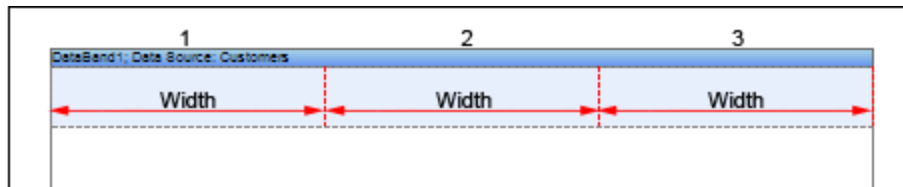
6. Put a text component with expressions on the **DataBand**. Where expression is a reference to the data field. For example, put one text component with the **{Customers.City}** expression.
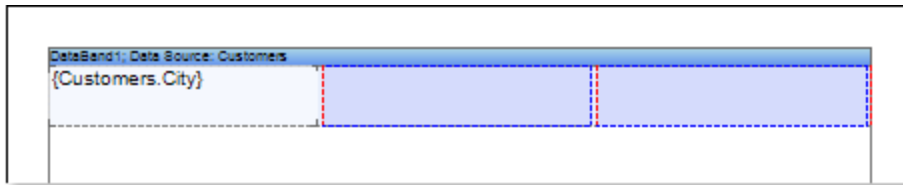
7. Edit expressions and text components:

    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.
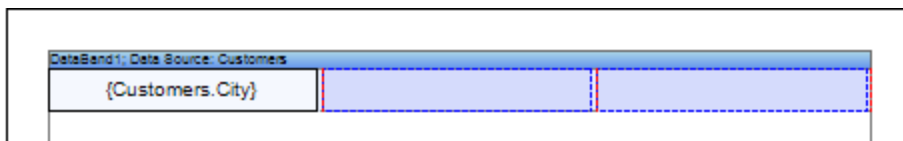
8. Set the columns direction of data output using the **Column Direction** property. Read about this property in section Report Internals -> Columns.

9. Click the **Preview** button or invoke the **Viewer**, 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.København	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**.

ColumnHeaderBand1

DataBand1; Data Source: Customers

{Customers.City}

ColumnFooterBand1

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.
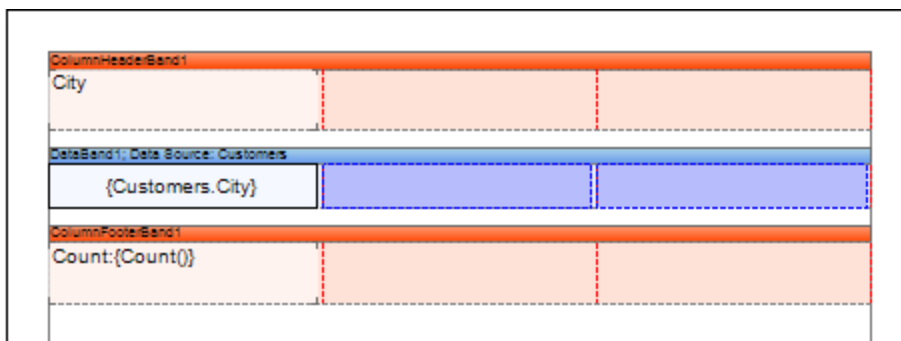
14. Edit **Text** and **TextBox** component:

    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.
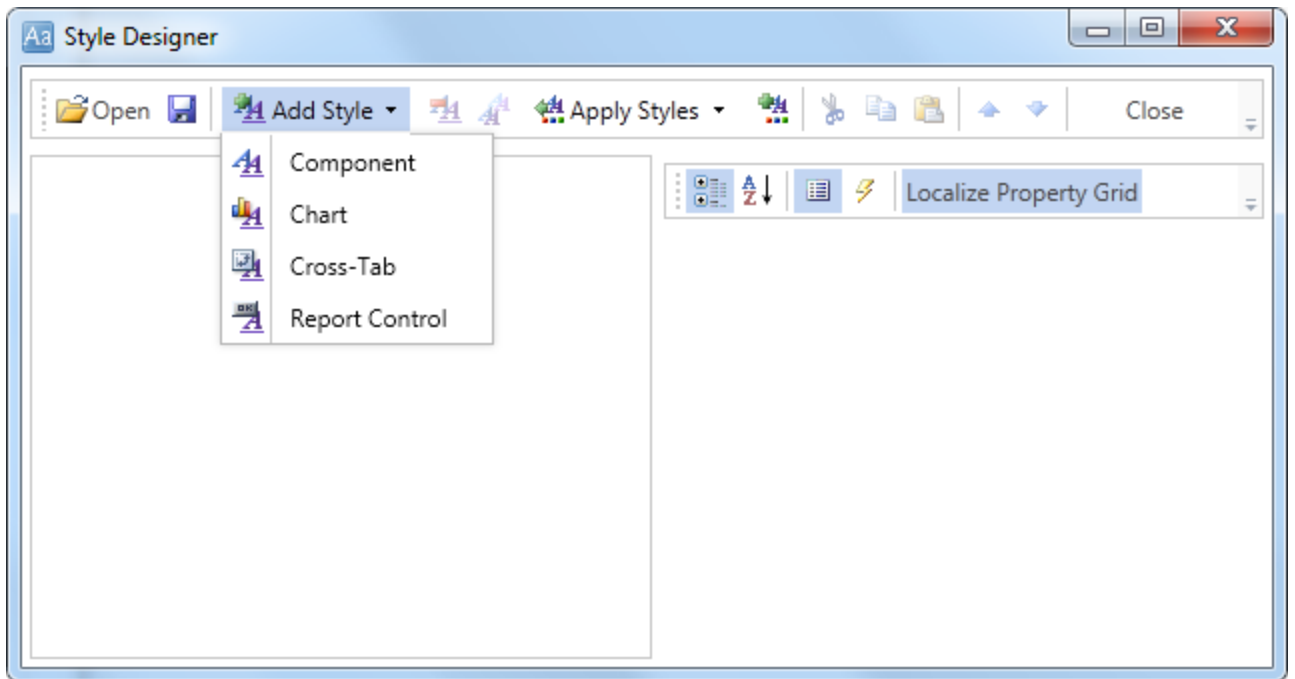
    14.8. Change the border color.
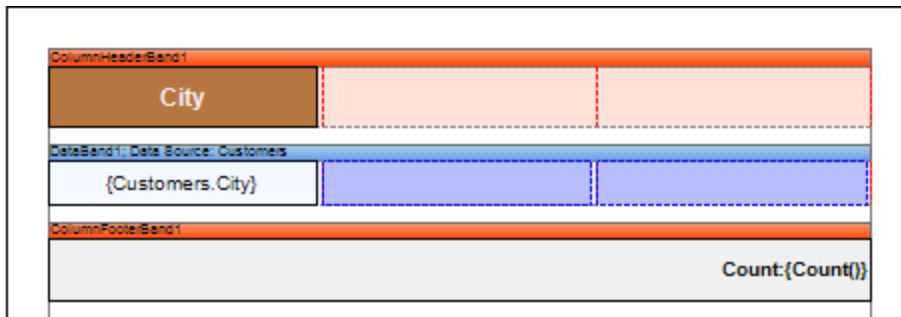
15. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies 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.København	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**, 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.København	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.København	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

2.6   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.
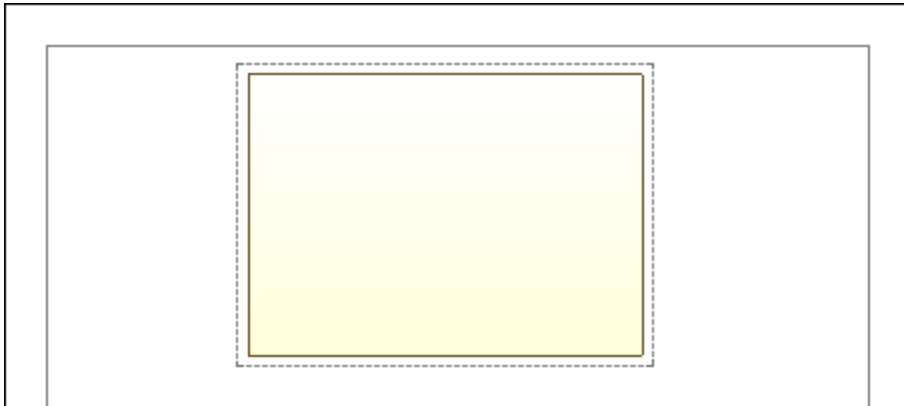
4. Edit the **Chart** component:

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**.
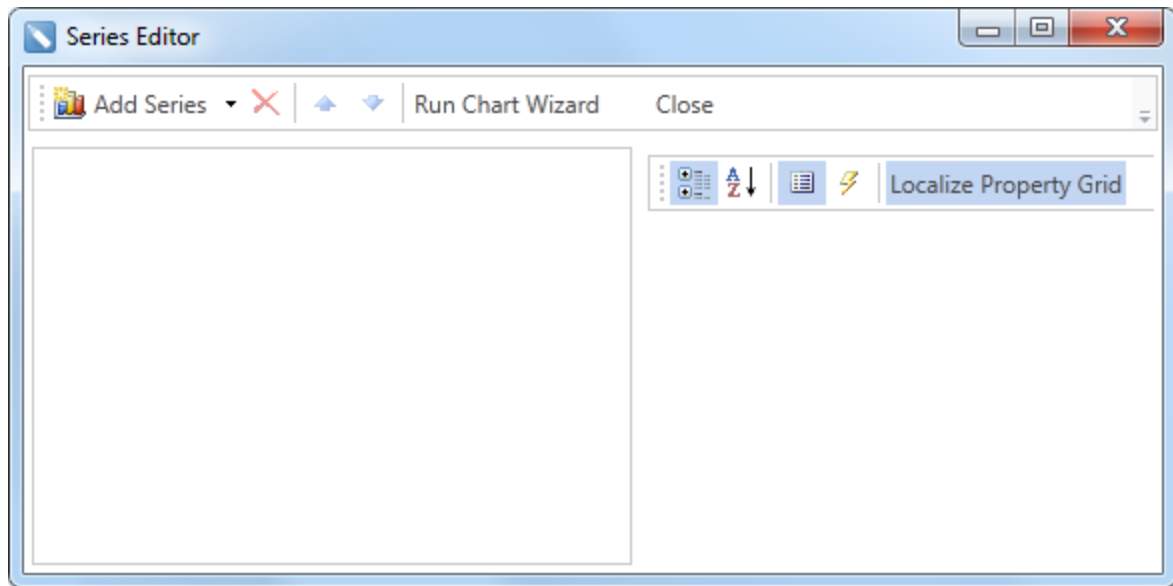
Click the **Add Series** button to add a series and select the type of series in the menu. The picture below shows the menu of the **Add Series** button:
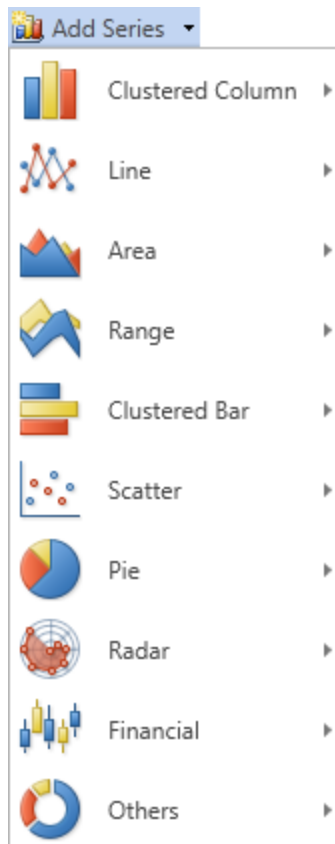
It should be noted that the type of number should match the type of chart, i.e. if the

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:
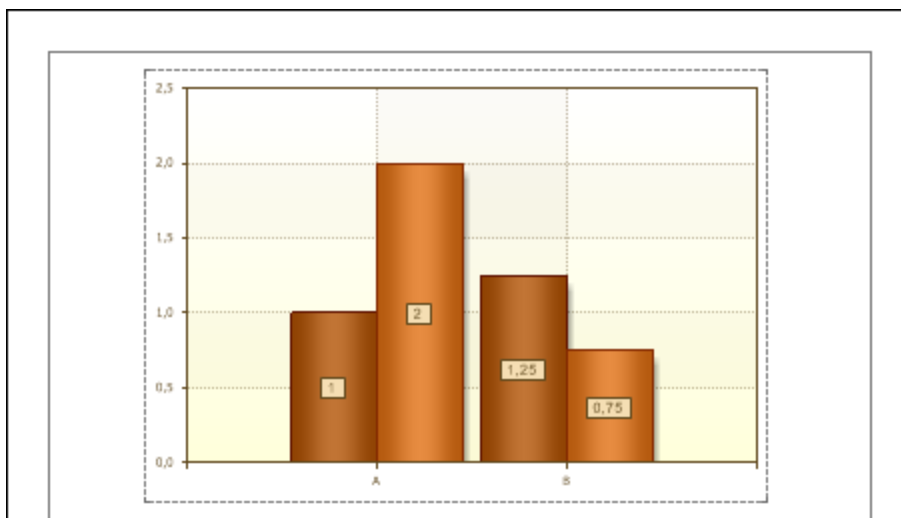
8. Edit **Legend**:

    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:
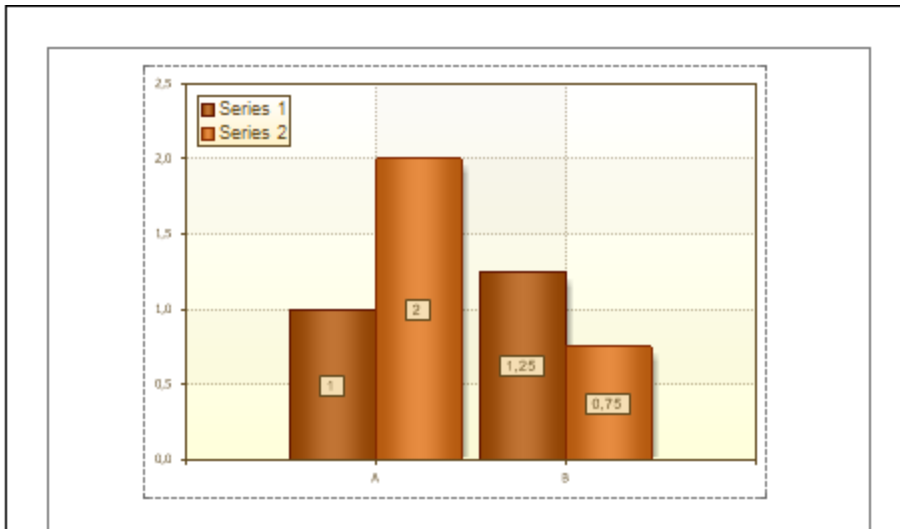
9. Change the style of the chart, i.e. completely change the appearance of the chart:

    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:
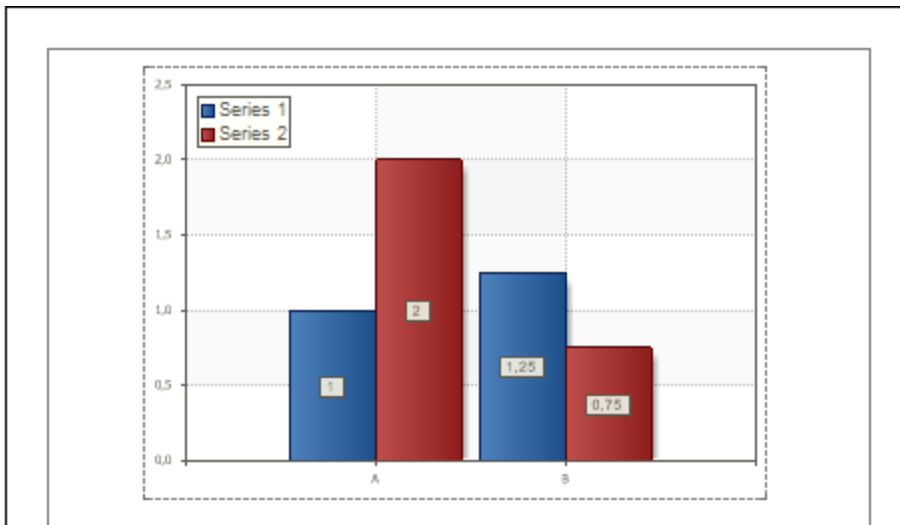
10. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. The picture below shows samples of reports with the chart:

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.
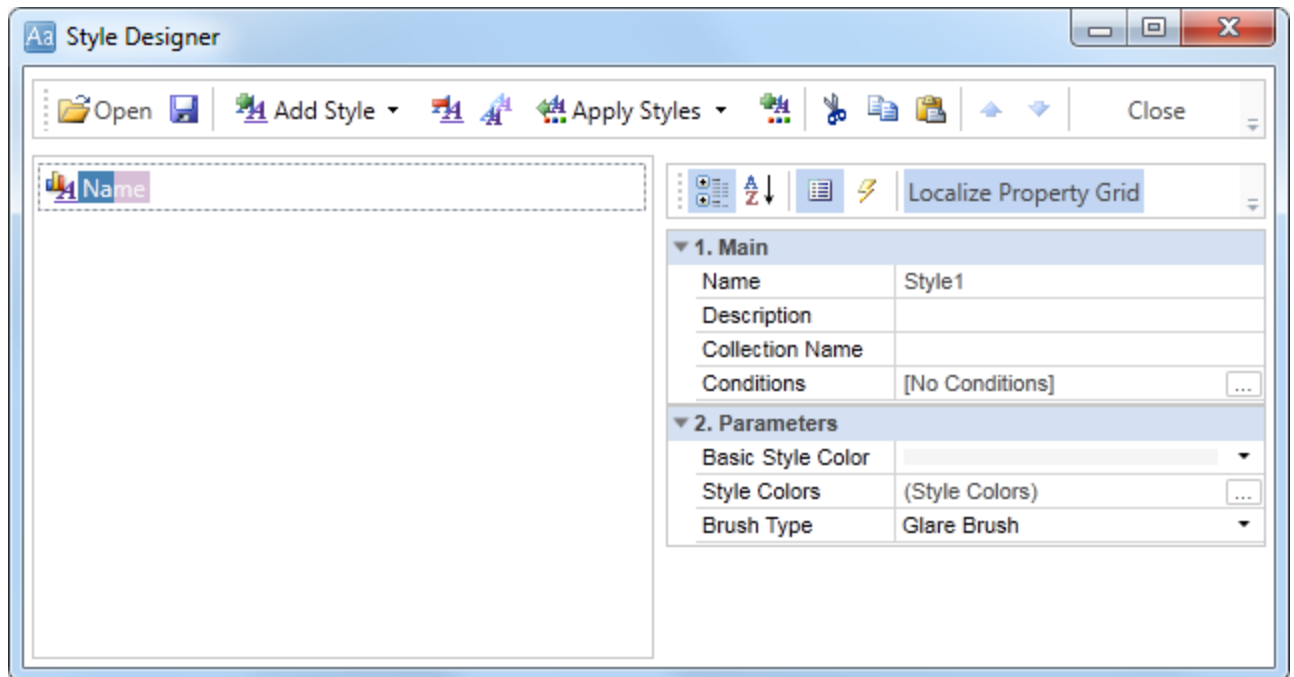
Click **Close**. In the list of values of the **Style** property of the chart component a custom style will be displayed. In our case, the value is **Style for Chart**. Select this value;

3. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. The picture below shows samples of reports with the chart with a style applied:

2.7   Report with Chart in Data Band

For better understanding this step-by-step instruction, please watch the <u>video file</u>.

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:
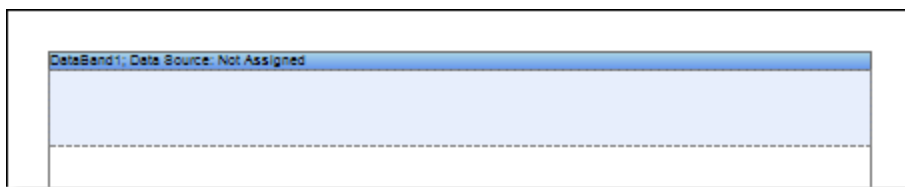
5. Edit **DataBand**:
     5.1. Align the **DataBand** by height;
     5.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
     5.3. Change the **DataBand** background;
     5.4. Enable **Borders** for the **DataBand**, 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:
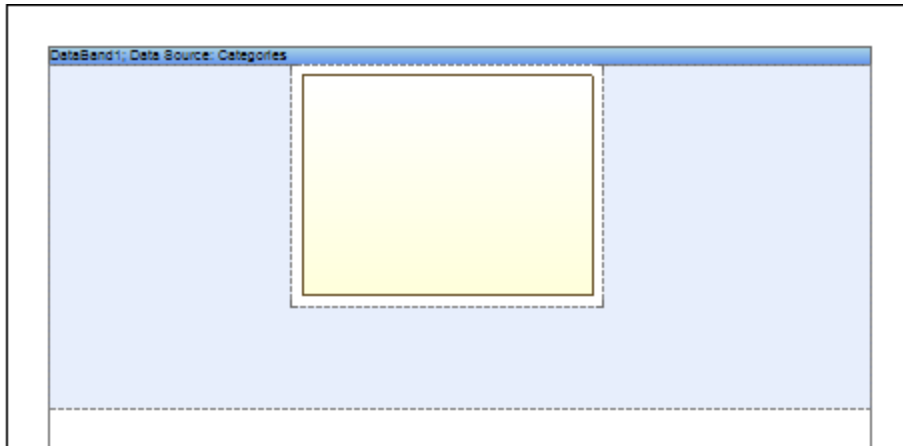
8. Edit the **Chart** component:
    8.1. Align it by width;
    8.2. Change properties of the **Chart** component. For example, set the **GrowToHeight** property to **true**, if it is required the Chart component be grown by height;
    8.3. Set **Borders**, if required, for the **Chart** component;
    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.
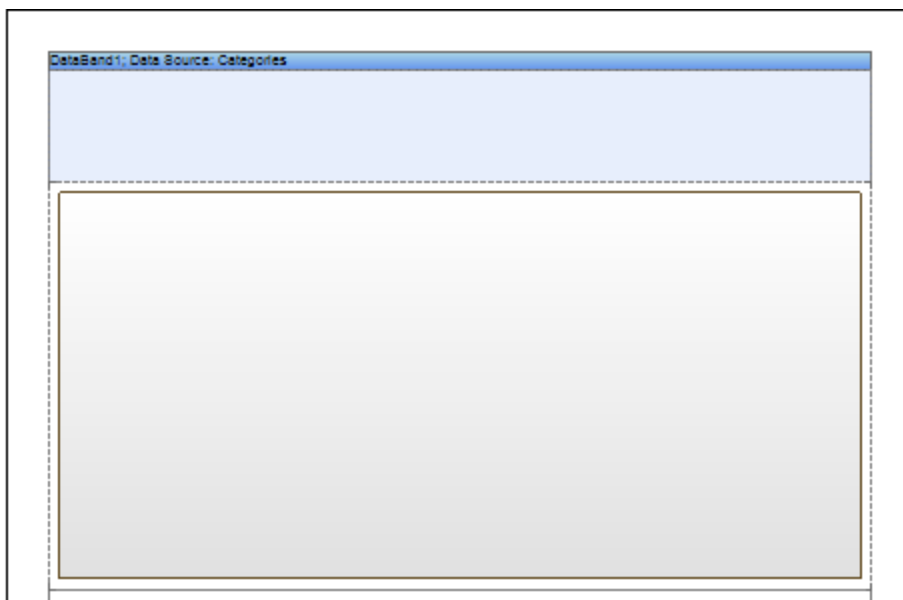
9. Change the type of a chart using the **Chart Type** property. For example, set it to **Clustered Column**:

10. Define the data source for the **Chart** component using the **Data Source** property

| Data Source | Products | ... |

11. Define the relation between data sources, using the **DataRelation** property of the **Chart** component:

| Data Relation | Categories | ... |

12. Add series. Invoke the **Series Editor**, for example, by double-clicking the **Chart**:

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:
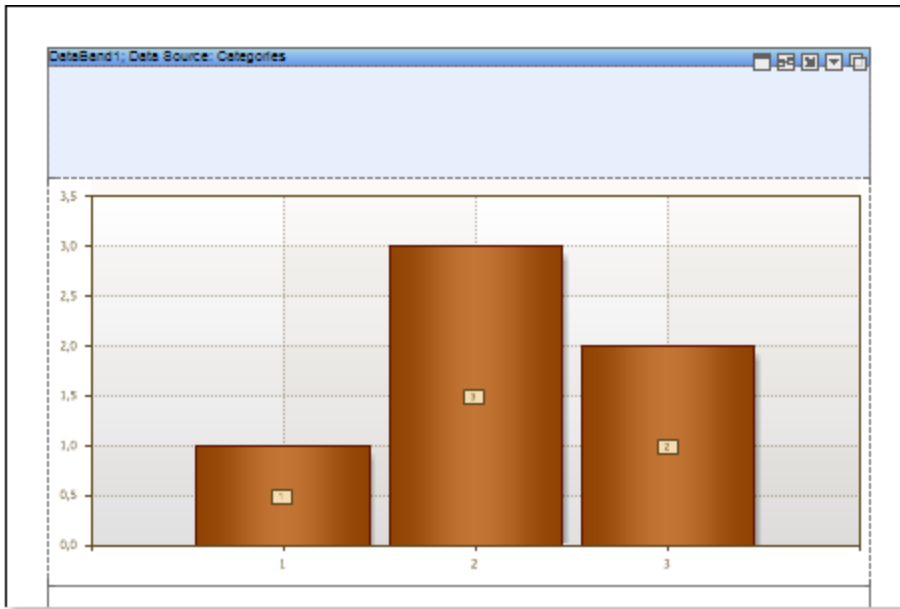
14. Edit **Legend**:

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:
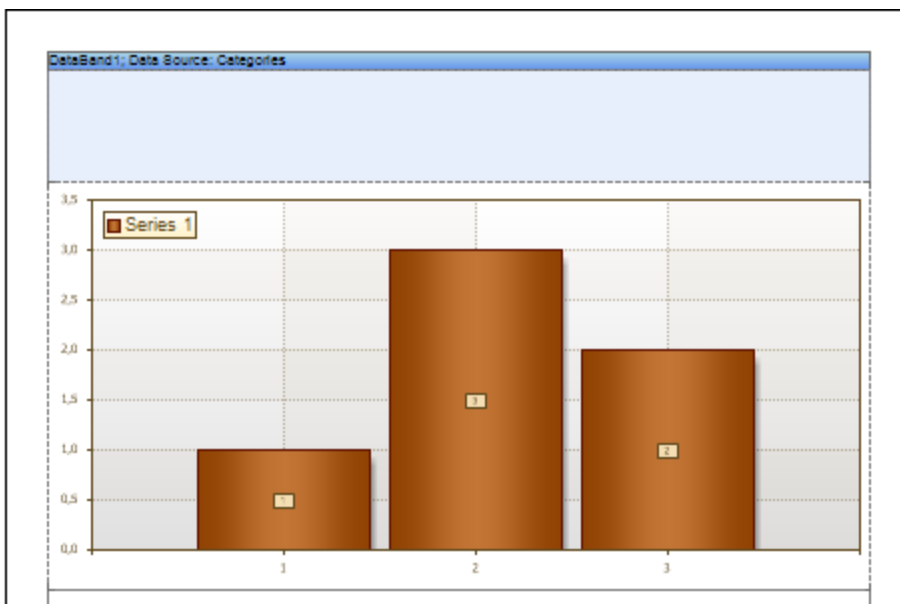
15. Change the style of the chart, i.e. completely change the appearance of the chart:

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:
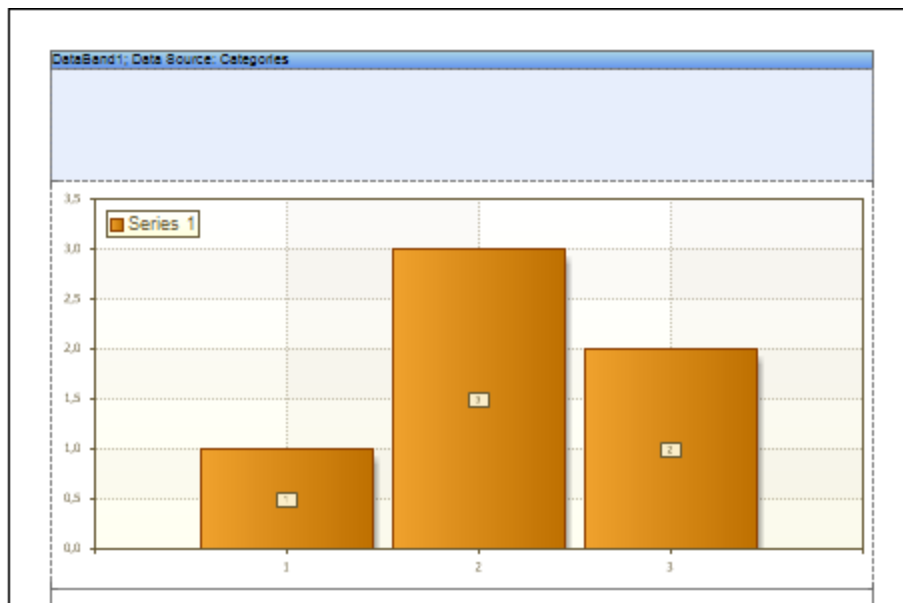
16. Put text components with an expression in the **DataBand**. Where the expression is a reference to the data field. For example, put a text component with the expression: **{Categories.CategoryName}**;

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**, 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.
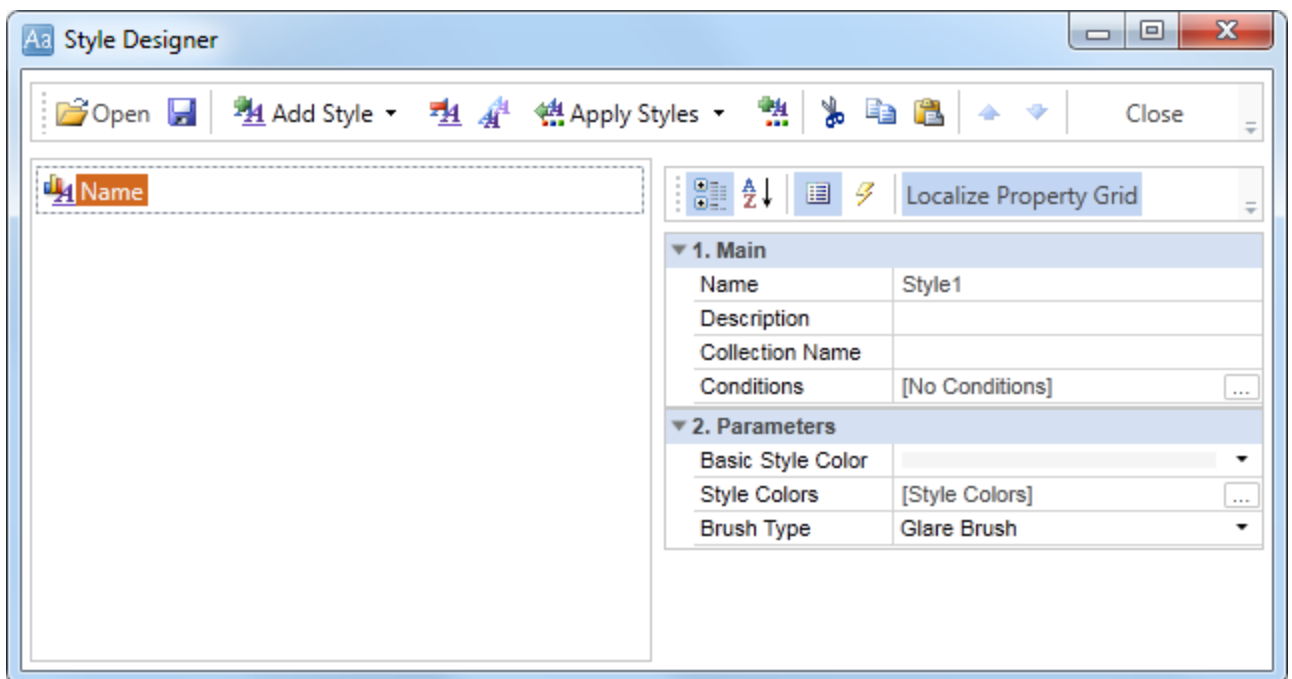
Click **Close**. In the list of values of the **Style** property of the chart component a custom style will be displayed. In our case, the value is **Style for Chart**. Select this value;

3. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. The picture below shows samples of reports with the chart with a style applied:

2.8 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;
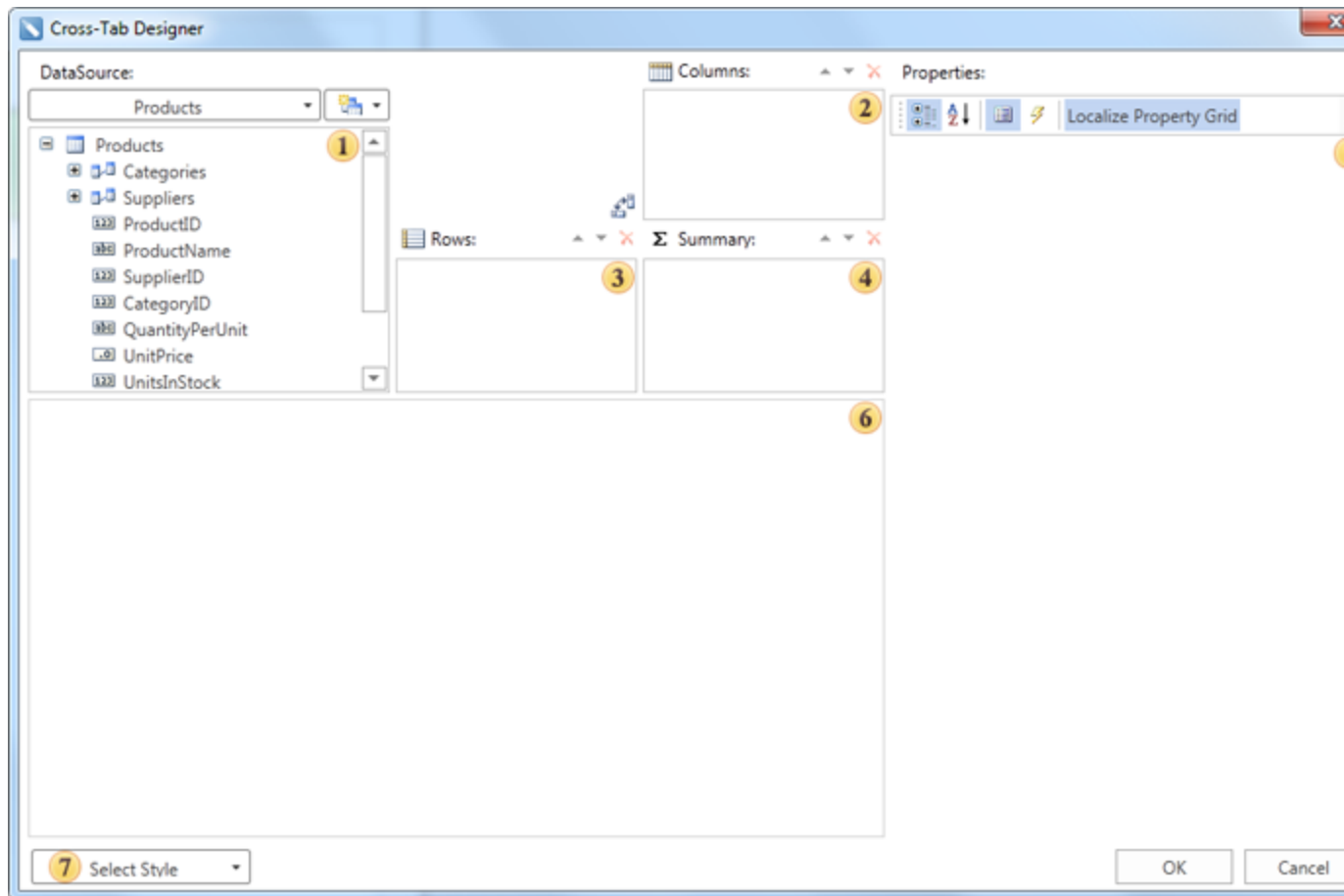
2. Connect data:
    2.1. Create **New Connection**;
    2.2. Create **New Data Source**;

3. Put the **Cross-Tab** component on a page of the report template.
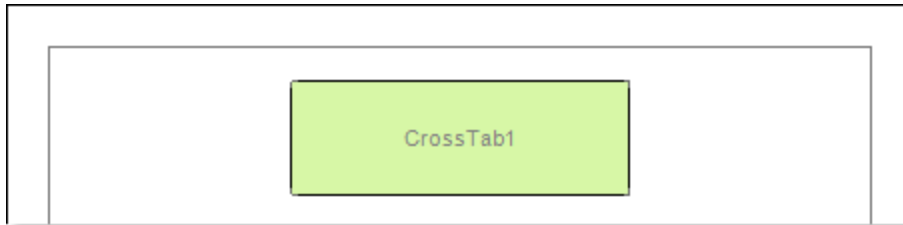
4. Edit the **Cross-Tab** component:
    4.1. For example, set the **GrowToHeight** property to **true**, to allow the **Cross-Tab** component to grow by height;

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 **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.
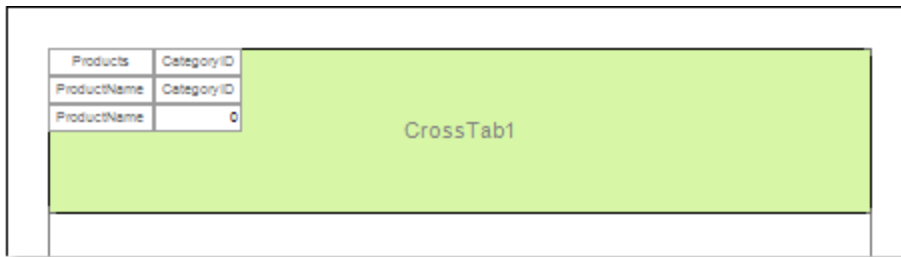
Products	CategoryID	
ProductName	CategoryID	
ProductName	0	CrossTab1

9. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. The picture below shows a rendered cross-tab report:

Products				Category ID				
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

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**, clicking the **Preview** menu item. The picture below shows a report of the rendered report with the cross table after editing report template cells:
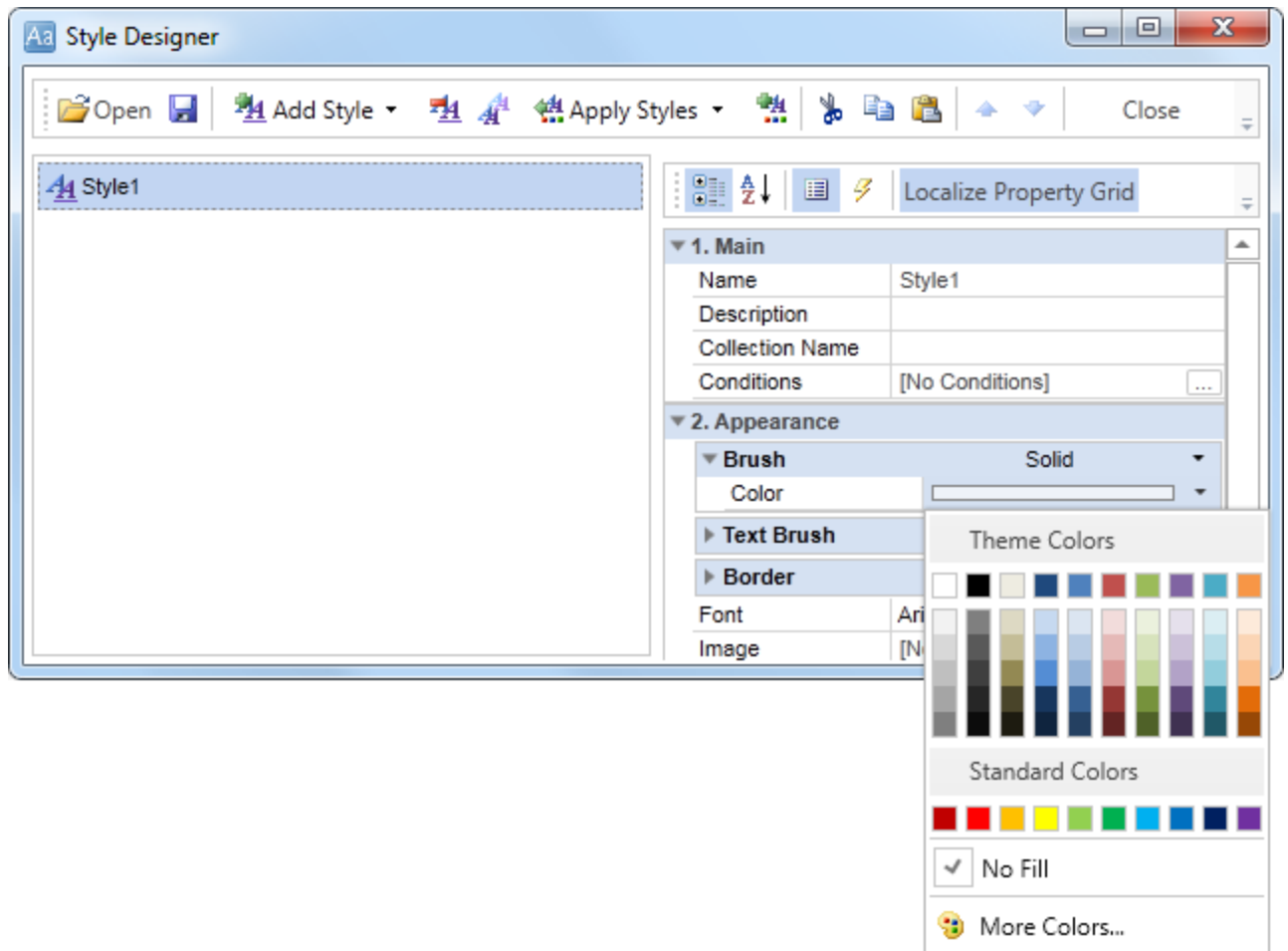
Adding styles

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**, 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

2.9   Cross-Tab Report in Data Band

For better understanding this step-by-step instruction, please watch the <u>video file</u>.

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;
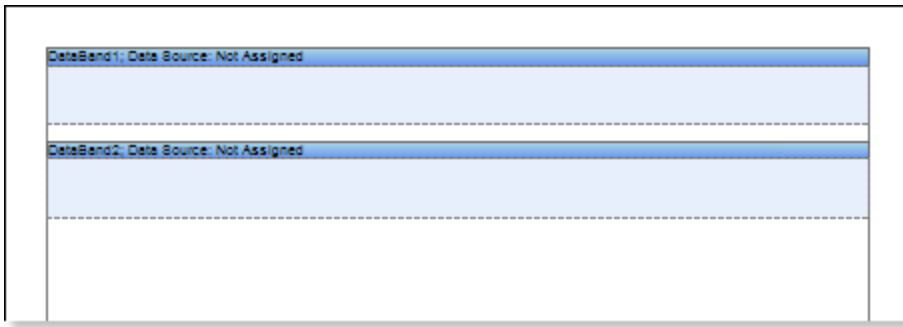
5. Edit **DataBand1** and **DataBand2**:

    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**:

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.
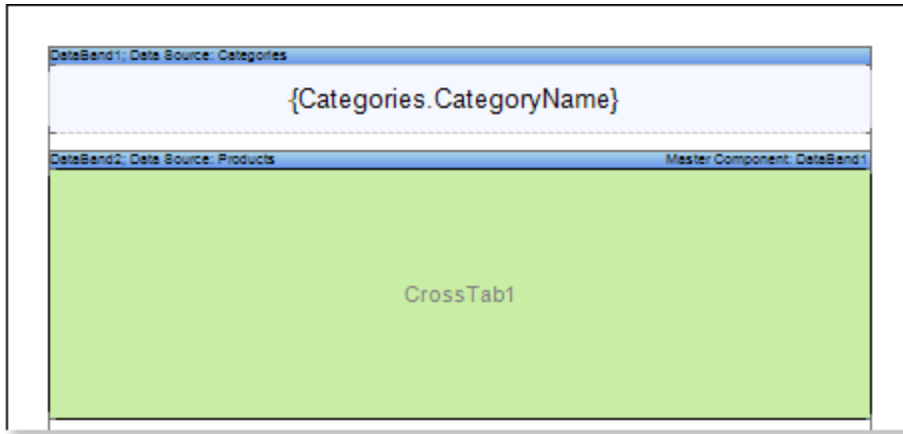
11. Edit the **Cross-Tab** component:

    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.
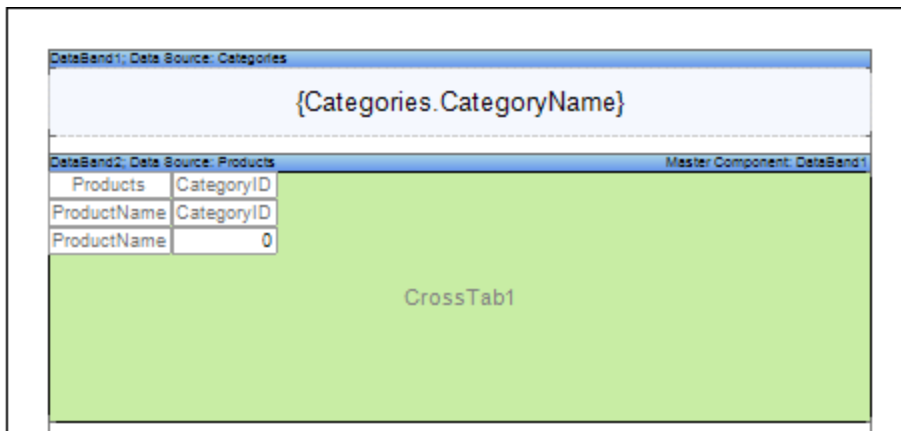
16. Render a report. Click the **Preview** button or call the **Viewer** by selecting the **Preview** of the menu item. The picture below shows an example of the cross-table report:

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.

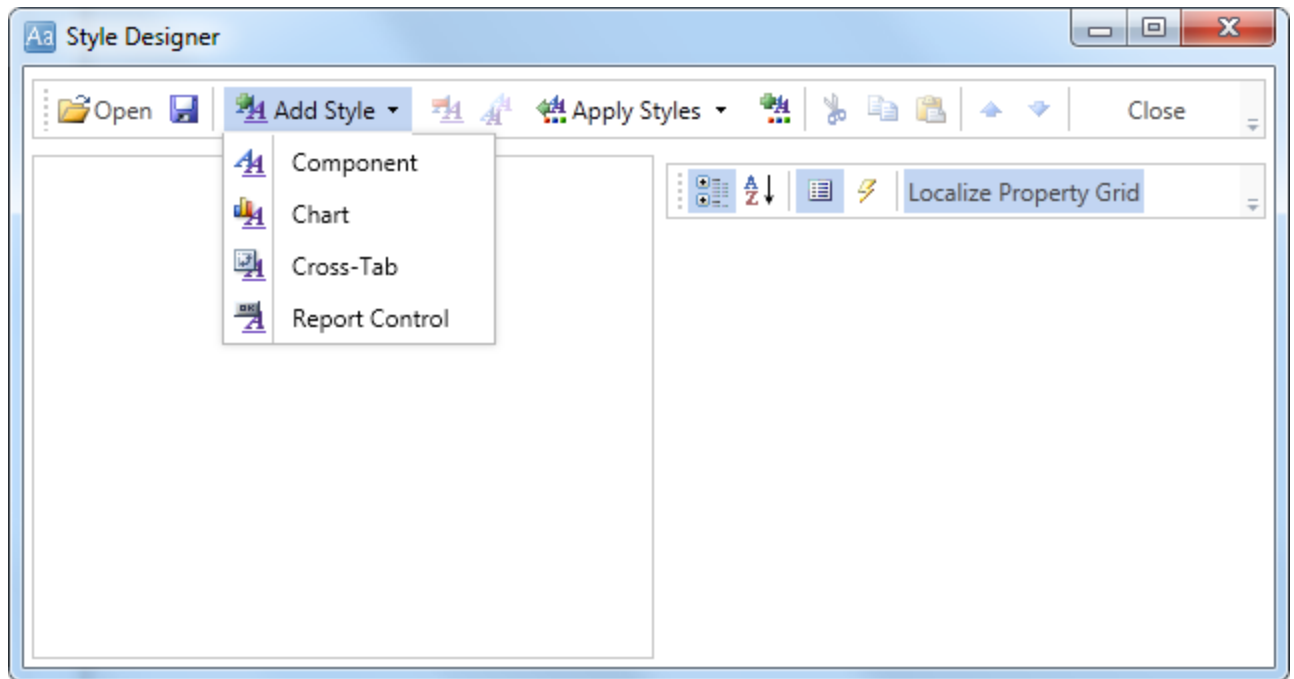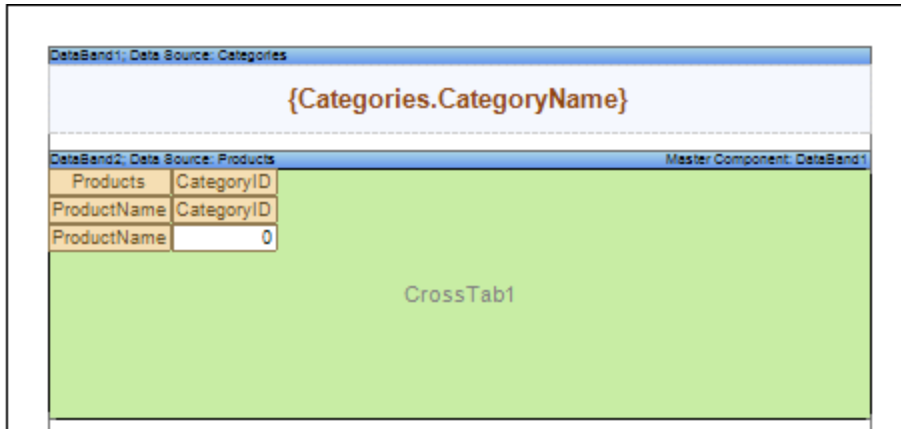20. Render a report. Click the **Preview** button or call the **Viewer** by clicking the **Preview** menu item. The picture below shows an example of the cross-table report after editing cells of the report template:

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**;
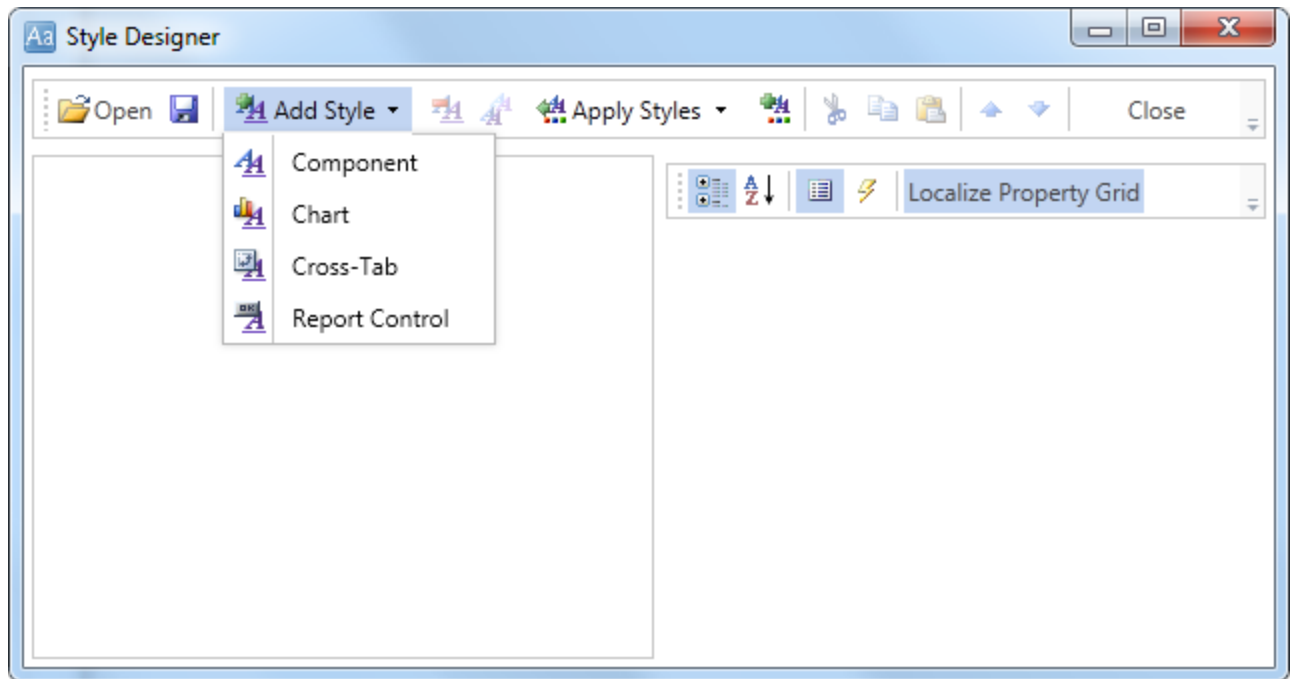
Click the **Add Style** button to start creating a style. Select **Cross-Tab** from the drop down list. Call the new style as **Style for Cross-Tab**. To create a custom style it is necessary to change the **Color** property, where the value of this property and is a color scheme.
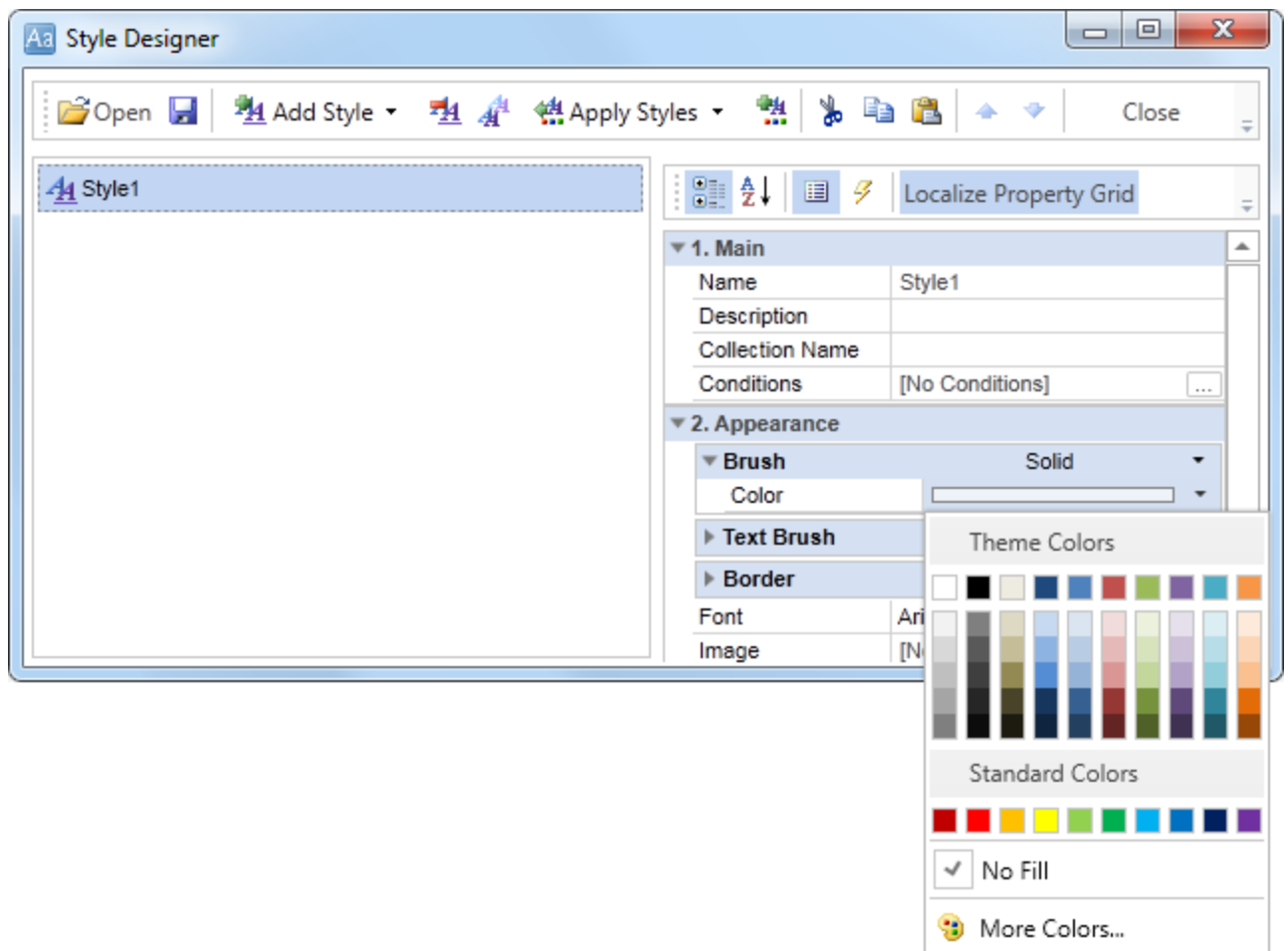
After the style is created, press the **Close** button. In the list of values of the **Select Style** button in the editor of the cross-table, a custom style will be displayed. In our case, this is the **Style for Cross-Tab**. Select this value;

3. Render a report. Click the **Preview** button or call the **Viewer** by selecting the **Preview** 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

2.10 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**.**
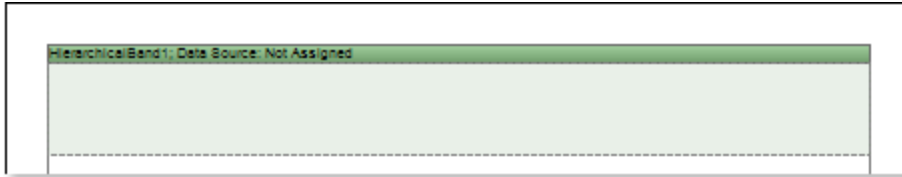
4. Edit the **HierarchicalBand**:
    4.1. Align the **HierarchicalBand** by height;
    4.2. Set the properties of the **HierarchicalBand**. For example, set the **Can Break** property to **true**, if it is necessary for the **HierarchicalBand** to be broken;
    4.3. Set the background of the **HierarchicalBand**;
    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:
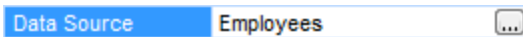
6. Put text components with expressions in the **HierarchicalBand**. Where the expression is a reference to the data field. For example, put three text component with expressions: **{Employees.LastName}, {Employees.City}**, and **{Employees.Region}**;
7. Edit text (**Text**) and text components (**TextBox**):
    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.
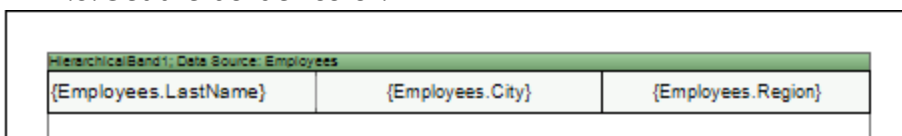
8. Set the **KeyDataColum** property, i.e. select a data column on which an identification number of the data row will be assigned. In this case, select the **EmployeeID** data column:

Key Data Column   [abc] **EmployeeID**   ▼

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:

Master Key Data Column [abc] **ReportsTo**   ▼

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);

Indent   20

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:

Parent Value

12. Click the **Preview** button or call **Viewer**, using 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.

Fuller	Tacoma	WA
Buchanan	London	
Dodsworth	London	
King	London	
Suyama	London	
Callahan	Seattle	WA
Peacock	Redmond	WA
Davolio	Seattle	WA
Leverling	Kirkland	WA

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.

HeaderBand1		
HierarchicalBand2: Data Source: Employees		
{Employees.LastName}	{Employees.City}	{Employees.Region}

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 the **Preview** menu item. After rendering a report, all references to data sources will be replaced with data from these sources:

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.
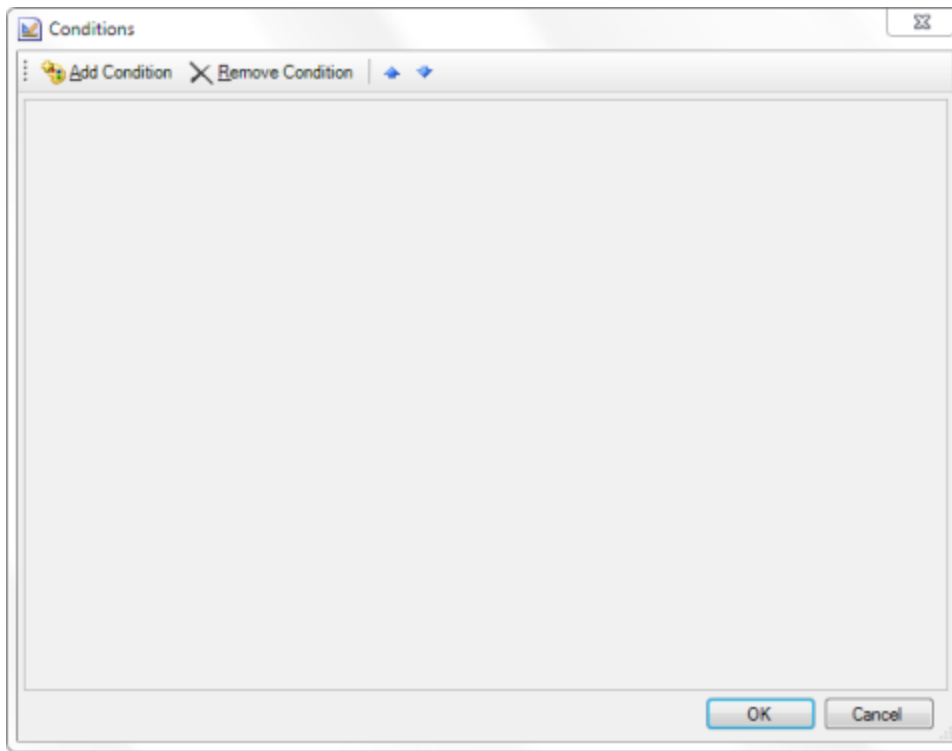
3. To get started, you must click the **Add Condition** button and in the **Conditions** dialog box the condition and formatting options will be displayed. The condition can be of two types: **Value** and **Expression**. In this case, consider an example of a condition, such as **Expression**. The picture below shows an example of **Conditions** dialog box with options and conditions of formatting:
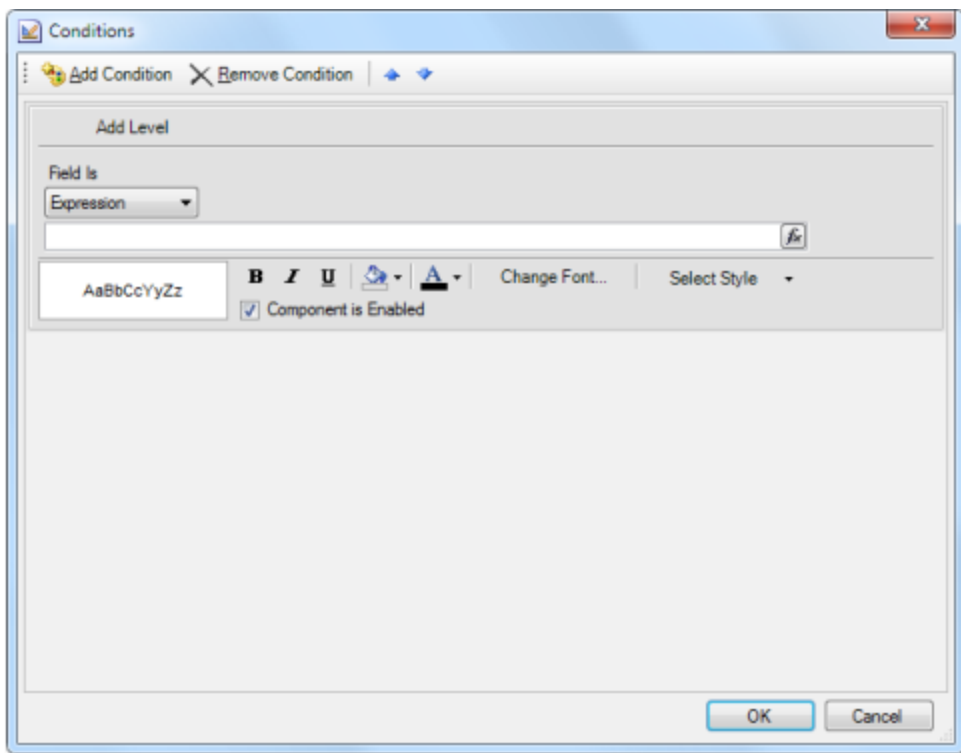
4. Specify the options of conditional formatting. In this case, to specify the condition means to specify the expression. For example, **Line% 2 == 1**, and set the formatting means to change the background, for example, by pressing the **Back Color** button and selecting the drop-down list of values 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** clicking the **Preview** menu item.

2.11 Report with Sub-Report

For better understanding this step-by-step instruction, please watch the <u>video file</u>.

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:
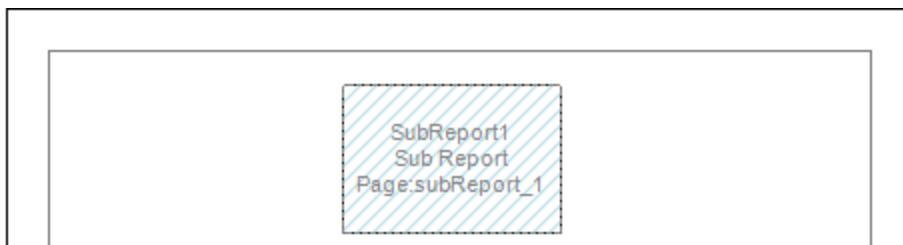
4. Edit the **Sub-Report** component:
   4.1. Stretch the **Sub-Report** component as seen on the picture below;
   4.2. Change the value of properties of **Sub-Report**. For example, set the **Keep Sub-Report Together** property to **true**, if you want the sub-report to be kept together;;
   4.3. Change the background color of the component.
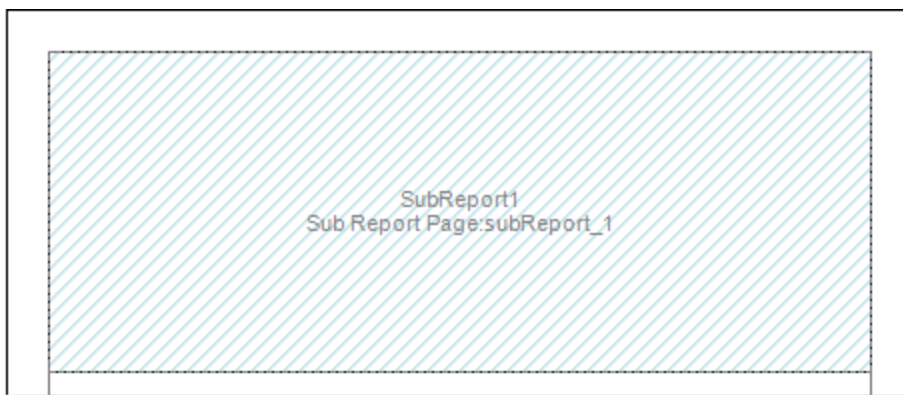
5. Go to the sub-report page;

6. Add to the **DataBand** to the sub-report page**;**
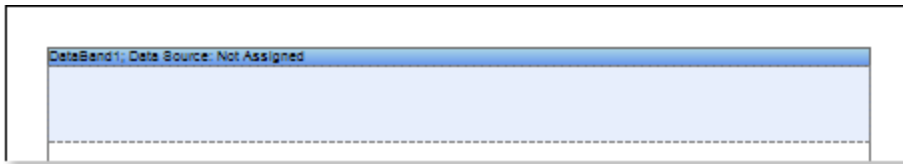
7. Edit the **DataBand**:
    7.1. Align the **DataBand** vertically;
    7.2. Change values of properties of the **DataBand**. For example, set the **CanBreak** property to **true**, if you want this band to be broken;
    7.3. Change background color of the band;
    7.4. Set **Borders**, if necessary;
    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 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 Taqueria	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 Taqueria	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.
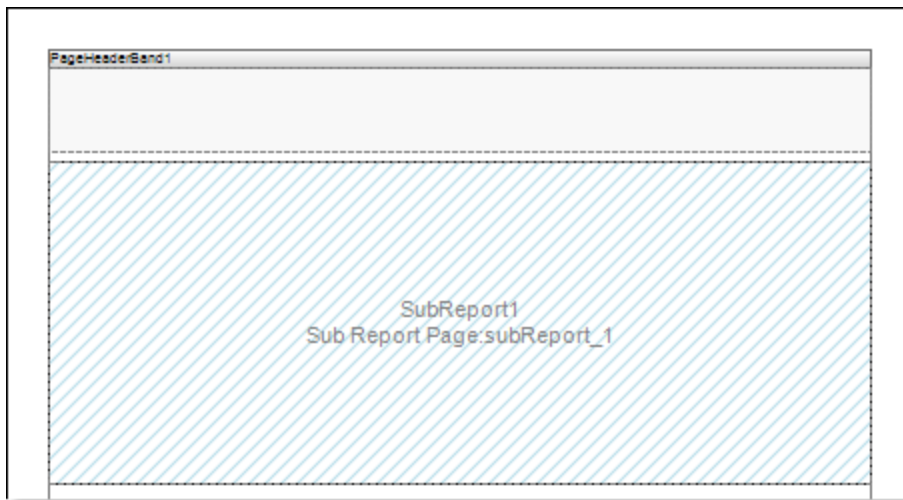
15. Put a text component with expression where the expression of the text component in the **PageHeaderBand** will be the page title.

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 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 fis	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**, 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 fis	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Markets	Tsawassen
B's Beverages	London
Cactus Comidas para llevar	Buenos Aires

2.12 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:
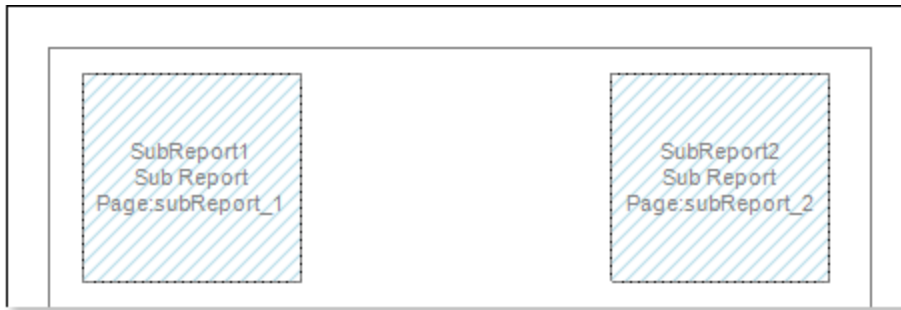
4. Edit **Sub-Report** components:

    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**;
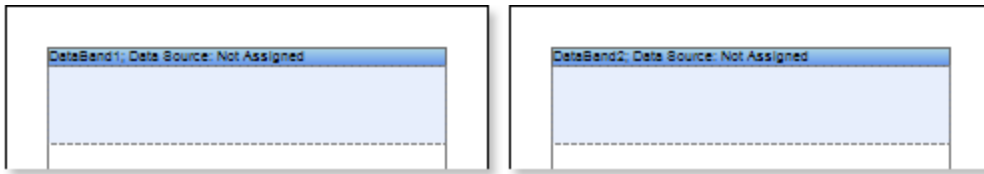
7. Edit the **DataBands**:
    7.1. Align the **DataBands** vertically;
    7.2. Change values of properties of the **DataBands**.
    7.3. Change background color of the band;
    7.4. Set **Borders**, if necessary;
    7.5. Change the border color.

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 the **Preview** menu item to see how the report will look like:

|			
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

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.
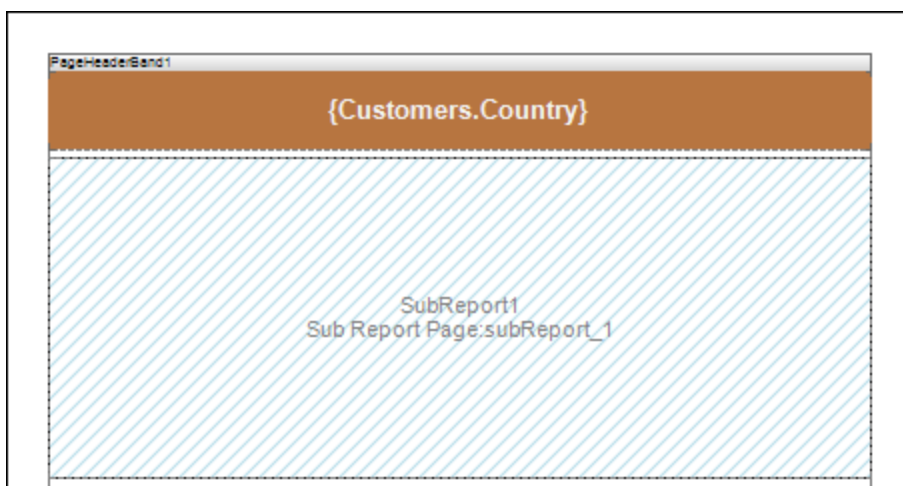
12. Go back to the report template;
13. If necessary, add some bands to the report template, for example, the **HeaderBand**;
14. Edit this band:
    14.1. Align vertically this band;
    14.2. Set values of the properties of the **HeaderBand**, if necessary;
    14.3. Set the background color;
    14.4. Set **Borders** of a text component.
    14.5. Set the border color.

HeaderBand1		HeaderBand2	
DataBand1; Data Source: Customers		DataBand2; Data Source: Products	
{Customers.CompanyNa me}	{Customers.City}	{Products.ProductName}	{Products.UnitPrice}

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;

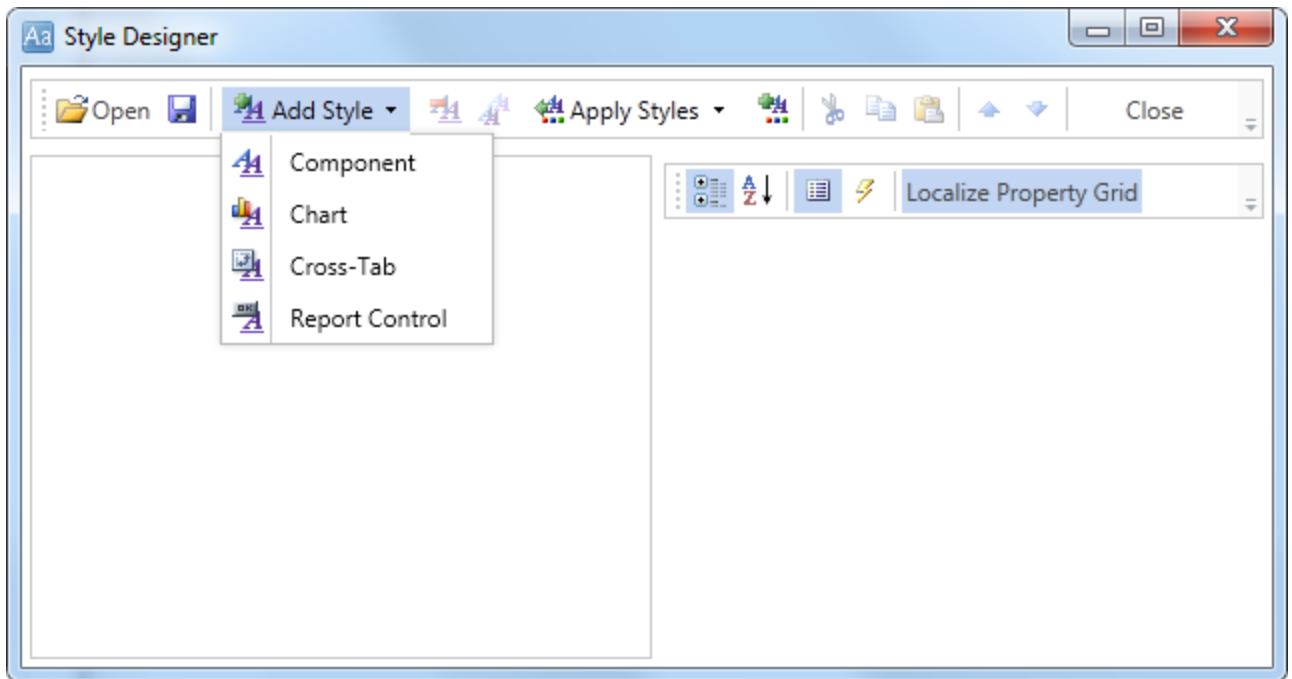16.7. Set **Borders** of a text component.

16.8. Set the border color.

CompanyName	City
{Customers.CompanyName}	{Customers.City}

ProductName	UnitPrice
{Products.ProductName}	{Products.UnitPrice}

17. Click the **Preview** button or call **Viewer**, using 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**, 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

2.13 Report with Sub-Reports in Data Band

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.

DataBand1: Data Source: Not Assigned
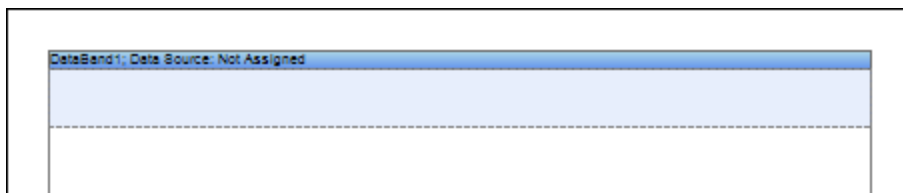
4. Edit **DataBand**:
    4.1. Align the **DataBand** by height;

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;
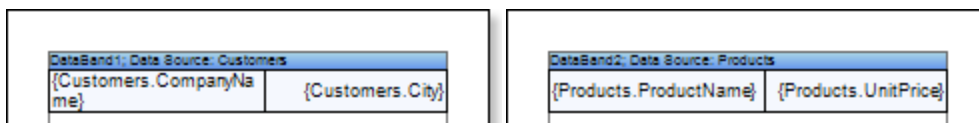
4.5. Change the border color.

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.
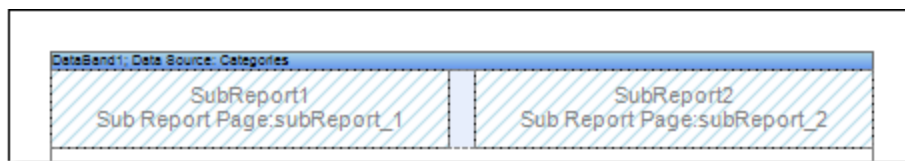
8. Go to the sub-report page;

9. Add two **DataBands** to the sub-report page. Add **DataBand1** to the **Sub Report1** and **DataBand2** to the **Sub Report2**;
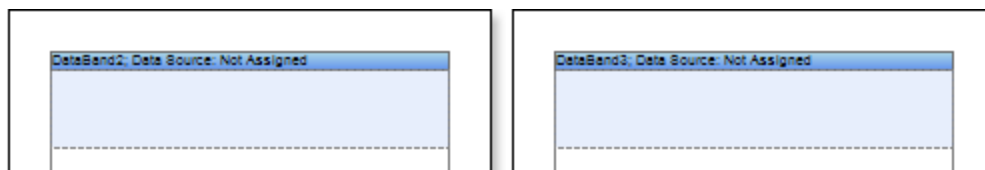
10. Edit the **DataBands**:

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**:

Data Source	Customers	...
Data Source	Products	...

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.

DataBand2: Data Source: Customers	
{Customers.CompanyName}	{Customers.City}

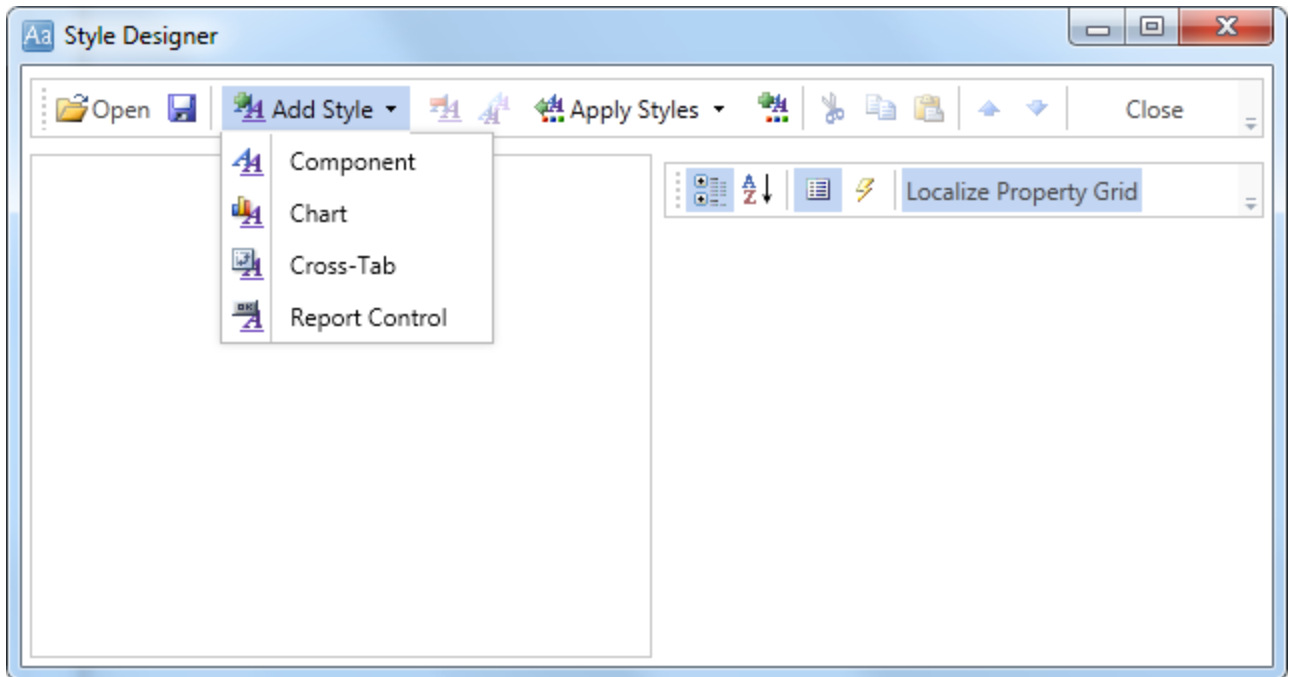| DataBand3: Data Source: Products	
{Products.ProductName}	{Products.UnitPrice}

14. Click the **Preview** button or call **Viewer**, using the **Preview** menu item to see how the report will look like:

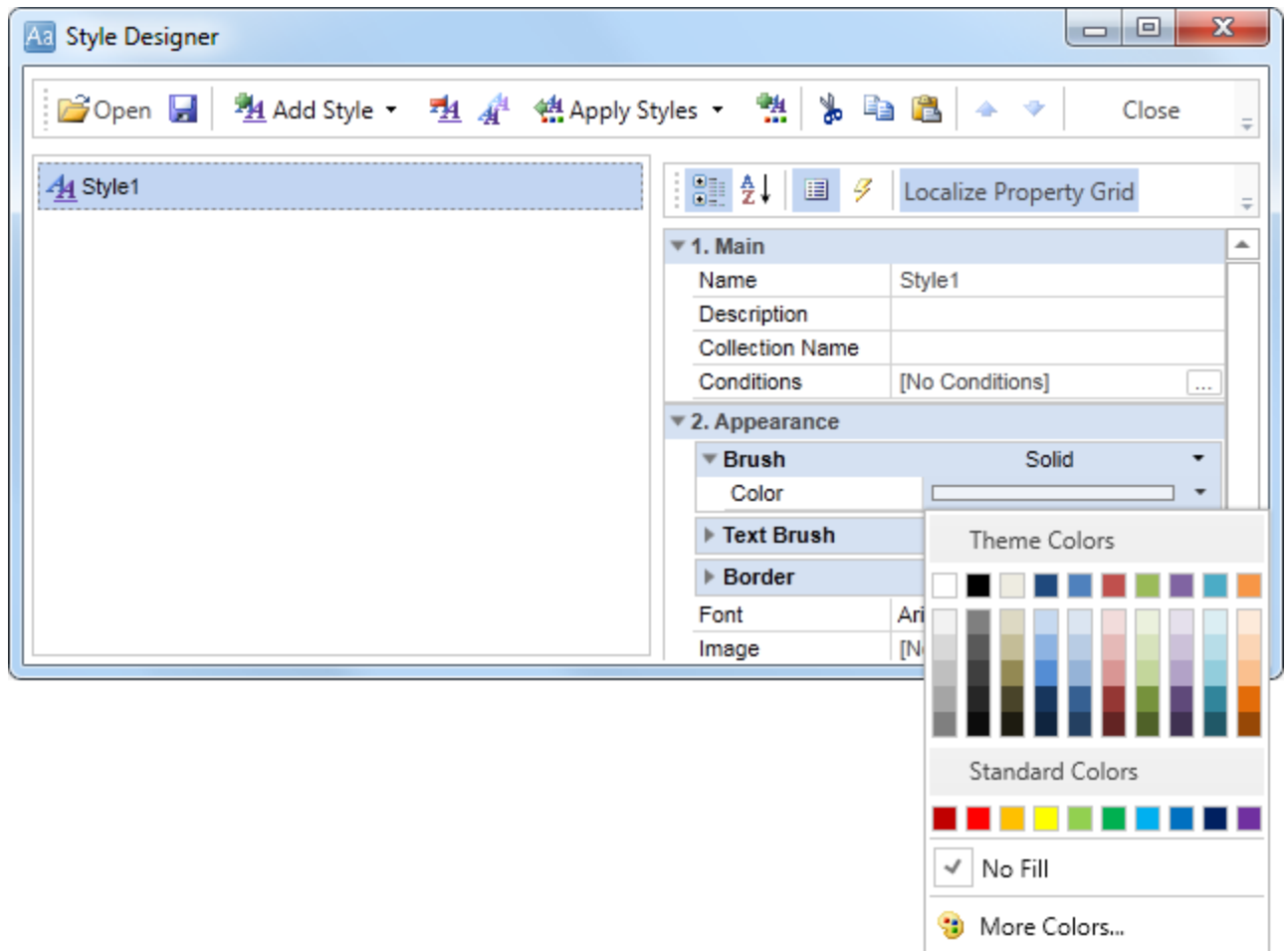|			
Alfreds Futterkiste	Berlin	Chai	18
Ana Trujillo Emparedados y helados	México D.F.	Chang	19
Antonio Moreno Taquería	México D.F.	Aniseed Syrup	10
Around the Horn	London	Chef Anton's Cajun Seasoning	22
Berglunds snabbköp	Luleå	Chef Anton's Gumbo Mix	21,35
Blauer See Delikatessen	Mannheim	Grandma's 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	NuNuCaNuß-Nougat-Creme	14
France restauration	Nantes	Gumbär Gummibärchen	31,23
Franchi S.p.A.	Torino	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.
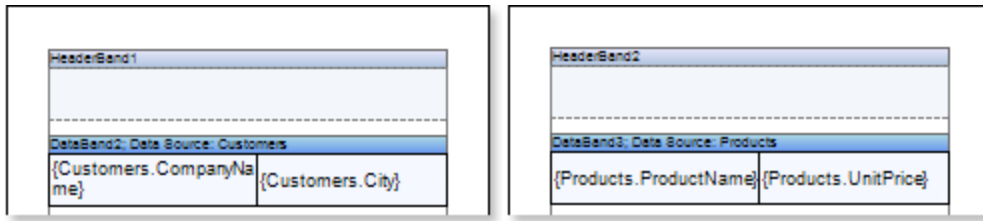
18. Put a text component with expression where the expression of the text component in the **HeaderBand** will be the page title.

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 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
Família 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**, 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
Família 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	NuNuCaNuß-Nougat-Creme	14

2.14 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:
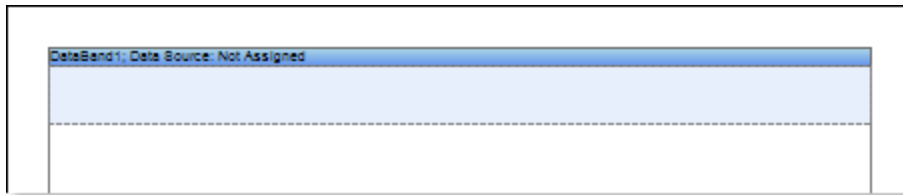
5. Edit **DataBand1**:
    5.1. Align the **DataBand1** by height;
    5.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
    5.3. Change the **DataBand1** background color;
    5.4. Enable **Borders** for the **DataBand1**, 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.
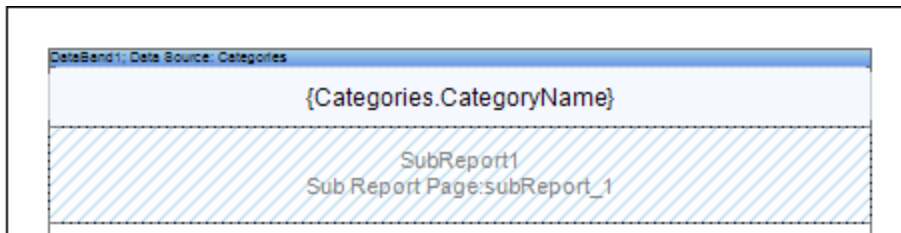
11. Go to the sub-report page;

12. Add to the **DataBand2** to the sub-report page.
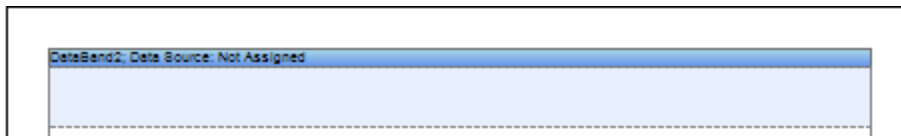
13. Edit **DataBand2**:

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.
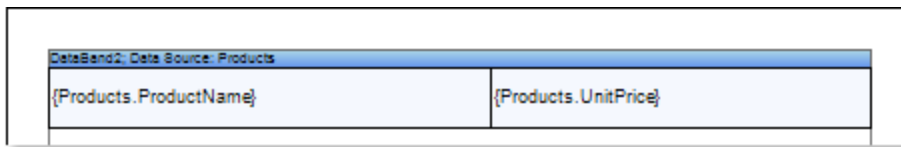
19. Click the **Preview** button or call **Viewer**, using the **Preview** menu item to see how the report will look like:

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**, 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:

2.15 Report with Empty Band

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:

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 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	

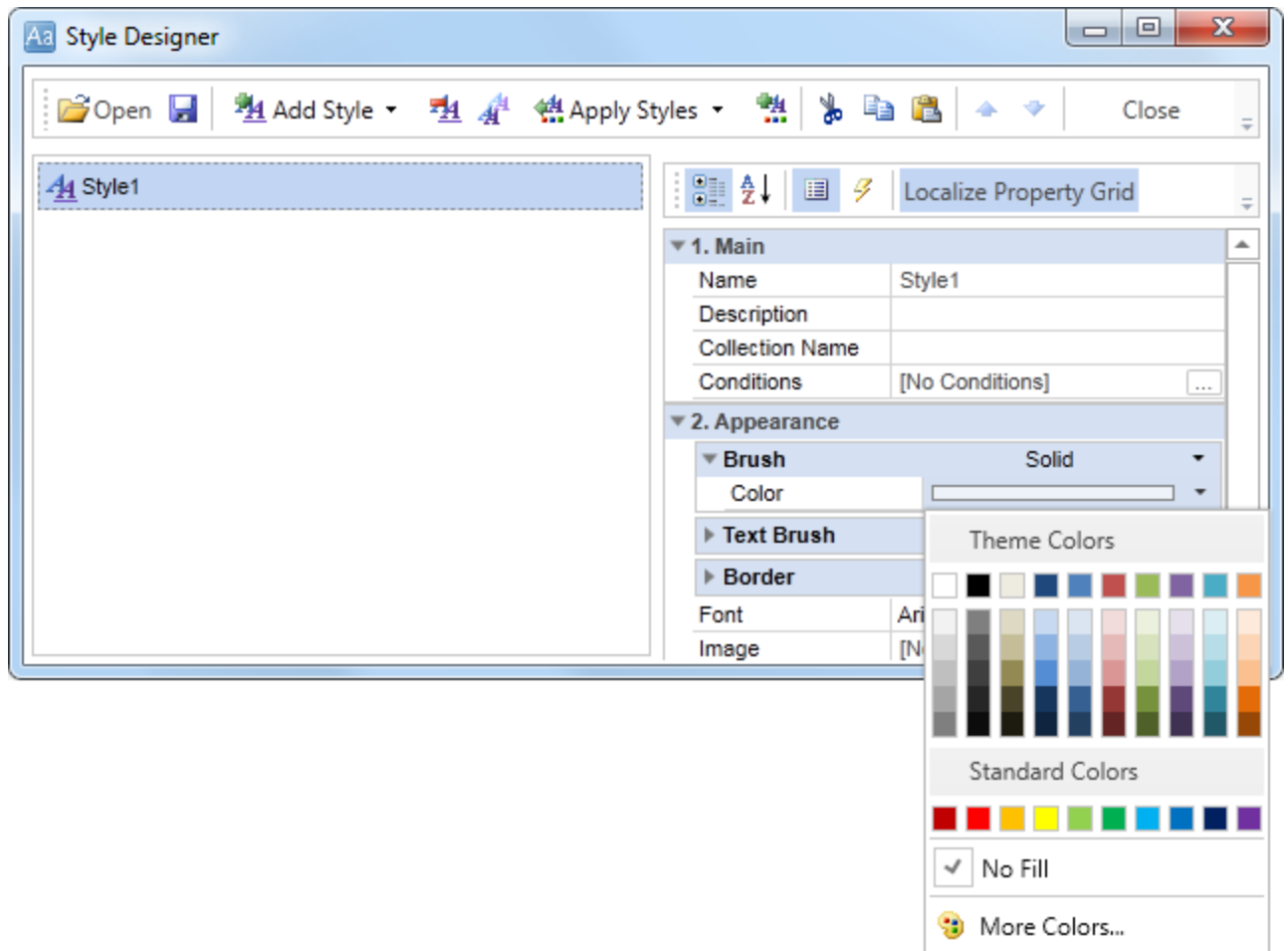As can be seen in the picture above blank lines will be numbered and output in the report.

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**, clicking the **Preview** menu item. The picture below shows a sample of a rendered report:

2.16 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.
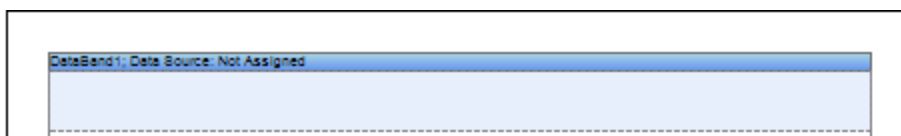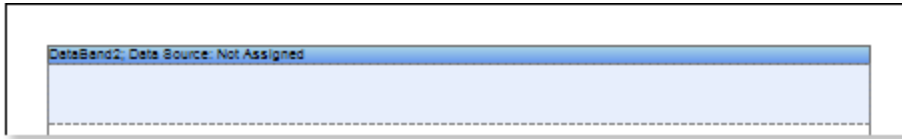
4. Edit **DataBand1** and **DataBand2**:
    4.1. Align the **DataBands** vertically;
    4.2. Change the value of the required properties;
    4.3. Change the background color of the **DataBand**;
    4.4. If necessary, set **Borders** of the **DataBand**;

5. Define a data source for **DataBands** using the **Data Source** property:
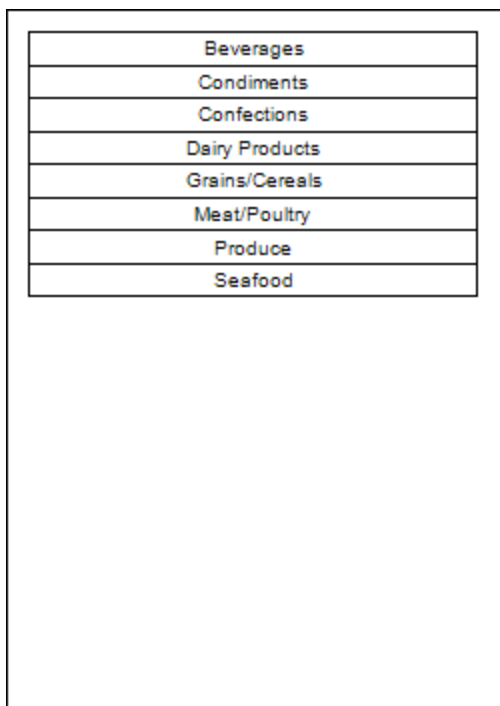
6. Put the text components with expressions. Where the expression is a reference to the data field. For example: put the text component with the **{Categories.CategoryName}** expression in the **DataBand1**, and put two text components with the **{Products.ProductName}** and **{Products.UnitePrice}** 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.
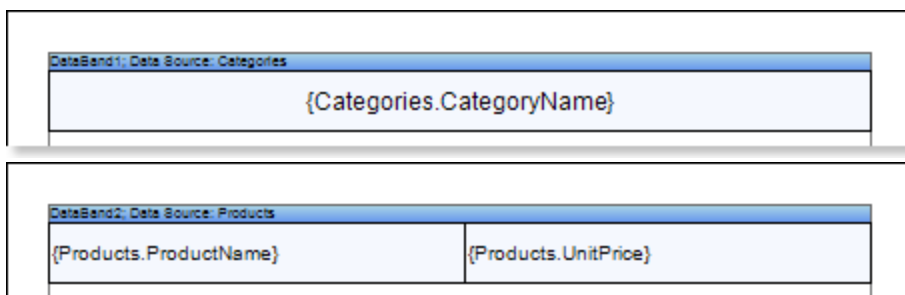
8. Select a text component in the **DataBand1**;
9. Set the **Interaction.Drill-Down Enabled** to **true**;
10. Set the Interaction.Drill-Down Page to **Page2**;
11. Edit **Drill-Down Parameter 1** for the text component of the **DataBand 1**:

11.1. The **Name** property should be set to **CategoryID**;

11.2. The **Expression** property should be set to **Categories.CategoryID**;

12. Set filter in the **DataBand2**, in this case, we specify the **(int) this ["CategoryID"] == Products.CategoryID** expression;

13. Click the **Preview** button or invoke the **Viewer**, 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:

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 Lager	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.

HeaderBand1	
DataBand2: Data Source: Products	
{Products.ProductName}	{Products.UnitPrice}

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.

ProductName	UnitPrice
{Products.ProductName}	{Products.UnitPrice}

19. Click the **Preview** button or invoke the **Viewer**, 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:

Page
Beverages
Condiments
Confections
Dairy Products
Grains/Cereals
Meat/Poultry
Produce
Seafood

Page 2 (Drill-Down Page)

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
Sirop d'érable	28,5
Vegie-spread	43,9
Louisiana Fiery Hot Pepper Sauce	21,05
Louisiana Hot Spiced Okra	17
Original Frankfurter grüne Soße	13

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**:

Page		Page 2 (Drill-Down Page)	
Beverages		**ProductName**	**UnitPrice**
Condiments		Pavlova	17,45
Confections		Teatime Chocolate Biscuits	9,2
Dairy Products		Sir Rodney's Marmalade	81
Grains/Cereals		Sir Rodney's Scones	10
Meat/Poultry		NuNuCa Nuß-Nougat-Creme	14
Produce		Gumbär Gummibärchen	31,23
Seafood		Schoggi Schokolade	43,9
		Zaanse koeken	9,5
		Chocolade	12,75
		Maxilaku	20
		Valkoinen suklaa	16,25
		Tarte au sucre	49,3
		Scottish Longbreads	12,5

2.17   Drill-Down Report Using External Report

For better understanding this step-by-step instruction, please watch the <u>video file</u>.

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**, 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;
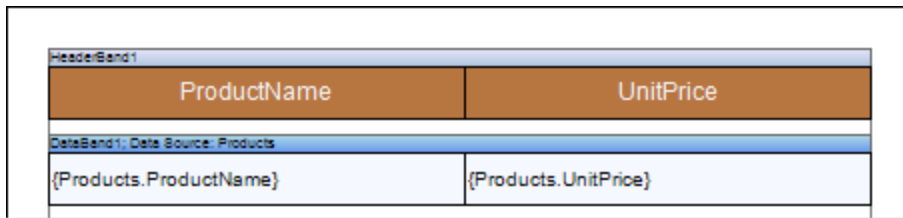
11.3. Change the background of the band;

11.4. Enable **Borders**, if required;

11.5. Set the border color.

HeaderBand1	
DataBand1; Data Source: Products	
{Products.ProductName}	{Products.UnitPrice}

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.

HeaderBand1	
ProductName	UnitPrice
DataBand1; Data Source: Products	
{Products.ProductName}	{Products.UnitPrice}

14. Click the **Preview** button or invoke the **Viewer**, 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**;
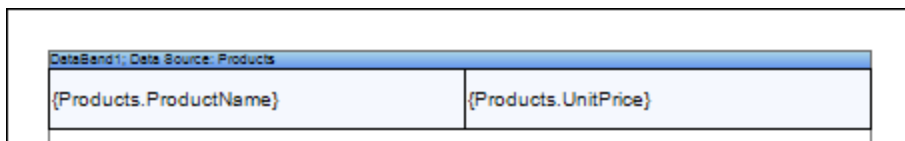   4.4. Set **Borders**, if required;
    4.5. Set the border color.

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**, 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

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**, 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**, 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 | Drill-Down Report

Beverages
Condiments
Confections
Dairy Products
Grains/Cereals
Meat/Poultry
Produce
Seafood

ProductName	UnitPrice
Mishi Kobe Niku	97
Alice Mutton	39
Thüringer Rostbratwurst	123,79
Perth Pasties	32,8
Tourtière	7,45
Pâté chinois	24

2.18   Report without Bands

For better understanding this step-by-step instruction, please watch the <u>video file</u>.

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**, clicking the **Preview** menu item:

8. Go back to the report template;
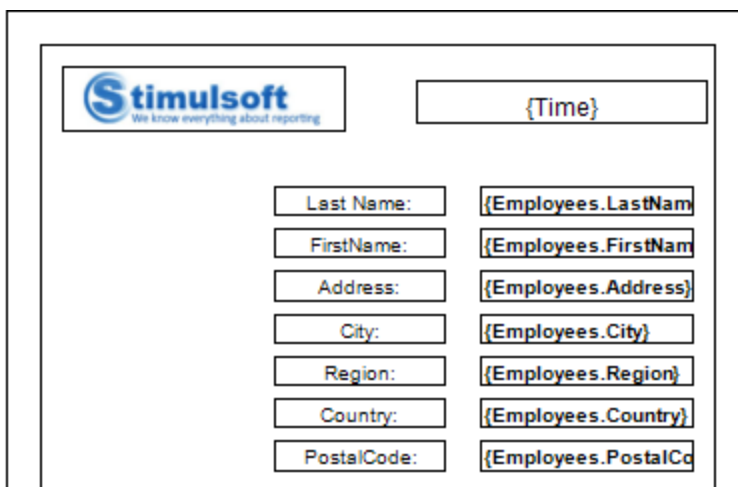
9. Disable **Borders** of all components. Enable bottom borders for **9-15 TextBoxes**:

10. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item.

2.19 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
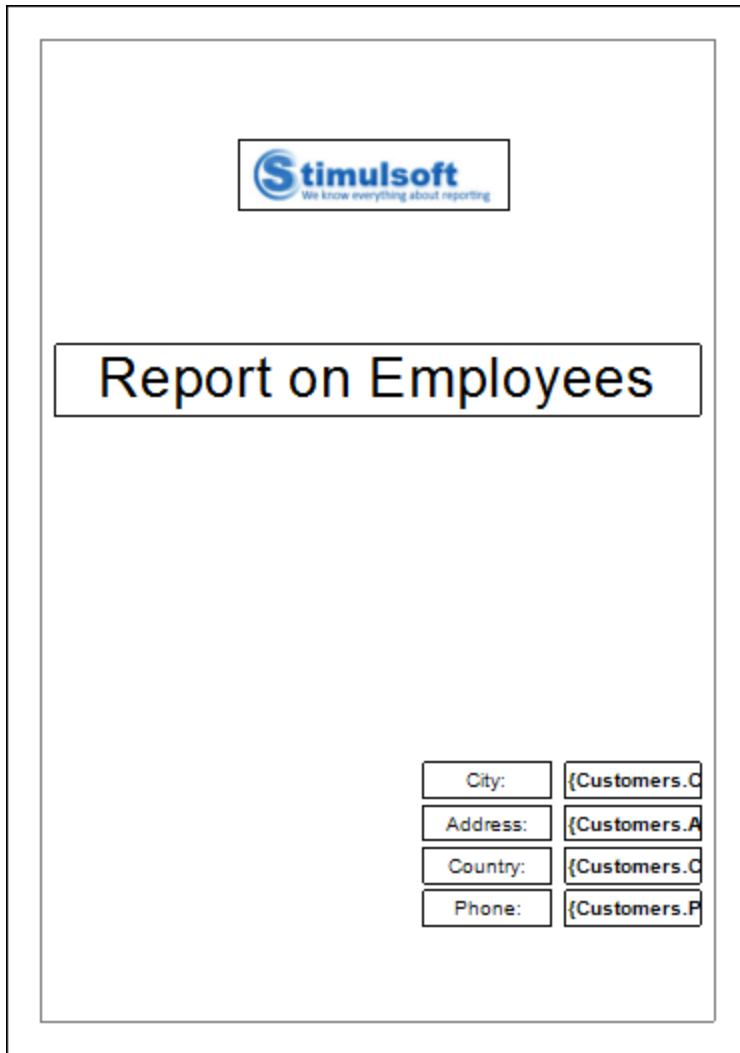
1. Run the designer;
2. Connect the data:
     2.1. Create a **New Connection**;

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**, 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.
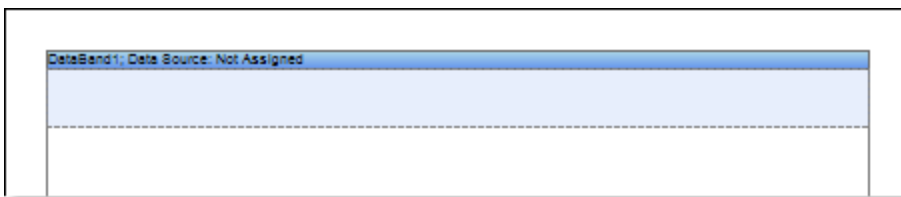
2. Edit **DataBand**:
    2.1. Align the **DataBand** by height;

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:

Data Source	Employees	...

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.

DataBand1: Data Source: Employees		
{Employees.FirstName}	{Employees.LastName}	{Employees.BirthDate}
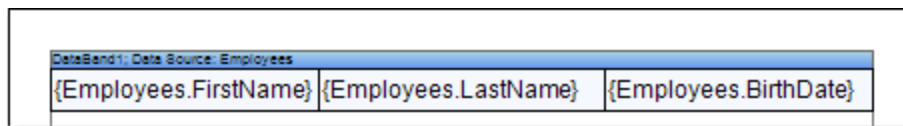
6. Add other bands to the report template, for example, the **HeaderBand**;

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 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**, 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.

Stimulsoft
We know everything about reporting

Report on Employees

City:	**Berlin**
Address:	**Obere Str. 57**
Country:	**Germany**
Phone:	**030-0074321**
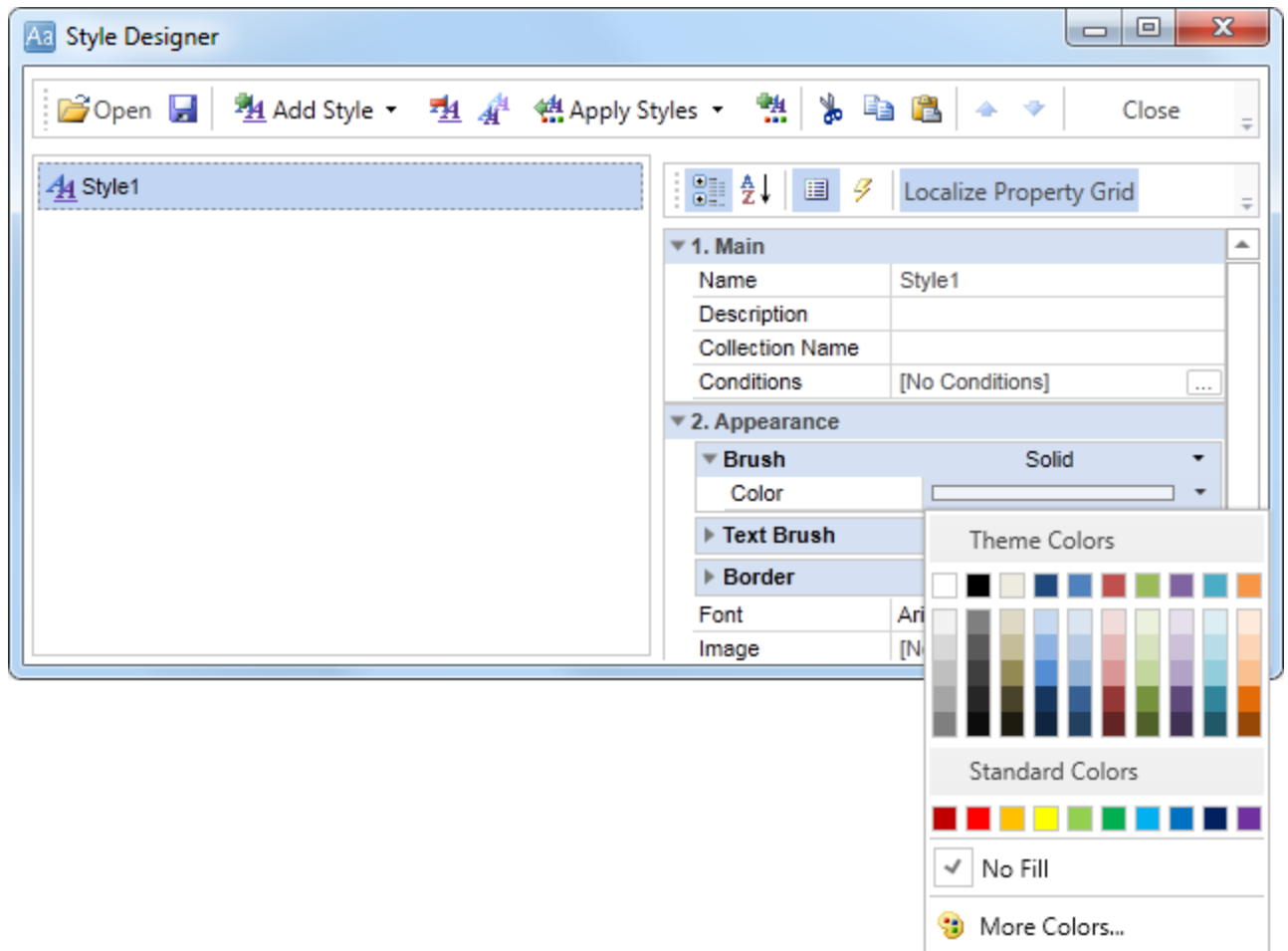
FirstName	LastName	BirthDate
Nancy	Davolio	08.12.1948 0:00:00
Andrew	Fuller	19.02.1952 0:00:00
Janet	Leverling	30.08.1963 0:00:00
Margaret	Peacock	19.09.1937 0:00:00
Steven	Buchanan	04.03.1955 0:00:00
Michael	Suyama	02.07.1963 0:00:00
Robert	King	29.05.1960 0:00:00
Laura	Callahan	09.01.1958 0:00:00
Anne	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**, clicking the **Preview** menu item.

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

2.20   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.
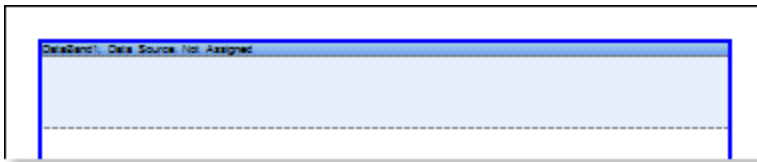
5. Edit **DataBand**:
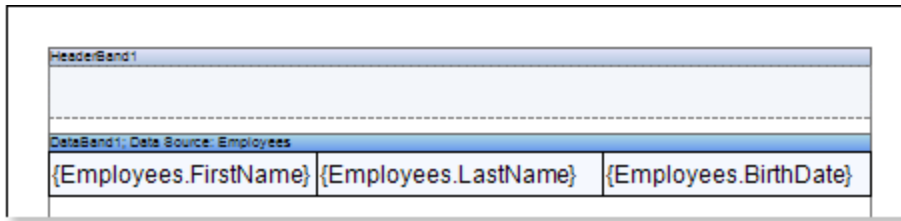
    5.1. Align the **DataBand** by height;

    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.
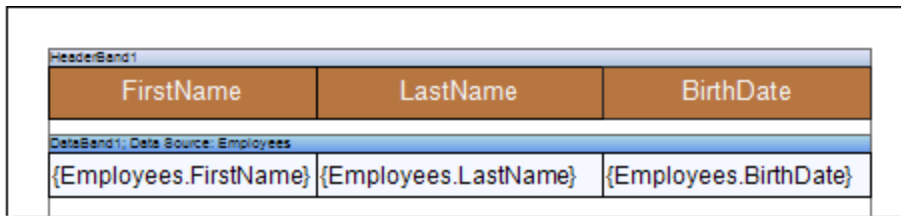
    8.8. Change the border color.
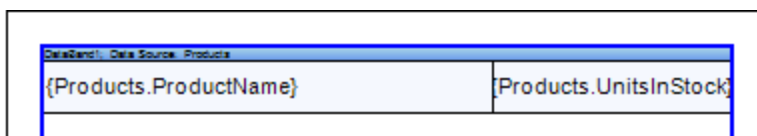
9. Click the **Preview** button or invoke the **Viewer**, pressing **F5** or clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields. Data will be output in consecutive order from the database that was defined for this report. The amount of copies of the **DataBand** in the rendered report

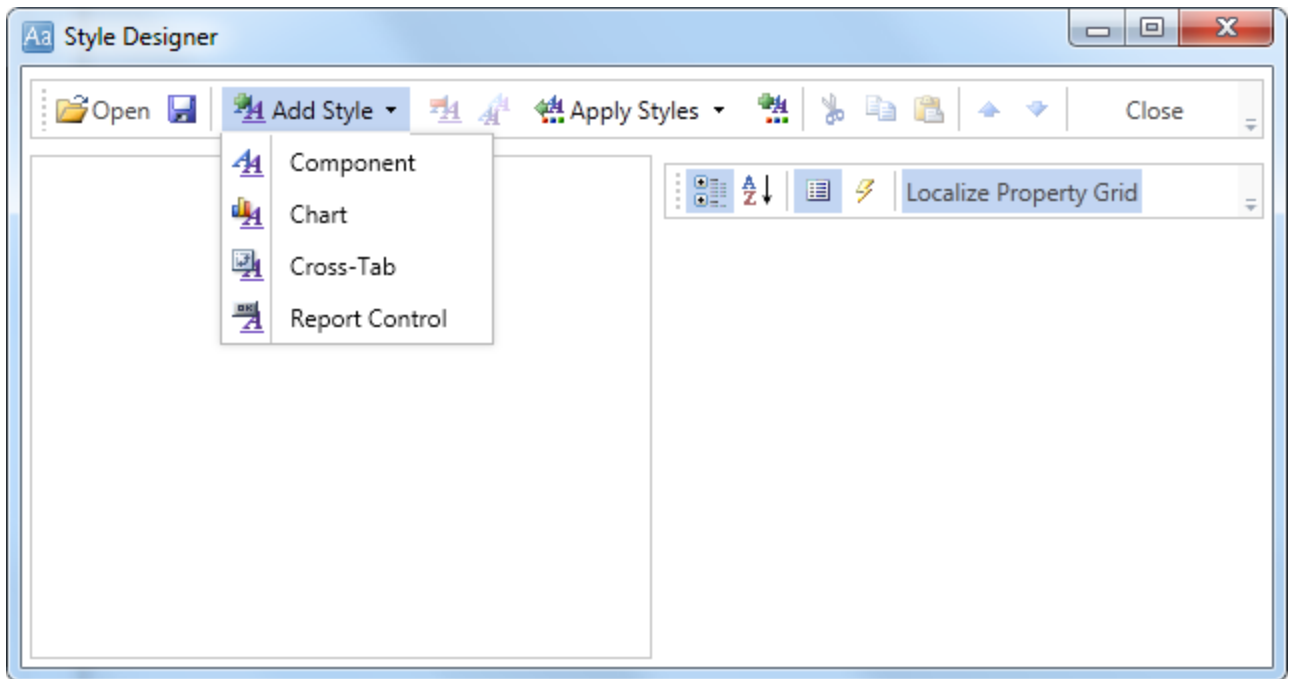will be the same as the amount of data rows in the database.

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 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
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.
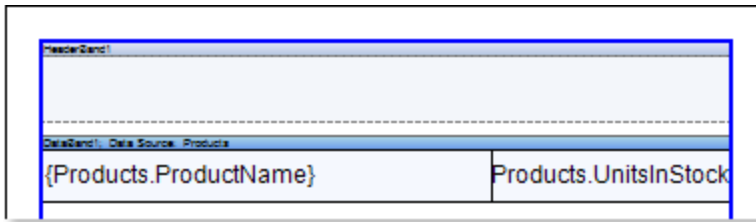
12. Put text components with expressions in the band. The expression in the text component is a header in the **HeaderBand**.

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**, 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**, 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

2.21   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.

DataBand1: Data Source: Employees	
{Employees.FirstName}	{Employees.City}

8. Click the **Preview** button or invoke the **Viewer**, 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**, 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**, 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

2.22   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**;

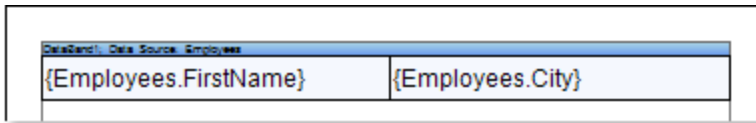3. Put the **DataBand** on a page of a report template.

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**, clicking the **Preview** menu item.

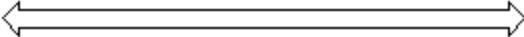|	
Nancy	Seattle
Andrew	Tacoma
Janet	Kirkland
Margaret	Redmond
Steven	London
Michael	London
Robert	London
Laura	Seattle
Anne	London

9. Go back to the report template.

10. Add the **Shape** component to a report template 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:
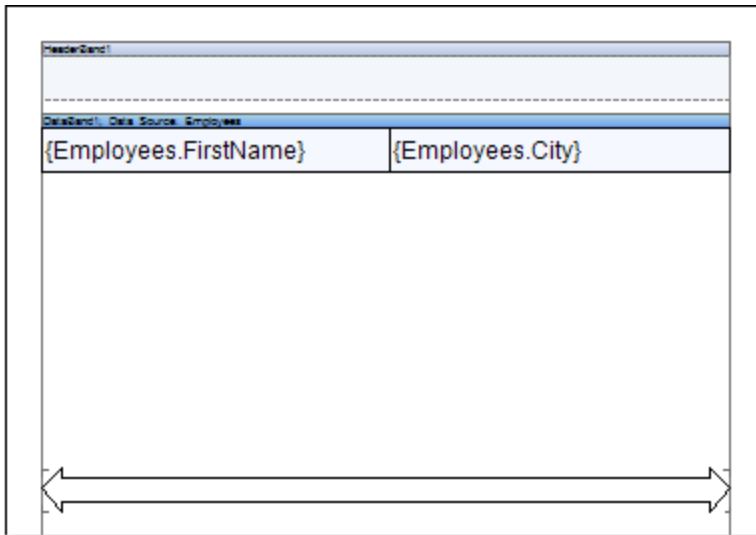
DataBand1: Data Source: Employees

{Employees.FirstName}   {Employees.City}

11. Click the **Preview** button or invoke the **Viewer**, 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**, 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**, 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

2.23   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.
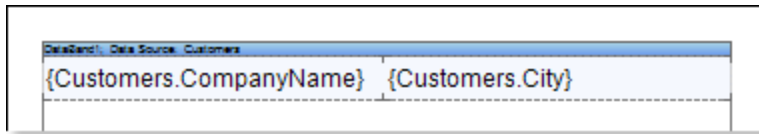
4. Add **GroupHeaderBand** and **GroupFooterBand** to a report template. The **GroupHeaderBand** should be placed above the **DataBand** to which it applies. The **GroupFooterBand** should be placed below the **DataBand**. And it is meant exactly the **DataBand**, that is associated with the **GroupHeaderBand**. Each **GroupFooterBand**, refers to a certain **GroupHeaderBand**. The **GroupFooterBand** will not be output without the **GroupHeaderBand**.
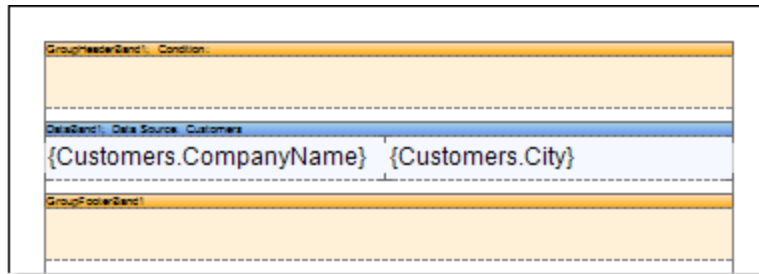
5. Edit the **GroupHeaderBand** and the **GroupFooterBand**:
    5.1. Align them by height;
    5.2. Change the values of the required properties. For example, set the **KeepGroupHeaderTogether** property for the **GroupHeaderBand** to **true**, if you want the **GroupHeaderBand** be kept with the group. 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.
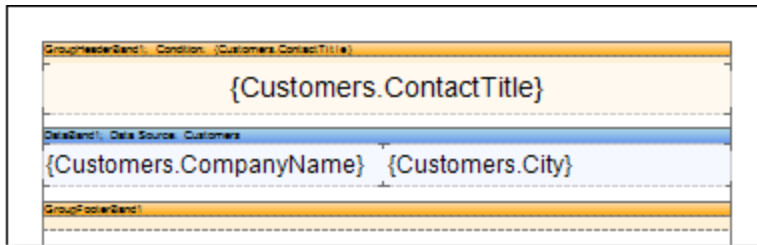
8. Edit expressions, and text components:

    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**, clicking the **Preview** menu item. After rendering a report all references to data fields will be changed on data from specified fields.

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**.
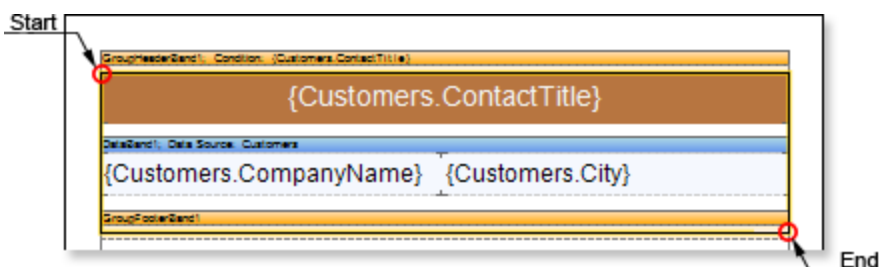
12. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. After rendering a report all references to data fields will be changed on data from specified fields. The picture below shows a rendered report page with grouping and the rendered **Rectangle** cross-primitive:

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**, 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.

2.24 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**, clicking the **Preview** menu item. After rendering a report all references to data fields will be changed on data from specified fields.

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**, 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

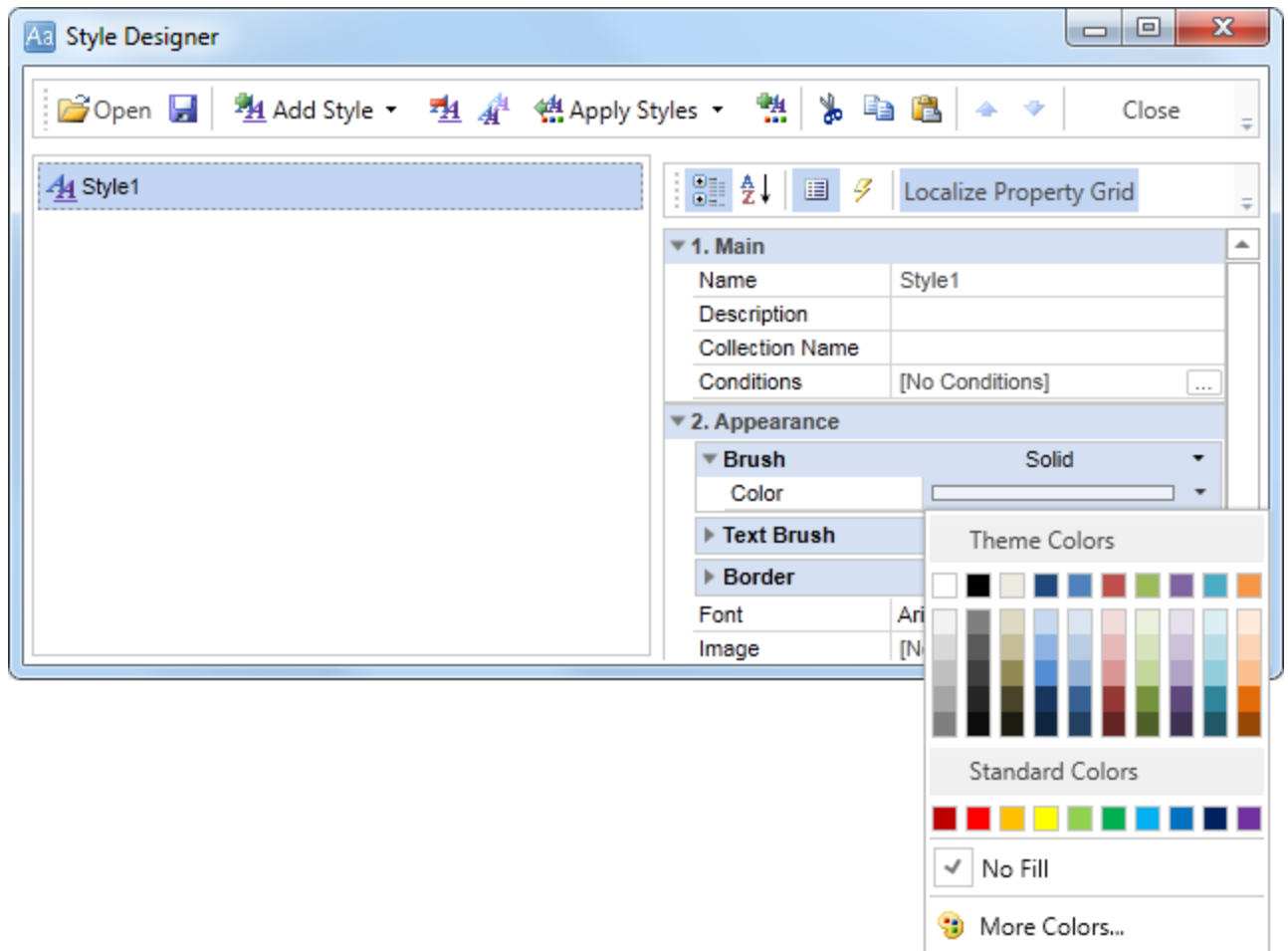| Assistant Sales Agent |
|---|
| Assistant Sales Representative |
| Marketing Assistant |
| Marketing Manager |
| Order Administrator |
| Owner |
| Owner/Marketing Assistant |
| Sales Agent |
| Sales Associate |
| Sales Manager |

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**, 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

2.25 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.
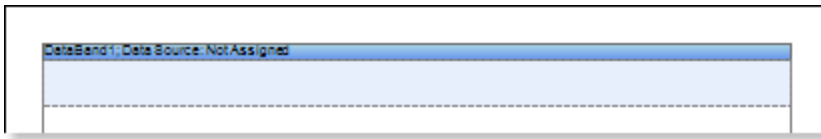
4. Edit **DataBand**:
    4.1. Align the **DataBand** by height;
    4.2. Change values of band properties. For example, set the **Can Break** property to **true**, if you wish the data band to be broken;
    4.3. Change the **DataBand** background;
    4.4. Enable **Borders** for the **DataBand**, if 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**, 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.

Chal	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.

HeaderBand1		
DataBand1: Data Source: Products		
{Products.ProductName}	{Products.QuantityPerUnit}	{Products.UnitsInStock}

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.

ProductName	QuantityPerUnit	UnitsIn Stock
{Products.ProductName}	{Products.QuantityPerUnit}	{Products.UnitsInStock}

14. Click the **Preview** button or invoke the **Viewer**, 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 Matjeshering	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 done. Set the **Interaction.Sorting Column** property to **DataBand1.ProductName**;

18. Click the **Preview** button or invoke the **Viewer**, 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 Matjeshering	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**, 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

2.26   Report With Dynamic Collapsing in Preview

The report with dynamic collapsing is an interactive report in what items can collapse/expand its contents by clicking the title of the block. In order to create a report with dynamic folding in the preview window, you should do the following:
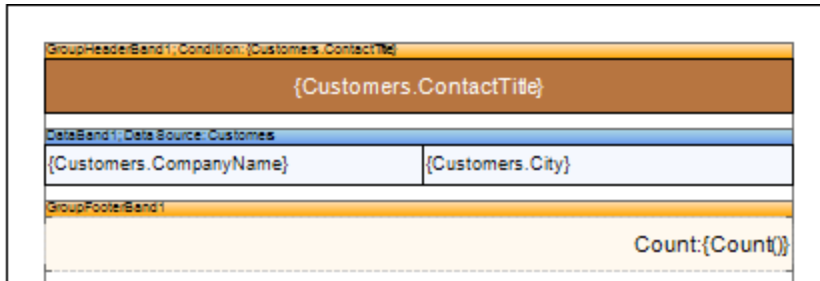
Run the designer;
Connect the data:
2.1. Create a **New Connection**;
2.2. Create a **New Data Source**;

3. Create a report or open a previously designed one. For example, open a report with grouping, which was reviewed in the chapter "Report from the groups." The picture below shows a report template with groups:

4. Render your report. Click on the **Preview** tab or invoke the report viewer clicking the Preview in the menu. After rendering a report, all references to the data field will be replaced with data from these fields. The picture below shows a report page with the grouping:

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**. In this case, set this property to **{GroupLine!=1}**, i.e. all the groups except the first one will be collapsed:

9. Render the report. Click on the **Preview** tab or invoke the report viewer clicking the Preview in the menu. After rendering a report, all references to the data field will be replaced with data from these fields. The picture below shows the rendered page of the report:

In order to expand or collapse the group, select the **GroupHeaderBand** in the rendered report. If you want to collapse the group together with the the group footer you should set the **Interaction.Collapse Group Footer** property set to **true**. The picture below shows a rendered report page with the collapsed items:

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**, clicking the **Preview** menu item.
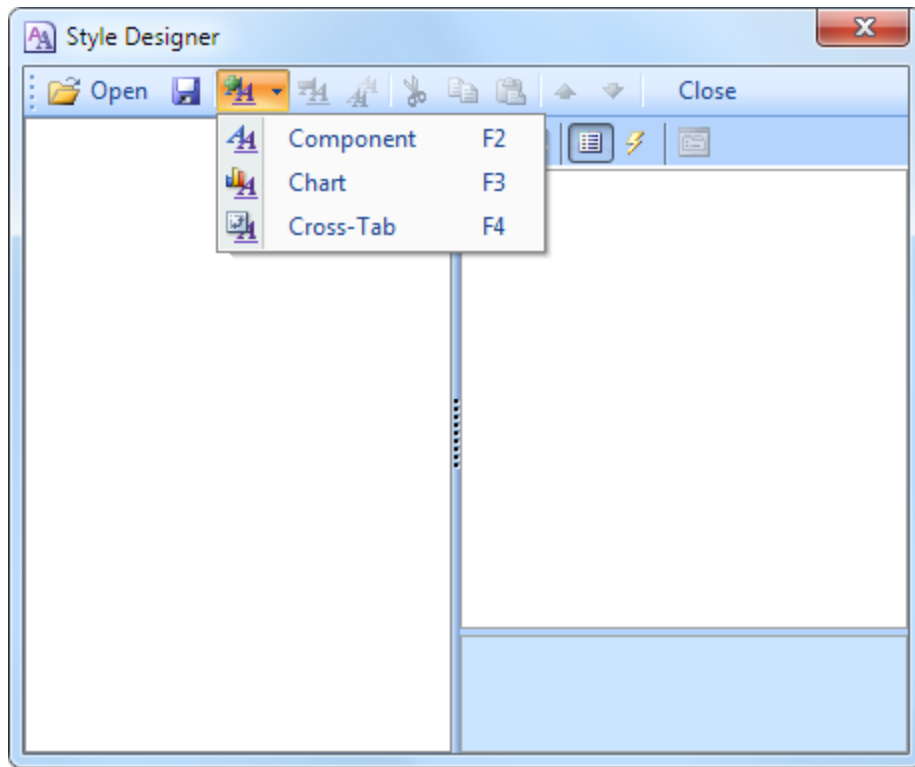
Accounting Manager	
Bottom-Dollar Markets	Tsawassen
Romero y tomillo	Madrid
Que Delicia	Rio de Janeiro
FISSA Fabrica Inter. Salchichas S.A.	Madrid
Suprêmes délices	Charleroi
QUICK-Stop	Cunewalde
LILA-Supermercado	Barquisimeto
Wartian Herkku	Oulu
Hanari Carnes	Rio de Janeiro
Vins et alcools Chevalier	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

2.27 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.
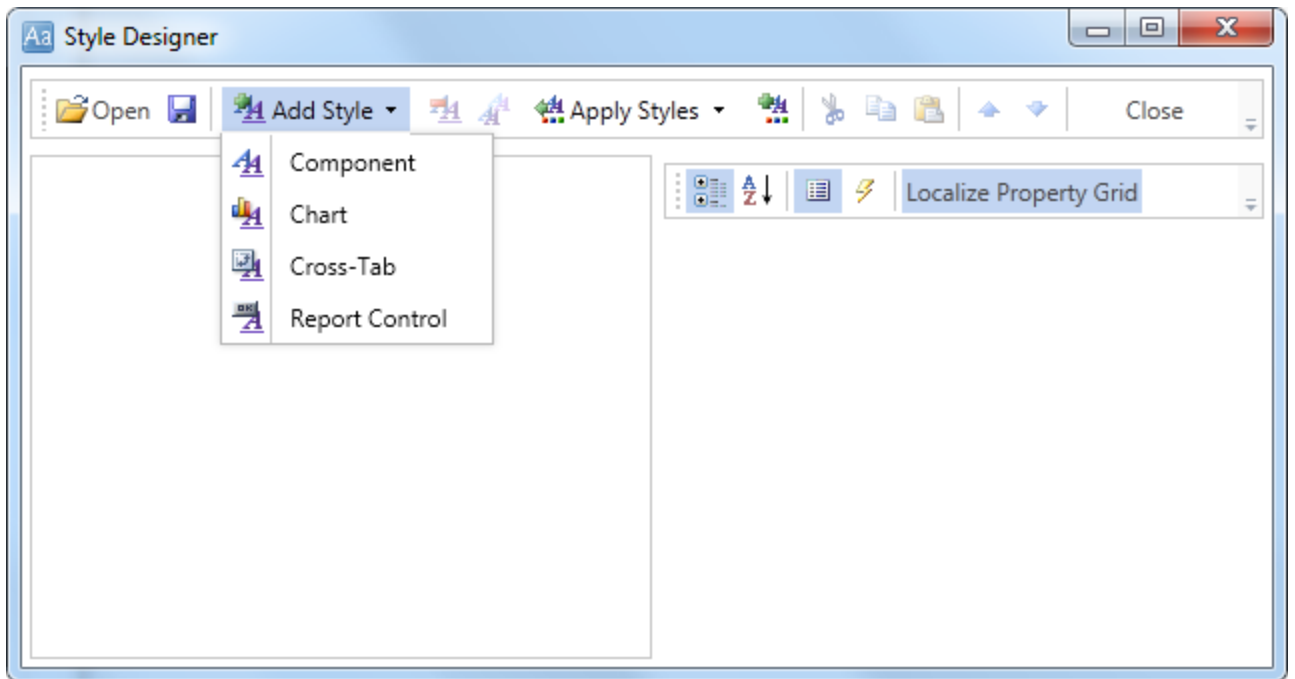

8. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified

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.
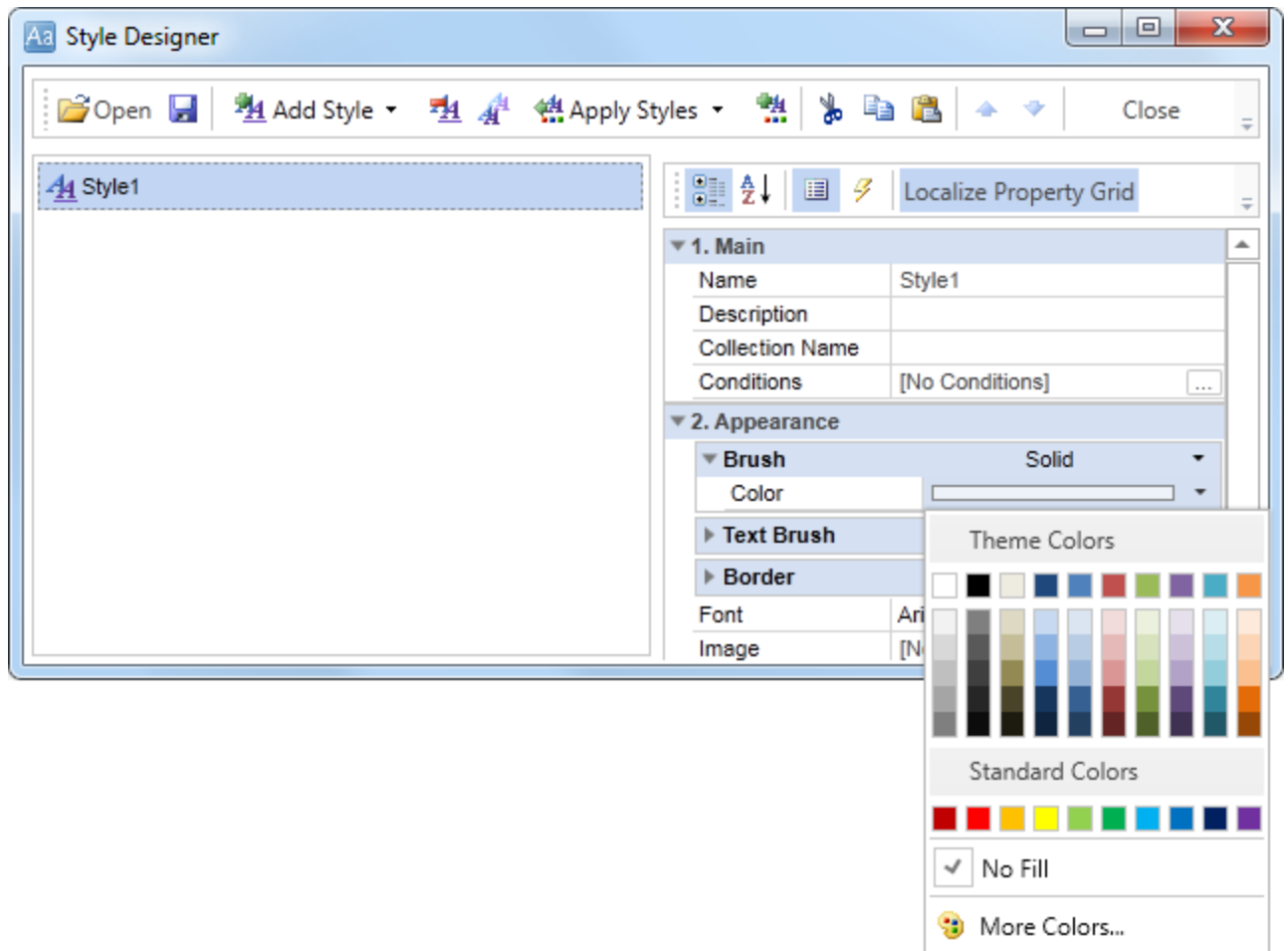
CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
	ContactName:	Maria Anders
	Phone:	030-0074321
	Fax:	030-0076545
Ana Trujillo Emparedados y helados	México D.F.	Mexico
	ContactName:	Ana Trujillo
	Phone:	(5) 555-4729
	Fax:	(5) 555-3745
Antonio Moreno Taqueria	México D.F.	Mexico
	ContactName:	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:	Martin 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**, clicking the **Preview** menu item.

2.28   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.
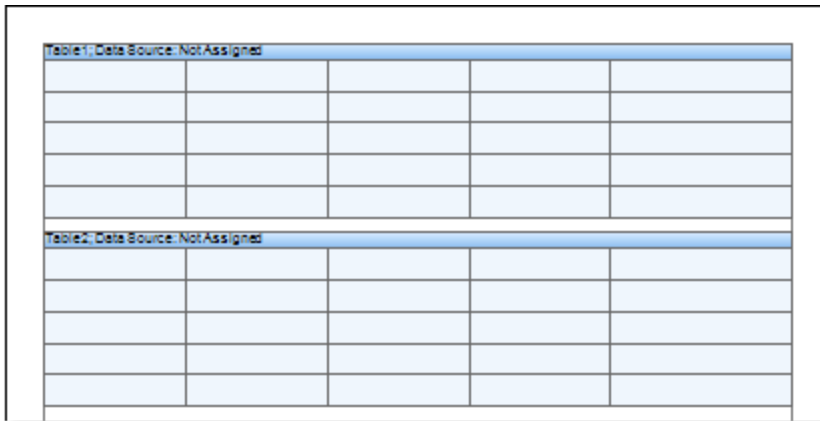4. Put two **Table** components on a page of a report template.

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:

Data Relation	Categories	...

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**, clicking the **Preview** menu item. After rendering all references to data fields will be changed on data form specified fields.

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**, clicking the **Preview** menu item.

2.29   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
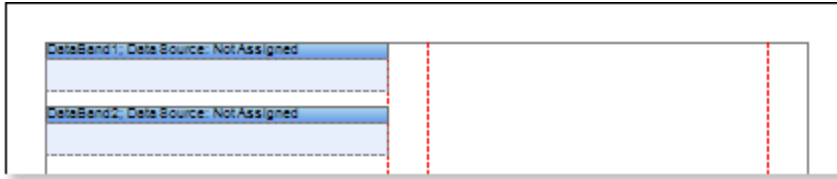
6. Edit **DataBand1** and **DataBand2**:

    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}**.
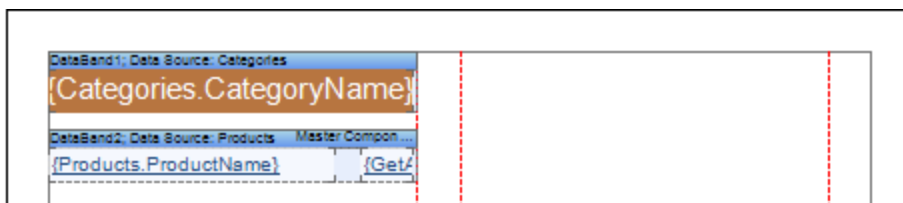
Creating a master list

11. Create a second page in the report template;
12. Put two **DataBands** on the page of the report template.
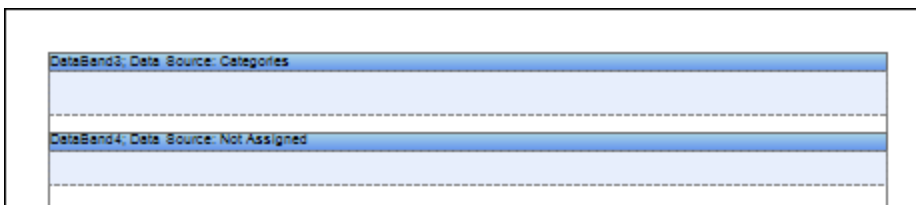
13. Edit **DataBand3** and **DataBand4**:
    13.1. Align the **DataBand** by height;
    13.2. Change the values of the required properties. For example set the **Print if Detail Empty** property of the **DataBand3**, which is the **Master** component in the Master-Detail report to **true**, if you 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:

Data Source	Categories	...
Data Source	Products	...

15. Fill the **DataRelation** property of the **DataBand4**, which is the **Detail** component:

Data Source	Categories	...

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**, 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**, 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.
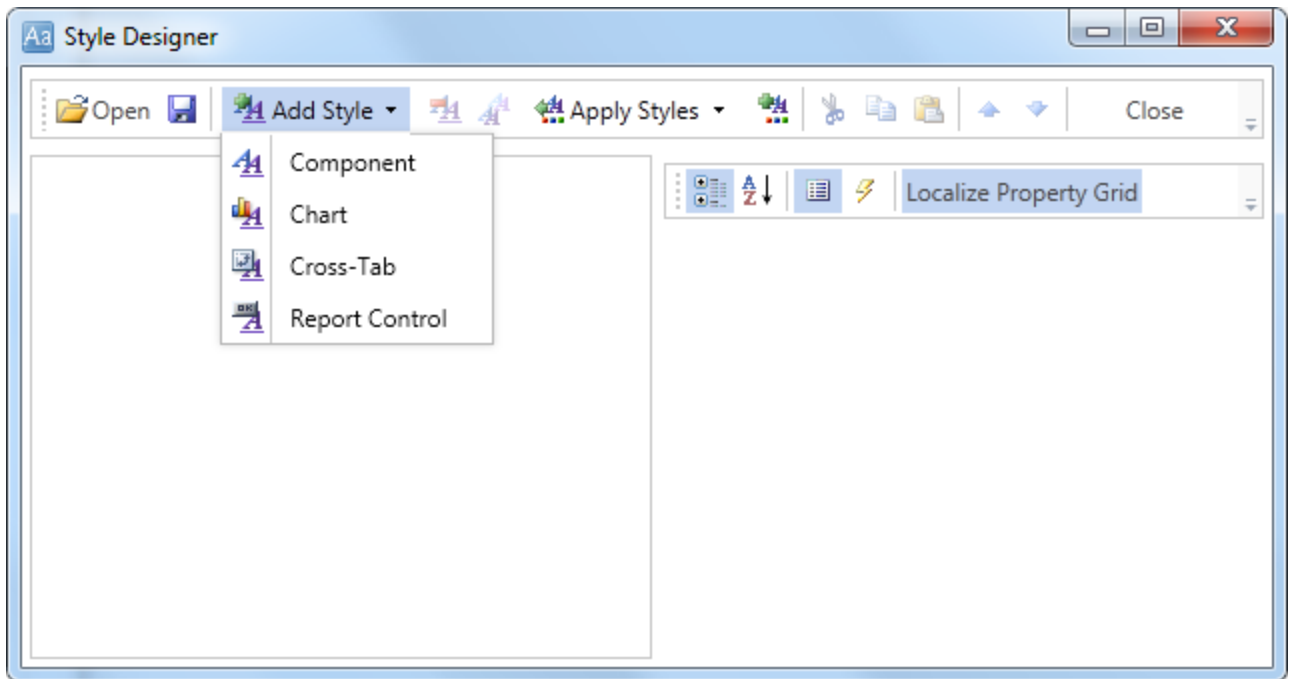
Beverages

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

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**, clicking the **Preview** menu item.

Beverages		Schoggi Schokolade	4
Chai	3	Zaanse koeken	4
Chang	3	Chocolade	4
Guaraná Fantástica	3	Maxilaku	4
Sasquatch Ale	3	Valkoinen suklaa	4
Steeleye Stout	3	Tarte au sucre	4
Côte de Blaye	3	Scottish Longbreads	4
Chartreuse verte	3	**Dairy Products**	
Ipoh Coffee	3	Queso Cabrales	4
Laughing Lumberjack Lager	3	Queso Manchego La Pastora	4
Outback Lager	3	Gorgonzola Telino	4
Rhönbräu Klosterbier	3	Mascarpone Fabioli	4
Lakkalikööri	3	Geitost	4
Condiments		Raclette Courdavault	4
Aniseed Syrup	3	Camembert Pierrot	4
Chef Anton's Cajun Seasoning	3	Gudbrandsdalsost	4
Chef Anton's Gumbo Mix	3	Flotemysost	4
Grandma's Boysenberry Spread	3	Mozzarella di Giovanni	4
Northwoods Cranberry Sauce	3	**Grains/Cereals**	
Genen Shouyu	3	Gustaf's Knäckebröd	4
Gula Malacca	3	Tunnbröd	4
Sirop d'érable	3	Singaporean Hokkien Fried Mee	4
Vegie-spread	3	Filo Mix	4
Louisiana Fiery Hot Pepper Sau	3	Gnocchi di nonna Alice	5
Louisiana Hot Spiced Okra	3	Ravioli Angelo	5
Original Frankfurter grüne Soße	3	Wimmers gute Semmelknödel	5
Confections		**Meat/Poultry**	
Pavlova	3	Mishi Kobe Niku	5
Teatime Chocolate Biscuits	3	Alice Mutton	5
Sir Rodney's Marmalade	3	Thüringer Rostbratwurst	5
Sir Rodney's Scones	4	Perth Pasties	5
NuNuCa Nuß-Nougat-Creme	4	Tourtière	5
Gumbär Gummibärchen	4	Pâté chinois	5

2.30 Invoice Report

The invoice is most often used in accounting for the tax (customs) control or in the international supply of goods. This document usually includes the cost of transportation, shipping operations, insurance, payment of export duties, as well as various taxes (fees), and more. If your activity requires constant creation of invoices, for optimization, time and cost savings, it is logical to assume that it is easier to create a document template. Using it, you change only the data, saving yourself from routine work to create the structure of the invoice and its design.

You can create templates and tools in many ways, but I want to help you save time in finding these resources. In this tutorial you will learn how to quickly create an invoice template, decorate it and get the finished document. This will take you some time. I will try as much as possible to describe in detail the process of creating such a report.

The product which is used in this tutorial is Stimulsoft Reports.Net which trial can be downloaded at http://www.stimulsoft.com/Downloads/ StimulsoftReports.Net_2012.1_Trial.zip .
The database to this tutorial is delivered with the product installation. I also attached the video file which shows how to create a report.
The ready invoice.mrt file is also attached to this article.

In order to create an invoice, you should do the following steps:

1. Run the designer;
2. Connect the data:
2.1. Create **New Connection**;
2.2. Create **New Data Source**;

3. Put the **DataBand** on the page of the report template;
4. Put the **HeaderBand** above the **DataBand**. The picture below shows an example of the report template with the bands on the page:
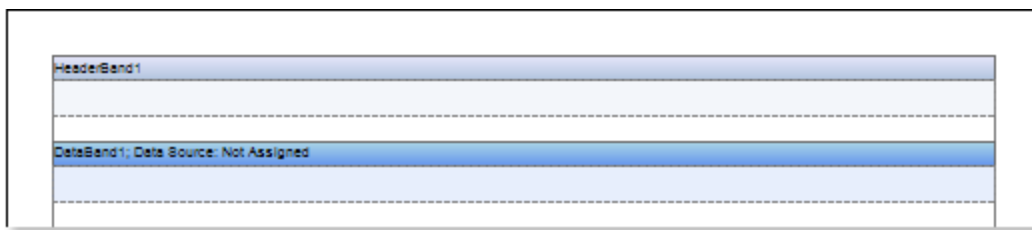
Edit the bands **DataBand** and **HeaderBand**:
5.1. Align them by height;
5.2. Set the properties of the **DataBand**. For example, set the **Can Break** property to **true**, if you want the band be broken;
5.3. Set the background color for the bands;
5.4. If necessary, set **Borders**;
5.5. Set the border color.

6. Specify the data source for the **DataBand** using the **Data Source** property from the object inspector:

7. Put text components in the **HeaderBand** with texts **Unit Name**, **Description**, **Qty**,

Item Price, **Total**;

8. Put text components in the **DataBand** with expressions. Where the expression is a reference to the data field. Put text components with the expressions: **{Products.ProductName}**, **{Products.QuantityPerUnit}**, **{Products.UnitsInStock}**, **{Products.UnitPrice}**, and **{Products.UnitsInStock * Products.UnitPrice}**;

9. Edit **Text** and **TextBox**:

9.1. Drag the text components on the **DataBand** and **HeaderBand** to the appropriate places;

9.2. Set the font parameters: size, style and color;

9.3. Align text components by height and width;

9.4. Set the background of text components;

9.5. Align text in text components;

9.6. Set the properties of text components. For example to set the **Word Wrap** property to **true**;

9.7. If necessary, include **Borders** of text components;

9.8. Set the border color.

The picture below shows the report template:

10. Click on the **Preview** button or invoke the report viewer, using the **Preview** item. After rendering a report, all references to the data fields will be replaced with data from the specified fields. That data will be taken sequentially from the data source that was specified for the given band. The number of copies of the **DataBand** in the rendered report will be equal to the number of rows in the data source. The picture below shows the rendered report:

Unit Name	Description	Qty	Item Price	Total
Chai	10 boxes x 20 bags	39	18	702
Chang	24 - 12 oz bottles	17	19	323
Aniseed Syrup	12 - 550 ml bottles	13	10	130
Chef Anton's Cajun Seasoning	48 - 6 oz jars	53	22	1166
Chef Anton's Gumbo Mix	36 boxes	0	21.35	0.00
Grandma's Boysenberry Spread	12 - 8 oz jars	120	25	3000
Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15	30	450
Northwoods Cranberry Sauce	12 - 12 oz jars	6	40	240
Mishi Kobe Niku	18 - 500 g pkgs.	29	97	2813
Ikura	12 - 200 ml jars	31	31	961

11. Go back to the report template;

12. Add the **FooterBand** on the report page and edit it;

13. Put text components in the band with the expression **Items per page: {cCount (DataBand1)}** and edit this text component;

14. Add **Rectangle**, so that the upper points are located on the **HeaderBand**, and the lower ones on the **FooterBand**;

15. Add cross-primitives, which start points are located at the top of the **HeaderBand**, and the end ones - on **FooterBand**. The picture below shows the report template with the **FooterBand**, rectangle and primitives:

16. Add the **ReportTitleBand** to the report template and **FooterBand** and edit them;

17. Put a text component in the **FooterBand** with the expression **Total: {Sum (Products.UnitsInStock * Products.UnitPrice)}**;

18. Put a text components in the **ReportTitleBand** with expressions:

18.1. The first text component has the text **BILL TO**;

18.2. The second one indicates **Name Street Address Address 2 City, ST ZIP Code**;

18.3. The third component with the text **SHIP TO**;

18.4. In the fourth component the text is the same as in the second one **Name Street Address Address 2 City, ST ZIP Code**;

18.5. Put the text **Invoice # 123456** in the next component;

18.6. Put the expression **Invoice date {Today.ToString ("d")}** in the sixth component in this band;

18.7. And in the last component put **Customer ID 123**;

The picture below shows a report template:

19. Click on the **Preview** button or invoke the report viewer, using the **Preview** item. After rendering a report, all references to the data fields will be replaced with data from the specified fields. That data will be taken sequentially from the data source that was specified for the given band. The number of copies of the **DataBand** in the rendered report will be equal to the number of rows in the data source. The picture shows a report with the report header and footer:

Adding styles

1. Go back to the report template;
2. Call the **Style Designer**;

The picture below shows the dialog **Styles Designer**:
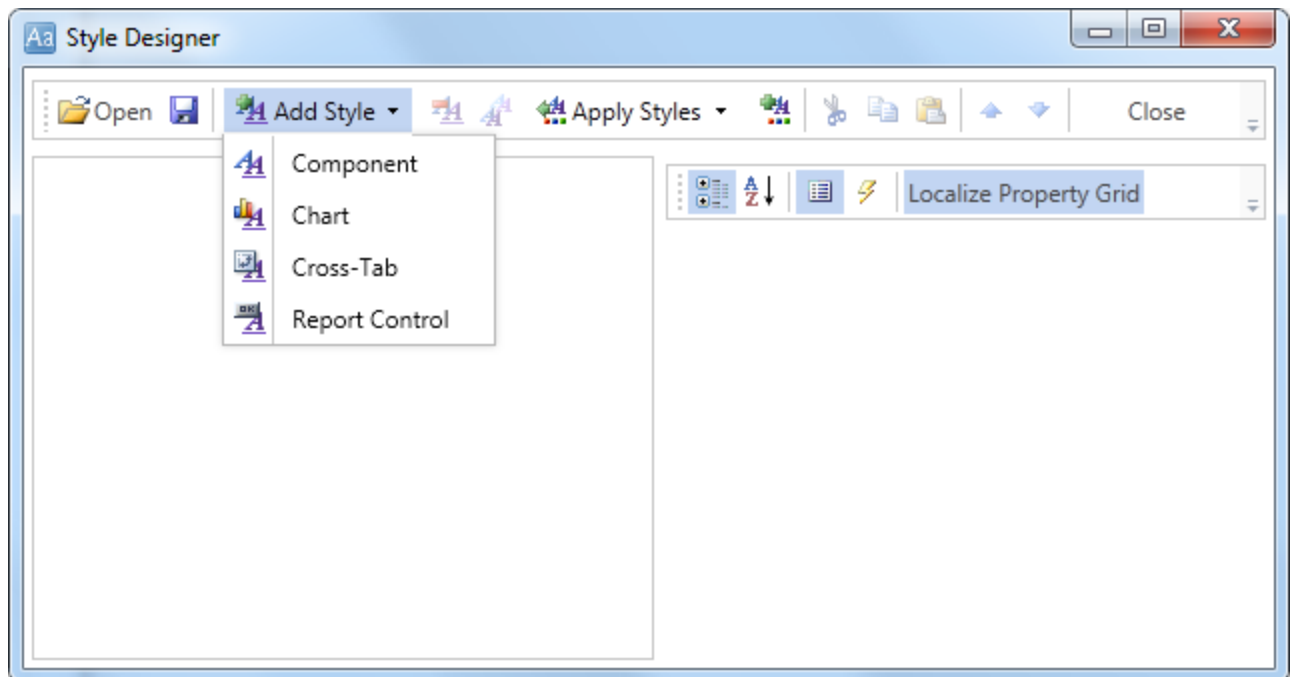
Click the **Add Style** button to start creating a style. Select **Component** from the drop

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
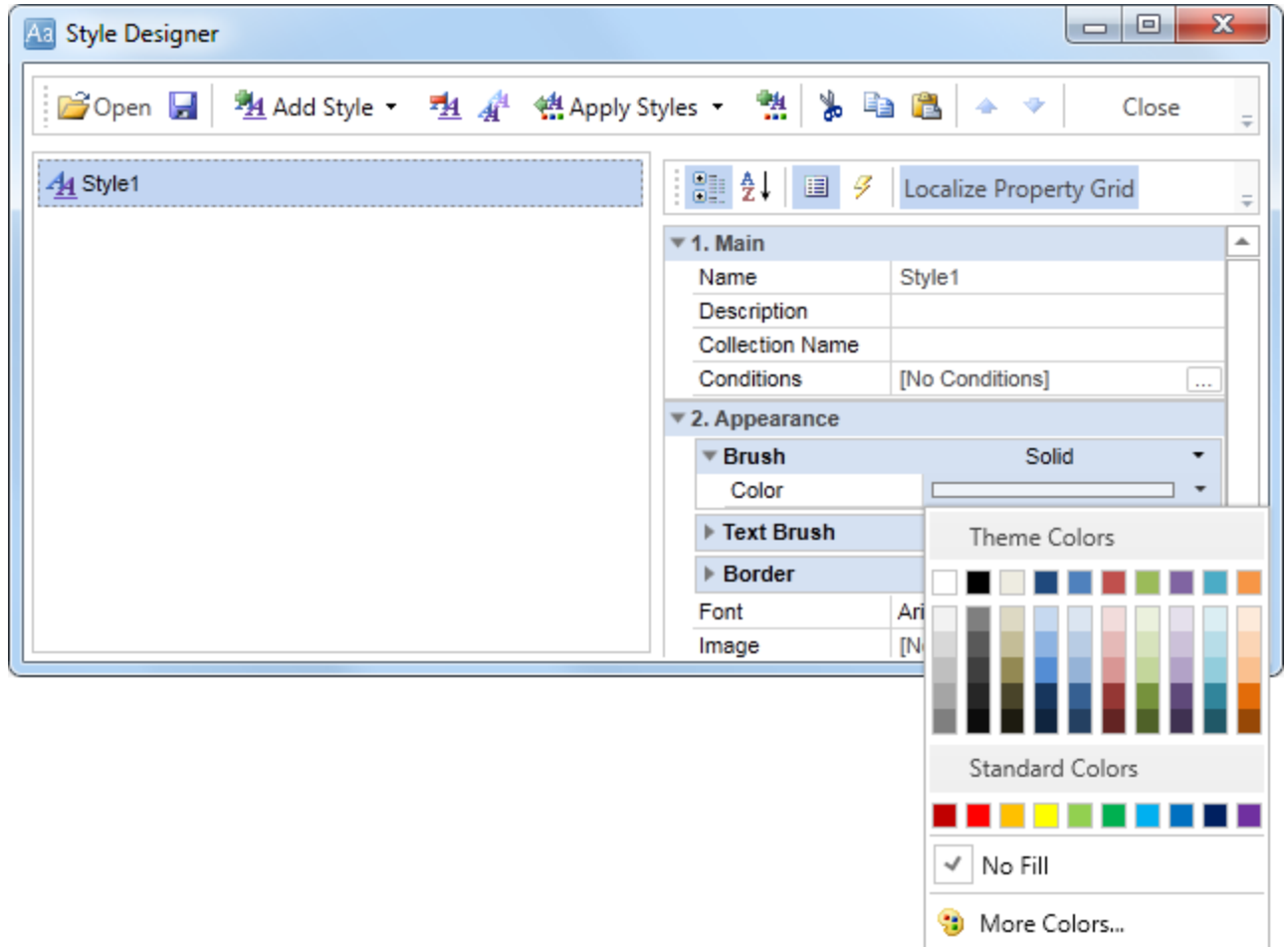
Press the **Close** button when the property is set. After that, in the list of values of properties **Even style** and **Odd style** the new values will appear, i.e. the new style of even/odd lines, respectively.

4. Render a report by clicking on the **Preview** tab or call the report **Viewer** using the **Preview** menu item. The picture below shows the rendered report with the invoice:

Unit Name	Description	Qty	Item Price	Total
Chai	10 boxes x 20 bags	39	18	702
Chang	24 - 12 oz bottles	17	19	323
Aniseed Syrup	12 - 550 ml bottles	13	10	130
Chef Anton's Cajun Seasoning	48 - 6 oz jars	53	22	1166
Chef Anton's Gumbo Mix	36 boxes	0	21.35	0.00
Grandma's Boysenberry Spread	12 - 8 oz jars	120	25	3000
Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15	30	450
Northwoods Cranberry Sauce	12 - 12 oz jars	6	40	240
Mishi Kobe Niku	18 - 500 g pkgs.	29	97	2813
Ikura	12 - 200 ml jars	31	31	961
Queso Cabrales	1 kg pkg.	22	21	462
Queso Manchego La Pastora	10 - 500 g pkgs.	86	38	3268

5. Go back to the report template;

6. Save the report template, for example, as **Invoice.mrt**.

The invoice, can be printed, saved to any of the available file formats, or sent via Email. The picture below shows a list of file formats available for saving or sending reports via Email:

	Document File...	Ctrl+S		
	Adobe PDF File...			
	Microsoft XPS File...			
	Microsoft PowerPoint 2007/2010 File...			
	HTML File...			
	HTML5 File...			
	MHT Web Archive...			
	Text File...			
	Rich Text File...			
	Microsoft Word 2007/2010 File...			
	OpenDocument Writer File...			

	Microsoft Excel File...			CSV File...	
	Microsoft Excel Xml File...			dBase DBF File...	
	Microsoft Excel 2007/2010 File...			XML File...	
	OpenDocument Calc File...			Data Interchange Format (DIF) File...	
	Data	▶		Symbolic Link (SYLK) File...	
	Image	▶			

	BMP Image...	
	GIF Image...	
	JPEG Image...	
	PCX Image...	
	PNG Image...	
	TIFF Image...	
	Windows Metafile...	
	Scalable Vector Graphics (SVG) File...	
	Compressed SVG (SVGZ) File...	

2.31 Invoice Report With Parameters

Do the following steps in order to create an invoice with parameters:

1. Run the report designer;
2. Open the saved report template and render a report. The picture below shows the rendered report with the invoice:

Pay attention to the report header. As can be seen from the picture above, information about payments and delivery are not specified. How to make it so you can easily specify these details? The constant editing of text components in the report template is not an option, but using the parameters in the report is quick and easy. Especially if there are more recipients of your invoices. So, in order to add parameters to the report, follow these steps:

3. Go back to the report template;

4. Add parameters to the report template. The parameters in the report are implemented using variables (a variable may have different values). To add a variable, in the tab Dictionary -> the menu item New Item -> select New Variable.... The picture below shows the New Item:
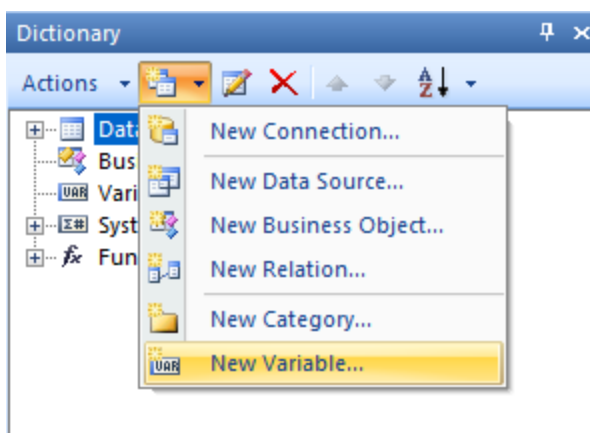
Details BILL TO and SHIP TO, by definition of fields (name, street, city, zipcode) are the

same, so when you create variables, there could be confusion. To avoid this, the variables can be created in different subcategories. So, to avoid this, create a sub-category of variables, which are called BILL TO and SHIP TO. For this purpose, in the context menu of the category Variables, click New Category...:
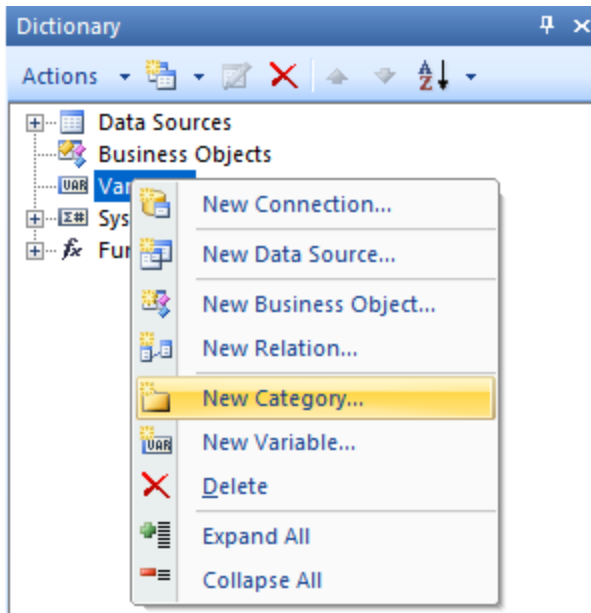
Then, in the box of the New Category you should specify a name for the category (BILL TO and SHIP TO). After that, we will create the variables in the category BILL TO. In principle, there is no difference where to create a variable, because it is always possible to move it to the appropriate subcategory. Yet, to save time, get used immediately to create the correct location. So, select a subcategory created by BILL TO command and call the new variable (New Variable) from the context menu or menu item New (New Item). The picture below presents a window to create a new variable:

Define the parameters created by the variable:

5.1. Change the name (Name) and Nick (Alias) variable, specify the description (Description), if necessary;

5.2. Choose the type of stored value (in this case string) and the type of the variable (we will approach the variable type value (Value)). Here is a very important step, which we have determined that our variable will store a single value (rather than a list of values or Range), and this value will be stored in a string type.

5.3. Set the default value. In our example, set the value of Name;

5.4. Get the answer options are installing from a user (Request from User), and use user values (Allow User Values). In this step, we allow the user to participate, as well as change the value stored in variable;

5.5. Press Ok.

In order to use this variable in the report, you must provide a link to it - {variable name}. In this case, we indicate in the text component {BILLTO_Name}. The picture below predstalen invoice template with a variable:

Render a report to check how works the newly created key in the final report. Click on the Preview button or bring up the Viewer, using the shortcut key F5 or the menu Preview. After building a report, all references to data sources will be replaced with data from these fields. With that data will be taken sequentially from a data source that was specified for a given band. The number of copies of the band Data in the rendered report will be equal to the number of rows in the data source. The picture below before your report with a parameter:

As can be seen from the picture, the report shows the specified field values of the parameter (in this case, Name). Note that in the first set of values stored in the variable value by default. Now change the value and click the Apply button (Submit). In the picture below a report with the modified parameter value:
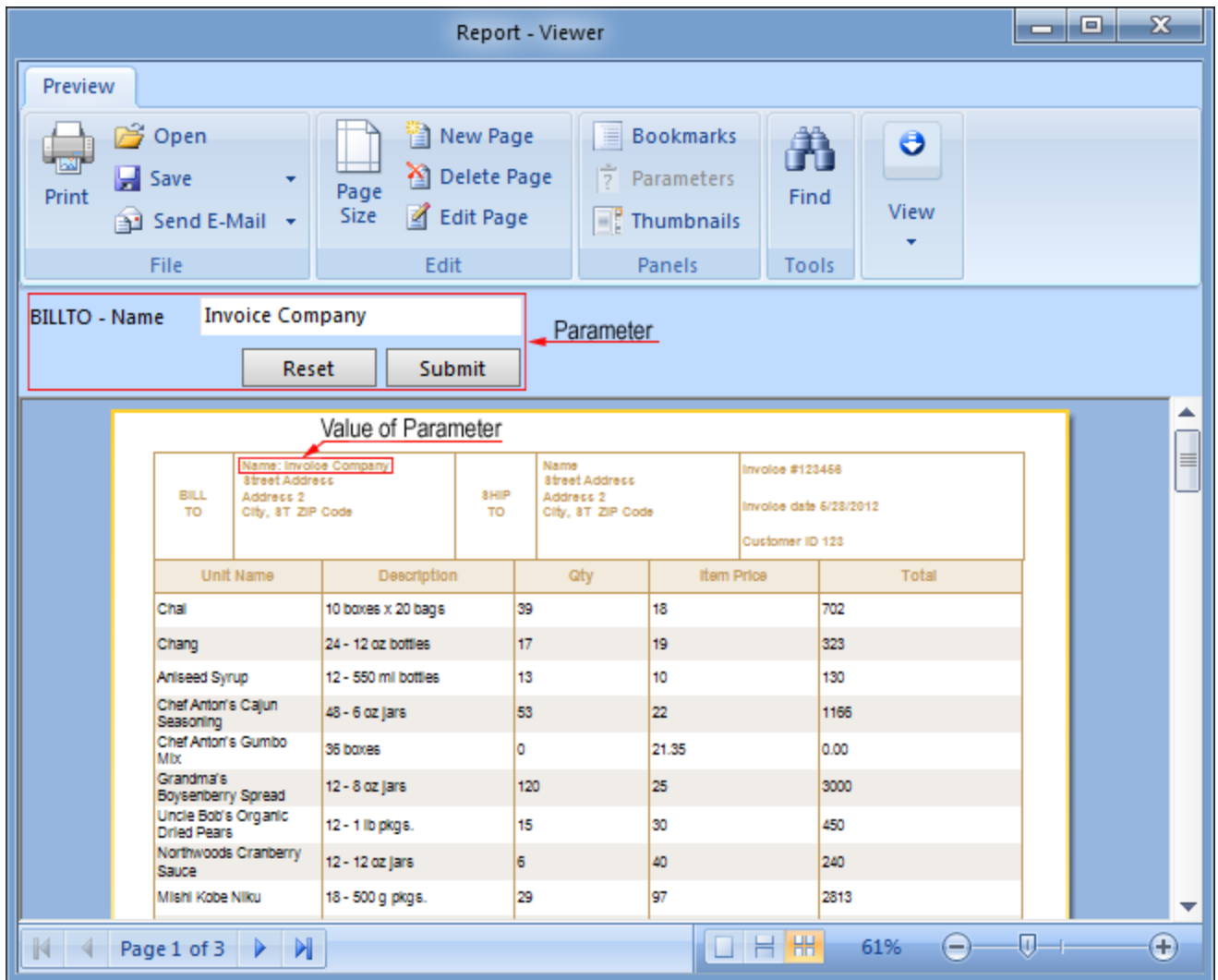
Add options for other fields. To do this:

Back to the template;
Create a similar variables in the sub-BILLTO named BILLTO_Street_Address, BILLTO_Address_2, BILLTO_City-ST-ZIP_Code;
In a similar sub-SHIPTO variables, with the names of SHIPTO_Name, SHIPTO_Street_Address, SHIPTO_Address_2, SHIPTO_City-ST-ZIP_Code;
Use these variables to the report, ie They point to the links in the template;
We construct a report to check how the newly created key in the final report. Click on the Preview button or bring up the Viewer, using the shortcut key F5 or the menu Preview. After building a report, all references to data sources will be replaced with data from these fields. With that data will be taken sequentially from a data source that was specified for a given band. The number of copies of band Data in the constructed report will be equal to the number of rows in the data source. The picture below before the

report prepared with the following parameters:

Now, in order to prepare an invoice with the required details and BILLTO SHIPTO, no need to alter permanently a template. Enough to simply specify the details and click the Apply button (Submit). Reset Button (Reset) resets the values stored in a variable and sets the value stored by default. In these two articles, I showed you how to use report generator Stimulsoft can facilitate their work in creating invoices. And also learned how to use this tool in a few steps and get a hard-structured, well-designed, dynamic report. I would like to add that this is only a small part of the potential reporting tool Stimulsoft. Stimulsoft Start learning today and you'll wonder how you can quickly and easily create reports. And I'll be sure to write articles to help you solve your questions.

3   Report Internals

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

3.1   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.

3.1.1   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.

3.1.2   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.

3.1.3   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.

3.1.4   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.

3.1.5   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")}

3.1.6 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}

3.1.7 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.

3.1.8   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**.

3.1.9   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]}

3.2 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.

3.2.1 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.
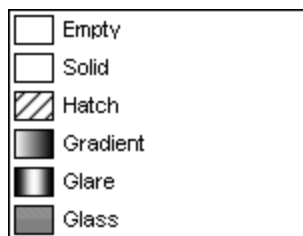
Six types of Brushes are available within Stimulsoft Reports:
   ✓ **Empty**;

✓ **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.

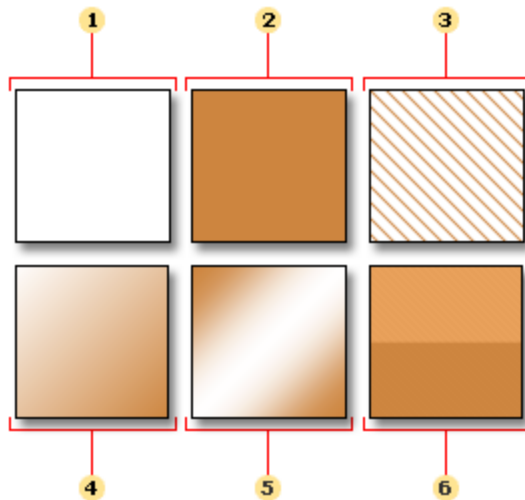**3** **Hatch.** The background of a component is filled with a texture. The background and foreground colors of the selected texture can be specified individually.

4 **Gradient.** The background of a component is filled with gradient. A Start color, an End color, and a Gradient angle can be 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.

3.2.2 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

Information

The font list that is available in the report designer contains the fonts installed in the current operating system. For web tools – from the fonts installed in the operating system of the server.

Any font that is installed on your machine and available from the .NET Framework can be used in a report. Most frequently these are OpenType and TrueType fonts. 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:

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:

Custom Fonts

When designing reports, it is possible to use custom fonts which are not installed by default in the operating system. Add the font files (*.ttf, *.otf) to the report resources. In

this case, the font will be added to the report and can be used the following way:

❯ Assign a font directly to a text component. To do this, follow these steps:

Step 1: Select the text component or several components;

Step 2: Select the added font from the drop-down menu in the **Home** ribbon tab of the report designer panel. Custom fonts are displayed at the top of the font list.
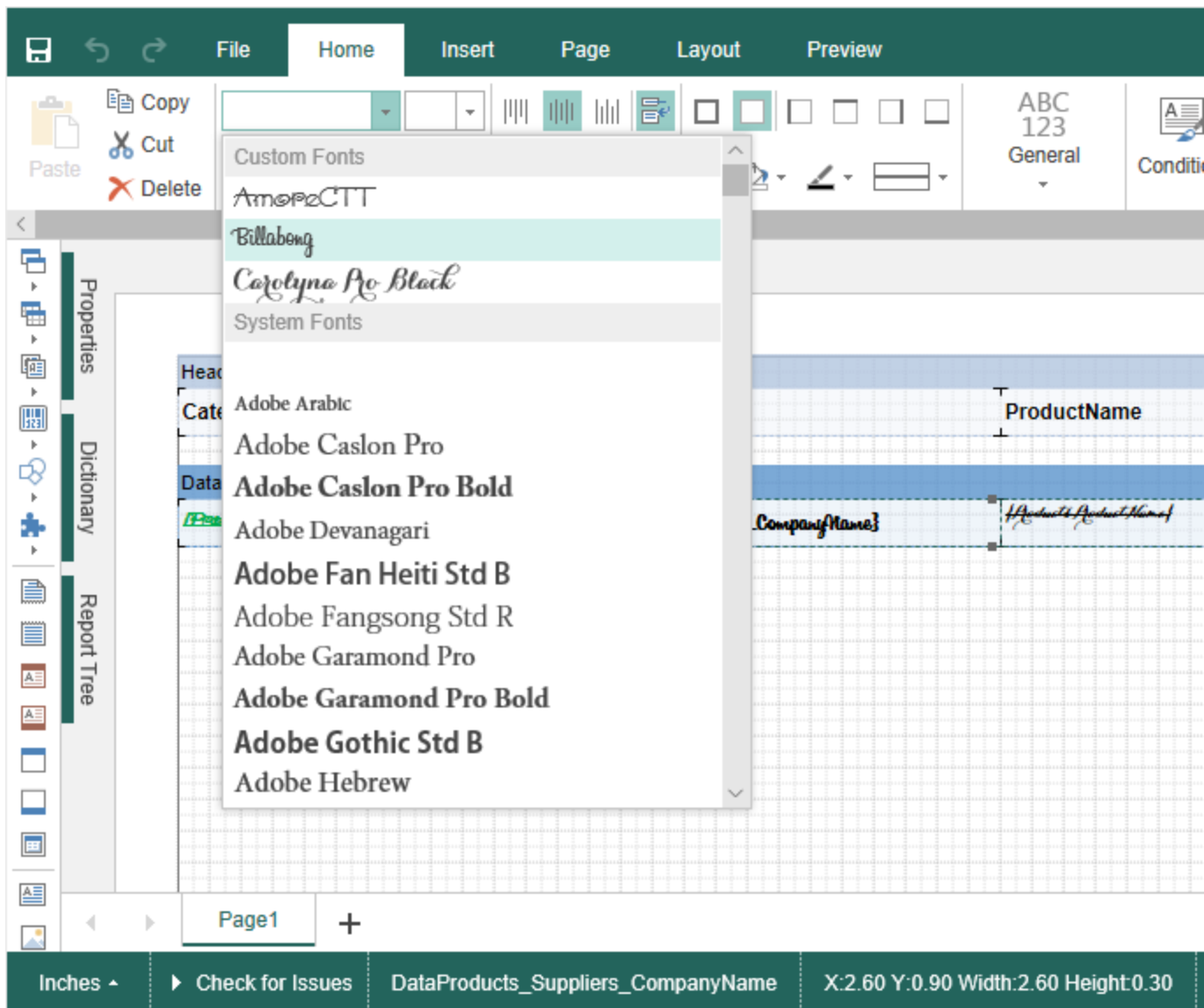
The selected font will be applied to all selected components.

❯ Use conditional formatting of the component. To do this, follow these steps:

Step 1: Select the text component;

Step 2: Click the **Conditions** button in the Home tab of the report designer panel;

Step 3: Add **Highlight Condition**;

Step 4: Specify the logical conditions for applying the design settings;

Step 5: Click the **Change Font** button and select the custom font. Custom fonts are displayed at the top of the font list.

Step 6: Click the **Ok** button in the Font dialog, and then click the button in the editor of a condition.

Now, when the logical condition is executed, a custom font will be applied to the component.

❯ In report styles. To do this, follow these steps:

Step 1: Click the **Style Designer** button in the **Home** ribbon tab of the report designer panel;

Step 2: Select the style or create a new one in the editor of a condition;

Step 3: In the properties panel, select the custom font in the **Font** property. Custom fonts are displayed at the top of the font list.

Step 4: Click the **Ok** button in the editor of a condition.

The font will be applied for all components for which this style is assigned unless other is specified by the style settings or conditional highlighting.

3.2.3   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.

3.2.3.1   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:
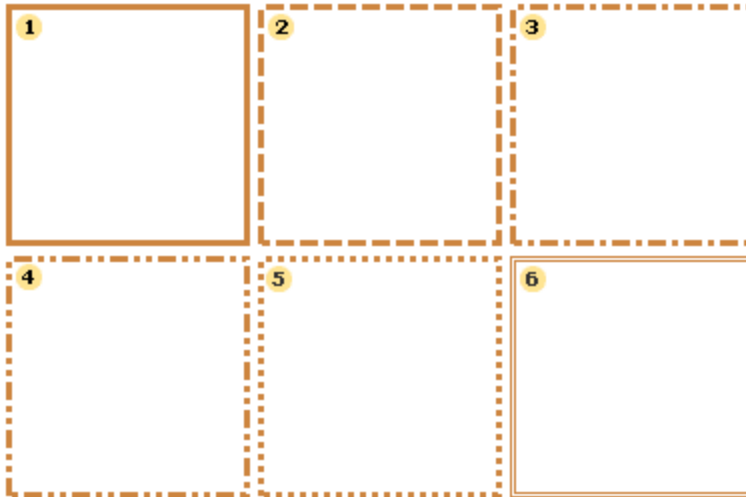
[1] **Solid**;
[2] **Dash**;
[3] **Dash Dot**;
[4] **Dash Dot Dot**;
[5] **Dot**;
[6] **Double**.

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.
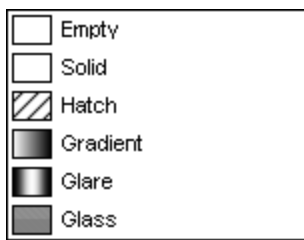
These styles can be combined with the other shadow properties to apply a wide range of different appearances to report components. A few examples:
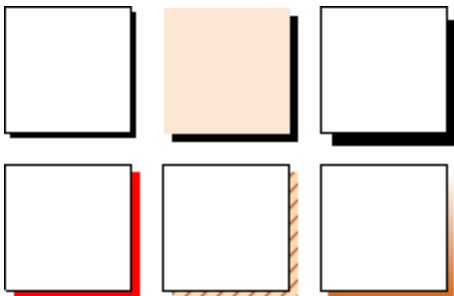
Setting Simple Border Properties
You can set simple Border properties directly from the Object Inspector, or using the Borders Toolbar.

Object Inspector

To set properties from the Object Inspector click the ellipsis button beside the Border property
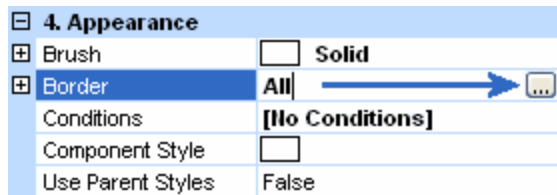
A new dialog will be displayed that allows you to set the options for the border of the component:
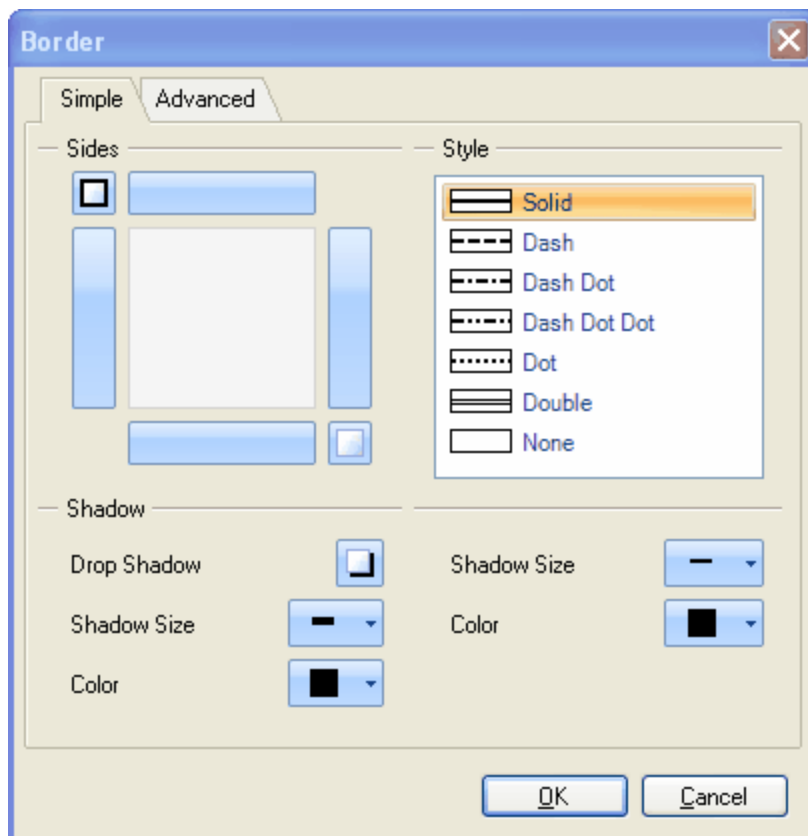
Simply select the settings you would like to apply and click the OK button to close the dialog and update the border.

3.2.3.2 **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:
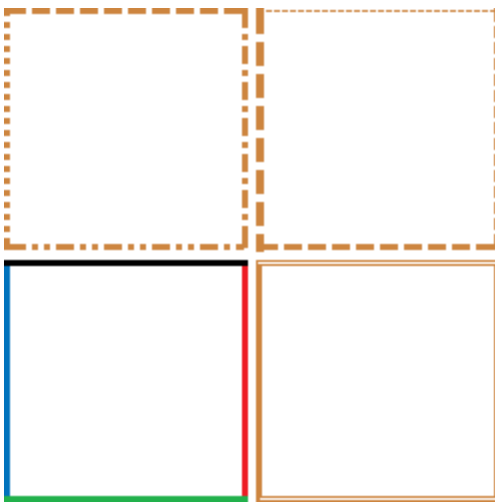
Setting Advanced Border Properties
You can set Advanced Border properties only from the Object Inspector.

❗ **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
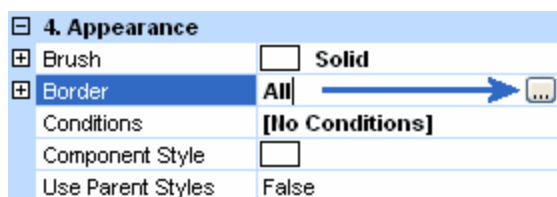
The simple **Border** dialog will be displayed. To access the advanced border features simply click the Advanced tab at the top to bring it to the front.
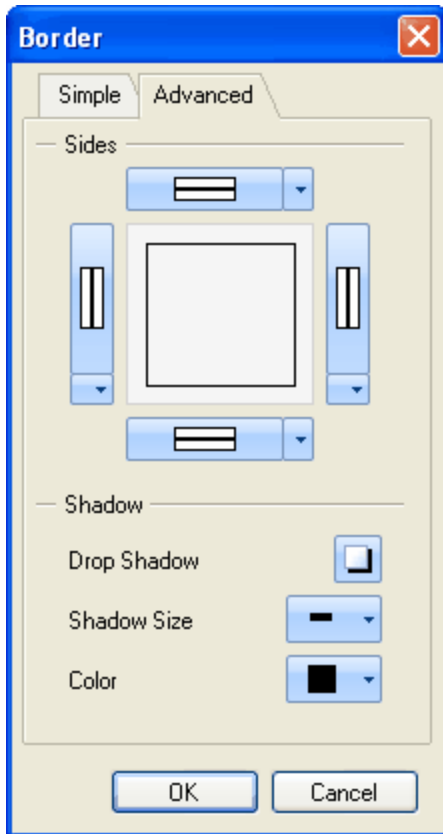
Simply select the settings you would like to apply and click the OK button to close the dialog and update the border.

3.2.3.3   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.

3.2.4   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.

3.2.4.1 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.
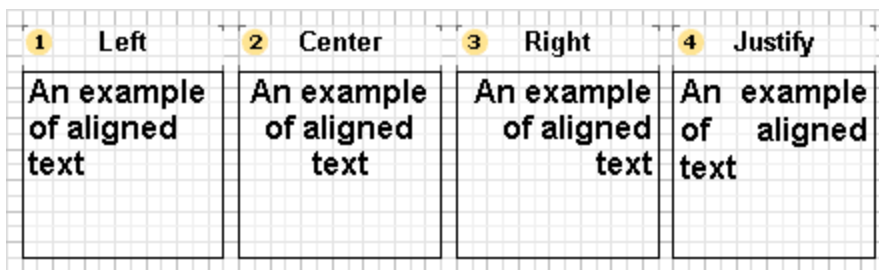
1 **Left.** The text is aligned on the right edge with a ragged right edge.
2 **Center.** The text is aligned centrally within the component with ragged left and right edges.
3 **Right.** The text is aligned on the right edge with a ragged left edge.
4 **Justify.** The text is aligned evenly across the 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 alignment property it is possible to set text alignment using **HTML** tags.

3.2.4.2 Horizontal Image Alignment

Stimulsoft Reports allows a choice of three alignments of an image within an image component: **Left**,**Center**, and **Right**.
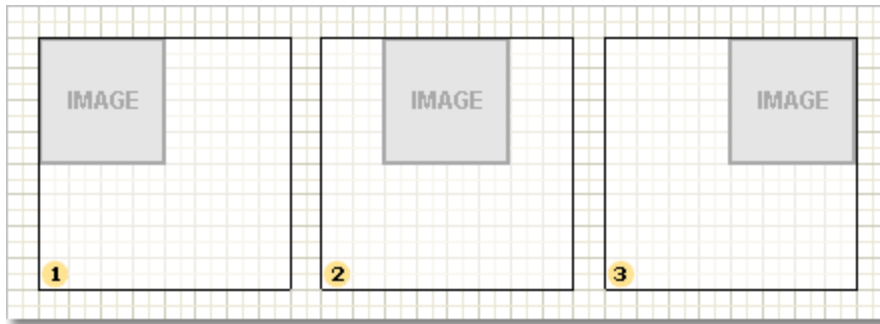
1 **Left.** The image is aligned on the right edge.

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**.

3.2.5   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.

3.2.5.1   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.
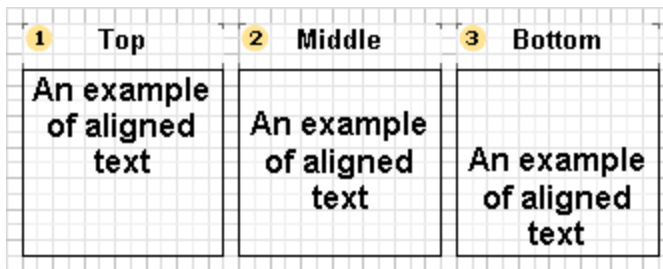
① **Top.** Text is aligned with the top edge of the component.

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.

3.2.5.2   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.
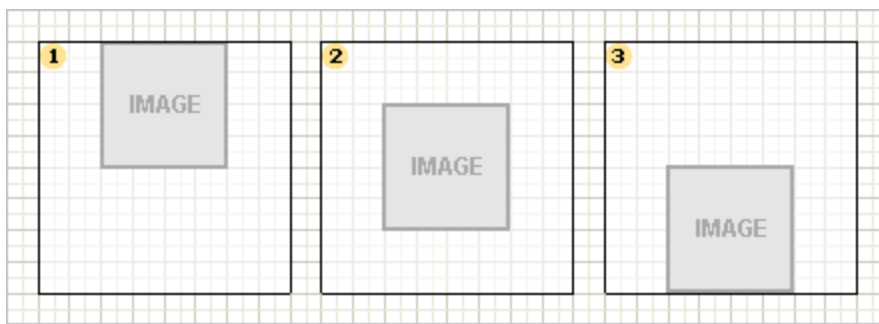
① **Top.** The image is aligned with the top edge of the component.

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.

3.2.6   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 style can 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 a description which can be used to explain 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 four types of styles:

✔ **Component**;
✔ **Cross-Tab**;
✔ **Chart**;
✔ **Report Control.**

A̲A̲	Component	F2
▫A̲	Chart	F3
▣A̲	Cross-Tab	F4
▣A̲	Report Control	F5

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.
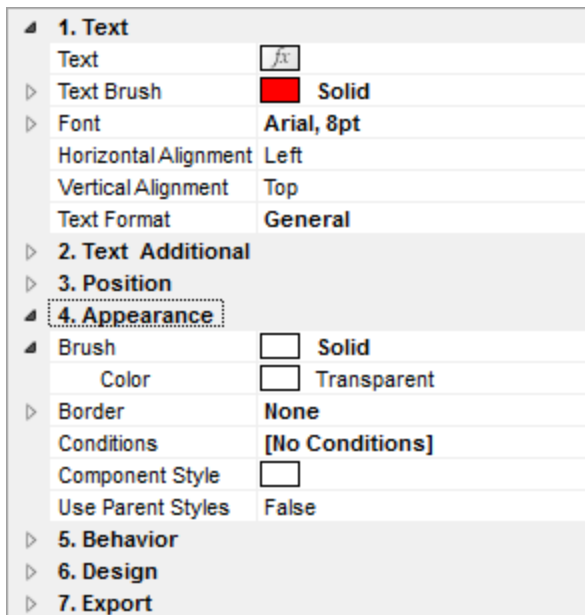
Applying Styles

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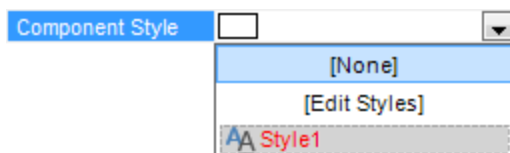
After a style has been assigned to a component the report generator will ensure that the appearance of the component consistently matches that of the specified style. Changes to the style will automatically cascade to all components to which the style has been assigned.

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.

3.2.7   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:

Company
Alfreds Futterkiste
Ana Trujillo Emparedados y helados
Antonio Moreno Taquería
Around the Horn
Berglunds snabbköp
Blauer See Delikatessen
Blondesddsl père et fils

3.2.8   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.

3.2.9 Style Designer

The **Style Designer** is a tool in the report designer that can be used to create and edit styles in reports. Using the styles and their collections you can change the appearance of the report. The picture below shows the Style Designer dialog:
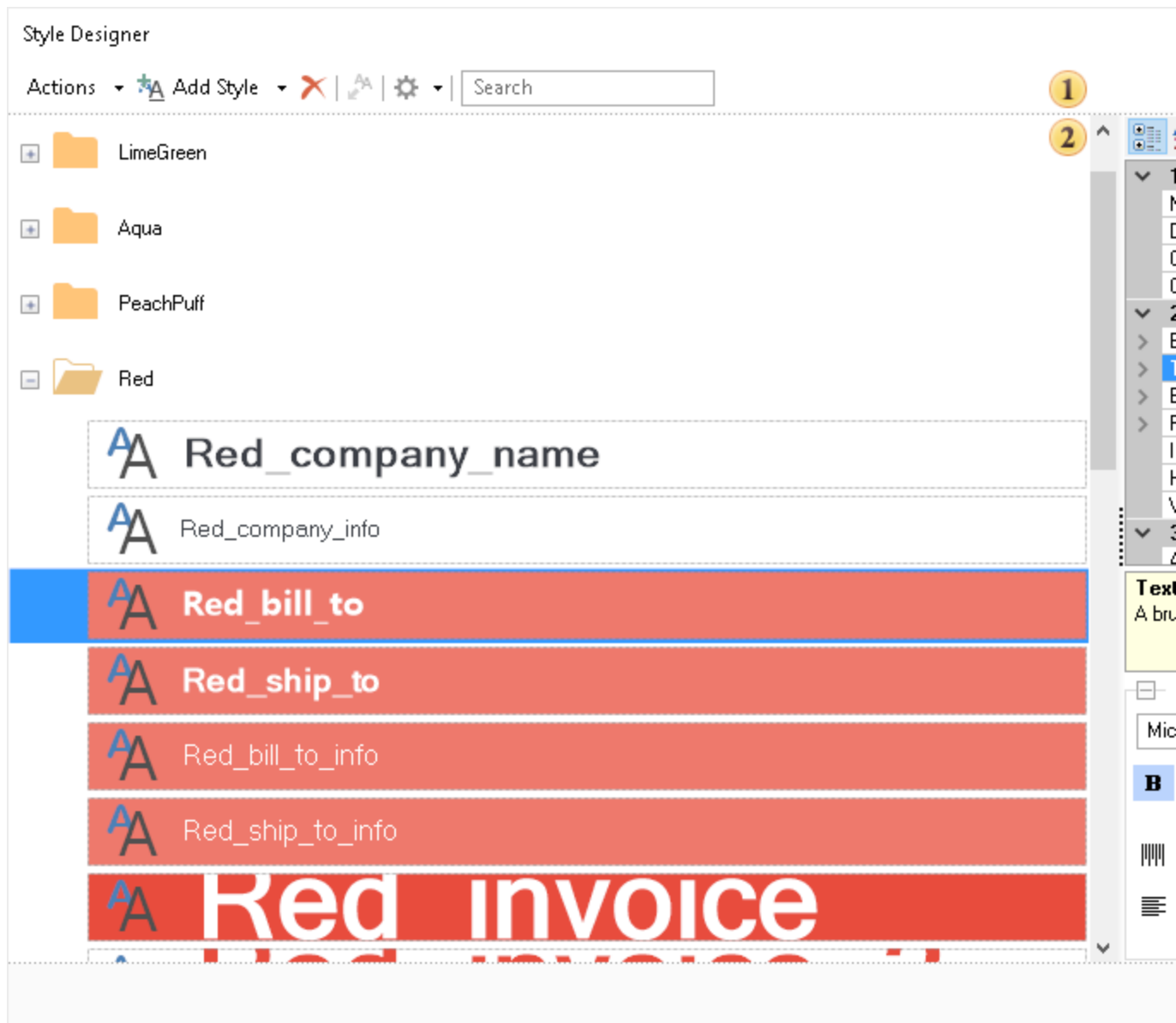
1 Toolbar. On this panel you can find commands used in the style designer.

2 List of collections and styles.

3 Properties panel of the selected style.

4 Description of the selected property.

5 The **Formatting** panel contains the basic controls for formatting components. This is actual for styles such as Component.

Toolbar

The toolbar contains the basic controls used in the style designer. The picture below shows the control panel:

1 The Actions menu contains basic commands to control styles and style collections:

    i The **Open...** command allows you to open a previously saved file (*.sts) with a list of styles and style collections.

    i The **Save As...** command saves styles and style collections to the *.sts file.

    i The **Create Style Collection...** command is used to create an automatic collection of styles.

2 The **Add Style** menu contains a list of styles for various report items that you can add to the list of report styles and collections:

    i The **Component** style can be applied to almost all report items that have the ability to select a style, except for maps, charts, gauges, tables, and cross-tabs.

    i The **Chart** style is used for any type of charts.

    i The **Map** style is used only for maps in reports.

    i The **Gauge** style is used for both linear and radial gauges.

    i The **Cross-Tab** style is used for the cross-tab components.

    i The **Table** style is applicable only to the table component.

3 The **Delete** button allows you to delete a selected style or collection from the list of report styles.

4 This command is used to get a style from a component. To do this, select the component in the report, call the style designer and click this button. The style will be created with the settings as the selected component in the report.

5 The **Settings** menu contains commands for managing the list of report styles:

    i Select the types of styles that you want to display in the list of report styles. By

default, all types are displayed. If the type is disabled, and you add a new style of the same type, then all styles of this type will be displayed in the list.

i Set the sorting of styles in the list. First, in the selected order (ascending or descending), collections of styles will be sorted, and then the styles in these collections will be sorted in the same order. Please note that to manually change the order of the list of styles (for example, when dragging a style in the list), you should disable the sorting mode.

6 The **Search** field. Simply enter a name or other characters from the name of the desired style, and the list of report styles will be filtered.

Context menu

The context menu of the style designer can be called by right-clicking in the list of styles:
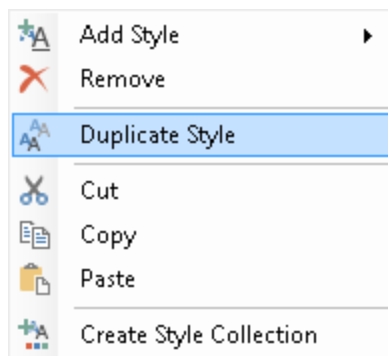
As you can see from the picture, the context menu contains:

i The commands are used to create styles of various types;

i The command is used to delete the selected item (collection or style);

i The **Duplicate style** command is used to create a copy of the style from the list. In this case, the duplicate style will be in the same collection, and the suffix Copy1,2,3 will be added to the name of the copied style.

i Commands are used to work with the clipboard (Cut, Copy, Paste);

i The command is used to create a collection of styles automatically.

You should know that, for the collection of styles, some commands from the context menu will be unavailable.

Hotkeys

You can use keyboard shortcuts to invoke a specific command when working with the

style designer. The table below shows the list the commands and key combinations.

Hotkeys	Command
Ctrl+x	Cut a style to the clipboard
Ctrl+c	Copy a style to the clipboard
Ctrl+v	Paste a style to the clipboard
F2	Rename the style or collection of styles
Delete	Delete the style or collection of styles
Shift+insert	Insert the style or collection of styles
Shift+Up	Move the style one position higher in the current collection. In this case, the sorting mode should be disabled.
Shift+Down	Move the style one position lower in the current collection. In this case, the sorting mode should be disabled.

Color Collection Editor

When creating a **Chart** and **Map** style, you can create a collection of colors for these styles. Applying this style in the report, you can use your own color for every **Chart** and **Map** component. You can create a collection of colors in a special editor. In order to call this editor, you should select the **Chart** or **Map** style in the style designer and click the button in the properties panel of the **Colors** property:
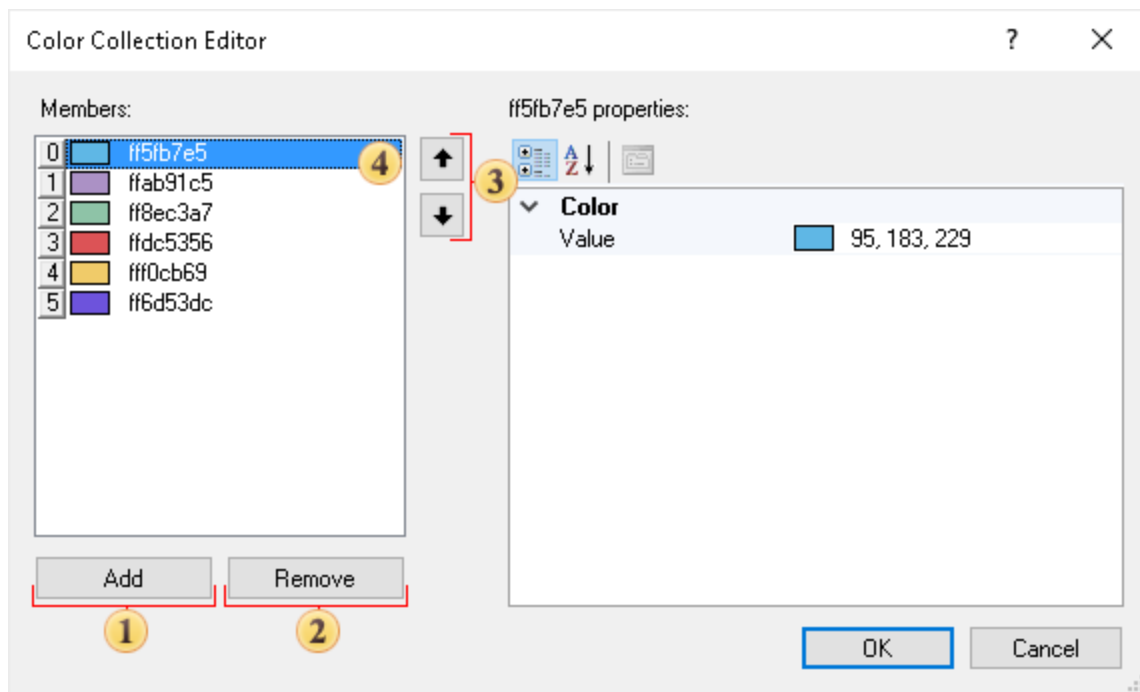
① The **Add** button is used to add the new color to the collection of colors. By default, the color is **White**. Then, you need to specify the color you want.

② The **Remove** button is used to remove the selected color from the collection of colors.

③ Buttons for moving the selected color up and down in the list. The sequence of colors from top to bottom determines the color of report items. The first color in the list is assigned to the first report item, the second color – to the second one, and so on. Therefore, by changing the order of colors in the list, you can adjust the color scheme of the Chart or Map components.

④ The list of color collections. The number of colors in the collection is unlimited.

3.2.9.1 Creating Collection of Styles

A collection of styles is a group of styles that are united by some criteria. The collection should contain at least one style. For example, if the styles are simultaneously used in the current report, they can be united into a collection. It should be noted that the report can have several collections of styles. But the current collection that is applied to the report can be only one collection.

You can create a collection of styles with the following options:

❯ Manually, by filling the **Collection Name** style property;

❯ Select the **Create Style Collection** command from the Actions menu, define the settings for the future collection, and click the OK button. In this case, the collection will be generated automatically.

It should be noted that, at the same time, there can be collections of styles created manually and generated in the automatic mode. Also, if necessary, the generated collection of styles, and all the styles in it, can be edited.

Creating a collection of styles manually
Step1. Run the Style Designer

Step 2. Create styles from the [Add Style](#) menu or open a previously saved file with styles;

Step 3. Fill the **Collection Name** property. The value of this property is the name of the future collection. For example, if you specify the name Black for five styles then, in the list of elements of the style designer, the collection with the name Black will be created. It will contain those five styles. When working with collections of styles, please note:
❯ If you drag a style from one collection to another, the value of the Collection name property will change.
❯ When you rename a collection of styles the value of the Collection name property will be changed for all styles in this collection.
❯ To ensure that the style does not belong to any collection of styles, you should delete the value of the Collection name property.

Step 4. Configure the style [conditions](#). Without conditions, when you apply a collection of styles to a report, the styles will be applied sequentially from top to bottom. As a result, the last style from the collection will be applied to the report. However, you can also change the style of the report component by choosing it in the report template and selecting the appropriate style from the style menu on the **Home** tab of the report designer.

Step 5. Set the style settings (brush, color, borders, alignment, etc.).

Step 6. Click the OK button in the style designer.

Creating a generated collection of styles
Step1. Run the Style Designer.

Step 2. Select the Create Style Collection command from the [Actions](#) menu or the

context menu of the style designer;

**Step3**. Set the parameters of the future collection of styles.

① The **Collection Name** field is used to specify the name of the collection.

② The **Color** field is used to select the main color for the style collection.

③ In the [Nested Level](#) field is used to select the value corresponding to the level of nesting in the report.

④ The **Nested Factor** field is used to specify the nesting factor that affects the clarity of colors in this collection of styles.

⑤ Enables or disables **Borders** for styles in the collection.

⑥ The **Remove Existing Styles** option. If this option is enabled (check box is selected) then, when creating the collection, existing styles and collections in the style designer will be deleted. If this option is disabled (the check box is cleared), a new collection of styles will be added to existing styles and collections.

⑦ The **Component** panel. In this panel, the bands of the report are checked. If a certain band is checked, then for the report components that are located on it, the styles will be applied. If no bands are checked, the collection will not be created.

⑧ The **Preview** panel. This panel previews a report with the collection of styles applied.

**Step 4**. Click the OK button in the Create Styles Collection dialog.

Step 5. Edit the styles from the collection, if necessary, and click the OK button in the style designer.

Applying a collection of styles in reports
Once the collections are created, you can select one of these collections in the report. To do this, click **Select Style** on the **Home** tab of the report designer. The drop-down list displays all report collections and lists of styles.
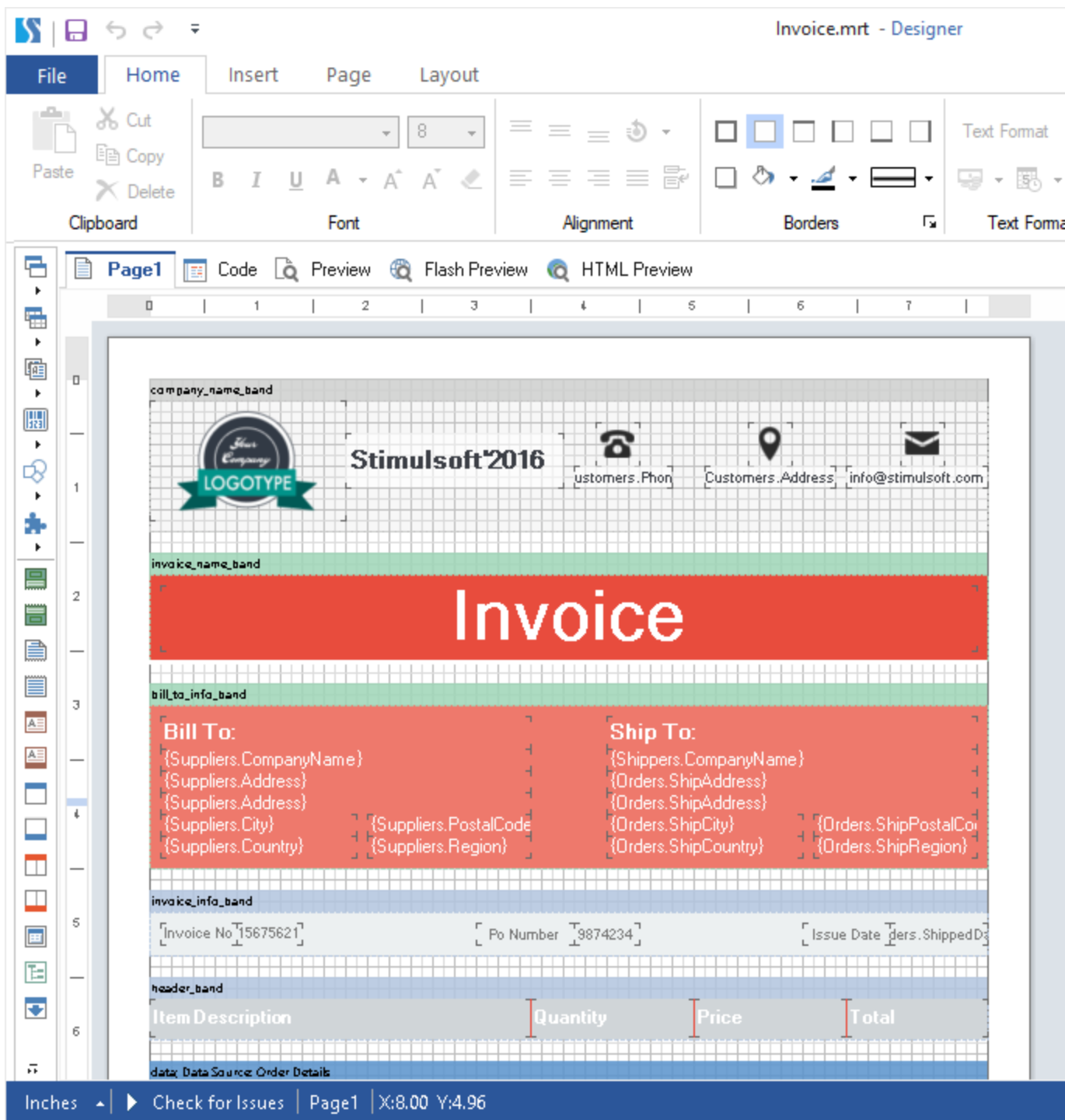
After selecting a collection, a message will be displayed. You need to confirm or cancel this action.
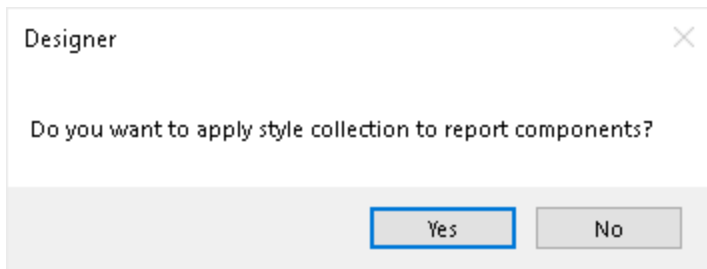
When you click **Yes**, the styles from the collection will be applied to the report components depending on the specified conditions in these styles.

3.2.9.2   Conditions

Applying of styles to the components of the report can be done:

❯ Manually. Select the report component, select a style from the style quick menu on the **Home** tab, or select the desired style in the **Component Style** property.

❯ Automatically by a **condition**. In other words, the style will be applied to the component if the **condition** (or **conditions**) is met.

Adding and changing the conditions for applying styles can be done in the **Conditions** dialog. To call this window, you need to click the **Browse** button in the properties panel in the style designer in the **Conditions** property.

As can be seen from the picture, the dialog contains one block, which shows the different types of conditions. Each type of conditions includes the following fields: name, which displays the name of the conditions, as well as the element of enabling this condition; operation type used to select an operation selected by means of which calculation of the value of the condition in the latter field specifies the values for which the operation will be performed.

1 The condition **Placement**. This type of conditions provides the opportunity to apply a style to a component, depending on its placement. In the values field containers (bands, panel, table, page) are selected. If to select operation is **equal to**, then the style will be applied to the components placed on the containers, which are selected in the value field. If you select the operation **not equal to**, then the style will be applied to the components placed in any container other than the selected value in the field. It should also be noted that in the value field, you can choose several containers.

2 The condition **Nested Level**. C using this kind of condition the components can be styled according to the nesting level of containers on which the components are placed.

In the value field you may specify the nesting level of the container (maximum 100). In this type of conditions the following operations are available:

> **equal to**. The style will be applied when the nesting level of containers will be equal to the specified level in the field of values;

> **not equal to**. The style will be applied to all components in containers, which the nesting levels will not be equal to the specified level in the field of values;

> **greater than**. The style is applied to the components in containers, which nesting level is greater than the level of the specified field of values;

> **greater than or equal to**. The style is applied to the components in containers, which the nesting level is equal to or greater than the specified level in the field of values;

> **less than**. The style will be applied to components in containers, which nesting level is less than the specified level in the field of values;

> **less than or equal to**. The style is applied to the components in containers, which nesting level will be equal to or less than the specified level in the value field.

It should be noted that this type of condition is only included if the condition of the type **Placement**.

3 The condition **Component Type**. If you want to apply style to components of a particular type, it can be done using this condition. Also, it should be noted that in the value field of this condition, you can select multiple types of components. In this condition the following operations are available: **equal to**, if you want the style applied to the components specified in the value field, and the operation **not equal to**, the choice of which style is applied to all components, except for the selected field value.

4 The condition **Location**. This type of condition provides an opportunity to apply a style to a component, depending on its location in the container. In the value field you may select the desired location of the component. The operations are available for the condition: **equal to**. The style is applied to all components, the location of which corresponds to the selected field values, and the operation **not equal to**. The style is applied to all components other than the location of which is selected in the value field. It should also be noted that in the value field you can simultaneously select multiple locations.

5 The condition **Component Name**. If you want to apply a style to a component with a certain name, it can be done using this condition. In the value field should specify the name of the component with respect to which the condition will be executed. The following operations are available when using this operation:

> **equal to**. The style is applied to a component with a name identical to the specified one in the value field;

> **not equal to**. The style is applied to all components, which name matches the

specified field of values;

❯ **containing**. The style will be applied to all components which contain the name of the specified field values in their name;

❯ **not containing**. The style will be applied to all components, which in its name do not contain the name of the specified field of values;

❯ **beginning with**. The style will be applied to all components for which the name starts with the name specified in the value field;

❯ **ending with**. The style will be applied to all components for which the name ends with the name specified in the value field;

Also, it is permissible to combine different types of conditions. In this case, the style will be applied if all the conditions will work. In other words, if you are using the conditions of the type **Placement** and **Component** name, then the style will be applied only if these conditions are fulfilled, i.e. properties of the component will meet two requirements - placement and naming. In addition, you can use the multi-conditions, i.e. when two or more blocks of conditions are involved. In this case, the component will be used only in the style, if all blocks in all conditions are met.

Nesting Level
The level of nesting is the level of subordination of a component to another component, i.e. to the component of the same type. The first level of nesting is organized when the component is added to the report template, i.e., if you add a component and it will not have a subordination, it will be a component of the first level of nesting.

If the report has, for example, two DataBands, one of which subordinates to the second one, then the subordinated band of the second level of nesting, and the subordinating one - of the first level of nesting. If the report contains three DataBands, where the third subordinates to the second band, and the second one the first one, then they will be components of the third, second and first level of nesting. It is also worth noting that there may be several components of one nesting level, i.e., one Data Band may subordinate a few bands.

It should be understood that it is impossible to create a nesting level between the DataBand and the ReportTitleBand, because they belong to different types of bands. The picture below schematically shows the levels of nesting of DataBands:
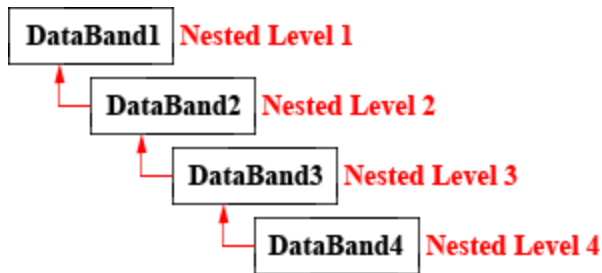
Nesting level of the DataBands

As previously mentioned, when the component is added to the report layout, it is given the first level of nesting. Changing the level of nesting can be done using the **Master Component** property. For this, in the field of the property you should select the DataBand to which it will be subordinated.

The nesting level of the subordinated band will be the next level of the subordinating one. If you select the DataBand of the third nesting level, the band will be assigned to the fourth nesting level. Also, remember that one band may subordinate to a few bands. In this case, the subordinated bands will have the same nesting level. The picture below shows an example of report organization with the three level nesting:
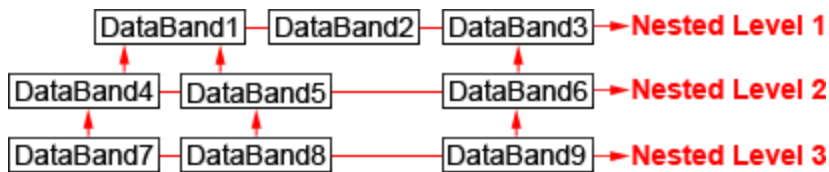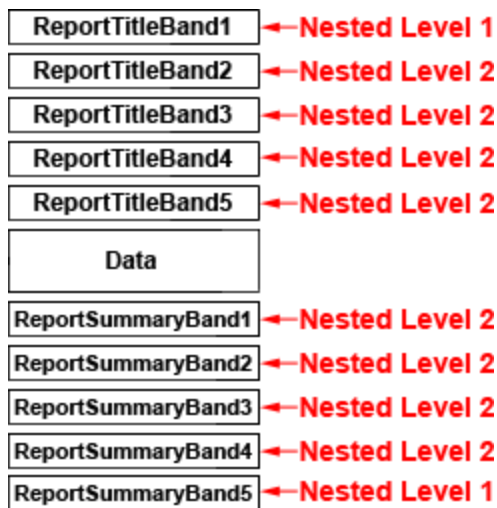

It should be noted that creating a collection of styles, in the dialog Create Style Collection, the tenth nesting level is the maximum to be specified. It can be increased to the 100th using the **Condition**. The **HeaderBand**, **FooterBand**, **GroupHeaderBand** and **GroupFooterBand** relate directly to the **DataBand** and, therefore, their level of nesting depends on the nesting level of the **DataBand** to which they relate . It is important to understand that the nesting level of the **DataBand** and the bands related to it, does not depend on their location in the report.

The nesting level of other bands

For the **ReportTitleBands** and **ReportSummaryBand**, you can create a collection of styles of only the first and second nesting level, i.e. for these bands is impossible to create a collection of styles of the third and subsequent nesting levels. In contrast to the **DataBand**, the subordination is done on the location of bands on the report page.

For the **ReportTitleBand** the nesting level is determined as follows: the first (top) band is assigned to the first nesting level, and all subsequent (located below) - the second nesting level. For the **ReportSummaryBand** the nesting level is determined slightly different: all the bands except the last (bottom) is assigned to the second nesting level, and the last (bottom) - to the first level.

 The picture below schematically shows the nesting levels for the **ReportTitleBands** and **ReportSummaryBands**:

ReportTitleBand1	←Nested Level 1
ReportTitleBand2	←Nested Level 2
ReportTitleBand3	←Nested Level 2
ReportTitleBand4	←Nested Level 2
ReportTitleBand5	←Nested Level 2
Data	
ReportSummaryBand1	←Nested Level 2
ReportSummaryBand2	←Nested Level 2
ReportSummaryBand3	←Nested Level 2
ReportSummaryBand4	←Nested Level 2
ReportSummaryBand5	←Nested Level 1

For the **PageHeaderBands** and **PageFooterBands**, you can create a collection of styles only of the first nesting level.

3.3   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 picture below shows the main elements of the editor of conditions:
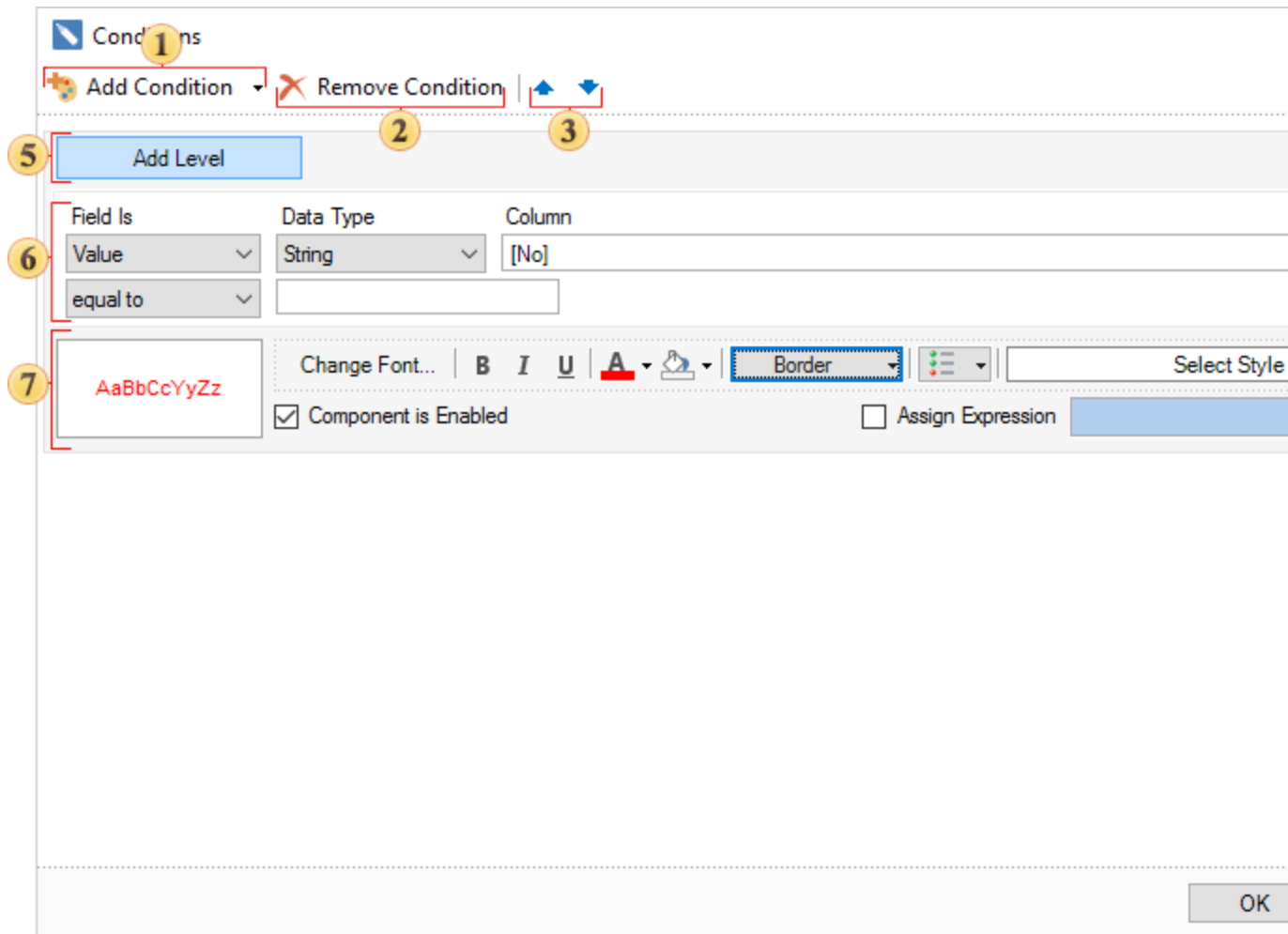
① **Add condition**

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 The buttons are used to move the selected level of conditions in the list. The higher the level is in the list, the higher is the priority of processing.

④ **Break if True**

By default, all the conditions of the levels are processed sequentially from top to bottom. Depending on the result, these or that format settings are applied. If you want to stop the processing of conditions so that the processing of the condition stopped when returning true, you should check this setting. In this case, the levels will be processed sequentially until to return the value **true**. Thereafter, subsequent processing of conditions (levels below) will be terminated.

⑤ **Add level**

This button adds one level of the condition parameter.

⑥ Parameters of a condition
Specify parameters of condition on this panel.
⑦ Format settings
Specify parameters of the appearance of the component on this panel.

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

3.3.1 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 reporting tool to 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).
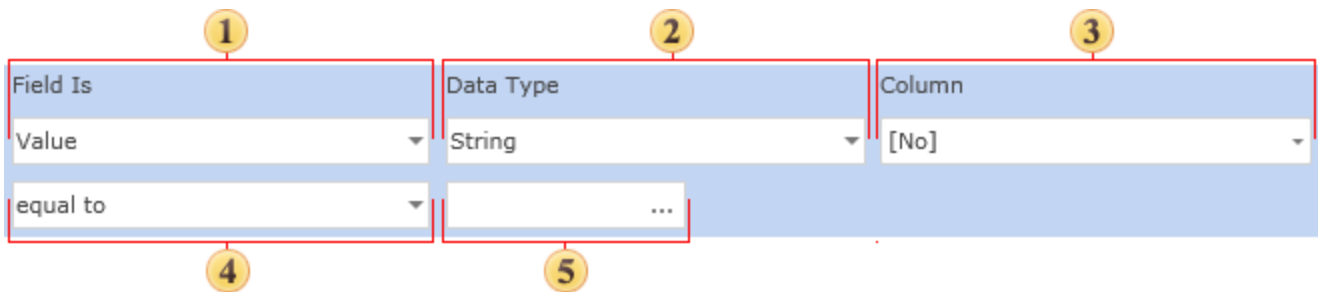
If you were writing a value condition in code, it would look like this:
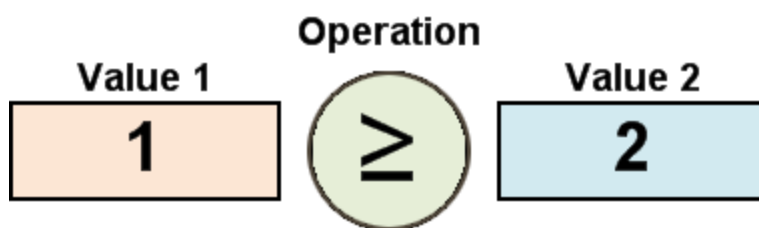
For several types of operation three values are used in calculating the condition. These are operations in which the value is checked to determine whether or not it is within a specified range, defined by two values. In addition to the elements described, the condition also includes a data type. The 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.

3.3.2 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:
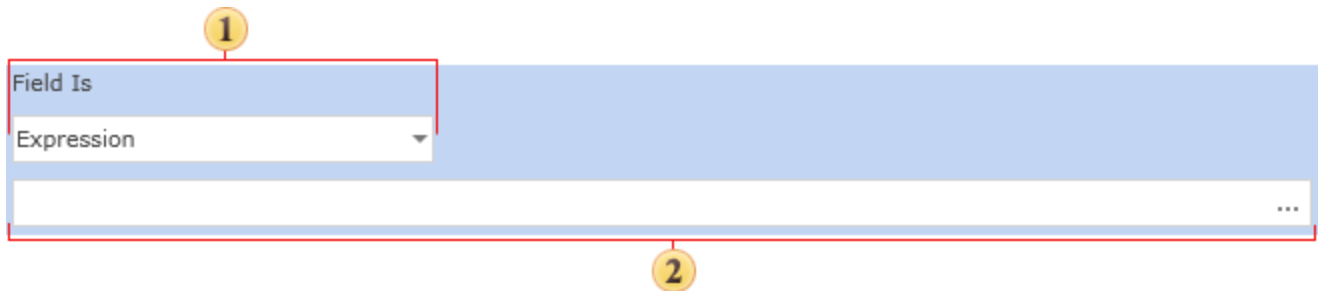
Operator	Types of data					Description
	String	**Numerical**	**Date**	**Logic**	**Expression**	
equal to	✓	✓	✓	✓	✓	If the first value is equal to the second, then the condition is true.
not equal to	✓	✓	✓	✓	✓	If the first value is not equal to the second, then the condition is true.
between		✓	✓		✓	If 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.
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.

3.3.3 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.

3.3.4   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:
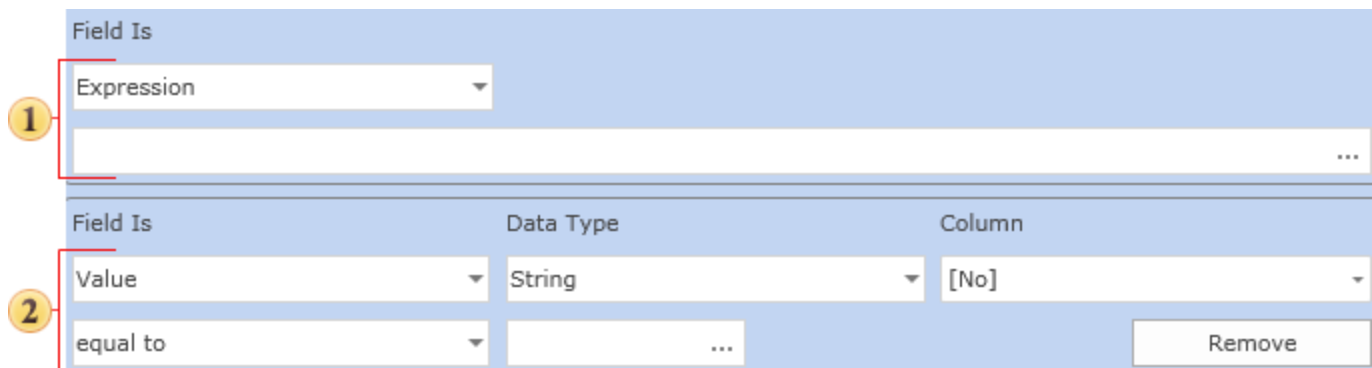
1 The first part of the condition.
2 The second part of the condition.

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
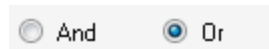
3.3.5   Defining Formatting

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:
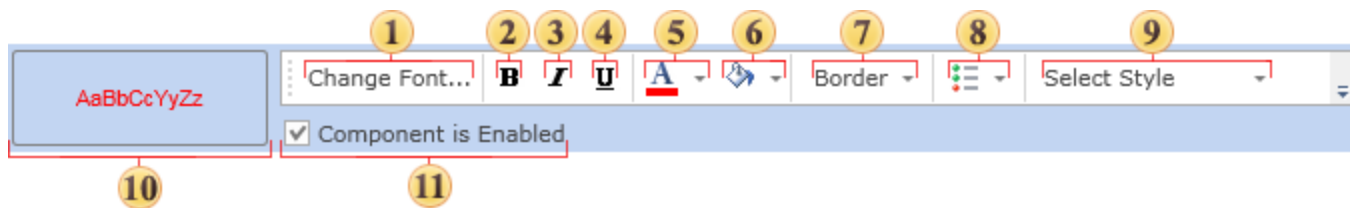
1 Font. Used to select the font.
2 Bold button. Used to define the bold font style.

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.
11 **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.

3.3.6   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.
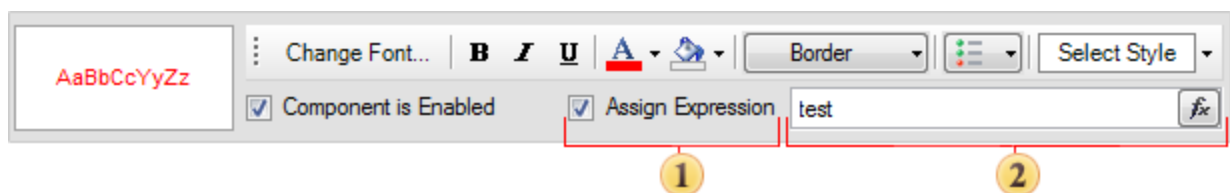
1 **Assign expression.** This flag controls whether or not a text expression is used in the condition. If it is disabled then the expression is not used.
2 **Text expression.** The text expression that will be assigned to a text component if the condition is true.

3.3.7 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"

3.3.8 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:
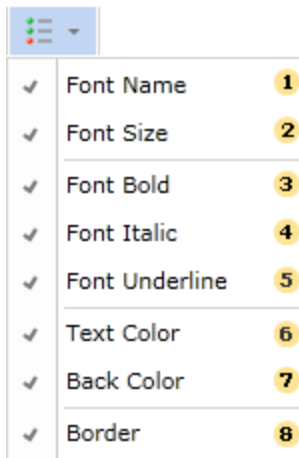
1 The **Font Name** menu item. Enabling/Disabling this item provides an opportunity to change/not change the font in the components that match the condition;

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;

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;

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;

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;

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;

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;

8 The **Border menu** item. Enabling of this item provides an opportunity to change the borders of components.

3.3.8.1 Font Name

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

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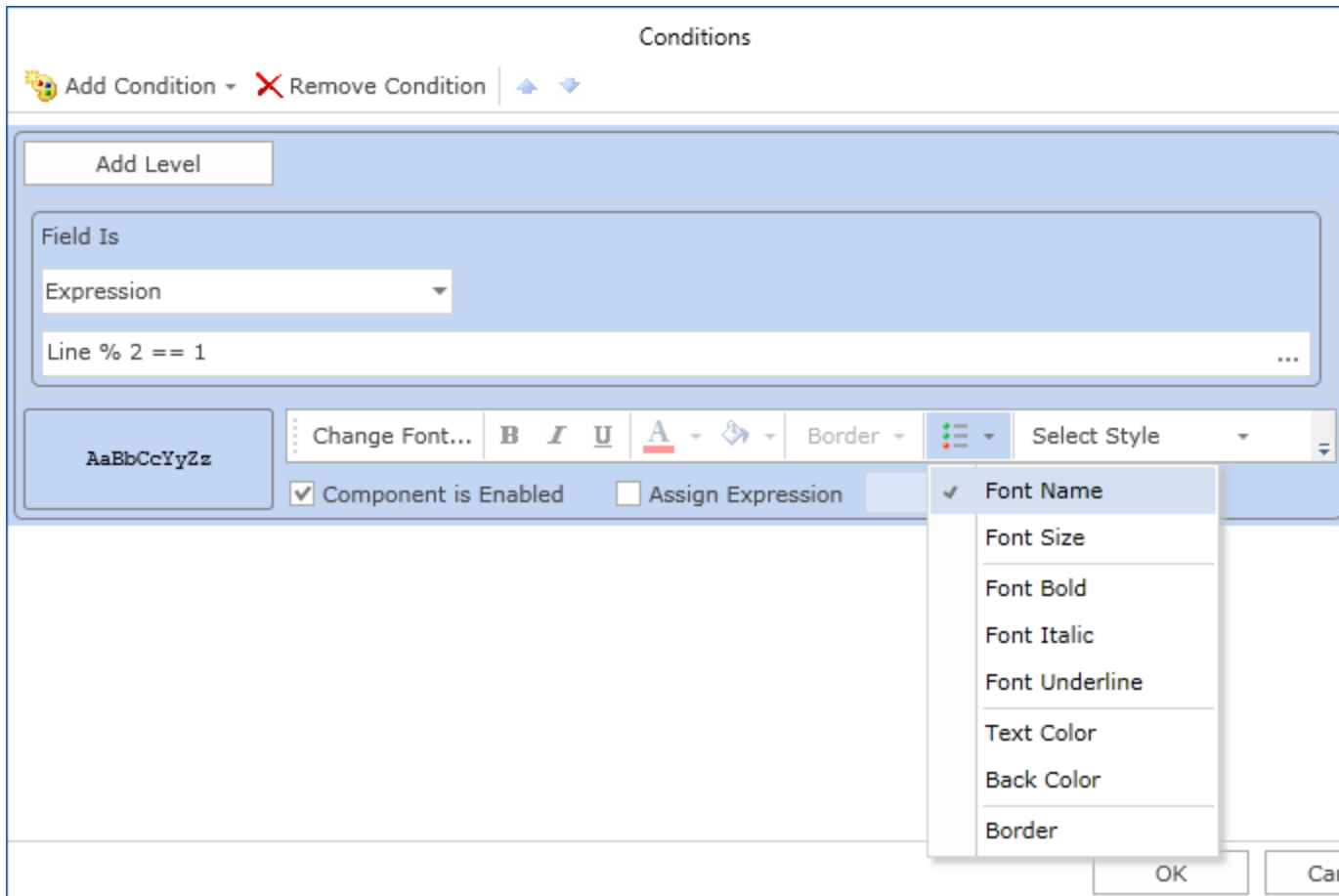
After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the font of the selected text component will be changed, depending on the condition. The picture below shows the page of the rendered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

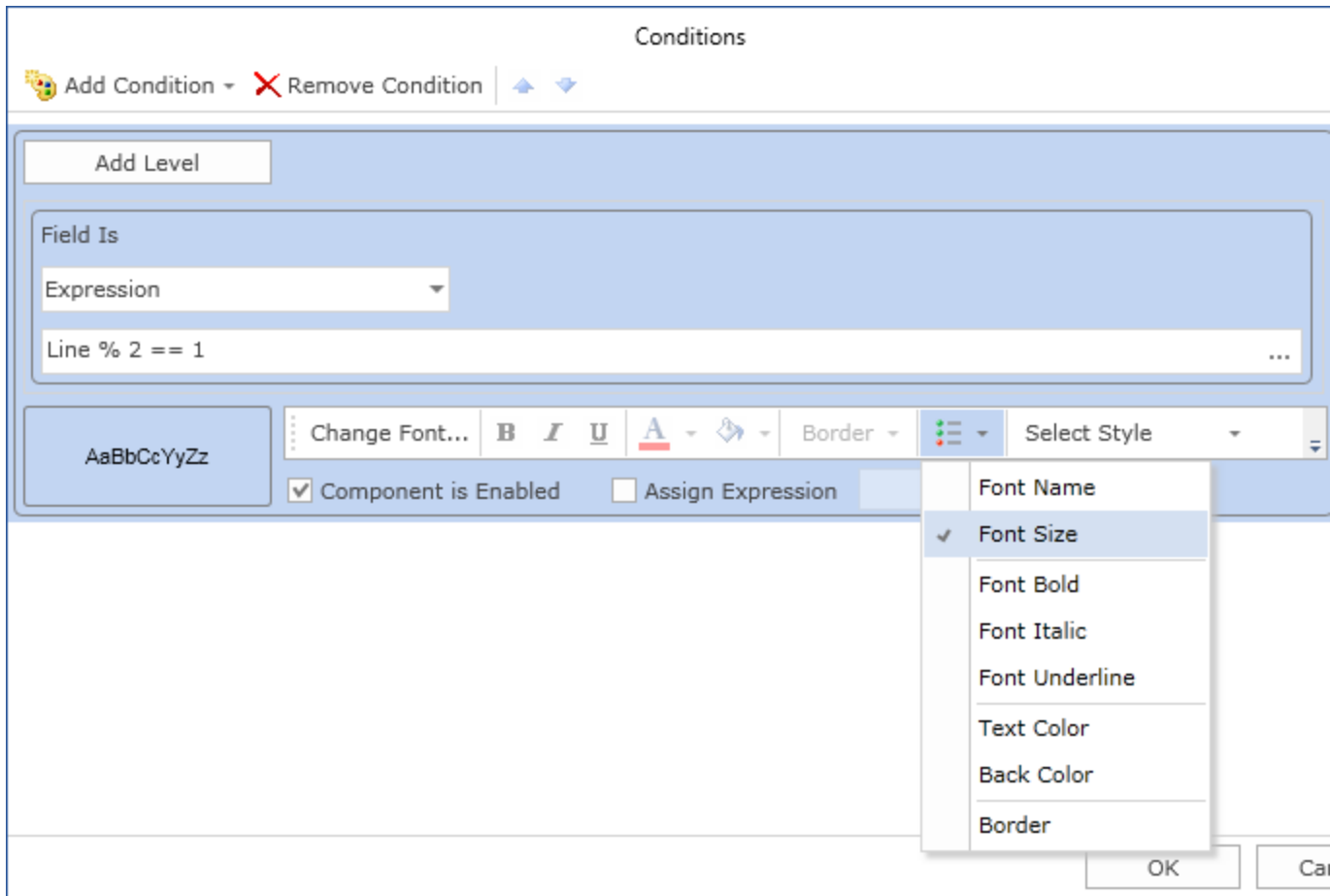| CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondesddsl père et fils	Strasbourg	France
Bólido Comidas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

3.3.8.2 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 report page:

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.

3.3.8.3   Font Bold

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

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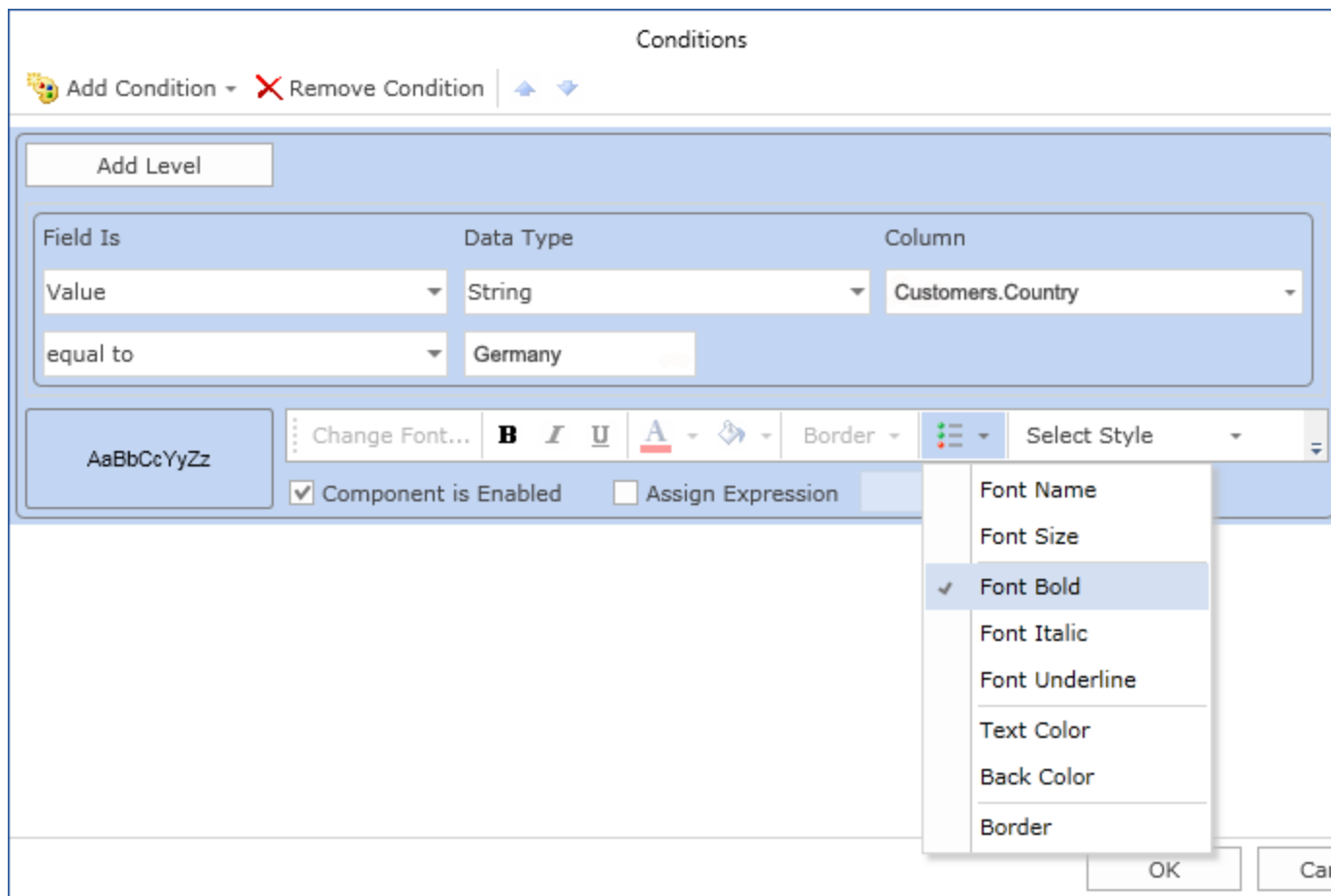
After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the bold font will be applied for the content of text components that match the specified condition. The picture below shows a page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

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

3.3.8.4   Font Italic

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

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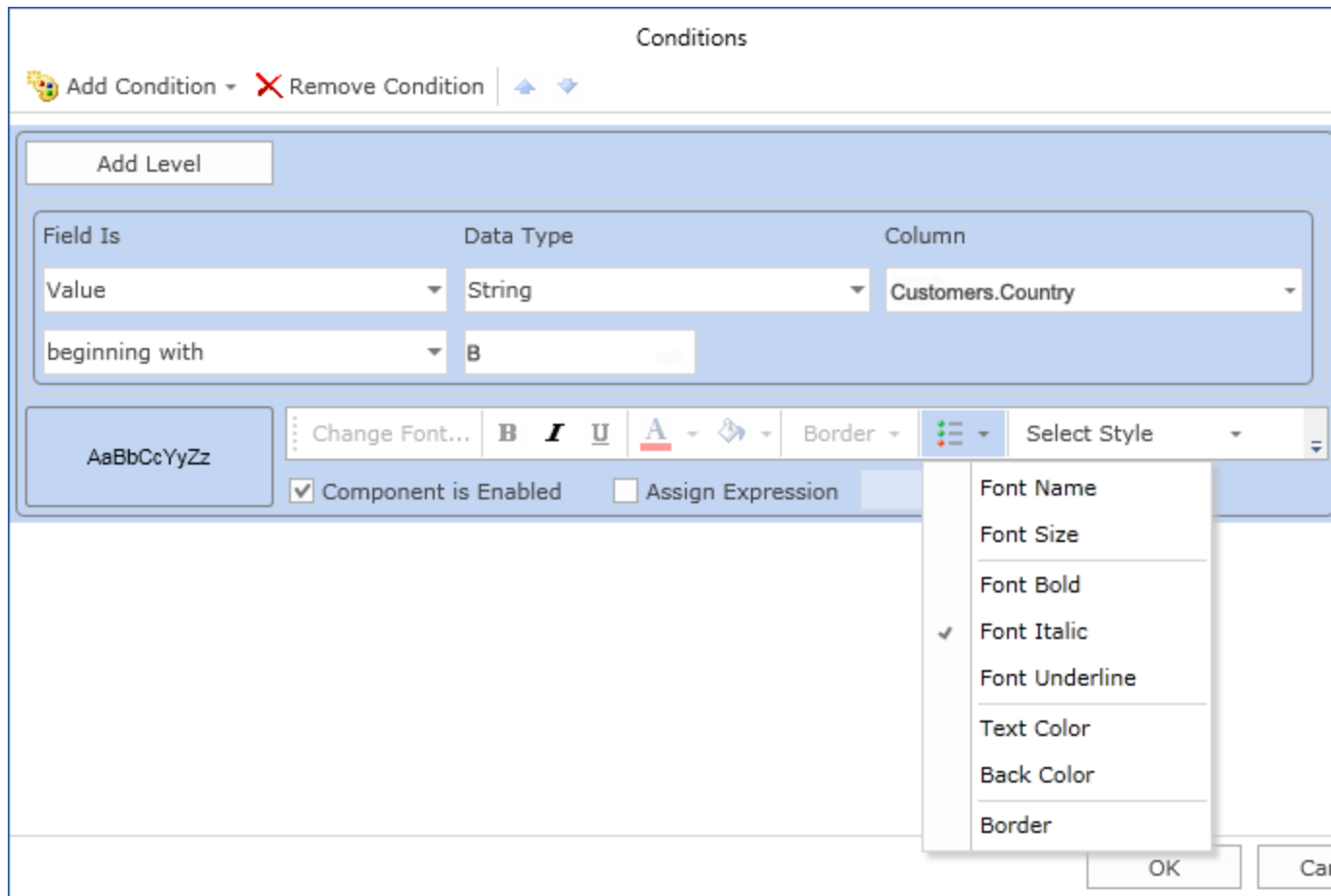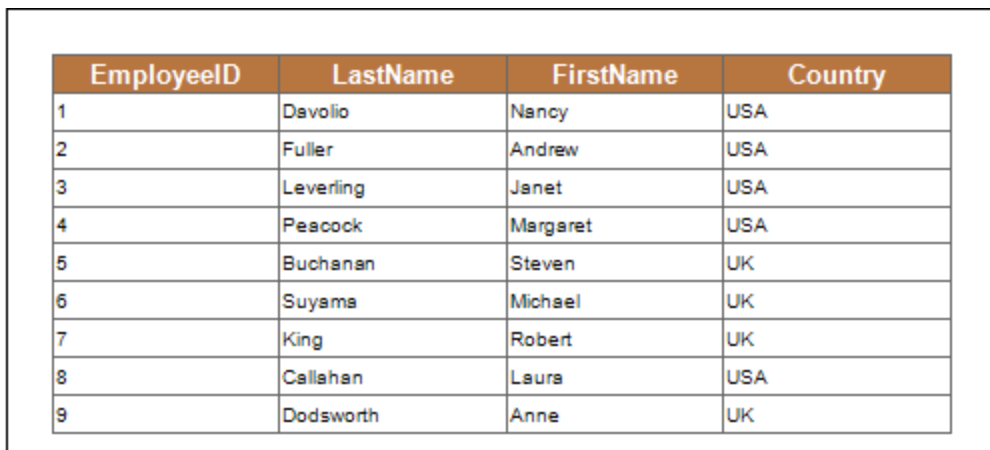
After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the italic font will be applied for the content of text components that match the specified condition. The picture below shows a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

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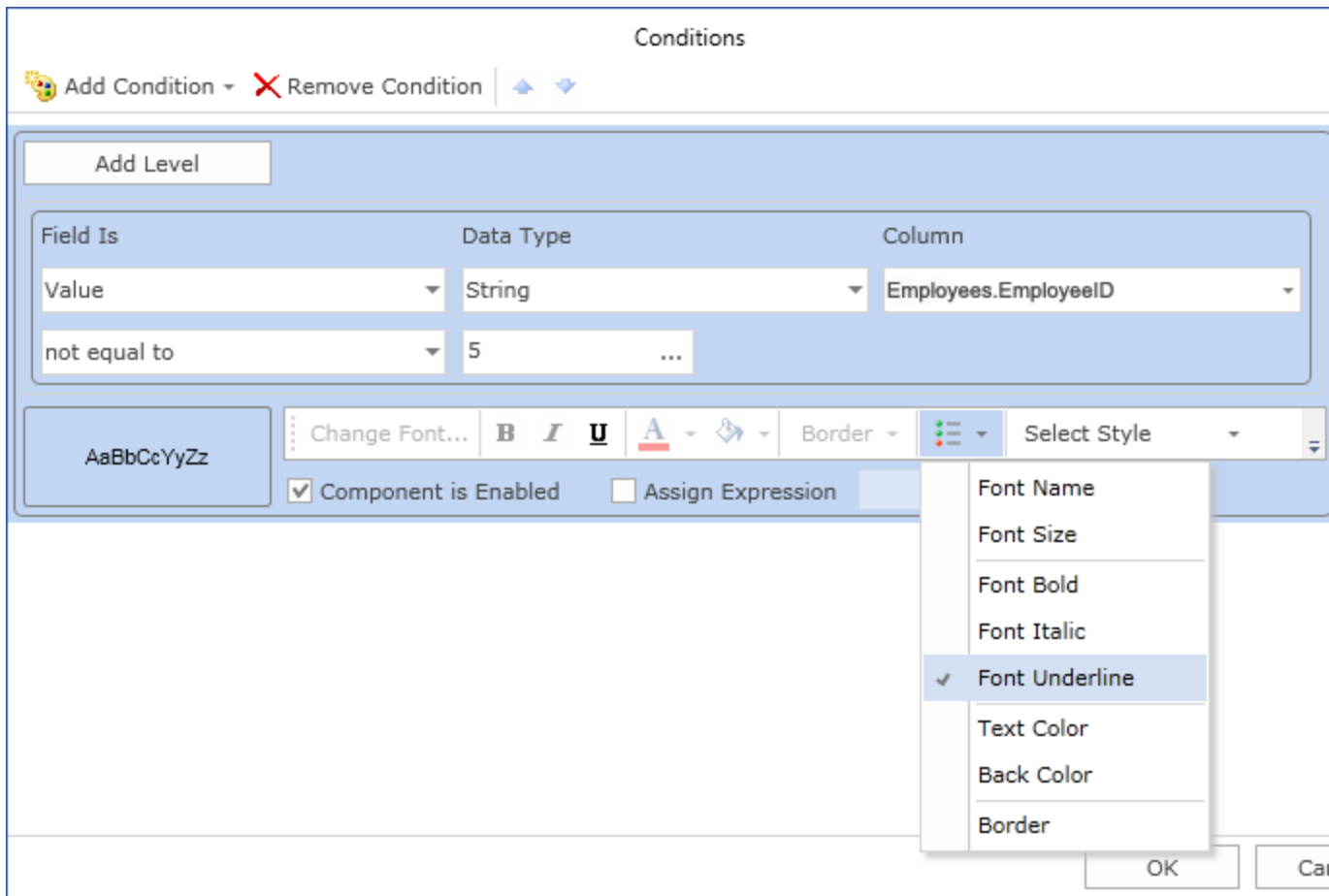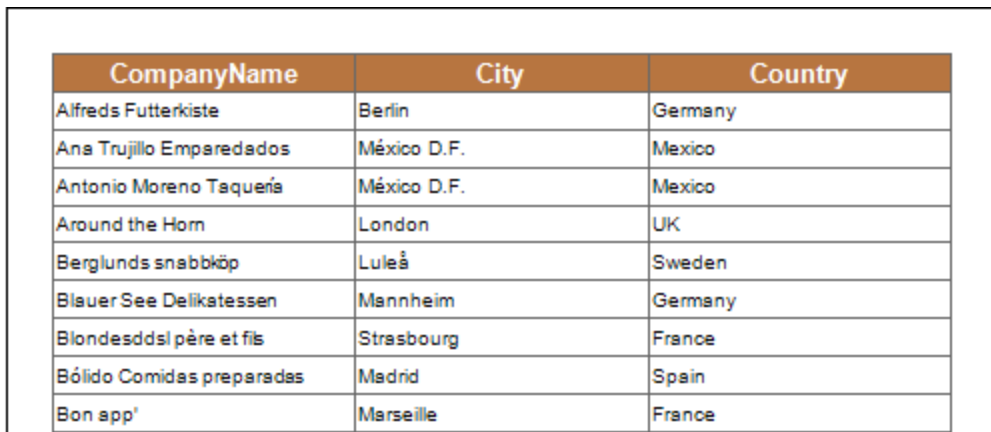
3.3.8.5   Font Underlined

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

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:

Conditions

Add Condition ▾   ✕ Remove Condition   ▲ ▼

Add Level

Field Is	Data Type	Column
Value ▾	String ▾	Employees.EmployeeID ▾
not equal to ▾	5 ...	

AaBbCcYyZz

Change Font... **B** *I* <u>U</u> A ▾ ◇ ▾ Border ▾ ≣ ▾ Select Style ▾

☑ Component is Enabled   ☐ Assign Expression

| Font Name |
| Font Size |
| Font Bold |
| Font Italic |
| ✓ Font Underline |
| Text Color |
| Back Color |
| Border |

OK   Ca

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	<u>Davolio</u>	Nancy	USA
2	<u>Fuller</u>	Andrew	USA
3	<u>Leverling</u>	Janet	USA
4	<u>Peacock</u>	Margaret	USA
5	Buchanan	Steven	UK
6	<u>Suyama</u>	Michael	UK
7	<u>King</u>	Robert	UK
8	<u>Callahan</u>	Laura	USA
9	<u>Dodsworth</u>	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.
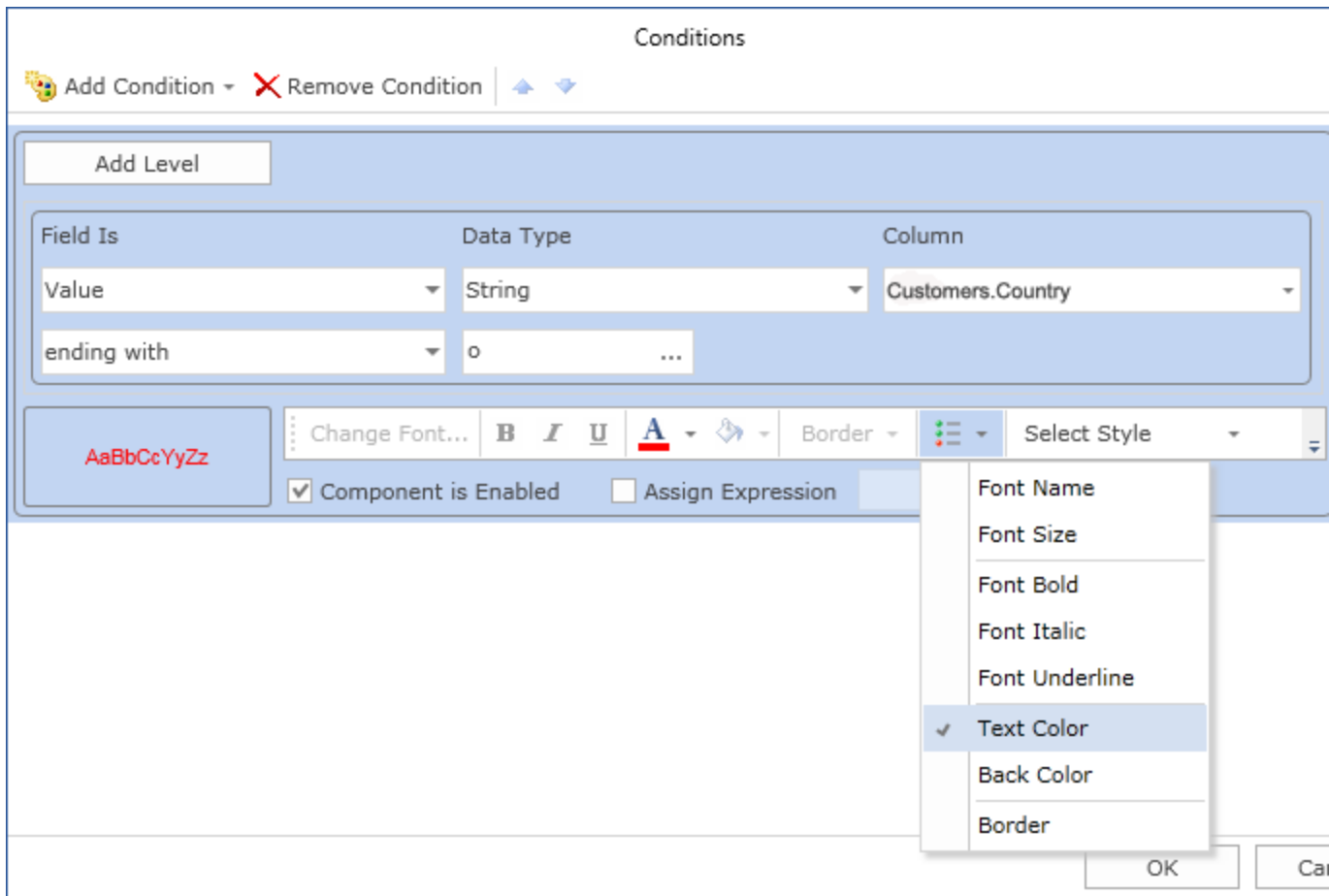
3.3.8.6 Text Color

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

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taqueria	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:

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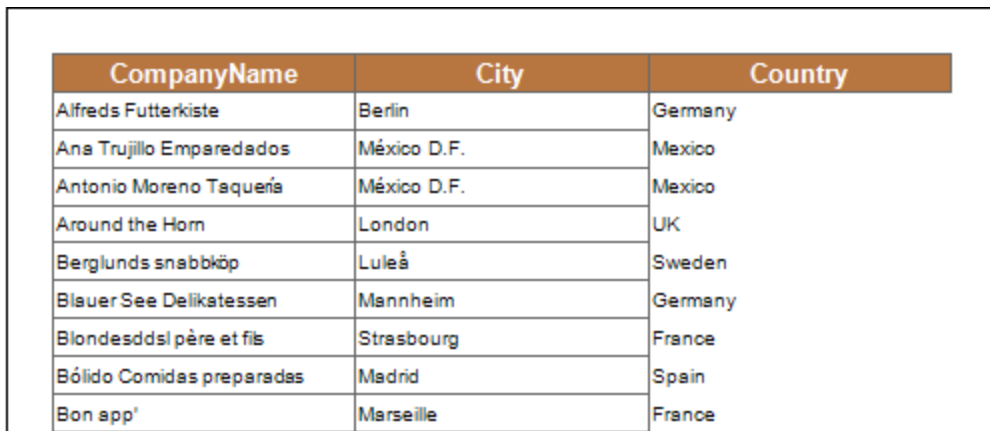
3.3.8.7   Back Color

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

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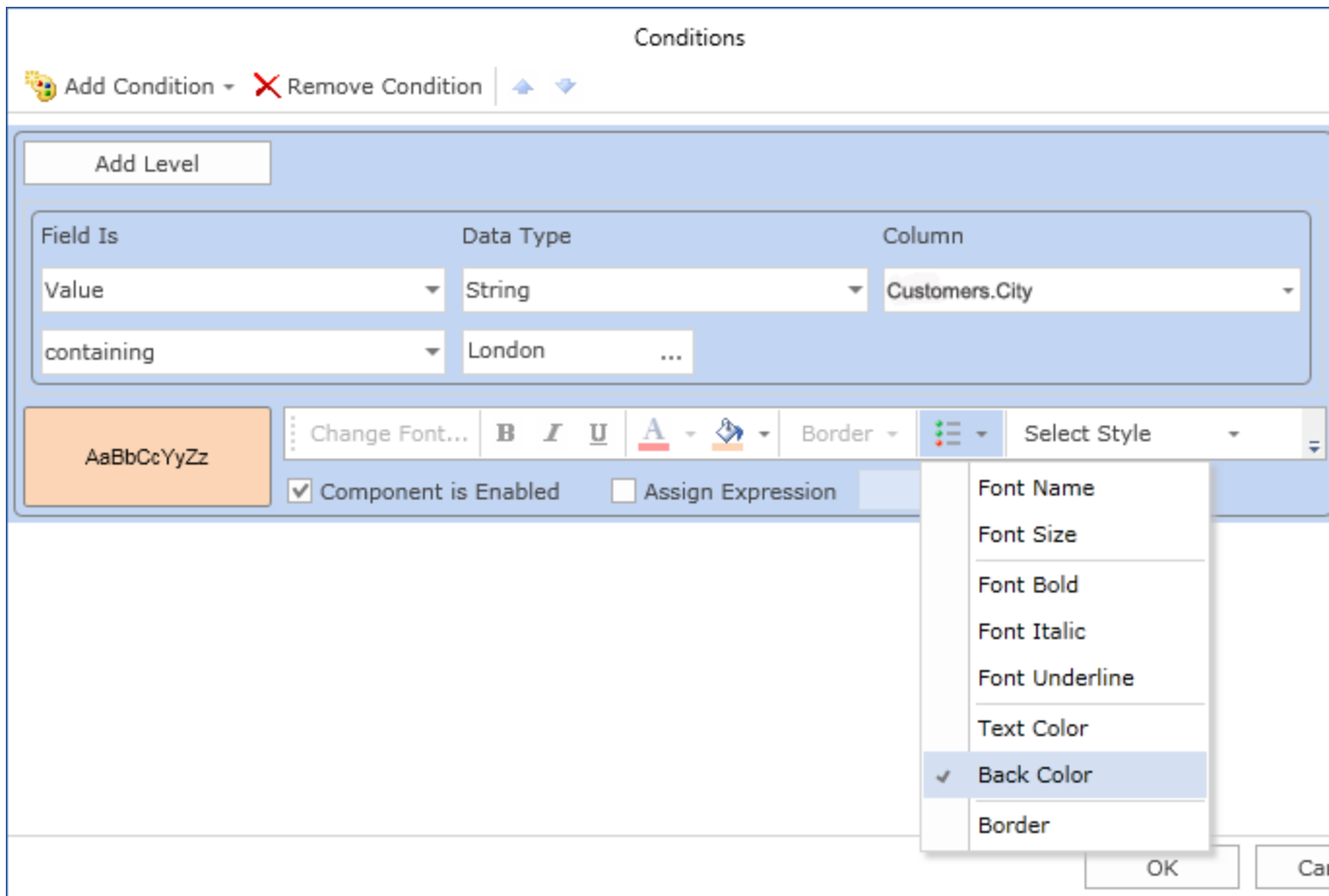
After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the background color will be applied for the content of text components that match the specified condition. The picture below shows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

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.

3.3.8.8 Borders

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

CompanyName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados	México D.F.	Mexico
Antonio Moreno Taqueria	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:
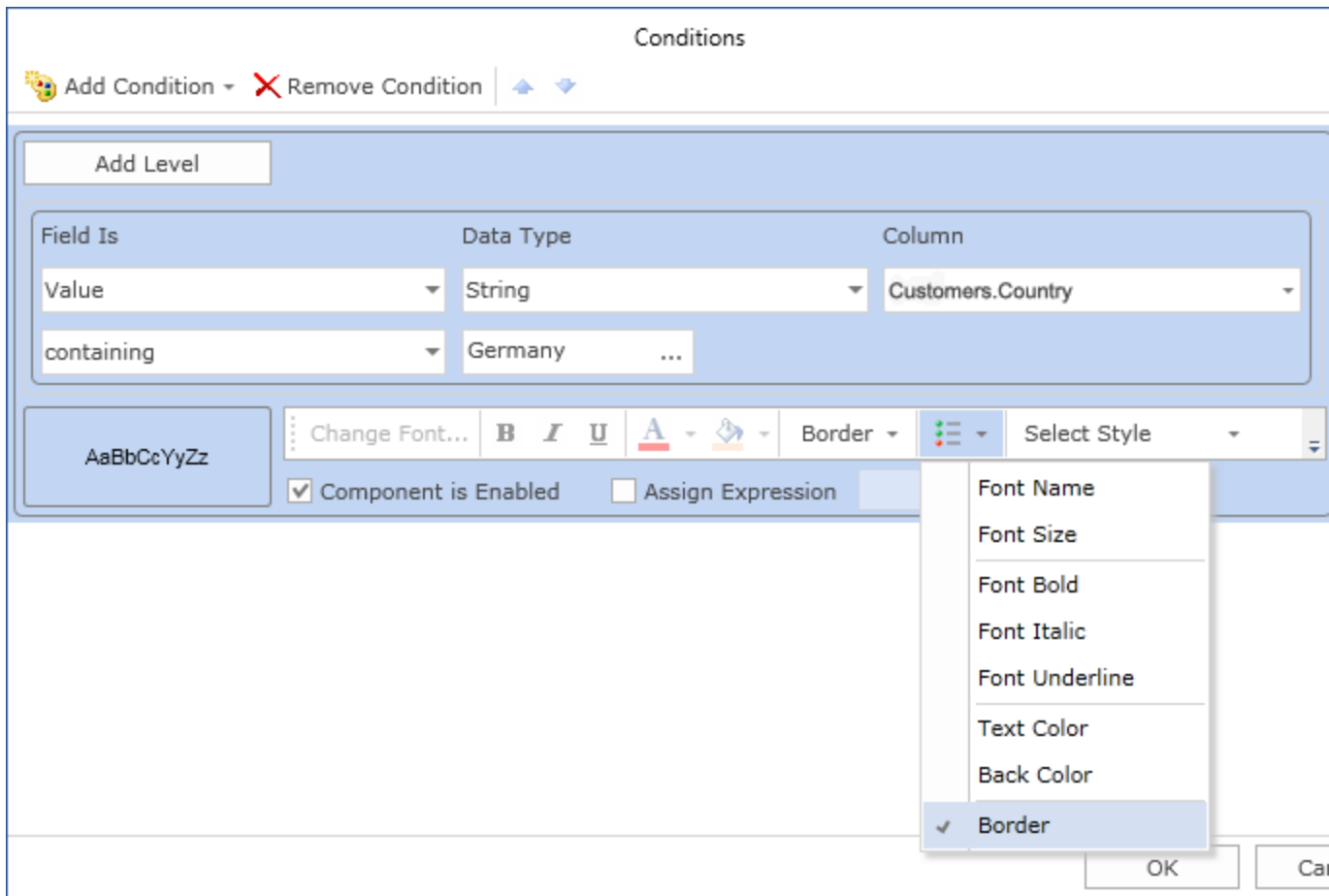
After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the borders will be set for the text components that match the specified condition. The picture below shows a page of the rendered report with conditional formatting:
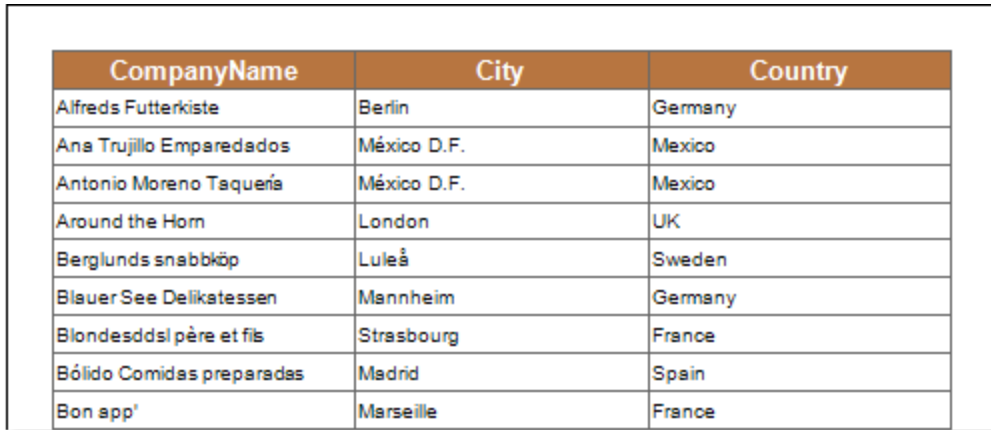

As can be seen in the picture above, borders of text components of the **Country**

column which contain the **Germany** word, will be set.

3.3.8.9   Enabling Component

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

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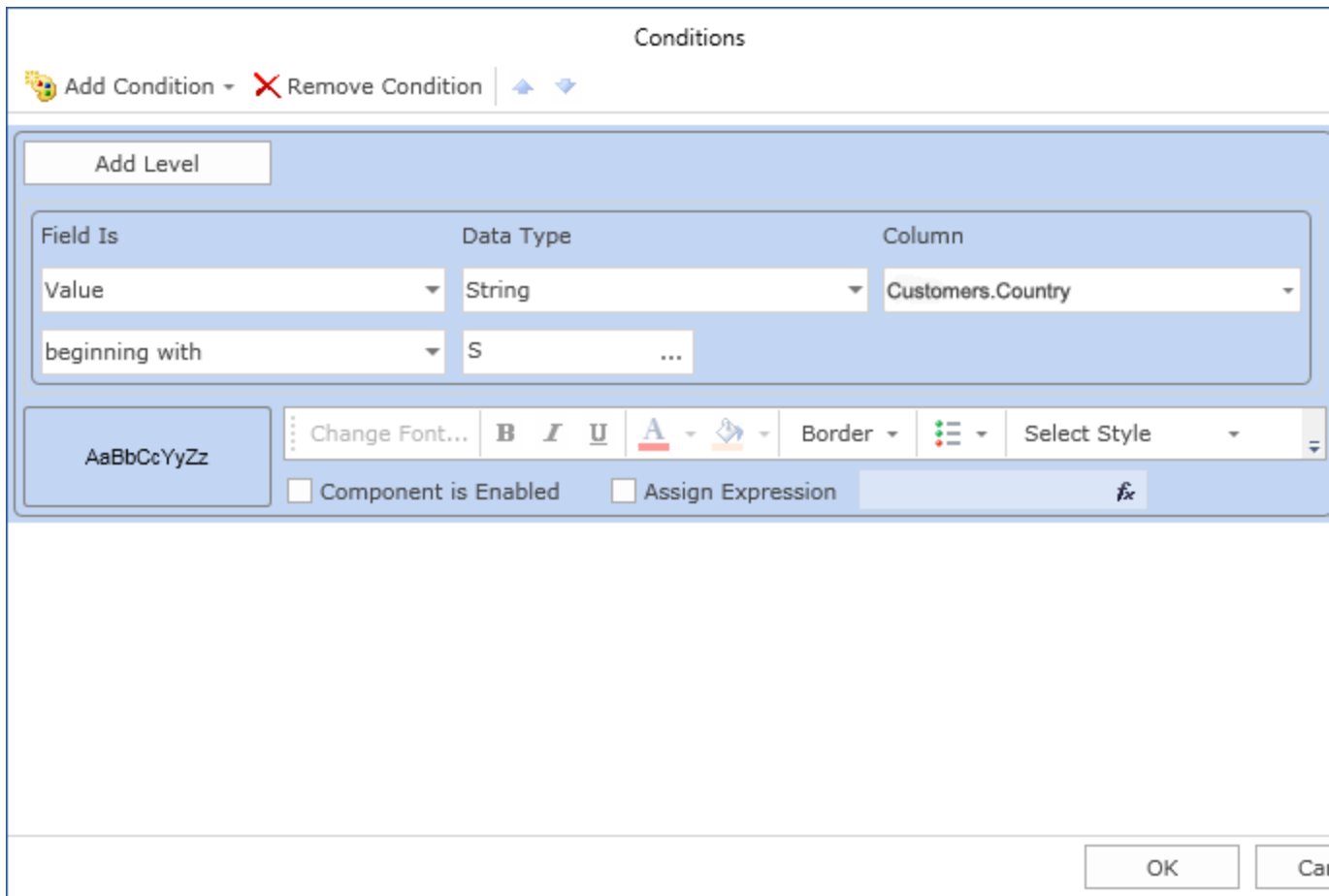
After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the borders the text components that match the specified condition will be hidden. The picture below shows a page of the rendered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

As can be seen in the picture above, the text components of the **Country** column which

lines start with the **S** letter are changed.

3.3.8.10 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 report page:

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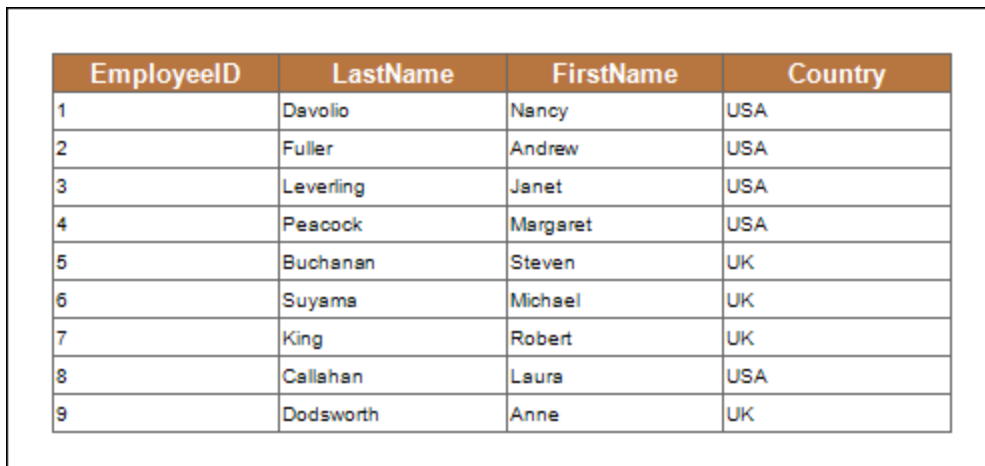
After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, assigning of the text expression in the text components that match the specified condition will be done. The picture below shows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

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.

3.3.9   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 report page:

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. Thus, if the data column has 25 as the minimum number and 100 as the maximum. In the component with a minimum number, the histogram will be rendered by 25 per cent. With this type, the extreme range of the value is 0.

> **Percentage** is used to specify a minimum value as a percentage;

> **Value** provides an opportunity to specify a minimum value as a numerical value,

> **Minimum** defines the minimum value in the selected data column and does not reset it to null. Thus, if the data column has 25 as the minimum number and 100 as the maximum. In the component with a minimum number, the histogram will not be rendered because 25 is the extreme value of the range.

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. Thus, if the data column has -25 as the maximum number and -100 as the minimum. In the component with a maximum number, the histogram will be rendered by 25 per cent. With this type, the extreme range of the value is 0;

➤ **Percentage** is used to specify a maximum value as a percentage;

➤ **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. Thus, if the data column has -25 as the maximum number and -100 as the minimum. In the component with a maximum number, the histogram will not be rendered because -25 is the extreme value of the range.

⚠ **Notice**: The difference between the **Auto** from the **Maximum** and **Minimum** may be noticeable only in a certain range of numbers.

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 data column 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 is determined by the minimum and maximum values. The minimum value is 0 per cent, maximum is 100 per cent. 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, and the

histogram of positive values of a color that is selected in the Positive. The picture 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	

The picture 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.

3.3.10   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 report page:

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:

**❶** The **Column** field. This field indicates the data column from which the value for the condition will be taken;

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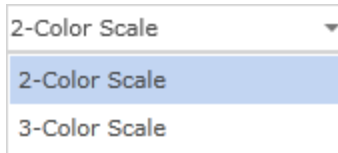
❸ The **Type** field provides an opportunity to change the type of a value that will be specified in the Value field for a minimum color scale. The picture below shows the menu to select the type of a value:
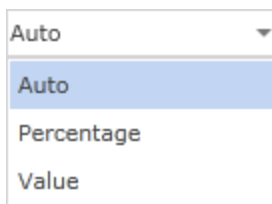
❹ The **Value** field. Used for a minimum color scale;

❺ The **Color** filed. Used for a minimum color scale;

❻ 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;

❼ A group of parameters (Type, Value, Color) of the medium color scale;

❽ 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.

3.3.11 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 report page:

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:

❶ 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;

❷ 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:
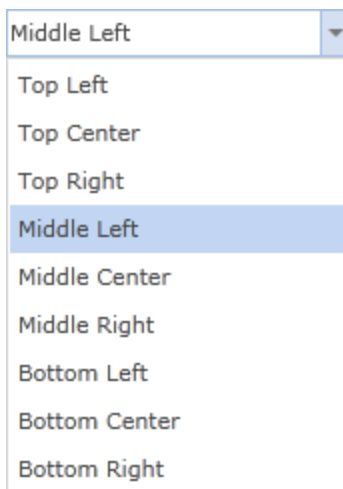

5 The **Icon** field shows the order of icons, and provides an opportunity to change the

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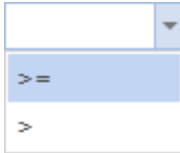
The Type field is used to change the type of a value of the first sub-condition. There are two values: Percentage and Value. The picture below shows the menu to select the type of a value:
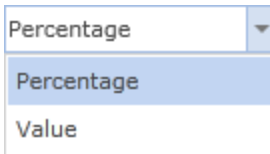
In the Value filed the value of a sub-condition is indicated.

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.

3.4   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:
✓ **Text** is the basic component to output text in the report. The component supports a large number of different settings, processing and displaying text;
✓ **RichText** is a component used to output of an RTF text;
✓ **Text in Cells** is a special component to output a text in a cell.

The text component can contain a simple text, and expression. Whether it is a calculation of an expression, or just a reference to a data column. Text components can be placed on other components, or directly on the report page. But no other components can be put into the text component.

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

3.4.1   Text Editor

Editing text components can be done in the **Text Editor**. This editor contains several tabs in which you can change an expression of the text component, select a column, system variable, specify the calculation results.

❯ The tab **Expression**

In the tab **Expression**, you can specify a text, expression, reference to any item in the data dictionary:
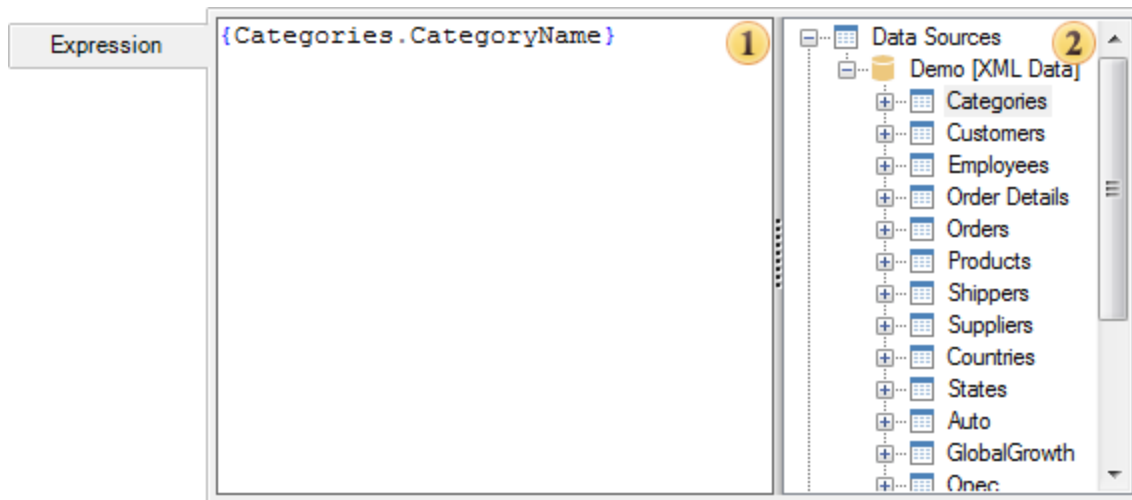
This tab has the following panels:

1 The panel **Text** where you can directly specify a text of the expression, reference to an item in the data dictionary.

2 The panel **Data Dictionary** contains items of a report data dictionary. It also supports **Drag and Drop** of items from the panel **2** to the panel **1**. At the same time, a reference will be automatically generated on the data dictionary item. In the picture above you see that the expression **{Categories.CategoryName}** is a reference to the description of the data columns **CategoryName** (data source **Categories**) in the report data dictionary.

> The tab **Data Column**
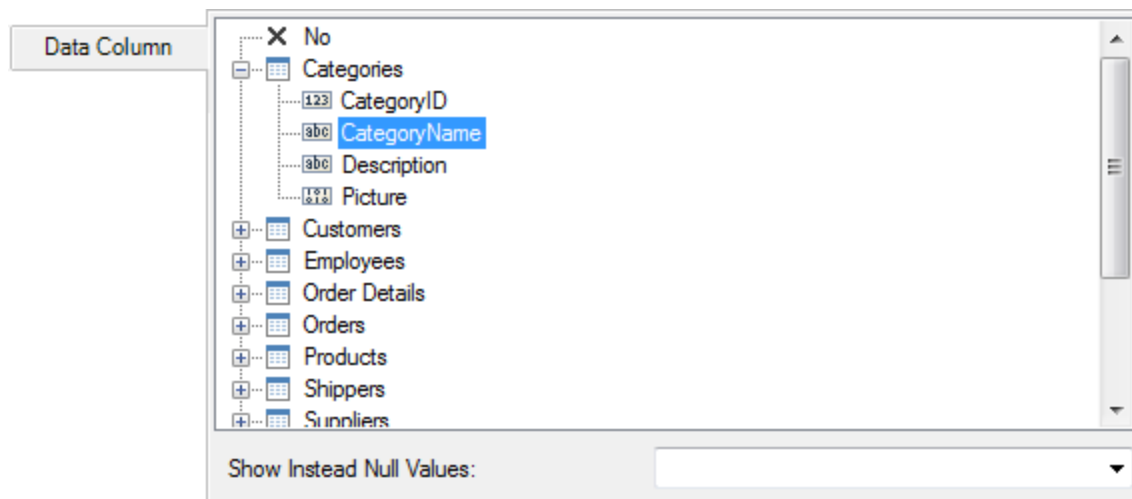
This tab is represented by a single panel, which displays only the data columns from the **Dictionary**. When you select a column, an expression will be formed. This

expression is a reference to the description of this column in the report data dictionary. Also on this tab you may find parameter **Show Instead Null Values**, using which you can specify the characters to be displayed instead of the zero values of selected data columns.

❯ The tab **System Variable**
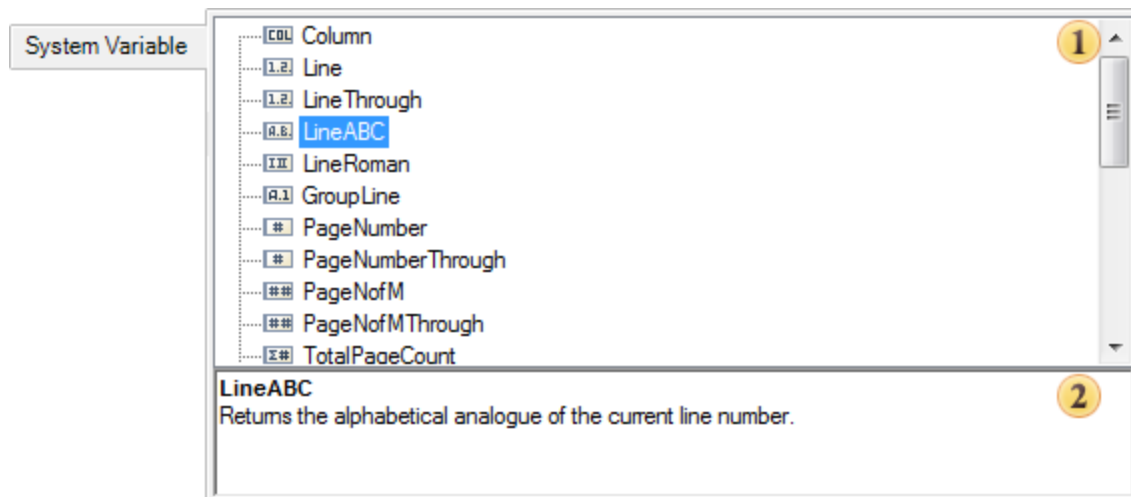
This tab has the following panels:

① The panel **System Variable**. This panel displays all the system variables of the data dictionary. A system variable is selected here, which will form the reference in the text component.

**②** The panel **Descriptions**. This panel displays a description of the selected variable.

❯ The tab **Summary**

On this tab, you can create an expression that calculates summary. The result of it will be displayed in this text component:

1 In this drop-down list you may determine the type of an aggregate function (operation) to calculate the summary.

2 In this drop-down list you can select the data band by which the summary will be calculated.

3 This list defines the data column, the values of which will be calculated totals.

4 This radio button sets the calculation function for the entire report. The value of the function in the any place of the report will be the same.

5 This radio button sets the calculation of the functions of the data column.

6 This radio button sets the calculation of the function by a report page. On each report page the total value will be calculated only on the page.

7 The checkbox sets the calculation mode with the running total. Each subsequent result includes all the previous ones.

8 The checkbox Condition allows you, when calculating totals, to take into account the value only when executing a certain condition.

9 The field is used for the condition expressions. Available when the checkbox Condition is enabled.

3.4.2   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.
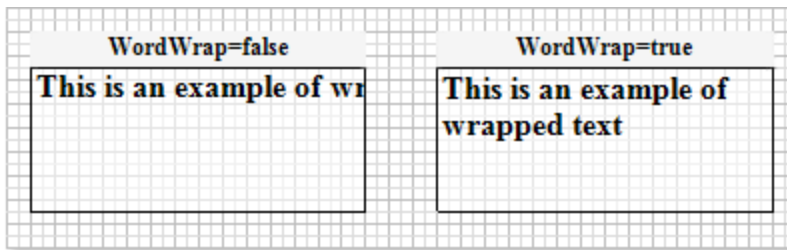
3.4.3   Trimming in the End of Text Line

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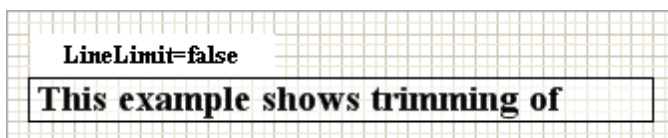
None - the text is trimmed strictly by the edge of a text component or, if it is a multiline text, by the last visible word;
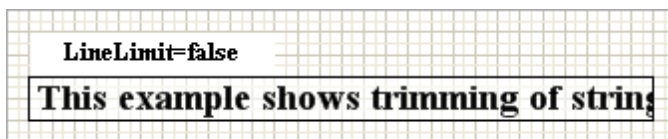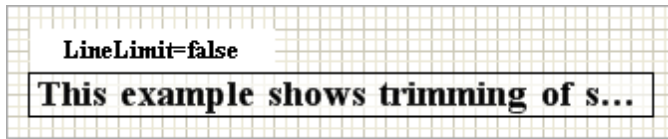

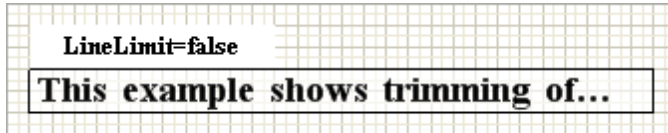**Character** - the line is trimmed after the last visible character;
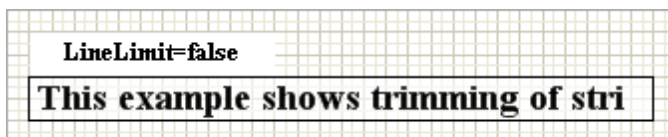
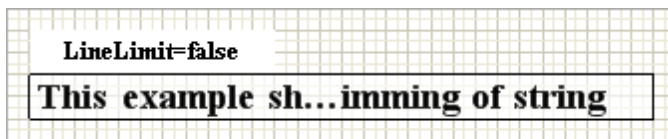**Word** - the line is trimmed by the last visible word;

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.

3.4.4 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.
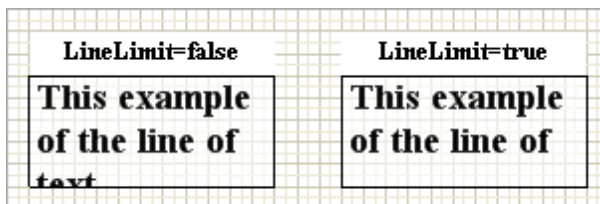
3.4.5 Lines of Underlining

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.
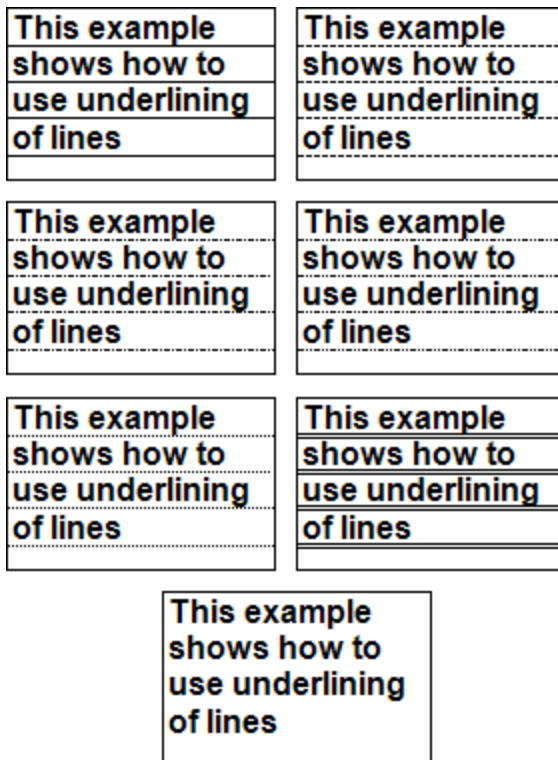
3.4.6 Maximal Number of Lines

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.
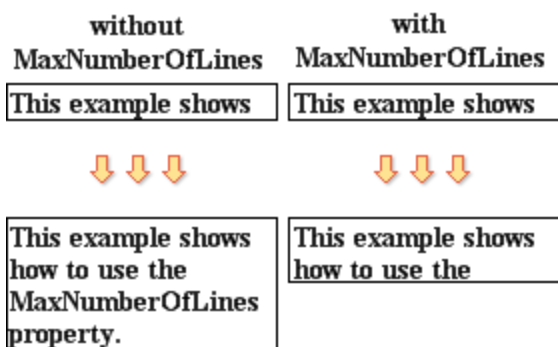
3.4.7   Text Rotation

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.
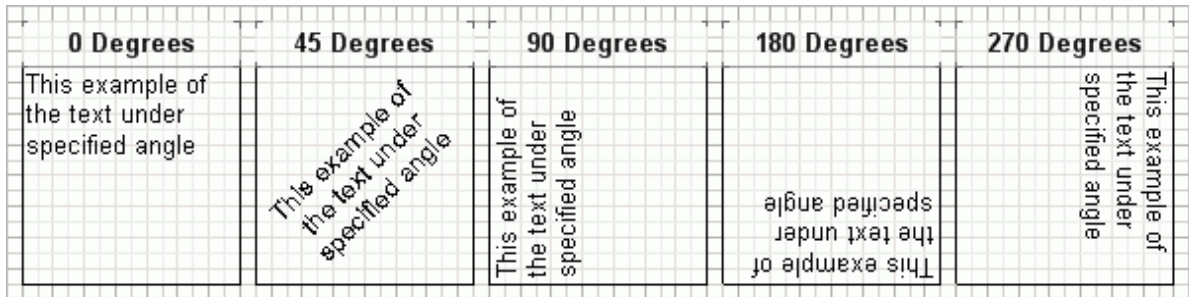
3.4.8   Processing Duplicates

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 pied	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.

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.

Remove Text - 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.

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.

3.4.9 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**).
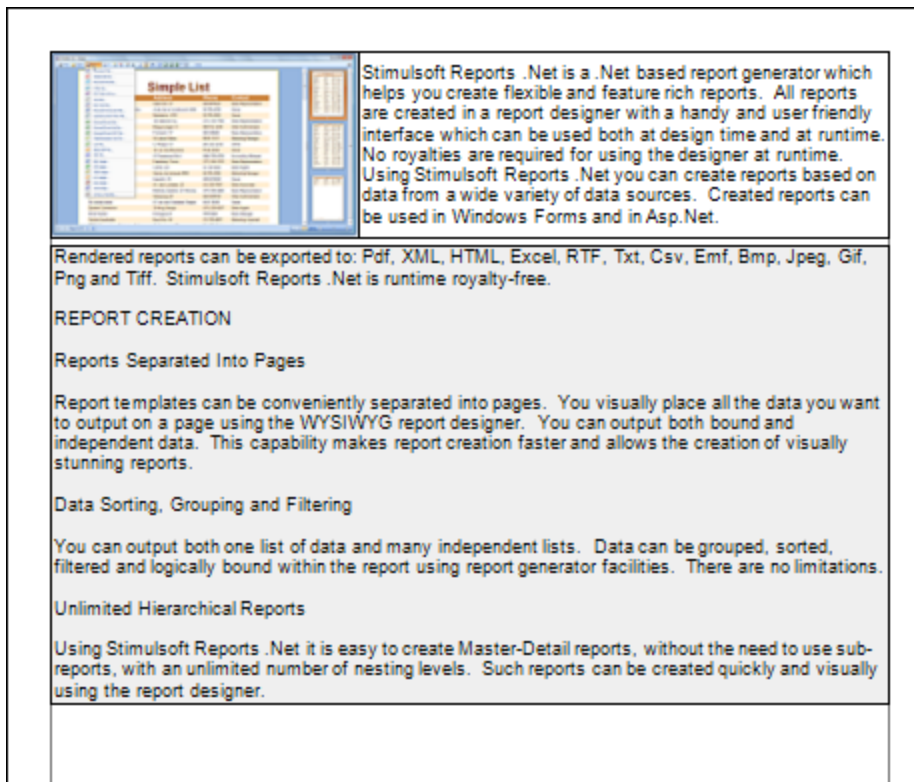
3.4.10 ReportTo Property

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.
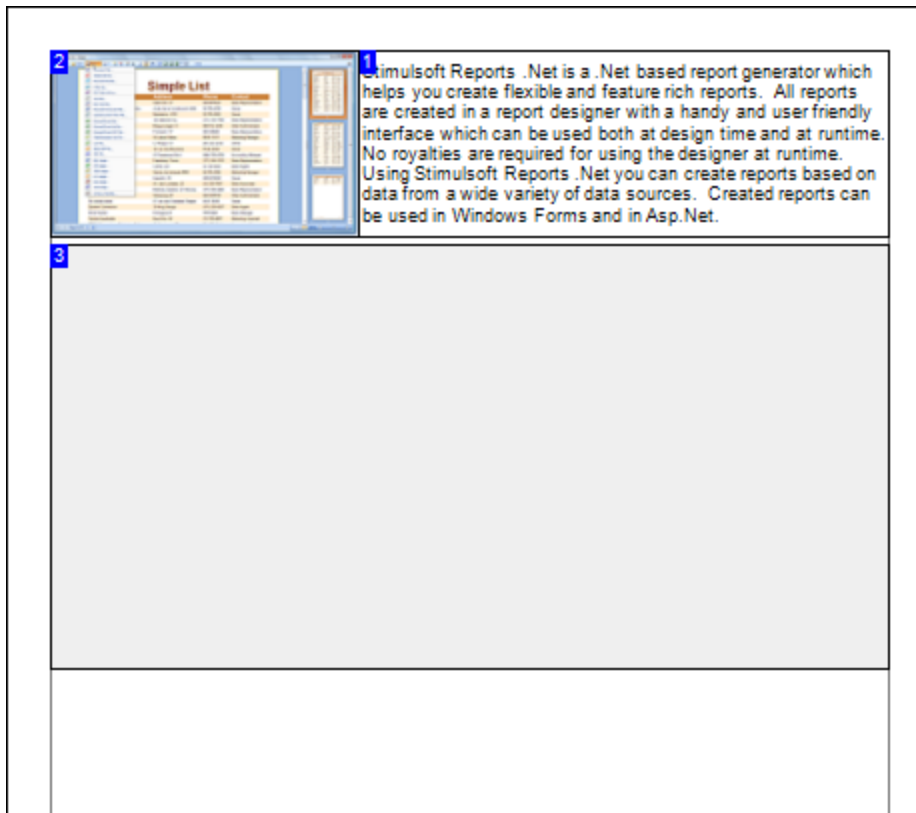
The result can be seen on the picture below.

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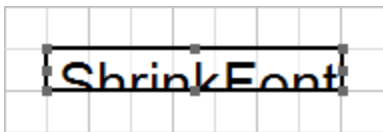
3.4.11 Shrink Font To Fit Property

The **Shrink Font To Fit** property of a text component is used when it is necessary to adjust the height of the text to the size of the text component. This property can be found on the Properties Panel.

◢	**1. Text**		
	Text	Σ	**ShrinkFontToFit Shri**
▷	Text Brush	■	**Solid**
▷	Font		**Arial; 8pt**
	Horizontal Alignment	Left	
	Vertical Alignment	Top	
	Text Format	**General**	
◢	**2. Text Additional**		
	Allow Html Tags	False	
	Angle	0	
	Editable	False	
	Hide Zeros	False	
	Lines of Underline	☐ None	
	Margins	**0;0;0;0**	
	Max Number of Lines	0	
	Only Text	False	
	Process at	None	
	Processing Duplicates	None	
	Render to		
	Shrink Font to Fit	**True**	▼
	Shrink Font to Fit Minimum Size	1	
	Text Quality	Standard	
▷	Text Options	**HotkeyPrefix=None; LineL**	
	Word Wrap	False	

The property can take two values: **true** and **false**, respectively, that means the property is enabled or disabled. By default, the property is set to false.
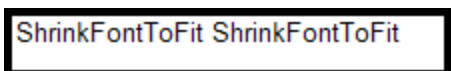
The picture below shows a component with the text, which is clearly larger than the size of the component.

When the **Shrink Font To Fit** property is set to **false**, the text in the viewer will look like on the picture below
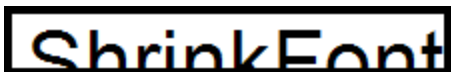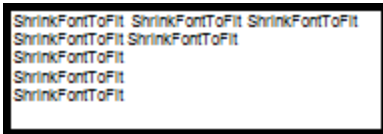

When the **Shrink Font To Fit** property is set to **true**, the text in the viewer will look like on the picture below

❗ **Notice:** The Shrink Font To Fit is a post-processing property and this should be taken into account when adjusting the text component. If you enabled CanBreak and CanShrink properties, then, when rendering a report, the text component will take a size corresponding to the height of the text on the basis of preset font size.

CanBreak and **CanShrink** properties are disabled, but **Shrink Font To Fit** is set to **true**

CanBreak and **CanShrink** properties are enabled, but **Shrink Font To Fit** is set to **true**

3.4.12   Shrink Font to Fit Minimum Size Property

The **Shrink Font to Fit Minimum Size** property of the text component is used to adjust the minimum size of the font to which the text should be reduced. This property can be found on the Properties Panel.

◢	**1. Text**	
	Text	Σ **ShrinkFontToFit Shrir**
▷	Text Brush	■ **Solid**
▷	Font	**Arial; 8pt**
	Horizontal Alignment	Left
	Vertical Alignment	Top
	Text Format	**General**
◢	**2. Text Additional**	
	Allow Html Tags	False
	Angle	0
	Editable	False
	Hide Zeros	False
	Lines of Underline	☐ None
	Margins	0;0;0;0
	Max Number of Lines	0
	Only Text	False
	Process at	None
	Processing Duplicates	None
	Render to	
	Shrink Font to Fit	**True**
	Shrink Font to Fit Minimum Size	1
	Text Quality	Standard
▷	Text Options	**HotkeyPrefix=None; LineL**
	Word Wrap	False

Images below show how this property works
The **Shrink Font to Fit Minimum Size** property is set to **1**. The font **Arial**, size **8pt**

ShrinkFontToFitMinimumSize
ShrinkFontToFitMinimumSize

The **Shrink Font to Fit Minimum Size** property is set to **4**. The font **Arial**, size **8pt**

ShrinkFontToFitMir
ShrinkFontToFitMir

❗ **Notice:** Works in association with the **Shrink Font To Fit** property set to **true**.

3.4.13   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.
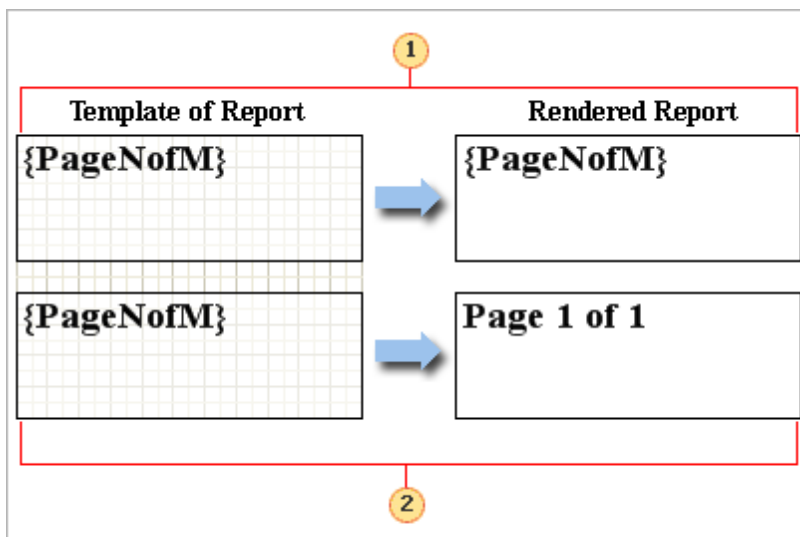
① The **TextOnly** property is set to **true**. The text is output "as is", without processing of expressions.

② The **TextOnly** property is set to **false**. The text is output with processing of expressions.

## 3.4.14 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 **Process At** 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.

3.4.15 Zip code

Zip code is used for mailing, in order to facilitate sorting. Stimulsoft Reports has a special component to display this code. It is called the Zip Code component. It can be placed on components, bands and pages. Setting the values of this component is

possible by means of the Code property. This value of the property can be any character, but the Zip Code component can only display numbers. The picture below shows a zip code with numbers "123456789":
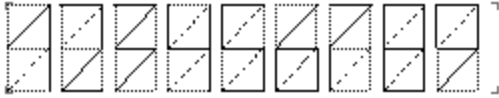

In order to increase the font size, change the value of the Size property, specifying the size with numbers, the higher the value is, the thicker is the width of the elements. The picture below shows a zip code with an increased width:
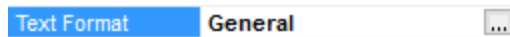

3.5 Text Formatting

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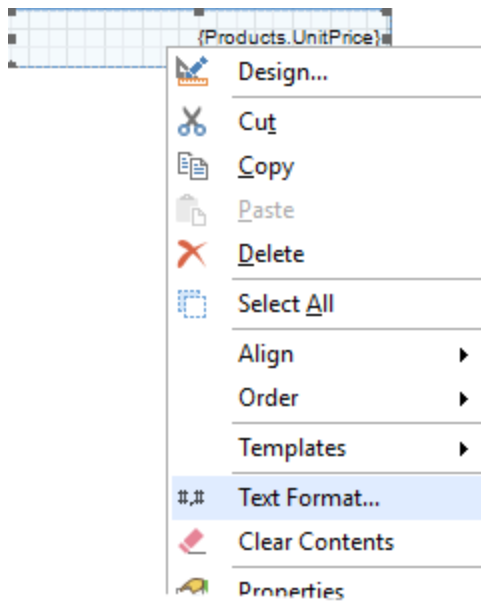
Also, using **TextFormat** properties, the dialog box can be called.

Text Format   **General**   ...

The Format window is divided into three parts.
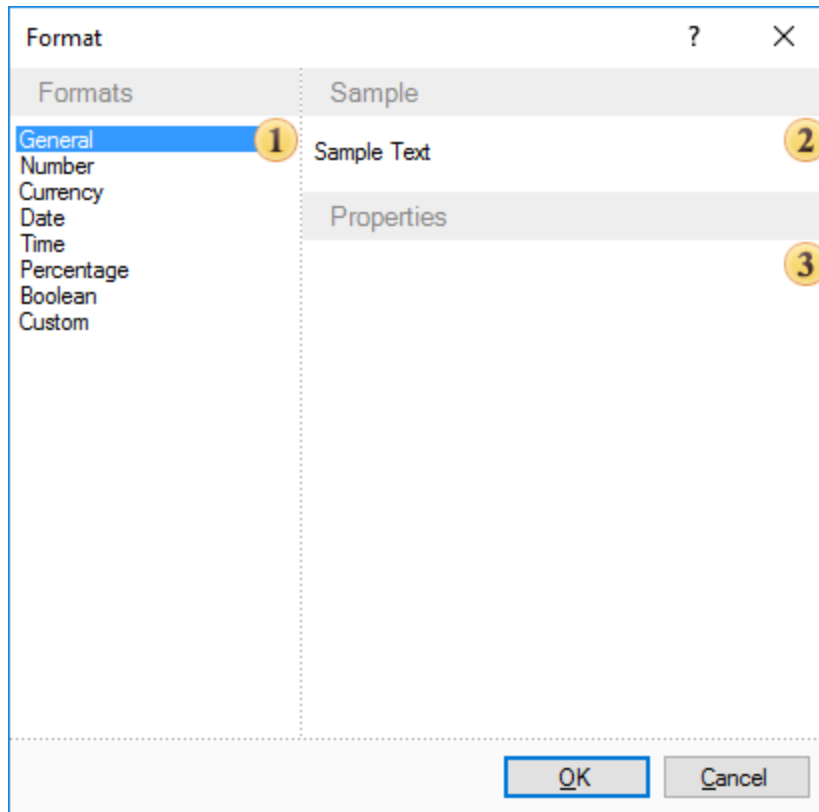
1 A section where the formatting type can be chosen.
There are some types of showing a text:

> ➤ **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.

3.5.1 Standard Formatting

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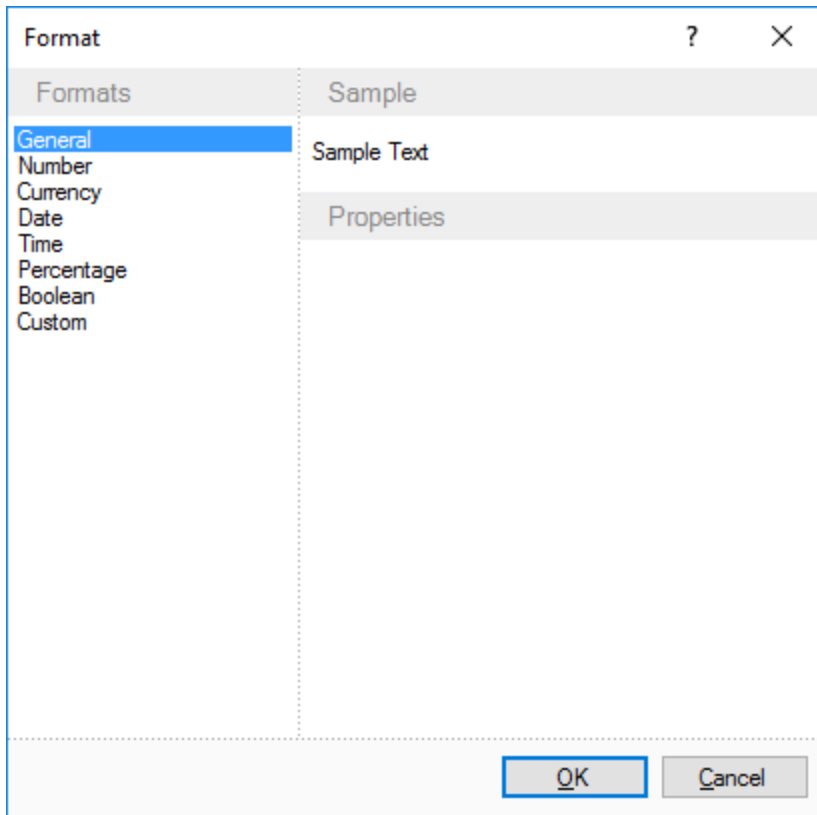
3.5.2 Numerical Formatting

To display numeric values, it is recommended to use a numeric format. Below is a report with a list of products, their price, as well as key product and category. By default, all text components use a text format General without any formatting.

ProductName	ProductID	CategoryID	UnitPrice
Chai	1	1	18
Chang	2	1	19
Aniseed Syrup	3	2	10
Chef Anton's Cajun Seasoning	4	2	22
Chef Anton's Gumbo Mix	5	2	21.35
Grandma's Boysenberry Spread	6	2	25
Uncle Bob's Organic Dried Pears	7	7	30
Northwoods Cranberry Sauce	8	2	40
Mishi Kobe Niku	9	6	97
Ikura	10	8	31
Queso Cabrales	11	4	21
Queso Manchego La Pastora	12	4	38

Set the numeric format for the values **ProductID**, **CategoryID**, **UnitPrice**. For this you should select the text components which contain references to the relevant data columns and click the ⬚ button of the **Text Format** property. In the **Format** dialog box you should go to the **Number** tab and define the settings.

ProductName	ProductID	CategoryID	UnitPrice
Chai	1.00	1.00	18.00
Chang	2.00	1.00	19.00
Aniseed Syrup	3.00	2.00	10.00
Chef Anton's Cajun Seasoning	4.00	2.00	22.00
Chef Anton's Gumbo Mix	5.00	2.00	21.35
Grandma's Boysenberry Spread	6.00	2.00	25.00
Uncle Bob's Organic Dried Pears	7.00	7.00	30.00
Northwoods Cranberry Sauce	8.00	2.00	40.00
Mishi Kobe Niku	9.00	6.00	97.00
Ikura	10.00	8.00	31.00
Queso Cabrales	11.00	4.00	21.00
Queso Manchego La Pastora	12.00	4.00	38.00

It should be noted that there were two ways available to determine the format mask:
➤ Use local settings. The text is formatted according to the current settings of the operating system.

❯ Each parameter is defined by the format mask manually.

Sometimes there were some disadvantages in both cases. For example, when using local settings to change the format parameters you should edit formats of the operating system. In the second case, when it is needed to change one parameter you should adjust others as well. Considering disadvantages of these methods, there is a third way to determine the format. Using the local settings you can change any parameter format. To do this, set the flag next to the parameter and set its value.
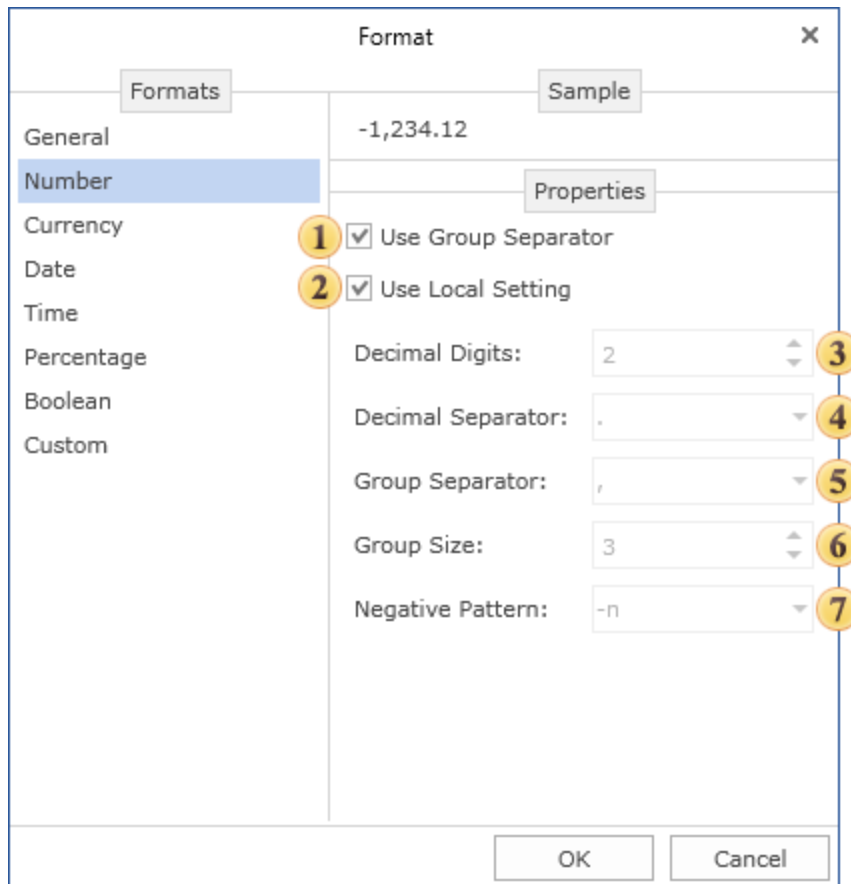
❶ Group separator
When the Group Separator is used then number will be separated into number positions.
❷ Local setting
When using the Local settings, numerical values are formatted according to the current OS installations.
❸ Decimal digits
Number of decimal digits, which are used to format numerical values.
❹ 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.

Thus, for columns ProductID, CategoryID we change only the number of digits in the fractional part.

ProductName	ProductID	CategoryID	UnitPrice
Chai	1.0	1.0	18.00
Chang	2.0	1.0	19.00
Aniseed Syrup	3.0	2.0	10.00
Chef Anton's Cajun Seasoning	4.0	2.0	22.00
Chef Anton's Gumbo Mix	5.0	2.0	21.35
Grandma's Boysenberry Spread	6.0	2.0	25.00
Uncle Bob's Organic Dried Pears	7.0	7.0	30.00
Northwoods Cranberry Sauce	8.0	2.0	40.00
Mishi Kobe Niku	9.0	6.0	97.00
Ikura	10.0	8.0	31.00
Queso Cabrales	11.0	4.0	21.00
Queso Manchego La Pastora	12.0	4.0	38.00

⚠ **Notice:** In order to display currency values you should use the Currency format. In the example above, for the **UnitPrice** column you should set the Currency format.

3.5.3   Currency Formatting

In order to display numeric values as a currency you should use the Currency format. This format is designed specifically to output monetary values.

ProductName	ProductID	CategoryID	UnitPrice
Chai	1.0	1.0	18.00
Chang	2.0	1.0	19.00
Aniseed Syrup	3.0	2.0	10.00
Chef Anton's Cajun Seasoning	4.0	2.0	22.00
Chef Anton's Gumbo Mix	5.0	2.0	21.35
Grandma's Boysenberry Spread	6.0	2.0	25.00
Uncle Bob's Organic Dried Pears	7.0	7.0	30.00
Northwoods Cranberry Sauce	8.0	2.0	40.00
Mishi Kobe Niku	9.0	6.0	97.00
Ikura	10.0	8.0	31.00
Queso Cabrales	11.0	4.0	21.00
Queso Manchego La Pastora	12.0	4.0	38.00

Set the currency format for the UnitPrice column.

Information: It is understood that when setting the currency format, the important point is the selection of the required currency. The same value can be either the US, European Union, China currency and the currency of any other country.
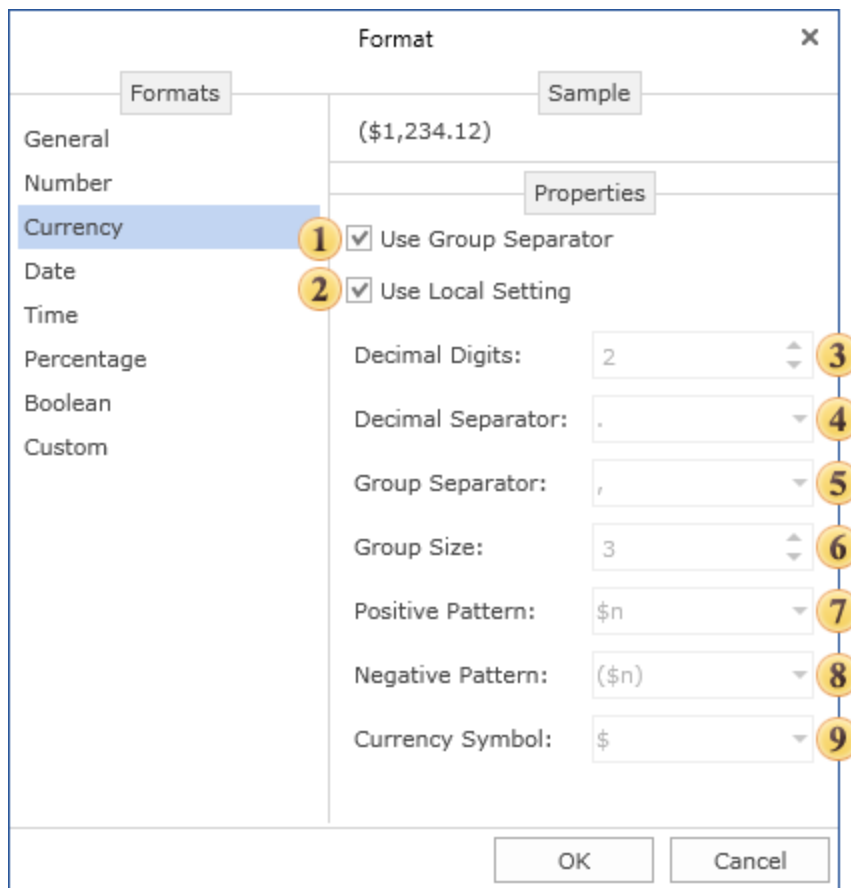
For example, the prices are in US dollars. Then, select the appropriate currency sign, and determine the parameters of the format.

ProductName	ProductID	CategoryID	UnitPrice
Chai	1.0	1.0	$ 18.00
Chang	2.0	1.0	$ 19.00
Aniseed Syrup	3.0	2.0	$ 10.00
Chef Anton's Cajun Seasoning	4.0	2.0	$ 22.00
Chef Anton's Gumbo Mix	5.0	2.0	$ 21.35
Grandma's Boysenberry Spread	6.0	2.0	$ 25.00
Uncle Bob's Organic Dried Pears	7.0	7.0	$ 30.00
Northwoods Cranberry Sauce	8.0	2.0	$ 40.00
Mishi Kobe Niku	9.0	6.0	$ 97.00
Ikura	10.0	8.0	$ 31.00
Queso Cabrales	11.0	4.0	$ 21.00
Queso Manchego La Pastora	12.0	4.0	$ 38.00

It should be noted that previously there were two ways to determine the format mask:
> Use local settings, i.e., the text is formatted according to the current settings of the operating system.
> Each parameter is defined by the format mask manually.

Sometimes there were some disadvantages in both cases. For example, when using local settings to change the format parameters you should edit formats of the operating system. In the second case, when it is needed to change one parameter you should adjust others as well. Considering disadvantages of these methods, there is a third way to determine the format. Using the local settings you can change any parameter format. To do this, set the flag next to the parameter and set its value.

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.

Let's go back to the example described above. Change the values only for the Positive Pattern and Currency Symbol parameters. Other parameters will be determined by local settings.

ProductName	ProductID	CategoryID	UnitPrice
Chai	1.0	1.0	€18.00
Chang	2.0	1.0	€19.00
Aniseed Syrup	3.0	2.0	€10.00
Chef Anton's Cajun Seasoning	4.0	2.0	€22.00
Chef Anton's Gumbo Mix	5.0	2.0	€21.35
Grandma's Boysenberry Spread	6.0	2.0	€25.00
Uncle Bob's Organic Dried Pears	7.0	7.0	€30.00
Northwoods Cranberry Sauce	8.0	2.0	€40.00
Mishi Kobe Niku	9.0	6.0	€97.00
Ikura	10.0	8.0	€31.00
Queso Cabrales	11.0	4.0	€21.00
Queso Manchego La Pastora	12.0	4.0	€38.00

3.5.4   Date Formatting

If the report contains text components which output date in the rendered report then the Date formatting can be applied to this text component. The date format is selected from a set of specified formats - short format, long format, etc. In the applied format, except the ones with an asterisk (*), the order of elements does not change. For example, the report contains the list of products and OrderDate, RequiredDate, ShippedDate.

ProductName	OrderDate	RequiredDate	ShippedDate
Queso Cabrales	8/3/2008 11:00:00 PM	8/31/2008 11:00:00 PM	8/15/2008 11:00:00 PM
Singaporean Hokkien Fried Mee	8/3/2008 11:00:00 PM	8/31/2008 11:00:00 PM	8/15/2008 11:00:00 PM
Mozzarella di Giovanni	8/3/2008 11:00:00 PM	8/31/2008 11:00:00 PM	8/15/2008 11:00:00 PM
Tofu	8/4/2008 11:00:00 PM	9/15/2008 11:00:00 PM	8/9/2008 11:00:00 PM
Manjimup Dried Apples	8/4/2008 11:00:00 PM	9/15/2008 11:00:00 PM	8/9/2008 11:00:00 PM
Jack's New England Clam Chowder	8/7/2008 11:00:00 PM	9/4/2008 11:00:00 PM	8/11/2008 11:00:00 PM
Manjimup Dried Apples	8/7/2008 11:00:00 PM	9/4/2008 11:00:00 PM	8/11/2008 11:00:00 PM
Louisiana Fiery Hot Pepper Sauce	8/7/2008 11:00:00 PM	9/4/2008 11:00:00 PM	8/11/2008 11:00:00 PM
Gustaf's Knäckebröd	8/7/2008 11:00:00 PM	9/4/2008 11:00:00 PM	8/14/2008 11:00:00 PM
Ravioli Angelo	8/7/2008 11:00:00 PM	9/4/2008 11:00:00 PM	8/14/2008 11:00:00 PM
Louisiana Fiery Hot Pepper Sauce	8/7/2008 11:00:00 PM	9/4/2008 11:00:00 PM	8/14/2008 11:00:00 PM
Sir Rodney's Marmalade	8/8/2008 11:00:00 PM	9/5/2008 11:00:00 PM	8/10/2008 11:00:00 PM
Geitost	8/8/2008 11:00:00 PM	9/5/2008 11:00:00 PM	8/10/2008 11:00:00 PM

By default, it displays the date and time. Set dates for the various formats. To do this, select the text component, call the **Format** dialog, go to the **Date** tab, and select the appropriate type.
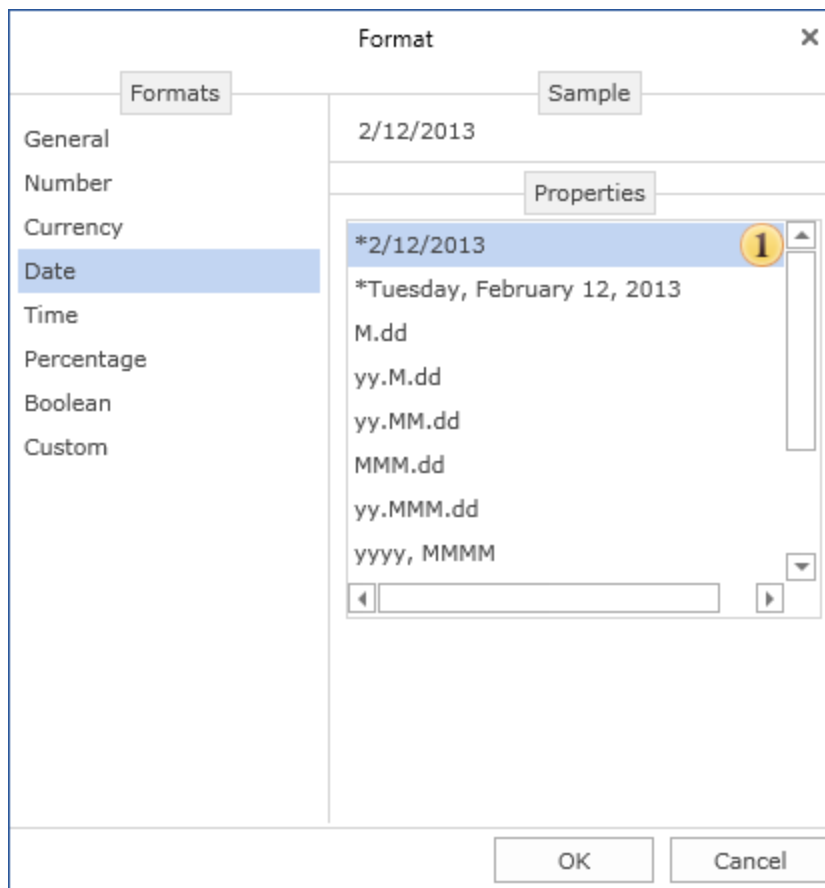
① **Date format**

The list of formatting types.

And then, the dates in the report will be displayed with certain formats.

ProductName	OrderDate	RequiredDate	ShippedDate
Queso Cabrales	8/3/2008	Sunday, August 31, 2008	15/08/2008
Singaporean Hokkien Fried Mee	8/3/2008	Sunday, August 31, 2008	15/08/2008
Mozzarella di Giovanni	8/3/2008	Sunday, August 31, 2008	15/08/2008
Tofu	8/4/2008	Monday, September 15, 2008	09/08/2008
Manjimup Dried Apples	8/4/2008	Monday, September 15, 2008	09/08/2008
Jack's New England Clam Chowder	8/7/2008	Thursday, September 4, 2008	11/08/2008
Manjimup Dried Apples	8/7/2008	Thursday, September 4, 2008	11/08/2008
Louisiana Fiery Hot Pepper Sauce	8/7/2008	Thursday, September 4, 2008	11/08/2008
Gustaf's Knäckebröd	8/7/2008	Thursday, September 4, 2008	14/08/2008
Ravioli Angelo	8/7/2008	Thursday, September 4, 2008	14/08/2008
Louisiana Fiery Hot Pepper Sauce	8/7/2008	Thursday, September 4, 2008	14/08/2008
Sir Rodney's Marmalade	8/8/2008	Friday, September 5, 2008	10/08/2008
Geitost	8/8/2008	Friday, September 5, 2008	10/08/2008

⚠ **Notice:** In addition to the formats on the **Date** tab, you can create a format on the **Custom** tab.

3.5.5   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).
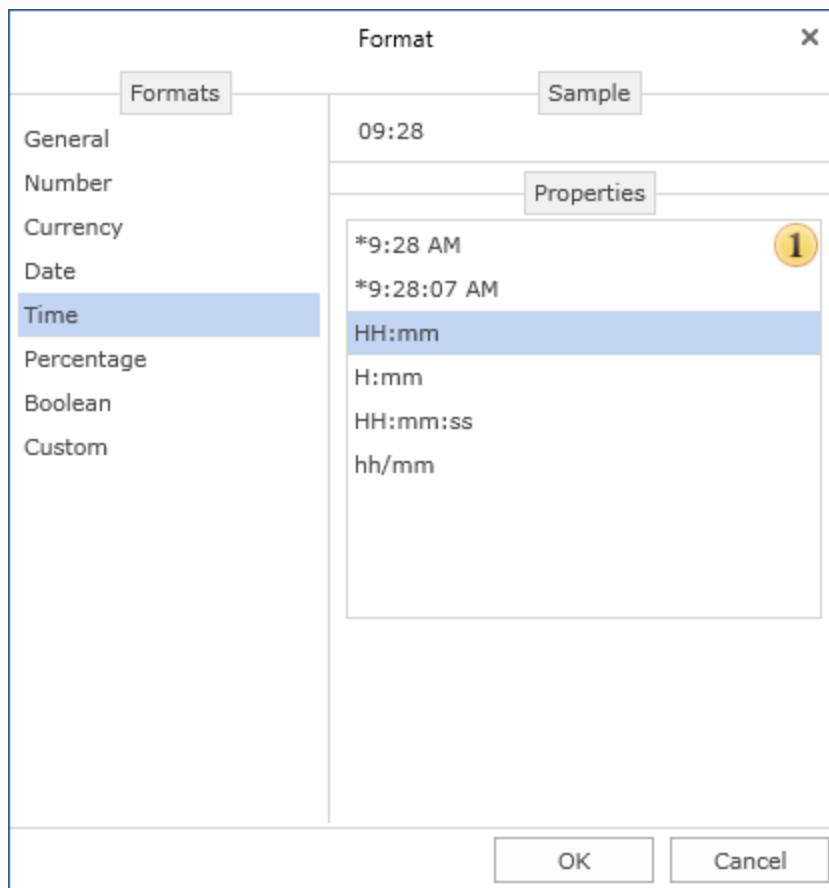
Time format

1 The list of formatting types

Below is an example of the report with the Time output and applied format to text components.

Report Rendered: 21/11/2015   10:40:39 PM   Report Created: 18/11/2015   20:33:14

ProductName	Unit Price	Units In Stock
Chai	$18.00	39.00
Chang	$19.00	17.00
Aniseed Syrup	$10.00	13.00
Chef Anton's Cajun Seasoning	$22.00	53.00
Chef Anton's Gumbo Mix	$21.35	0.00
Grandma's Boysenberry Spread	$25.00	120.00
Uncle Bob's Organic Dried Pears	$30.00	15.00
Northwoods Cranberry Sauce	$40.00	6.00
Mishi Kobe Niku	$97.00	29.00
Ikura	$31.00	31.00
Queso Cabrales	$21.00	22.00
Queso Manchego La Pastora	$38.00	86.00

3.5.6   Percentage Data Formatting

If the report uses the relative values, the current data can be output as a percentage text format. Consider the example of a report with relative values. Let's have a report that contains a list of products (standard format), their price (currency format) and the profitability index (number format).

ProductName	UnitPrice	Profitability Index
Chai	$ 18.00	0.01
Chang	$ 19.00	0.01
Aniseed Syrup	$ 10.00	0.00
Chef Anton's Cajun Seasoning	$ 22.00	0.01
Chef Anton's Gumbo Mix	$ 21.35	0.01
Grandma's Boysenberry Spread	$ 25.00	0.02
Uncle Bob's Organic Dried Pears	$ 30.00	0.02
Northwoods Cranberry Sauce	$ 40.00	0.03
Mishi Kobe Niku	$ 97.00	0.09
Ikura	$ 31.00	0.02
Queso Cabrales	$ 21.00	0.01
Queso Manchego La Pastora	$ 38.00	0.03

Now let's add a column with profitability. In this case, the profitability is the ratio as a percentage value. To do this, add the text component on the right with the reference to the Products.ProfitabilityIndex column and set the format as percent for this text component. The header of this column will be Profitability.

ProductName	UnitPrice	Profitability Index	Profitability
Chai	$ 18.00	0.01	0.84 %
Chang	$ 19.00	0.01	0.95 %
Aniseed Syrup	$ 10.00	0.00	0.00 %
Chef Anton's Cajun Seasoning	$ 22.00	0.01	1.26 %
Chef Anton's Gumbo Mix	$ 21.35	0.01	1.19 %
Grandma's Boysenberry Spread	$ 25.00	0.02	1.58 %
Uncle Bob's Organic Dried Pears	$ 30.00	0.02	2.11 %
Northwoods Cranberry Sauce	$ 40.00	0.03	3.16 %
Mishi Kobe Niku	$ 97.00	0.09	9.16 %
Ikura	$ 31.00	0.02	2.21 %
Queso Cabrales	$ 21.00	0.01	1.16 %
Queso Manchego La Pastora	$ 38.00	0.03	2.95 %

It should be noted that previously there were two ways to determine the format mask:
➤ Use local settings, i.e., the text is formatted according to the current settings of the operating system.
➤ Each parameter is defined by the format mask manually.

Sometimes there were some disadvantages in both cases. For example, when using local settings to change the format parameters you should edit formats of the operating system. In the second case, when it is needed to change one parameter you should adjust others as well. Considering disadvantages of these methods, there is a third way to determine the format. Using the local settings you can change any parameter format. To do this, set the flag next to the parameter and set its value.

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.

The symbol will used as a percent sign.

3.5.7   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**.

3.5.8   Custom Formatting

If, for some reason there are no predefined formats appropriate for you, then you can customize the format according to your needs. For example you have a report with a list of products, Order Date, Shipped Date, and the price of the product. Let's apply to

them predefined date formats and local settings for the price.

Product Name	Order Date	Shipped Date	Unit Price
Queso Cabrales	03/08/2008	15/08/2008	$14.00
Singaporean Hokkien Fried Mee	03/08/2008	15/08/2008	$9.80
Mozzarella di Giovanni	03/08/2008	15/08/2008	$34.80
Tofu	04/08/2008	09/08/2008	$18.60
Manjimup Dried Apples	04/08/2008	09/08/2008	$42.40
Jack's New England Clam Chowder	07/08/2008	11/08/2008	$7.70
Manjimup Dried Apples	07/08/2008	11/08/2008	$42.40
Louisiana Fiery Hot Pepper Sauce	07/08/2008	11/08/2008	$16.80
Gustaf's Knäckebröd	07/08/2008	14/08/2008	$16.80
Ravioli Angelo	07/08/2008	14/08/2008	$15.60
Louisiana Fiery Hot Pepper Sauce	07/08/2008	14/08/2008	$16.80
Sir Rodney's Marmalade	08/08/2008	10/08/2008	$64.80

Now let's set the format mask for each text component. To do this, select the text component, call the **Format** dialog, go to the Custom tab and create a mask.
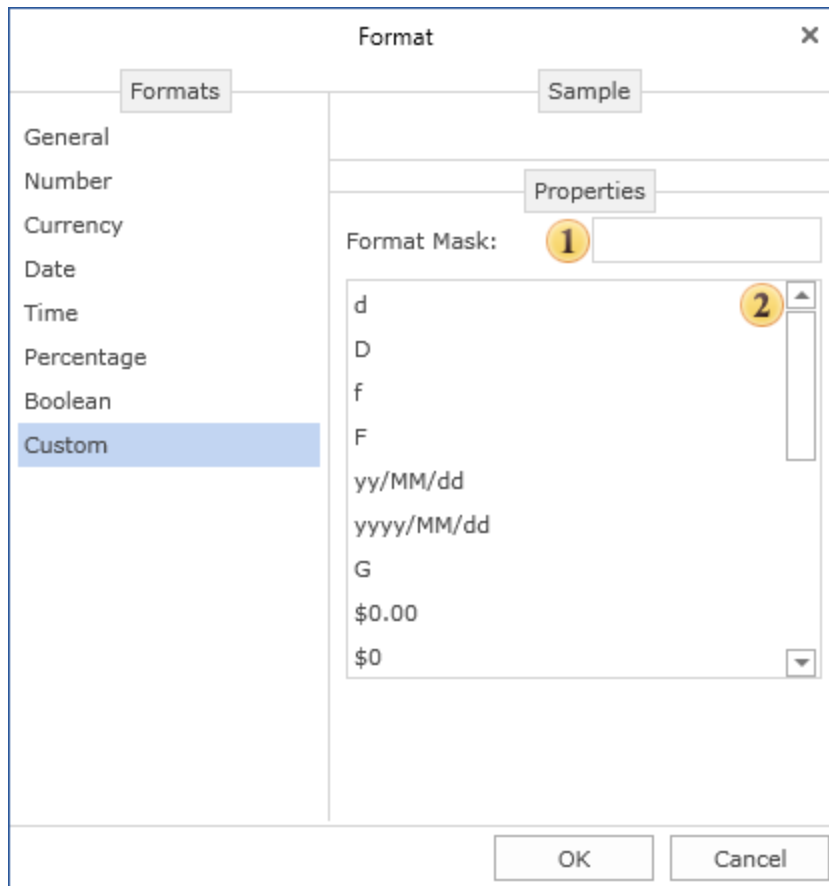
① Mask

A string or an expression that set formatting mask.

② Predefined values

The list of predefined values to format a string.

For the Order Date the mask has the form **yyyy-MM-dd**, Shipped Date - **MM-dd-yyyy**. For the price of a product the mask is **0.00 dollars of USA**. The data in the rendered report will be formatted as in the picture below.

Product Name	Order Date	Shipped Date	Unit Price
Queso Cabrales	2008-08-03	08-15-2008	14.00 dollars of USA
Singaporean Hokkien Fried Mee	2008-08-03	08-15-2008	9.80 dollars of USA
Mozzarella di Giovanni	2008-08-03	08-15-2008	34.80 dollars of USA
Tofu	2008-08-04	08-09-2008	18.60 dollars of USA
Manjimup Dried Apples	2008-08-04	08-09-2008	42.40 dollars of USA
Jack's New England Clam Chowder	2008-08-07	08-11-2008	7.70 dollars of USA
Manjimup Dried Apples	2008-08-07	08-11-2008	42.40 dollars of USA
Louisiana Fiery Hot Pepper Sauce	2008-08-07	08-11-2008	16.80 dollars of USA
Gustaf's Knäckebröd	2008-08-07	08-14-2008	16.80 dollars of USA
Ravioli Angelo	2008-08-07	08-14-2008	15.60 dollars of USA
Louisiana Fiery Hot Pepper Sauce	2008-08-07	08-14-2008	16.80 dollars of USA
Sir Rodney's Marmalade	2008-08-08	08-10-2008	64.80 dollars of USA

Thus, you can create masks of different formats.

3.5.9   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**.

3.6　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 div span. 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 bold

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 simple <i>expression {1+2}</i>

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.

3.6.1   HTML 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:

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 attributes have to be used. The default value is set within the attributes of the text component, so if the 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 attributes.

Example:
If you enter the following expression:

Test Test Test

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

Test Test Test

3.6.1.1 Color Attribute

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

By Name
You can define the color 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** attribute. For example:

<font color="red" ...>

<font color="black" ...>

<font color="white" ...>

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">

3.6.1.2 Face Attribute

The face attribute defines the name of the font of the text within the font element. To use this attribute 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** attribute:

<font face="Arial" ...>

Alternative Attributes
Instead of the "**face**" attribute the attributes "**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** attribute. For example:

<font-face="Arial">

3.6.1.3   Size Attribute

The size attribute 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 attribute is ignored.

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

 <font-size="14">

3.6.2   HTML Tags to Change Font Style

The report generator supports nine tags for changing a font style: ****, **<i>**, **<u>**, **<s>**, **<sup>, <sub>, <strong>, <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 text 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>text</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:

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

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

3.6.2.1   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 Test 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.

3.6.2.2   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.

3.6.2.3 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 Emphasis 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.

3.6.2.4 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 <u>Test</u> Test

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

3.6.2.5 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.

3.6.2.6 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 ^{Test} 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.

3.6.2.7   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 _{Test} Test

The result of output:

Test Test Test

3.6.2.8   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 Text Text

The result of output:

Text **Text** Text

3.6.2.9   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.

**3.6.2.10   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
 works.

The result of output:

How it
works.

3.6.2.11   HTML <ol> Tag

The <ol> tag inserts an ordered list, which is a block level element consisting of a sequence of numbered items, usually displayed with a number on the left margin.

<p>How it works!</p>

```
<ol>
<li>How</li>
<li>it</li>
<li>works.</li>
</ol>
```

The result of output:

How it works!

1. How
2. it
3. works.

3.6.2.12 HTML <ul> Tag

The <ul> tag inserts an unordered list, which is a block level element consisting of a sequence of items, usually displayed with a bullet on the left margin.

```
<p>How it works!</p>
<ul>
<li>How</li>
<li>it</li>
<li>works.</li>
</ul>
```

The result of output:

How it works!

- How
- it
- works.

3.6.3   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

3.6.4   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
<text-align="right">Test</text-align>
Test

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

3.6.5   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
<letter-spacing="0.5">Test</letter-spacing>

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

Test
T e s t

3.6.6   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

3.6.7   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">
</line-height>Test<line-height="0.7">
</line-height>Test

The result of output:

Test

Test
Test

3.6.8 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 "**<**" 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

3.7   Rich Text

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.

! **Note**: This component does not work in the product line **Stimulsoft Reports.Fx**.

3.7.1 Rich Text Editor

The **RichText** component has a special editor. This editor can load and save the **RTF** text, change the font, size, color, paste expressions etc. With this editor you can edit the RTF text without using third-party editors. The editor is called by double-clicking on the RichText component. This editor contains the following tabs:

> **Expression**. Specify here some text. You can edit the text here using a set of special tools.
> **Data Column**. Specify the data column that contains the Rich text.
> **File**. Load a file that contains the Rich text.
> **Url**. Specify the URL the source with the Rich text.

The picture below shows the **Rich** text editor with **Expression** tab open:
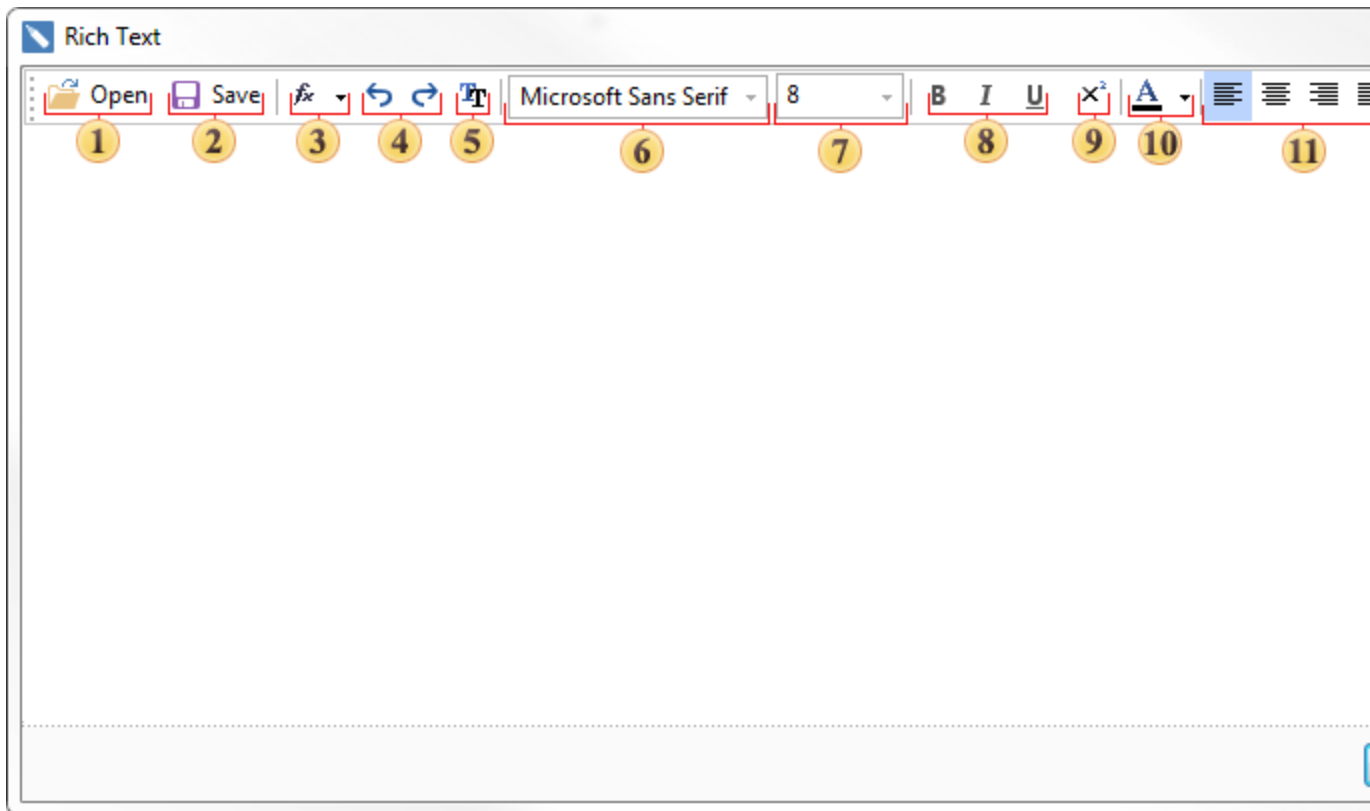
1 The **Open** button. Opens the dialog to load the saved *.rtf file.

2 The **Save** button. Opens the dialog to save the text as *.rtf.

3 The **Insert** button. Shows the data dictionary tree.

4 The **Undo** and **Redo** buttons.

5 The **Font** button Calls the window to setup the font.

6 Font face field. This field displays the current type of the font name. Also, this field contains a drop-down list of values that provides the ability to change the font type without calling the font settings dialog box.

7 Font size field. This field displays the font size value. Also, this field contains a drop-down list of values that provides the ability to change the font size without calling the font settings window.

8 **Bold**, **Italic**, **Underline** buttons.

9 The **Superscript** button. It provides the ability to place text on top with respect to the previous one. For example, the exponents.

10 The **Color** button. Calls the menu to change the text color.

11 Alignment of text: **Align Left**, **Align Center**, **Align Right**, **Justify**.

12 The **Bullets** button. Enables bullets in text.

3.7.2   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 color attribute is applied to the full expression from the opening brace "{" to the closing brace "}" including those symbols. For example:

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

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

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

> 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 help of the DataColumn.

> The property **Full Convert Expression** provides the ability to handle expressions in the RTF component in different ways. If this property is set to **false**, then the expression will be processed quickly, simply and consistently. If this property is set to **true**, then processing of expressions in the RTF component will be more thorough. This method slows report rendering, but allows converting expressions more thoroughly. Especially if the expression uses characters other than the numbers and Latin alphabet.

3.7.3 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.
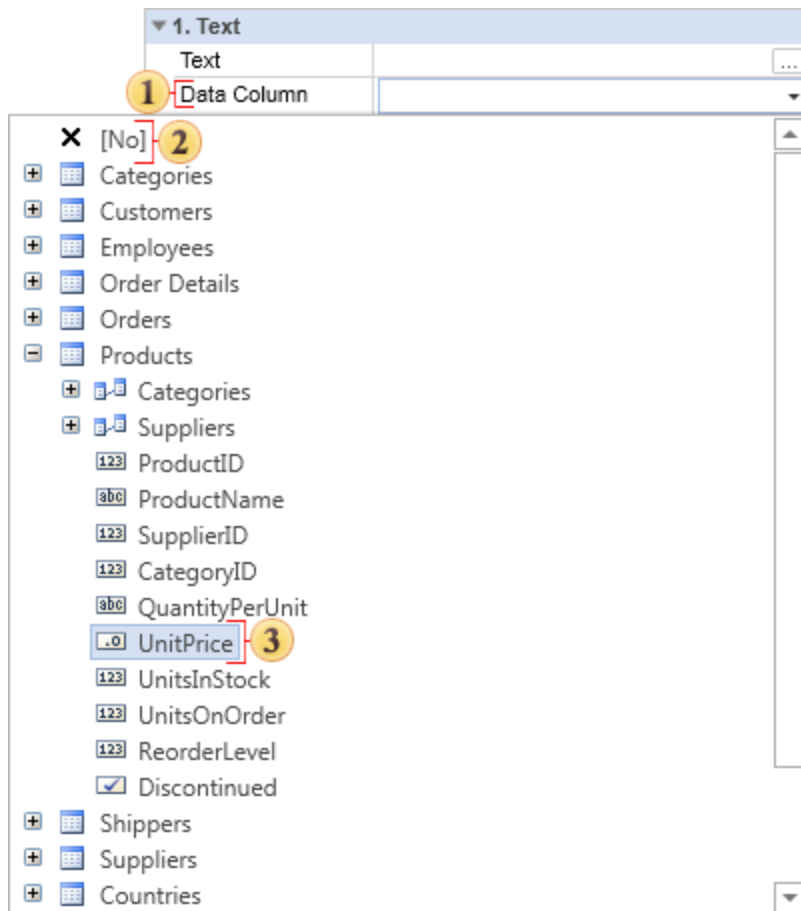
1 The **DataColumn** property. This property is used to indicate from which data field the RTF text should be loaded. Click the button beside to select the relevant column.
2 **Null node.** Selecting this node means that the RTF text is not loaded from a data field.
3 **Selected** field. The Data field from which the RTF text will be loaded.

3.8 Graphic Information Output

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	

3.8.1 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:

      ✓ Load an image from a file;
      ✓ Load an image from the report code;
      ✓ Load an image from the data field.
      ✓ Load an image from the URL.

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.

3.8.2 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**.
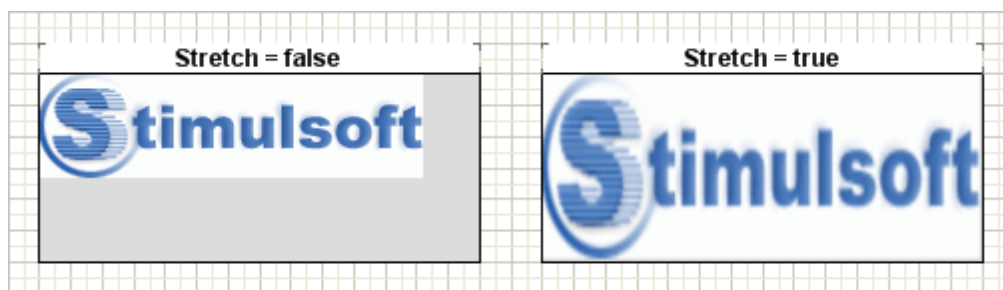
After setting the **Stretch** property to **true** the image will fill all free space of the component. When stretching, the image its proportions can be broken. To stretch an image and keep its proportions it is necessary to set the **AspectRatio** property to **true**.

And the **Image** component will always keep proportions of images.
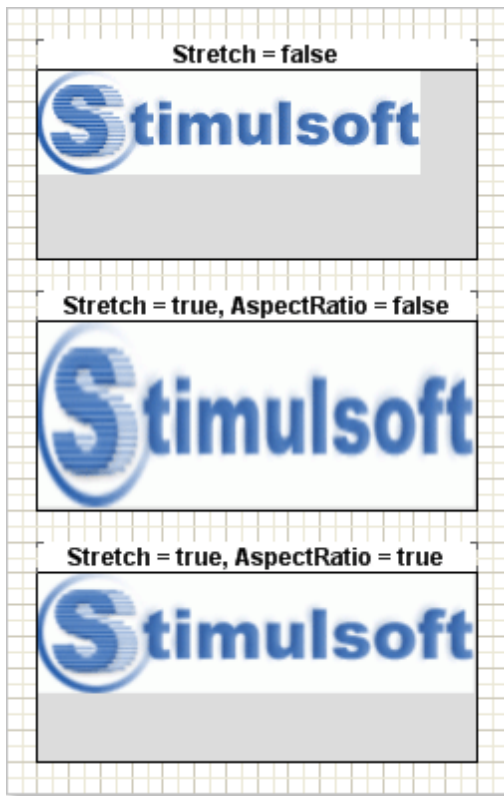
❗ **Important:** The **AspectRatio** property is in process only when the image stretching is enabled.

3.8.3 Resources of Images

Sometimes you need to add some image to the report. It could be images of goods, personnel, statistics, etc. Images can be added from different sources. To insert images, photos in a report in the Report Designer, you should use the **Image** component. The Image component should be put in the report where you want the image be placed (report page, data band, header band, footer band, etc.). When you add this component in the report, the dialog will be called:
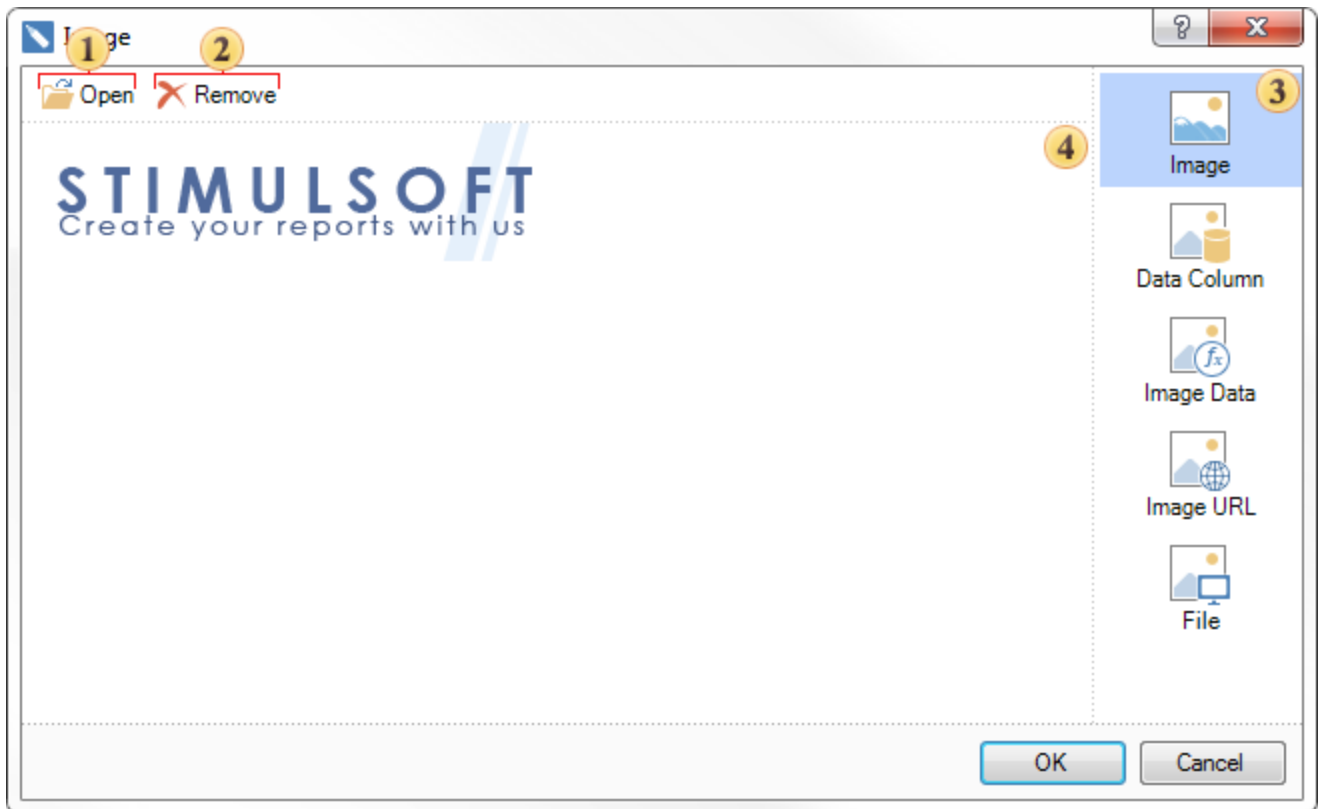
① Opens the dialog to select an image for the report.
② Remove the selected image.
③ The list of sources from which to load an image.
④ The window where the uploaded image is shown.

As seen from the above picture, the images can be downloaded from various sources. Consider them in more detail

> **Image**

Click the **Open** button and select the required image. This is a procedure of loading the image from the local source.

> **Data Column**

An image can be placed in the data table, for example, as a separate data column. Select the data column from which to extract the image.

> **Image Data**

Load an image from the expression. In this case specify some expression for this.

> **Image URL**

You can upload a picture from a **URL**. When rendering a report, the image will be retrieved from the specified URL. Consequently, in this type of source, you must specify the URL of the image.

> **File**

In addition to loading images directly, it can be retrieved from a file that is downloaded from a local source. With this type of source, press the […] button and select the file.

3.9　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:

❗ **Note**: The Can Grow property does not work in the product line **Stimulsoft Reports.Fx**.

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.
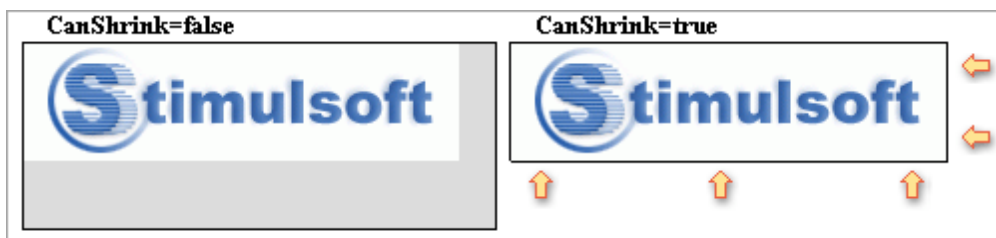
Using this property will help you to prevent wasted space on report pages
The report generator allows you to set both **CanGrow** and **CanShrink** properties. If

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.

3.9.1 Automatically Resizing Text Component

The automatic resizing of text components 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:
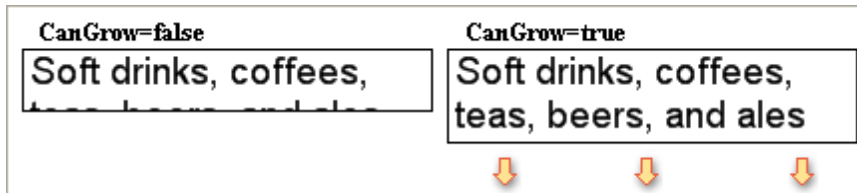
The **CanShrink** property works in the opposite way, so if it is set to true and there is more space than is needed for the text the report generator will automatically decrease the height of the text component.
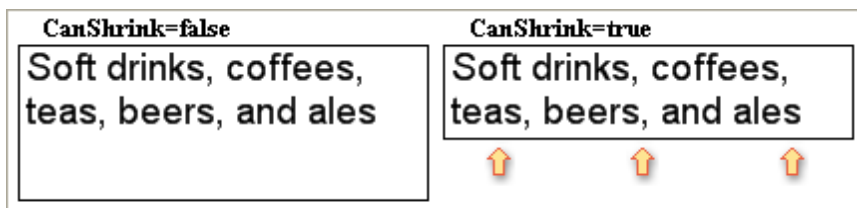
As with other components it is possible to set both properties to true. In this case, the height will automatically increase or decrease depending on the size of a text.

WordWrap 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.

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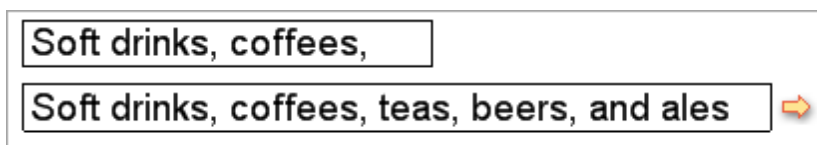
If the **AutoWidth** property is set to false, then the height of the text depends on settings of the **CanGrow** and **CanShrink** properties. If the **AutoWidth** property is set to true, then the width will be automatically changed.

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

3.9.2   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:
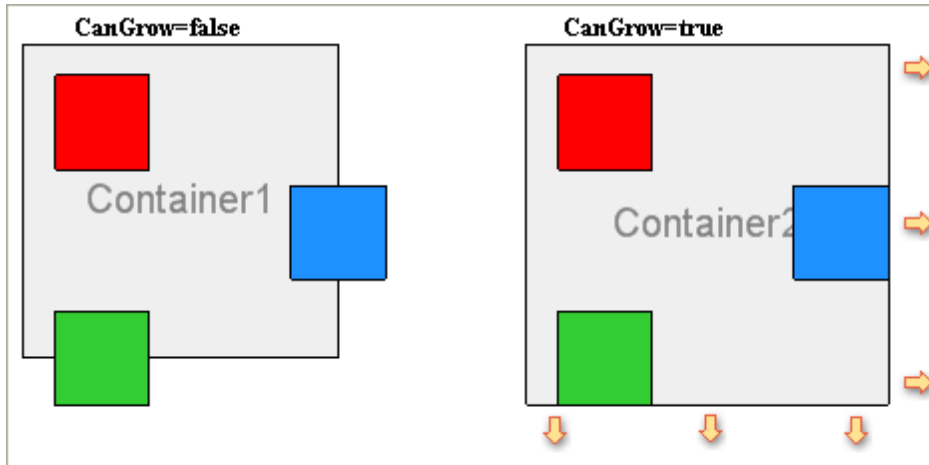
If the **CanShrink** property is set to true and the bounds of the combination of all the components contained within it are less than the bounds of the panels the panel size will automatically reduce to match the overall size of all components.
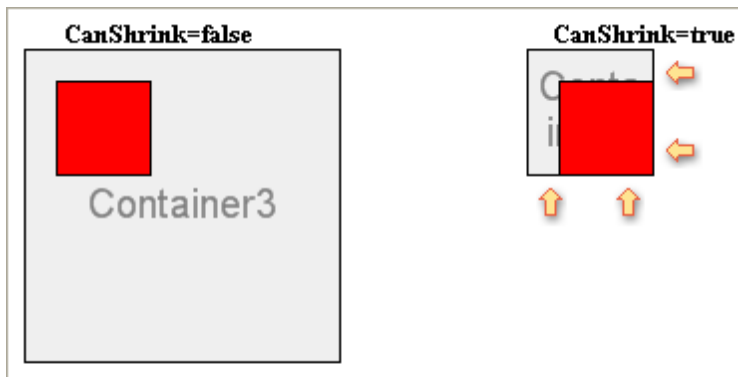
3.9.3   Automatically Resizing Bands

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:
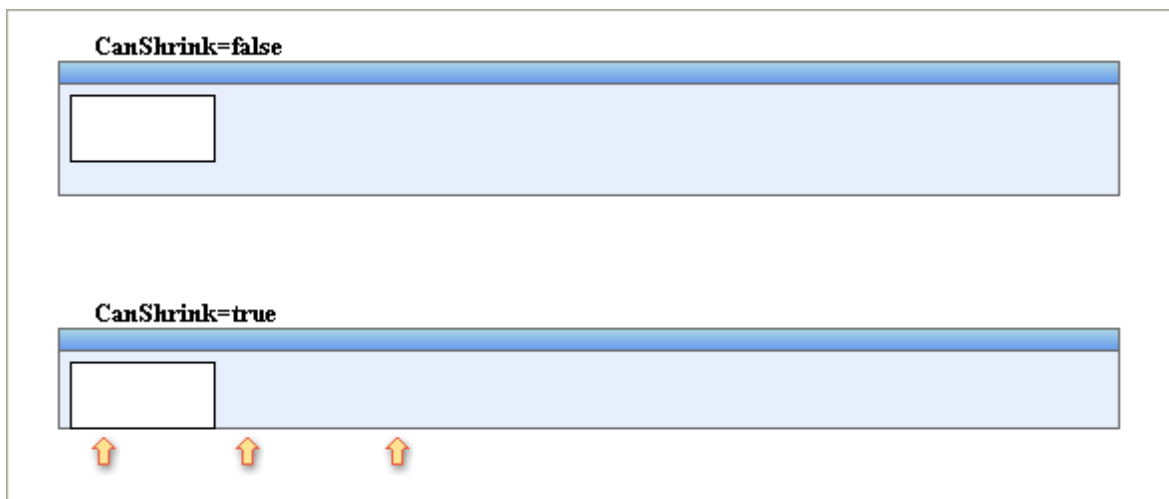
3.9.4 Binding Bottom Border of Component

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:

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.

3.9.5 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:

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.

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:

Decreasing Size = false   Decreasing Size = true
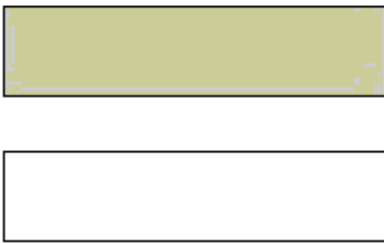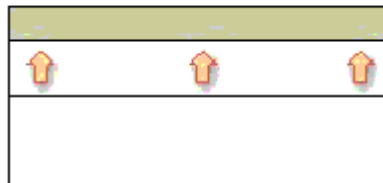
By default, this flag is set to false.

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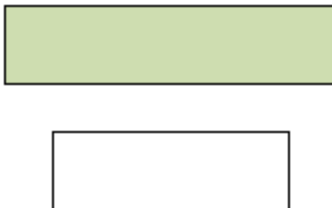
Or:
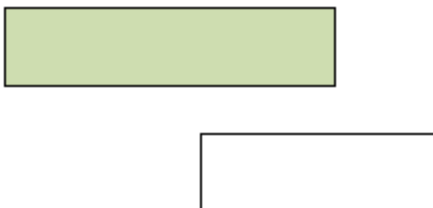
If this flag is disabled, the location of the left border of this component is ignored. For example:

By default this flag is disabled.

3.10 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 barcode.

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:
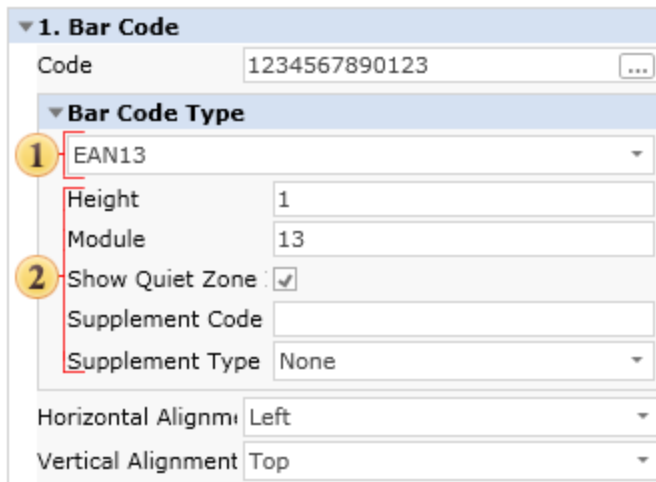
 {Items.Code}

❗ **Important:** When using the expression in the Code property in the design mode the expression will be displayed. When viewing the report, it will be replaced by the value.

Using Barcode Components

When using the Barcode components it is 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 BarCodeModules.

① The barcode type.
② The barcode properties.

3.10.1 Barcode Editor

When you add the Barcode component in the report template the bar code editor is called.

> **Information:** If, in the designer settings, the **Edit After Insert** option is disabled (unchecked), then, to call the editor, you need to double-click the component.

The Barcode editor consists of two tabs:

> The **Bar Code** tab. Select the bar code you need to use in the report. For example, QR Code:

❯ The **Settings** tab. Setup the barcode. The tabs has three panels: barcode parameters, preview, barcode properties.

⚠ **Notice**: In the **web** report designer, editing the barcode goes using the parameters and properties that are located on the properties panel. They are completely identical to those described below. In the **web** report designer you should select a component, go to the properties panel and set the component settings. When you double-click the component you will call a menu in which you need to specify values for the barcode (custom value, data column, variable, etc.).

Consider the barcode parameters in detail:

① The Code field. Specifies a value that a bar code will have. For example, you can specify a custom value. For QR Code it may be some text and numeric value. Also in this field, you can specify an expression. Then, the result of this expression will be the value of the barcode.

② The **Angle** parameter. It provides an opportunity to rotate the graphical barcode information on 90, 180, and 270 degrees.

③ The **Auto Scale** parameter. It provides the ability to determine the optimal scale of the bar code, taking into account the volume of information. It should be borne in mind that the larger is the amount of information in a bar code, the more graphical elements

is in it. So, if the bar code contains large volume of data and, at the same time, you minimize the component size, the barcode reader could read it incorrectly. Therefore, the size of the component in the report should always be defined taking into account the amount of information that the barcode contains.

4 The **Fore Color** parameter. Specifies the color of the graphical elements in the barcode

5 The **Back Color** parameter. Specifies the color of a font in the barcode.

6 The **Font** parameter. Specifies a type and style of the font for the barcode.

7 The **Show Label Text** parameter. Allows showing/hiding the label text of the barcode. This is applicable not for all barcodes but only for those who have a label. The label shows a value of the barcode. For example, the picture below shows two Code128 barcodes. One with a title, the other without it.

8 The **Quiet zone** parameter. It provides the ability to display or hide a quiet zone of a barcode. The Quiet zone is an empty space on the left and right side of the barcode. It is a conditional border of the beginning and end of the barcode for barcode readers. One example of the use of the Quiet zone is the case when there are several barcodes. If the Quiet zone is disabled then the barcodes can be read wrongly. Below is an example of two barcodes with enabled and disabled Quiet zones.
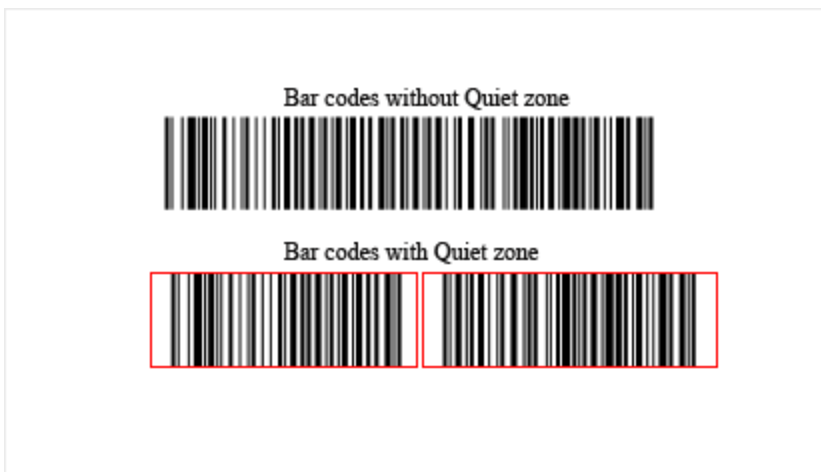
9 The preview panel.

10 The barcode property panel. Depending on the type of the barcode, the number of

properties and their name may vary.

3.10.2 Barcode 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 barcode 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 barcode.

Graphics Density
The graphics density of the barcode is connected with the barcode 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 barcode. They are required for the scanner to identify the barcode measurements.

3.10.3 Linear Barcodes

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

3.10.3.1 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.

3.10.3.1.1 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** barcode (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 barcodes. 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.

3.10.3.1.2 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.

3.10.3.1.3 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.

3.10.3.1.4 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 barcode 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 barcode 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.

3.10.3.1.5 EAN-8

The **EAN-8** barcode was developed for use on small packages. It is used instead of the EAN-13 barcode 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** barcode is in the same as the structure of the **EAN-13** barcode. The check digit is calculated automatically irrespective of input data.

The barcode contains the following elements:
    ✓ 3 digits - a prefix of the national organization.
    ✓ 4 digits - product code.
    ✓ 1 digit - check digit.

This barcode 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** barcode 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.

3.10.3.1.6 Add-On Symbols

Add-on Symbols (barcodes) can be used in some applications together with the EAN-13, UPC-A, and UPC-E barcodes. 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" barcode with the "02" Add-On Symbols

The "EAN-13" barcode with the "00321" Add-on Symbols

3.10.3.1.7 EAN-128

The **EAN-128** barcode 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 sub-codes, 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** barcode is the same as for the **Code128** barcode. Elements of the barcode 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** barcode and the **Code128** barcode 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.

3.10.3.1.8 ITF-14

The **ITF-14** barcode was developed to encode a Global Trade Item Number. The ITF barcode 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 barcode contains the following elements:
- ✓ 1 digit - logic.
- ✓ 3 digits - Global Trade prefix.
- ✓ 6 digits - Producer code.
- ✓ 3 digits - Product code.
- ✓ 1 digit - Check digit.
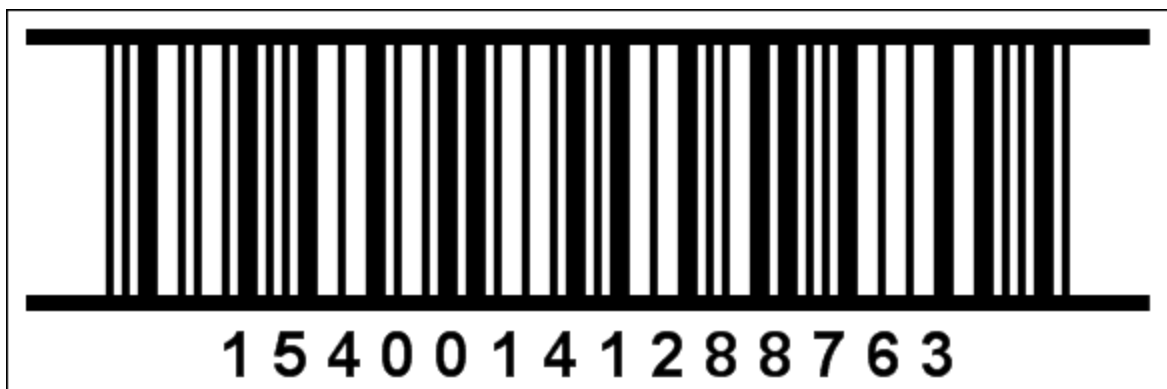
1 5 4 0 0 1 4 1 2 8 8 7 6 3

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.

3.10.3.1.9 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.

3.10.3.1.10 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 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.

3.10.3.1.12 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.

3.10.3.2 Other Barcodes

3.10.3.2.1 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.

3.10.3.2.2 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 barcode 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 barcode 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 barcode has 1 module width and a space has 2 modules. If the element is the binary 1 the bar has 2 module width and a space has 1 module. So, each character has 12 modules length. Therefore, this barcode has very low data density.

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

3.10.3.2.3 Msi

The **Msi** barcode developed by the MSI Data Corporation. It is based on the original Plessey symbology. Sometimes the **Msi** barcode is called the **Modified Plessey**. The basic implementation of the **Msi** barcode 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 barcode consist of 4 elements. If the element is the binary 0 then the barcode has the 1 module width and a space has 2 modules. If the element is the binary 1 the bar the 2 module width and a space has 1 modules. So, each character has 12 modules length. Therefore, this barcode has very low data density.

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

3.10.3.2.4 2of5

The **2of5** barcode 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 barcode 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.

3.10.3.2.5 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.
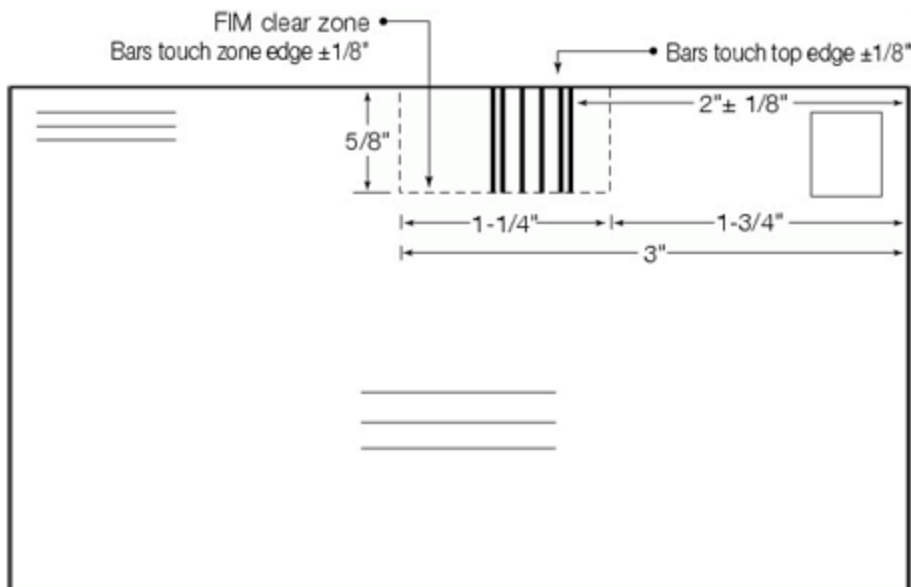
The FIM barcode on a card

The table below shows basic parameters of the **FIM** barcode.

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

The **FIM** barcode consists of nine elements. Each element can be 1 (vertical bar) or 0 (space). Four barcodes 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

3.10.3.2.6 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 barcode 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 barcode.

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

3.10.3.2.7 Postnet

The **POSTNET** (POSTal Numeric Encoding Technique) barcode 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 barcode can encode 0-9 digits. The barcode consist of short and long bars. Each symbol of data is encoded using five bars. This barcode contains only one check symbol, that is calculated by the modulo-10 algorithm.

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

3.10.3.2.8 Australia Post 4-state

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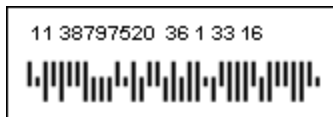
A "Australia Post 4-state" barcode. "1138797520" is a number encoded in the barcode.

3.10.3.2.9 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:
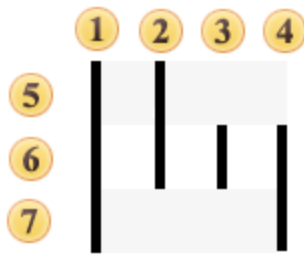
1 Full bar;
2 Ascender;
3 Tracker;
4 Descender;
5 Ascending Region;
6 Tracking Region;
7 Descending Region.

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

3.10.3.2.10 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.

1234567890123

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

3.10.3.2.11 Code11

The **Code 11** barcode 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 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.

3.10.3.2.12 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 barcode 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** barcode which also supports the ASCII set of characters. The 0-9, A-Z, "." and "-" characters are encoded the same as of

the **Code 39** barcode.

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.

3.10.3.2.13 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 barcode.

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 barcode.

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

3.10.3.2.14 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 sub-codes, 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.

3.10.3.3 Barcode Comparison Table

The table below shows the list of linear barcodes 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-	+	+	+	ASCII 32 to

			103				127	
Code128c	var	1	modulo-103	+				

Explanation:

✔ "Length" - is the data length, it is the number of characters, which can the barcode 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 barcode can encode characters of this range.

✔ "other symbols" - other symbols which the barcode can encode.

Barcode Sizes

Below is a comparison of barcodes of variable length, which can encode the numbers 0 to 9. All barcodes 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.
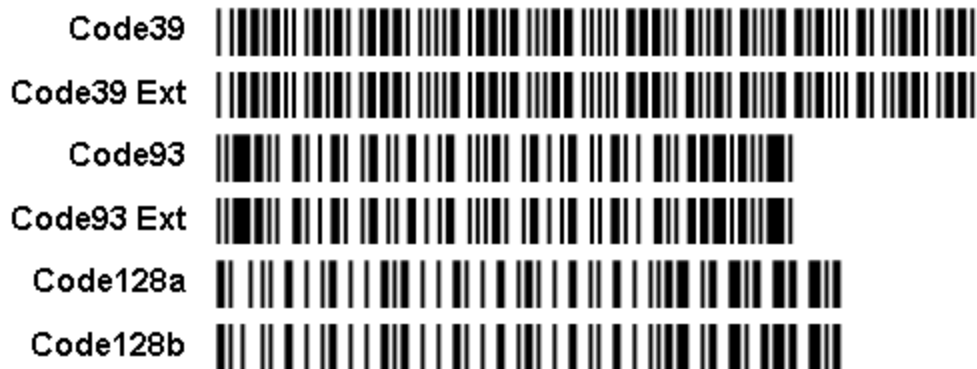
Coding English Lowercase Letters

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

3.10.4 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 barcodes are available. Stimulsoft Reports supports three most popular barcodes: **PDF417 Datamatrix**, and **QR Code**.

3.10.4.1 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 barcode: each barcode 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 barcode. The length of data depends on the encoding mode and can reach 1100 bytes, or 1800 text characters, or 2600 digits.

The barcode 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 barcode 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 barcode size will be fixed. If one of these properties is set to true, then the barcode 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 barcode is set automatically, considering the "AspectRatio" parameters (the ratio of the barcode width to the barcode 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 barcode contains levels of error corrections: even if the barcode 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.
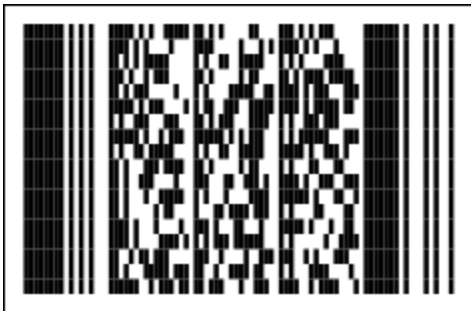
A "PDF417" barcode.

3.10.4.2 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.

For compatibility of the **DataMatrix barcode with GS1**, it is necessary to do the following:
> Set the **Process Tilde** property to **true**;
> Add the prefix **~FNC1** in the **Code** field. For example, the expression will be like this:

~FNC1{your_datasource.field_name}.

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

The barcode contains error correction codes so the barcode 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 barcode is used.

The barcode 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:

Barcode size	Length, bites	Barcode 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 barcode size can be set using the **MatrixSize** property. If this property is used to specify the specific size of the barcode, 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 barcode. If it is required to get a square barcode 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 ASCII character 128 to 255 ASCII numeric	8 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" barcode.

3.10.4.3 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:	ABCDEFGHIJKLMNOPQRSTUVW XYZ 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 barcode 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 barcode can be recovered, but fewer information can be encoded in the barcode of the same size.

A "QR Code" barcode.

3.11 Report

The report is a way of representing the data in printed form and in a user-defined form. Any report before rendering is a report template. The report template is an item of the report writer built under the rules of building a report in the designer. Elements are objects in the designer. Parameters are settings in the report designer. The picture below shows a diagram of the construction of the report.
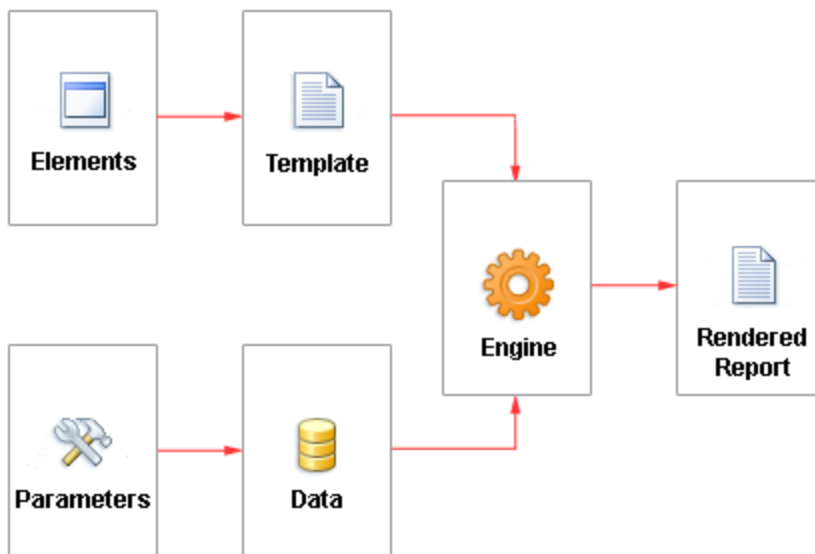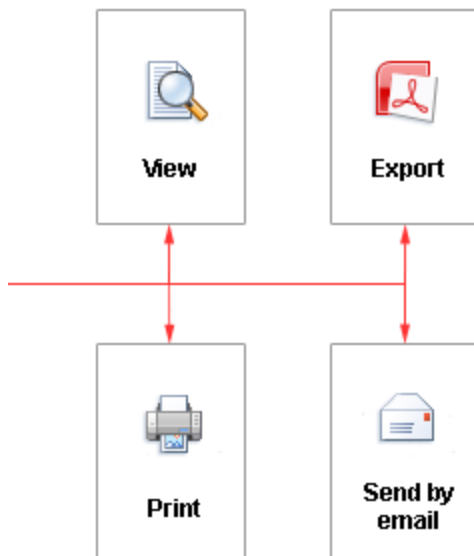
3.11.1 Report Structure

When you create a report in the designer, a report template as a page or just as a form is created. No other elements can be placed directly on the template. All other report template elements are arranged on the page or form. The picture below shows the hierarchy of the report:
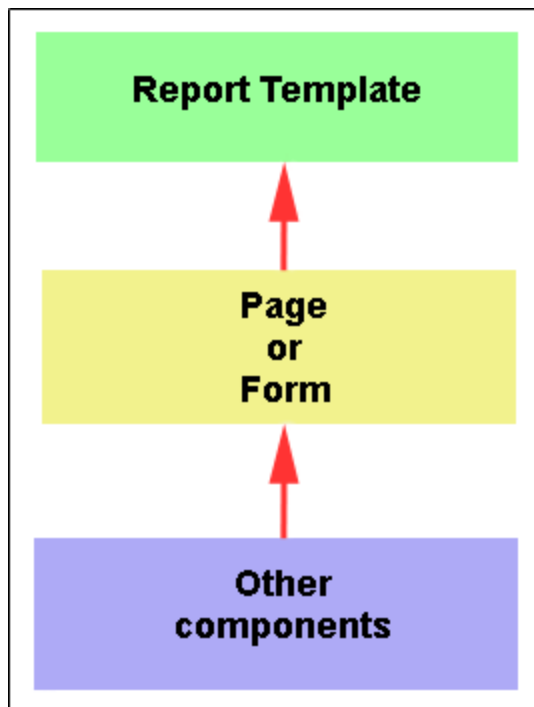
All elements of the template are divided into two categories - components and containers. The fundamental difference between the component and the container is that the container can be embedded in the container or another container and it is impossible to embed anything into the component. For example, the Text is a component. The Text component is located on a page or on another container. But it is not possible to place any container or a component in the Text component. The Form, for example, can be a container. You can place a component or container on the Form.

3.11.2 Report Rendering

Unlike most other reporting tools, Stimulsoft Reports report template is divided into pages. Each page can have their sizes and printing field. All components placed are on the report pages. When rendering, Stimulsoft Reports sequentially processes all the pages of the report.
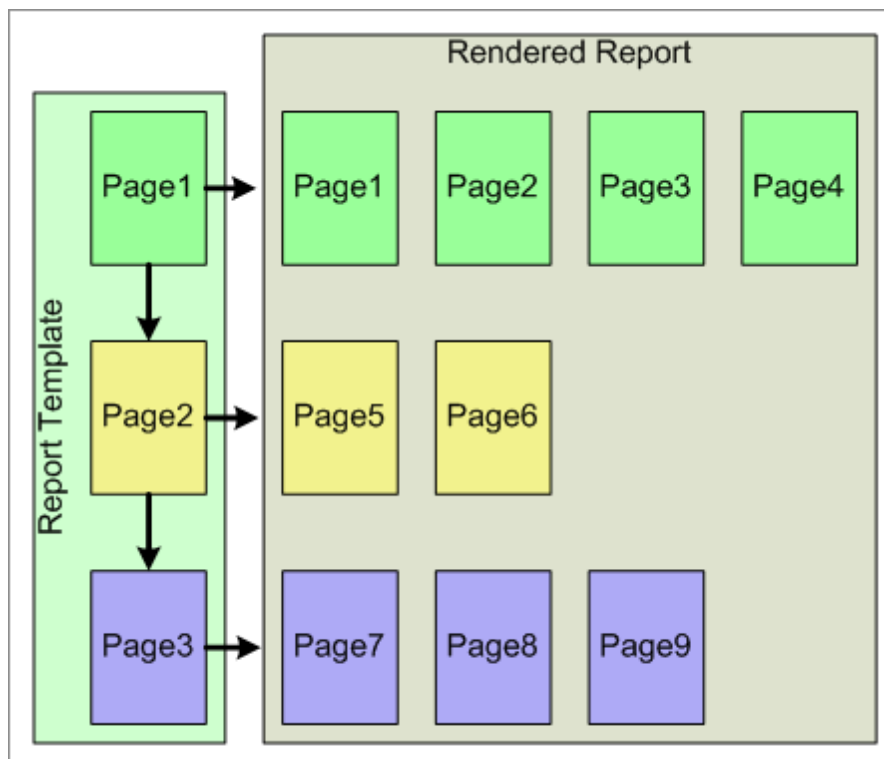
Such a structure of the report gives more flexibility in the construction of the report. You can turn off some of the pages, or vice versa, include. You can change the order of inclusion of pages in the report. You can organize the relationship between pages. When using the Sub-Report component, it is not necessary to rely on external reports, because Sub-Report in Stimulsoft Reports is also one of the pages of a report.

3.12 Pages

A page is any of the two sides of paper. The page in the reporting tool is the main component, as well as the designer workspace. The page in the report designer acts as a container. Other components of the report generator can be put on it. The page cannot be placed into any component.

3.12.1 Print On Previous Page Property

Pages of a report template are processed and printed in sequence, the first page of the template is processed first, then the second, etc. Processing order of pages can be found on the **Report Tree** tab, the higher the page is in the tree, the higher is its priority of processing. In the case with copies of pages the first page will be processed and the original page will be printed, and then copies of it. You should know that the report template page construction begins on a new page in the rendered report. For example, the first page of the report template was deployed on 14 and a half pages. In this case, the construction of the second page of the report template will begin with the 15-th page in the rendered report.

First Page of Report Template

1	Beverages
2	Condiments
3	
4	
5	
6	
7	
8	

Second Page of Report Template

Davolio	Nancy	Ms.
Fuller	Andrew	Dr.
Leverling	Janet	Ms.
Peacock	Margaret	Mrs.
Buchanan	Steven	Mr.
Suyama	Michael	Mr.
King	Robert	Mr.
Callahan	Laura	Ms.
Dodsworth	Anne	Ms.
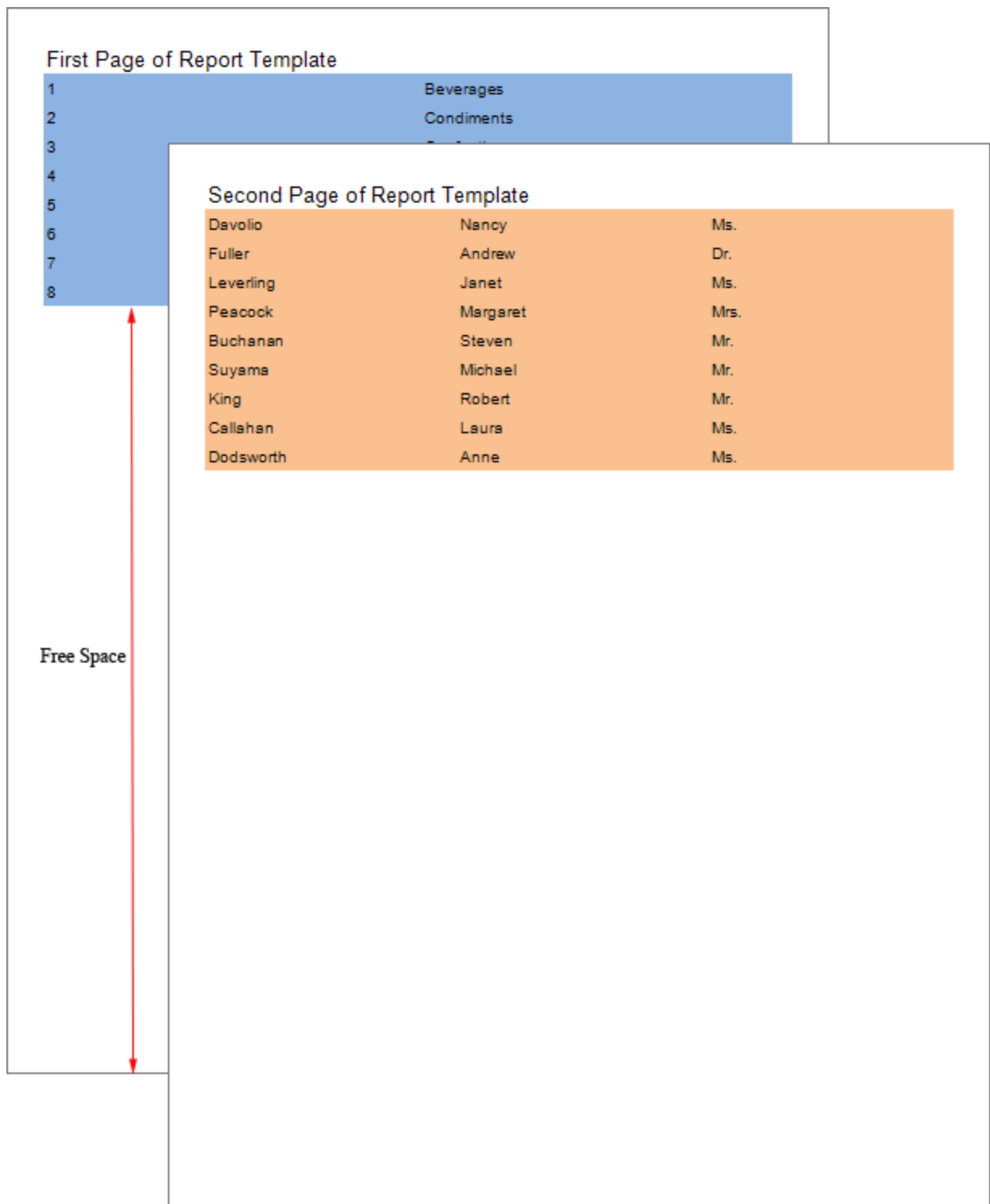
Free Space

As can be seen on the picture, after data from the first page of a template is processed, too much free space appeared in the output page. The data from the second page of the report template, was printed on the new page. In order for the data from the second page of the report template be printed immediately after the list of the first page of the

template, you should set the **Print On Previous Page** property of the second page of the template to **true**.

First Page of Report Template

1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood

Second Page of Report Template

Davolio	Nancy	Ms.
Fuller	Andrew	Dr.
Leverling	Janet	Ms.
Peacock	Margaret	Mrs.
Buchanan	Steven	Mr.
Suyama	Michael	Mr.
King	Robert	Mr.
Callahan	Laura	Ms.
Dodsworth	Anne	Ms.

By default, the **Print On Previous Page** property is set to **false**.

3.12.2 Margins

When you print the report, the situation usually occurs when the printer cannot print to the edges of the paper and a loss of information happens. In other words, the page can be maximally filled with text, but, due to the technical characteristics of the printer, part of information on the edges will not be printed. To avoid such issues you should set report margins. The margins divide the print area and the remaining empty space around the edges of the page, which are called fields.
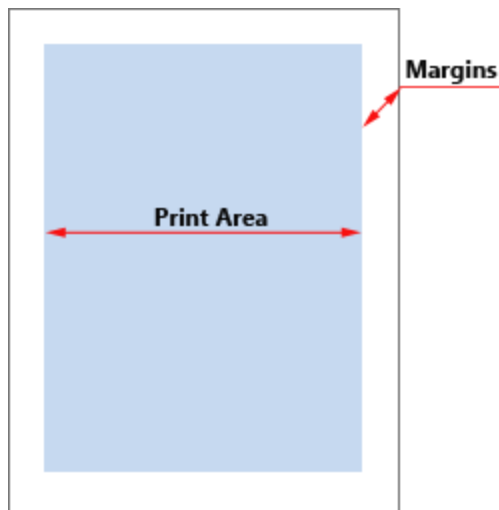
> ⓘ **Information:** Borders in the created report are not displayed. The page consists of the print area, and margins..

Generally, text and other report elements are placed in the print area. At the same time, you can place elements on margins. For example, the text component with the function to output the page number. The size of the fields can be changed by selecting one of the preset fields or set it the way you want. Preset options for fields can be selected on the **Page** tab -> **Margins** menu. Custom fields are defined by using the **Margins** report property.

> ⚠ **Notice**: Units of fields correspond to units of the report (centimeters, millimeters, inches, hundredths of inches).

Sometimes you need to create a report to staple it in a book. This requires a wider field of one of the page side.
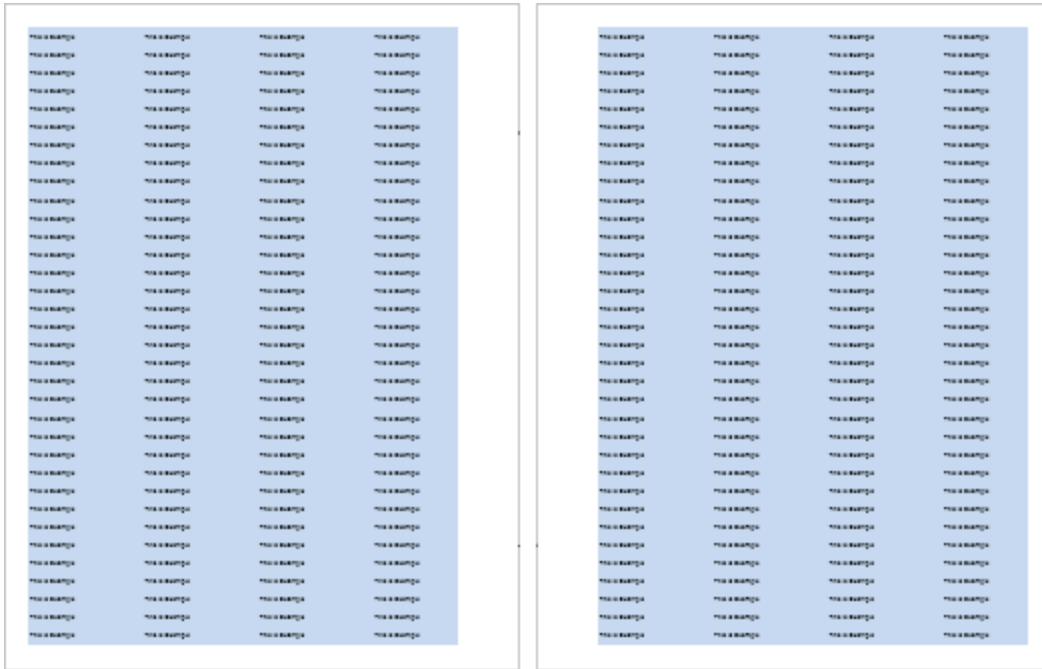
As can be seen from the picture, the right margin of the left page is wider than the left margin, while the left margin of the right page is wider than the right margin. This arrangement provides the opportunity to staple pages in a book. Location of fields in contiguous pages, as shown above, is called a mirror arrangement of margins. To activate the mirror margins you should set the **Mirror Margins** property set to true.

Information: If the margins have the same values (right margin is equal to the left), their mirrored margins will be the same.

Now consider the example of setting margins. Predefined fields can be changed on the **Page** tab with help of the **Margins** command.
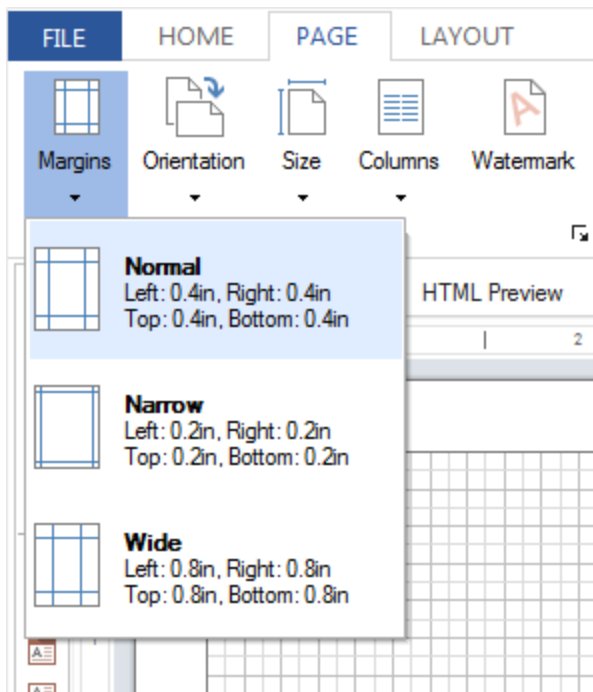
> **ⓘ Information**: In some types of interface the Page tab is missing. In this case, only is one margin size is set by default, and no other preset fields.

Setting of custom fields is carried out on the property panel. Depending on the type of the interface there can be on a single Margins property. In this case, the values of the properties will be of four numeric values from 0 or more, through ";" the separator.
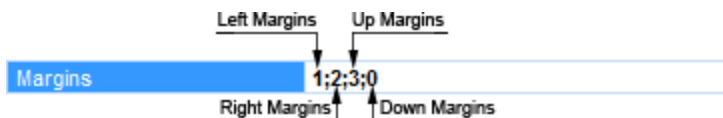
In some types of interface, the **Margin** group of properties will be located, where each margin is a separate property.

To activate the mirror fields you should be the **Mirror Margins** property to true.

> ⚠ **Notice**: The minimum size of margins depend on the printer used, the printer driver and the paper size. For information about the minimum size of the margins see the user manual of your printer.

3.13 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.
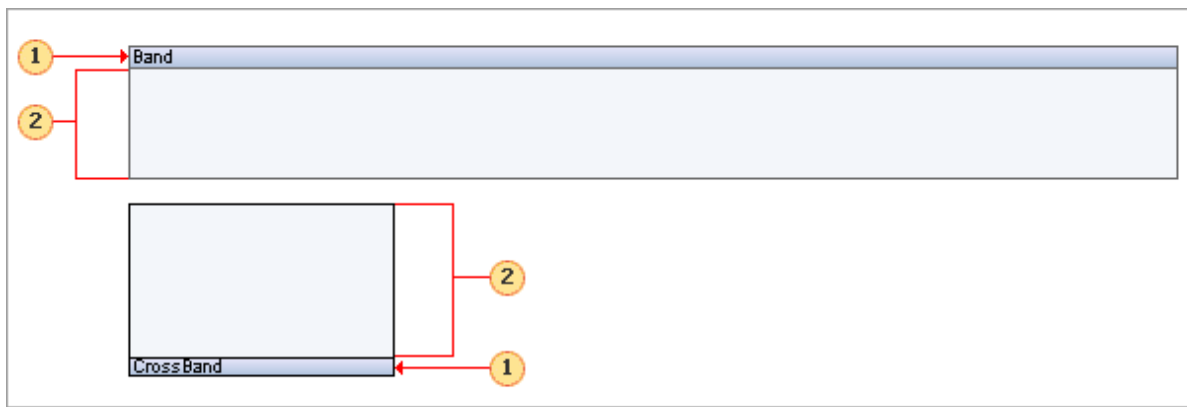
① The band header;
② The band working area.

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.

3.13.1 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 and 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.

3.13.1.1 Standard Bands

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

> ⚠ **NOTE**: This article lists the bands which are used to create reports. In order to become familiar with how they are processed when rendering the report, please read the article Rendering Order of Bands.

Icon	Band Name	Description
	Report Title	This band is printed in the beginning of a report
	Report Summary	This band is printed in the end of a report
	Page Header	This band is printed on the top of each page
	Page Footer	This band is printed on the bottom of each page
	Group Header	This band is printed in the beginning of a group
	Group Footer	This band is printed in the end of a group
	Header	This band is printed before data
	Footer	This band is printed after data

	Column Header	This band is printed before a column is output
	Column Footer	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:

PageHeaderBand1

ReportTitleBand1

GroupHeaderBand1; Condition:

ColumnHeaderBand1

HeaderBand1

DataBand1; Data Source: Not Assigned

ChildBand1

FooterBand1

ColumnFooterBand1

GroupFooterBand1

ReportSummaryBand1

PageFooterBand1

3.13.1.2 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 take the full height of its parent component so it is not recommended to put them on the page. If the band does not fit one page then it is not wrapped but a new page segment is added.

The list below shows types of cross-bands:

Icon	Name	Description
	Cross-Group Header	This band is printed in the beginning of a group
	Cross-Group Footer	This band is printed in the end of a group
	Cross-Header	This band is printed before data
	Cross-Footer	This band is printed after data
	Cross-Data	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.
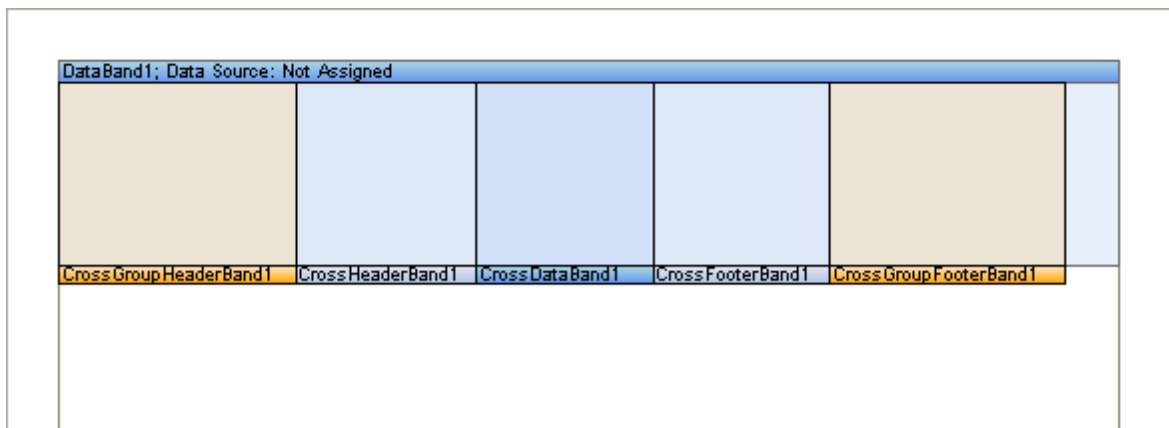
3.13.2 Rendering Order of Bands

In this article let's review the procedure of rendering the bands of the report, as well as define their relationship for the first level of nesting. Under the first level of nesting meant that the report will not have a hierarchy, only simple lists, only simple groups, etc. All bands can be divided into the following types.

❯ Page bands are **Page Header** and **Page Footer**, **Overlay**. These bands are related to the report pages, and are displayed on each page of the report;

❯ Report bands are **Report Title** and **Report Summary**. As is clear from their group name, these bands are interconnected with the report and are used to display the title and summary in reports. They are displayed only once.

❯ List bands are **Data Band**, **Hierarchical Band**. In the text below we will be referring to the Data Band, at the same time meaning that it can be used instead of the Hierarchical band;

❯ Bands associated with the **Data Band** are **Header Band**, **Footer Band**, **Group Header Band**, **Group Footer Band**, **Column Header Band**, **Column Footer Band**, **Empty Band**.

❯ The **Child Band**.

✖ The order of bands in the report template

All bands are displayed in the strict order. This is due to the fact that each band has a specific function in the report. And it is very important in which order bands are printed.

Order	Band name	Description
1	**Page Header**	On each page. Output on the first page is optional.
2	**Report Title**	Once at the beginning of a report. The Report Title band can be output before the Page Header band if the Title Before Header property of the page on which both bands are placed is set to true.
3	**Header, Column Header**	Once before data output (for the Column Header - once for every column. Output on each new

		page is optional.
4	**Group Header**	At the beginning of each group. Output on each new page is optional.
5	**Data**	Once for every row of data.
6	**Empty Band**	For each empty row on every page of the report.
7	**Group Footer**	At the end of each group.
8	**Footer,** **Column Footer**	After all data has been output (for the Column Footer - once for every column). Output on each new page is optional.
9	**Report Summary**	Once at the end of a report.
10	**Overlay**	Once on every page of the report.
11	**Page Footer**	On every page. Output on the first page is optional.

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

The **Child Band** can be placed on any Band except the **Page Header**, **Report Summary**, **Page Footer**. The picture below shows the report page template with the location of bands.

Rendering Order

When rendering a report, the report template pages are processed sequentially. The order of page processing is determined by the position of the page in the report tree. The higher the page is in the report tree, the higher is its priority (the sequence) of processing.
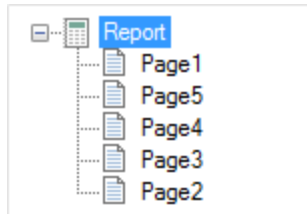
For the report tree shown in the picture above, the processing order of the pages will be as follows: the first will be processed **Page1**, then **Page5**, **Page4**, **Page3**, and finally **Page2**. Suppose that all the bands are placed on **Page1** (see an example of the report template page with the location of bands above). In this case, the bands are processed in several steps:

❯ On the first stage go the preliminary analysis of all the bands and the location of the next page bands **PageHeaderBand1**, **PageFooterBand1**, and **OverlayBand1**. These bands will always be primarily processed and added to each new page in the rendering of the report. Also, on the first page of the rendered report the ReportTitleBand1 will be added.

⚠ **Notice:** If the **Title Before Header** property is set to true, then the **ReportTitleBand1** will be processed and added to the first page first, and then **PageHeaderBand1**.
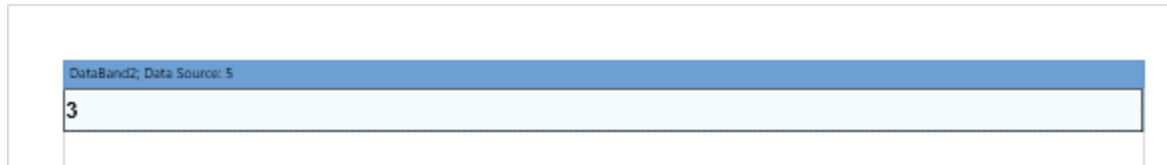
❯ In the second stage goes the analysis of other bands.

ℹ **Information:** It should be understood that other bands are in the relationship with the **Data Band** and their rendering depends on it. So and the **Data Band** is found and analyzed first, and then the other bands.

After the analysis, the report rendering will start. The ReportSummaryBand1 will be processed last.
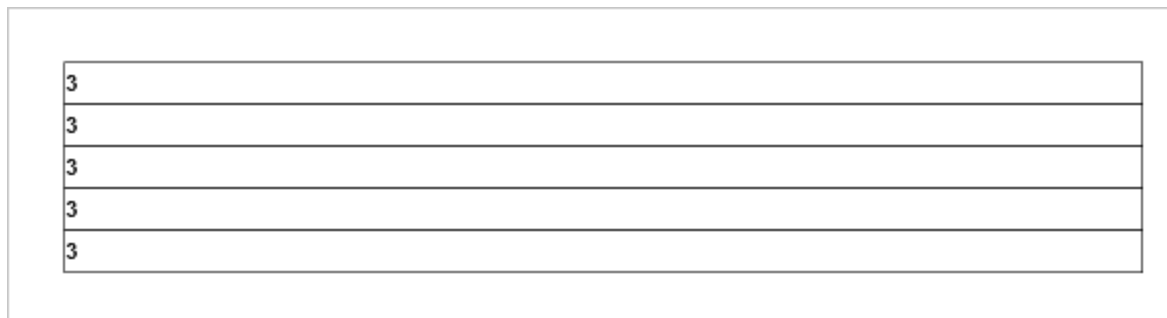
Relationships of bands

As mentioned above, all bands (except **PageHeaderBand1**, **PageFooterBand1**, **OverlayBand1**, **ReportTitleBand1**, **ReportSummaryBand1**) in the report rendering depends on the DataBand1. Consider these relationships in more detail and start with a simple example. The **Data Band** is placed on the template page.

The number of records in the data source is five, and this means that the Data Band is printed 5 times.

Almost all of the bands can be divided into two categories: **Headers** and **Footers**, for each header corresponds to the same type of Footers.

> ⚠ **Notice:** If there is equal number of headers and footers each header corresponds to its own footer. "Header - Footer" correspondence is considered not from top to bottom of the page but from the data band. Let's say there is one data band, two headers and two footers.

The order of the bands on the page from top to bottom.

Order	Band name
1	**HeaderBand3**
2	**HeaderBand2**
3	**DataBand2**
4	**FooterBand3**
5	**FooterBand2**

In this case, the **HeaderBand3** corresponds to **FooterBand2**, and **HeaderBand2** corresponds to FooterBand3. In other words, the first header of the data band corresponds to the footer of the first data band. Here is an example of a rendered report.

It often happens that the number of headers and footers of a particular type is different. For example, let's change the example above, adding **HeaderBand4** between **HeaderBand2** and **DataBand2**. Now HeaderBand4 corresponds to **FooterBand3** (color - yellow), **HeaderBand2** - **FooterBand2** (color - turquoise), but the band **HeaderBand3** (color blue) has no footer.

⚠ **Notice:** Just headers/footers are output only once before/after the data band and the number of them is not affected on anything. Headers and footers are displayed for each group and each group header strictly corresponds to the footer of the group. In complex reports with different number of headers and footers of the group there may be the erroneous relation with headers and footers.

Therefore, we recommend have the same number of bands, headers and footers of the groups in the report template.

Information: In order the band present in the report template but do not appear in a report you should set it height to zero.

For the example above, let's equalize the number of data headers and footers.
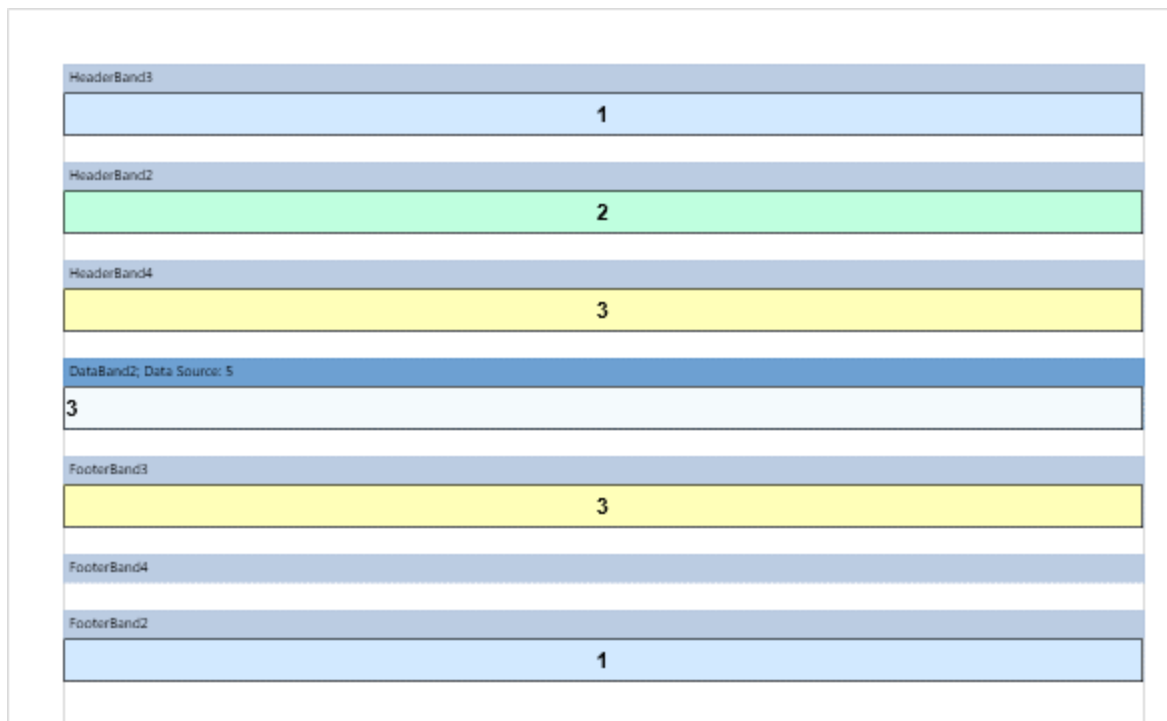
In this case, **HeaderBand4** corresponds to **FooterBand3** (yellow), **HeaderBand2** - **FooterBand4** (turquoise), **HeaderBand3** (blue) - **FooterBand2** (zero height). At the same time, FooterBand4 will not be printed (displayed) in the rendered report.

So there is an equal amount of header and footers in the report and it is easy to determine their correspondence. At the same time, you can turn off (do not display) certain bands. All of the examples above were considered for **Header Bands** and **Footer Bands**. The same principle applies to **Group Header Bands**, **Group Footer Bands Column Header Bands** and **Column Footer** Bands.

Here is an example below where there are a few data bands in the report.

These bands have no connection with each other. Therefore, they are processed sequentially. At first, **DataBand1** (category list) will be processed, and then - **DataBand2** (list of products).

DataBand1 List of Categories		
Beverages	Soft drinks, coffees, teas, beers, and ales	
Condiments	Sweet and savory sauces, relishes, spreads, a seasonings	
Confections	Desserts, candies, and sweet breads	
Dairy Products	Cheeses	
Grains/Cereals	Breads, crackers, pasta, and cereal	
Meat/Poultry	Prepared meats	
Produce	Dried fruit and bean curd	
Seafood	Seaweed and fish	

DataBand2 List of Products		
Chai	18	39
Chang	19	17
Aniseed Syrup	10	13
Chef Anton's Cajun Seasoning	22	53
Chef Anton's Gumbo Mix	21.35	0
Grandma's Boysenberry Spread	25	120
Uncle Bob's Organic Dried Pears	30	15
Northwoods Cranberry Sauce	40	6
Mishi Kobe Niku	97	29
Ikura	31	31
Queso Cabrales	21	22
Queso Manchego La Pastora	38	86
Konbu	6	24
Tofu	23.25	35
Genen Shouyu	15.5	39
Pavlova	17.45	29
Alice Mutton	39	0
Carnarvon Tigers	62.5	42
Teatime Chocolate Biscuits	9.2	25
Sir Rodney's Marmalade	81	40
Sir Rodney's Scones	10	3
Gustaf's Knäckebröd	21	104
Tunnbröd	9	61
Guaraná Fantástica	4.5	20
NuNuCa Nuß-Nougat-Creme	14	76

Now add the **Header Band** to the report template. The **Header Band** will refer to the **Data Band** above what it is located. In order the **HeaderBand1** corresponds to **DataBand1** (list of categories), it must be placed above this data band.

In order **HeaderBand2** be related to **DataBand2** (list of products), it should be placed directly above this **Data Band**.

And then the first page of the report will look the following.

Category	Description
Beverages	Soft drinks, coffees, teas, beers, and ales
Condiments	Sweet and savory sauces, relishes, spreads, a seasonings
Confections	Desserts, candies, and sweet breads
Dairy Products	Cheeses
Grains/Cereals	Breads, crackers, pasta, and cereal
Meat/Poultry	Prepared meats
Produce	Dried fruit and bean curd
Seafood	Seaweed and fish

ProductName	UnitPrice	UnitsInStock
Chai	18	39
Chang	19	17
Aniseed Syrup	10	13
Chef Anton's Cajun Seasoning	22	53
Chef Anton's Gumbo Mix	21.35	0
Grandma's Boysenberry Spread	25	120
Uncle Bob's Organic Dried Pears	30	15
Northwoods Cranberry Sauce	40	6
Mishi Kobe Niku	97	29
Ikura	31	31
Queso Cabrales	21	22
Queso Manchego La Pastora	38	86
Konbu	6	24
Tofu	23.25	35
Genen Shouyu	15.5	39
Pavlova	17.45	29
Alice Mutton	39	0
Carnarvon Tigers	62.5	42
Teatime Chocolate Biscuits	9.2	25
Sir Rodney's Marmalade	81	40
Sir Rodney's Scones	10	3
Gustaf's Knäckebröd	21	104
Tunnbröd	9	61

Header1
DataBand1
List of Categories
Header2
DataBand2
List of Products

Now consider the relationships of footers and multiple data bands. As mentioned above, footers in the report template refers to this data band and only below of which they are directly positioned. At the same time the **Footer Band** is a closing one to the **Header Band**. Suppose you want to display the total by the number of categories. In this case **FooterBand1** must be placed below the data band with a list of categories but above **HeaderBand2** for a list of products.

The report page will look the following way.

Header1

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

DataBand1 List of Categories

Footer1

Header2

ProductName	UnitPrice	UnitsInStock
Chai	18	39
Chang	19	17
Aniseed Syrup	10	13
Chef Anton's Cajun Seasoning	22	53
Chef Anton's Gumbo Mix	21.35	0
Grandma's Boysenberry Spread	25	120
Uncle Bob's Organic Dried Pears	30	15
Northwoods Cranberry Sauce	40	6
Mishi Kobe Niku	97	29
Ikura	31	31
Queso Cabrales	21	22
Queso Manchego La Pastora	38	86
Konbu	6	24
Tofu	23.25	35
Genen Shouyu	15.5	39
Pavlova	17.45	29
Alice Mutton	39	0
Carnarvon Tigers	62.5	42
Teatime Chocolate Biscuits	9.2	25
Sir Rodney's Marmalade	81	40
Sir Rodney's Scones	10	3
Gustaf's Knäckebröd	21	104

DataBand2 List of Products

In order to display the total by the data band with a list of products, **FooterBand2** must be placed below **DataBand2**. For this example, let's calculate the total cost of all the products using the Sum function. The result will be displayed on each page of the report (set the **Print on All Pages** property to true). Below is a page template with the footer by the data band and the list of products.

And then the first page of the report will look the following way.

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

ProductName	UnitPrice	UnitsInStock
Chai	18	39
Chang	19	17
Aniseed Syrup	10	13
Chef Anton's Cajun Seasoning	22	53
Chef Anton's Gumbo Mix	21.35	0
Grandma's Boysenberry Spread	25	120
Uncle Bob's Organic Dried Pears	30	15
Northwoods Cranberry Sauce	40	6
Mishi Kobe Niku	97	29
Ikura	31	31
Queso Cabrales	21	22
Queso Manchego La Pastora	38	86
Konbu	6	24
Tofu	23.25	35
Genen Shouyu	15.5	39
Pavlova	17.45	29
Alice Mutton	39	0
Carnarvon Tigers	62.5	42
Teatime Chocolate Biscuits	9.2	25
Sir Rodney's Marmalade	81	40
Sir Rodney's Scones	10	3

Sum Total: 22

Notice

⚠ **Notice:** For the example described above, the placement of the **FooterBand1** under the **HeaderBand2** is not quite correct.

In this case, **FooterBand1** and **HeaderBand2** do not refer to any **Data Band**. When rendering a report, all data bands will be defined first. Then, for each data band, headers which relate to this band are defined, i.e. all headers located above some footer band or another data band. Footers that relate to this data band are defined next, i.e. these are the footers which are placed below the next header or another data band. Therefore, **DataBand1** in the rendered report will be without a footer, **DataBand2** - without a header, and **HeaderBand2** and **FooterBand1** will not be displayed because they do not belong to any of the data bands.

Category	Description		
Beverages	Soft drinks, coffees, teas, beers, and al		
Condiments	Sweet and savory sauces, relishes, spre seasonings		
Confections	Desserts, candies, and sweet breads		
Dairy Products	Cheeses		
Grains/Cereals	Breads, crackers, pasta, and cereal		
Meat/Poultry	Prepared meats		
Produce	Dried fruit and bean curd		
Seafood	Seaweed and fish		
Chai	18		39
Chang	19		17
Aniseed Syrup	10		13
Chef Anton's Cajun Seasoning	22		53
Chef Anton's Gumbo Mix	21.35		0
Grandma's Boysenberry Spread	25		120
Uncle Bob's Organic Dried Pears	30		15
Northwoods Cranberry Sauce	40		6
Mishi Kobe Niku	97		29
Ikura	31		31
Queso Cabrales	21		22
Queso Manchego La Pastora	38		86
Konbu	6		24
Tofu	23.25		35
Genen Shouyu	15.5		39
Pavlova	17.45		29
Alice Mutton	39		0
Carnarvon Tigers	62.5		42
Teatime Chocolate Biscuits	9.2		25
Sir Rodney's Marmalade	81		40
Sir Rodney's Scones	10		3
Gustaf's Knäckebröd	21		104
Tunnbröd	9		61
			Sum To

Labels: Header1, DataBand1 List of Categories, DataBand2 List of Products, Footer2

The same principle of correspondence applies to **Group Header Band**, **Group Footer Band**, **Column Header Band**, and **Column Footer Band**.

➤ Headers are placed above the Data Band to which they relate and Footers are placed below. Headers and Footers cannot be printed themselves because they must refer to the specific data band.

> ❯ Always check the number of headers and footers, particularly in the report with groups. Sometimes it is easier to add a specific band (header or footer) in order to equalize their number and clearly trace the line. -Set zero height for the band in the report template if you want to hide it in the rendered report.

3.14   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.

3.14.1   Data Band

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:

DataBand1; Data Source: Not Assigned
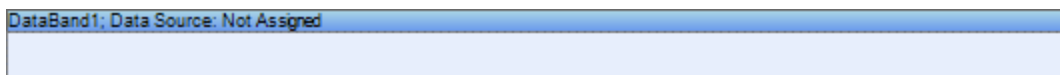
Virtual Data Band
Sometimes it is necessary to print a **Data** band several times without specifying a data source. The **CountData** property is used for this purpose.
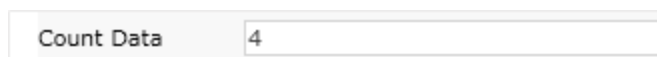
Count Data   4

It is possible to specify number of elements in the **Data** band editor. On the picture below the **Data** editor is shown.
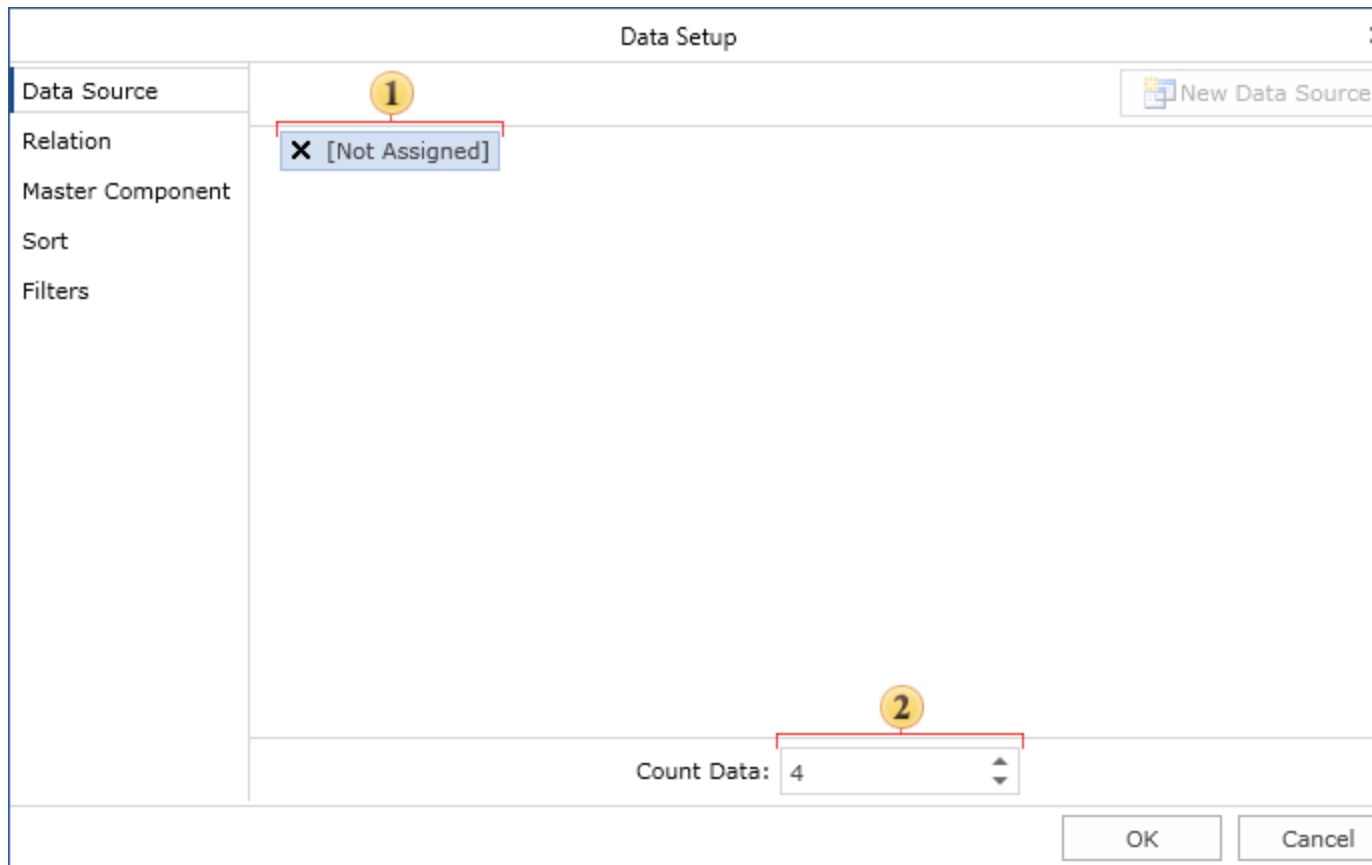
① The field in what number of elements for the **Data** band can be specified.
② A data source is not specified.

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.

3.14.2 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.
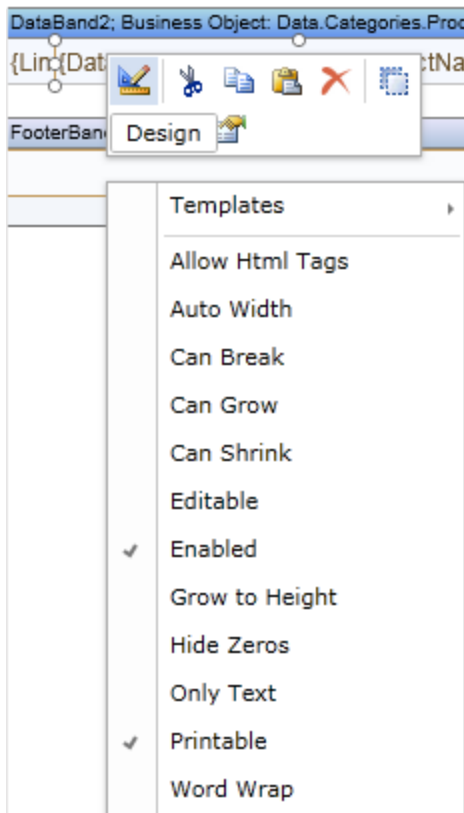
Also the editor can be called using the **DataSource** property of the **Data** band.
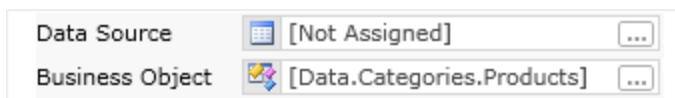
Data band editor allows quickly selecting data source. Data source is selected on the first bookmark of the **Data** band editor. All data sources are grouped in categories. Each category is one data connection with data in the Dictionary of Data. The picture below shows data in the **Data** band editor.
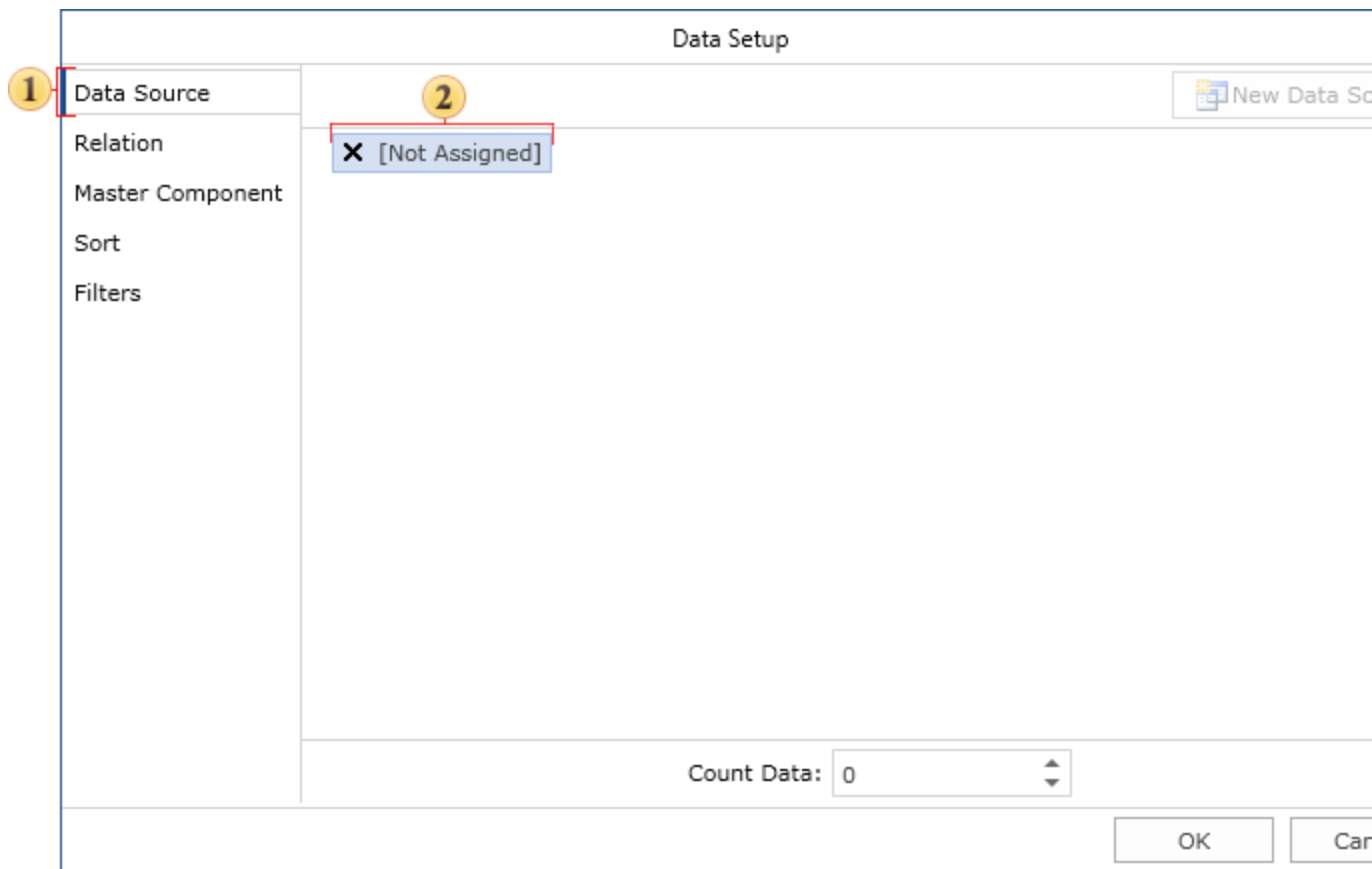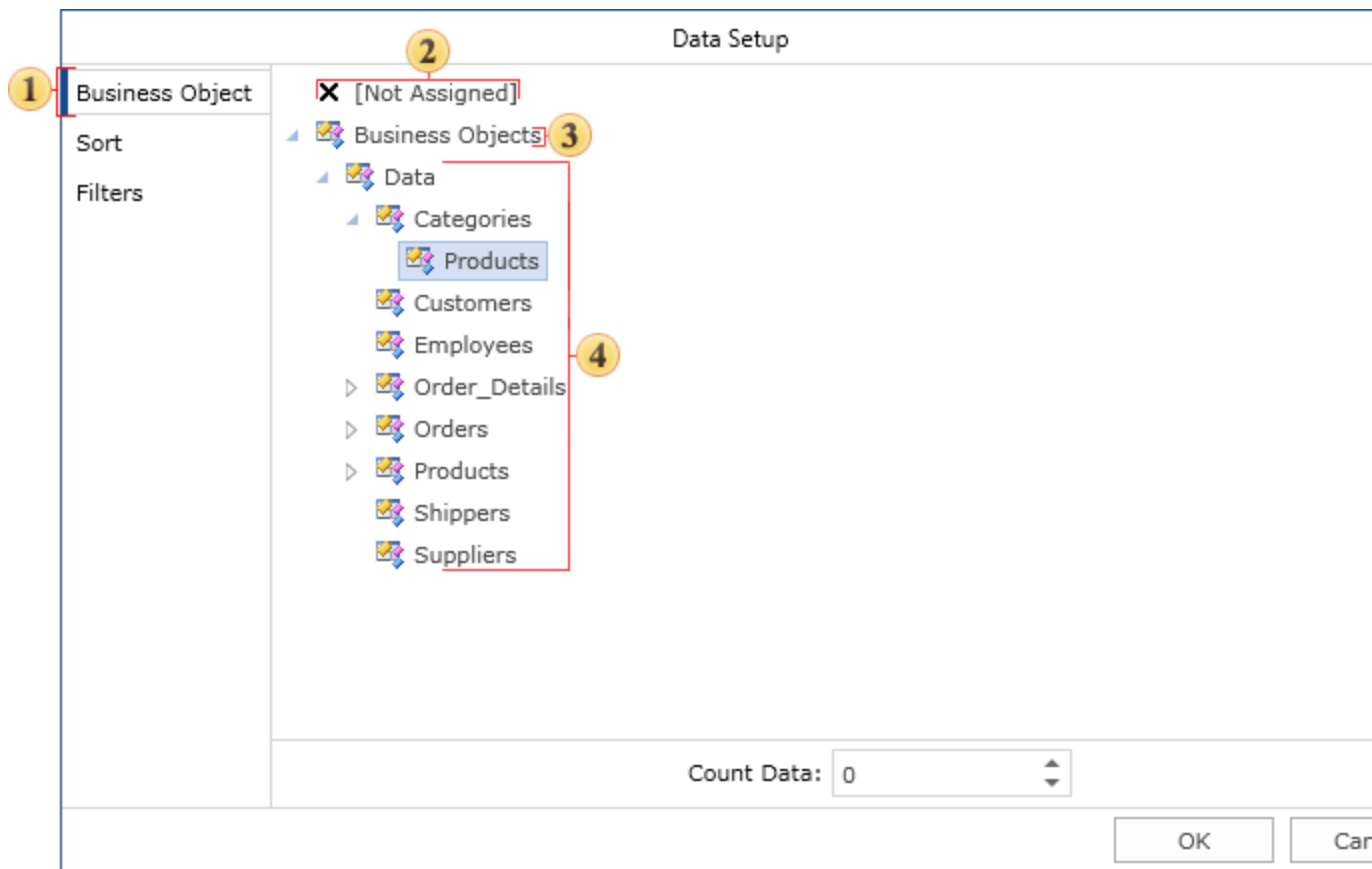
1 Select data source bookmark of the **Data** band.
2 Select this node if there is no need to specify any data source.
3 The "Demo" category of data.
4 The "Demo" category of data source.

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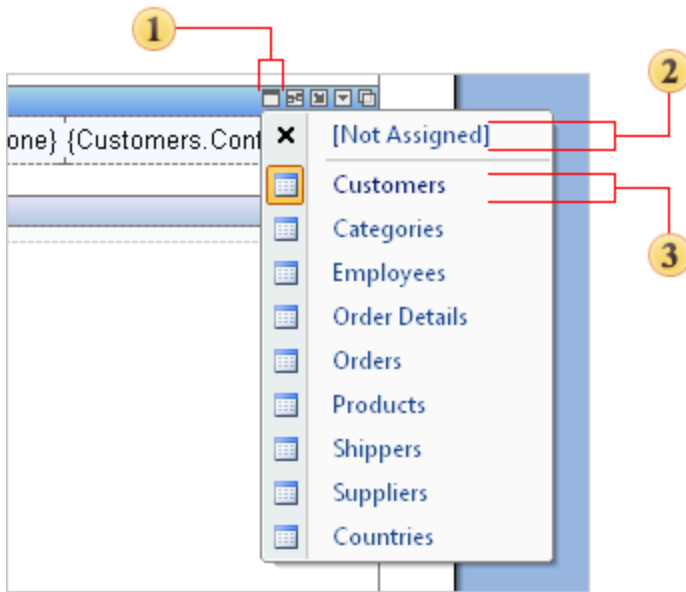
1 Quick button the select data source.
2 This menu item is used to reset data source selection.
3 The **Customers** data source is selected.

3.14.3 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.
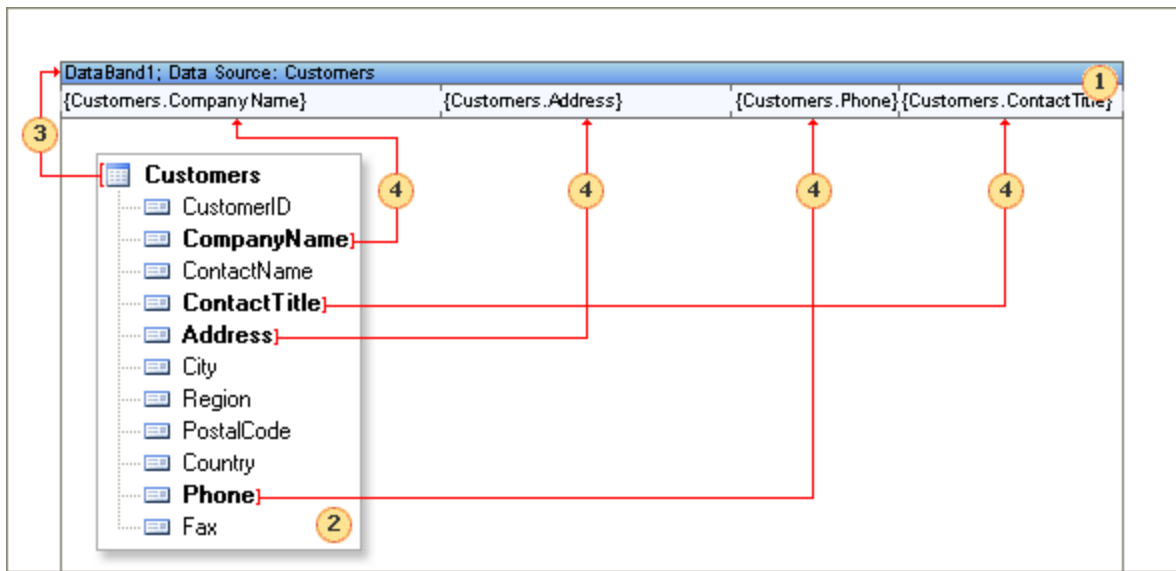
1 **Data** band that outputs a table.

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.

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

3.14.4 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 Taqueria	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.

3.14.5 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:

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.

3.14.6 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.

By default, the **KeepHeaderTogether** property is set to **true.** So headers will be kept together with the first row of data.

3.14.7 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.

And the same report with keeping footer together with the last row of data.

By default, the **KeepFooterTogether** property is set to **true**, so totals of data will be kept together with last row of data.

3.14.8 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.

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:

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}

3.14.9 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.

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.
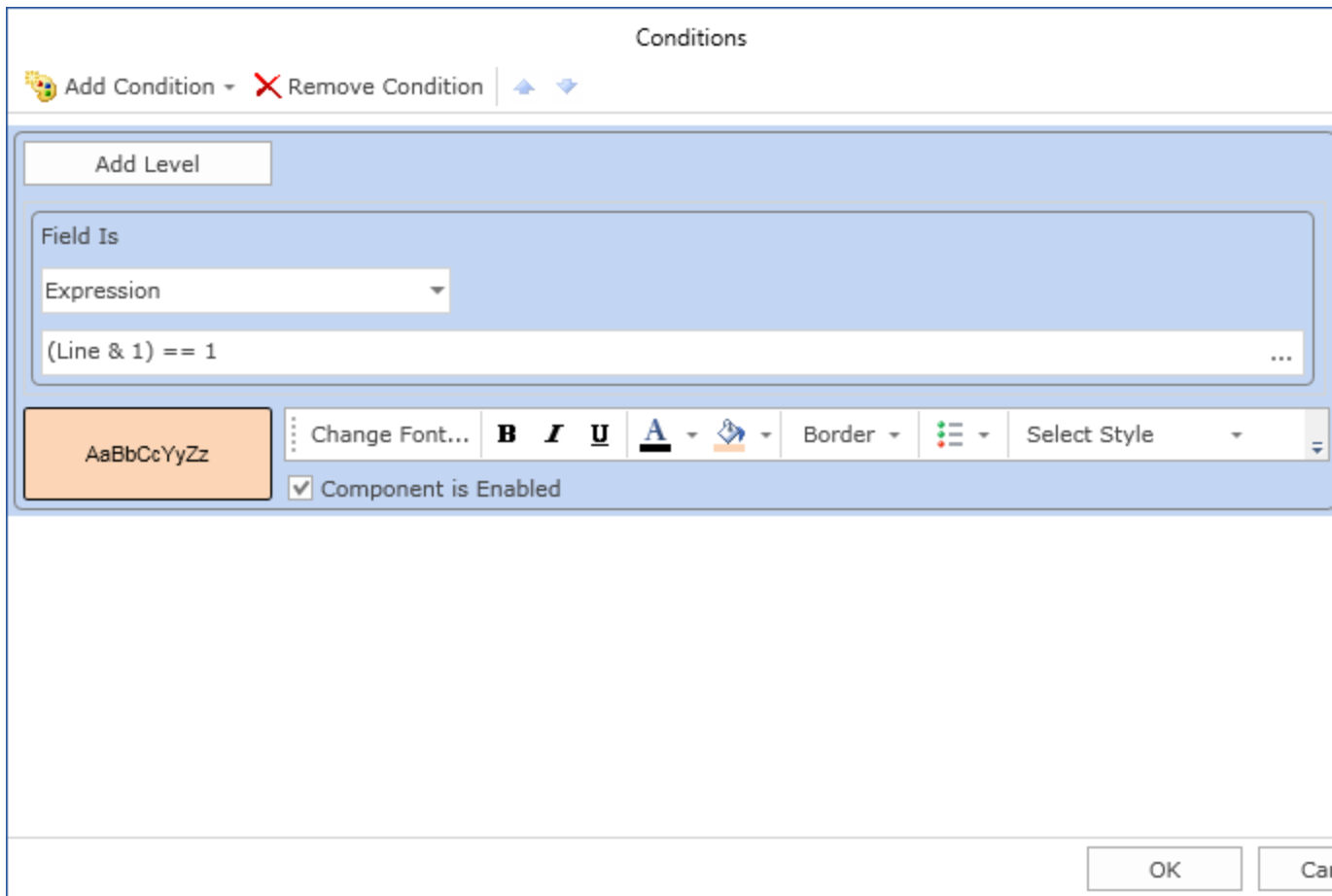
After adding a condition to the data band a report will look as it shown on the picture below.

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.
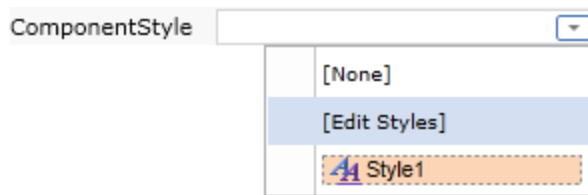
The report looks the same as the one where the first way was used.

3.14.10 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** 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** 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. The **RenderingEvent** 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** will be output 10 times.

Calculate a 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.**

3.14.11 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	[Sort]	...

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.
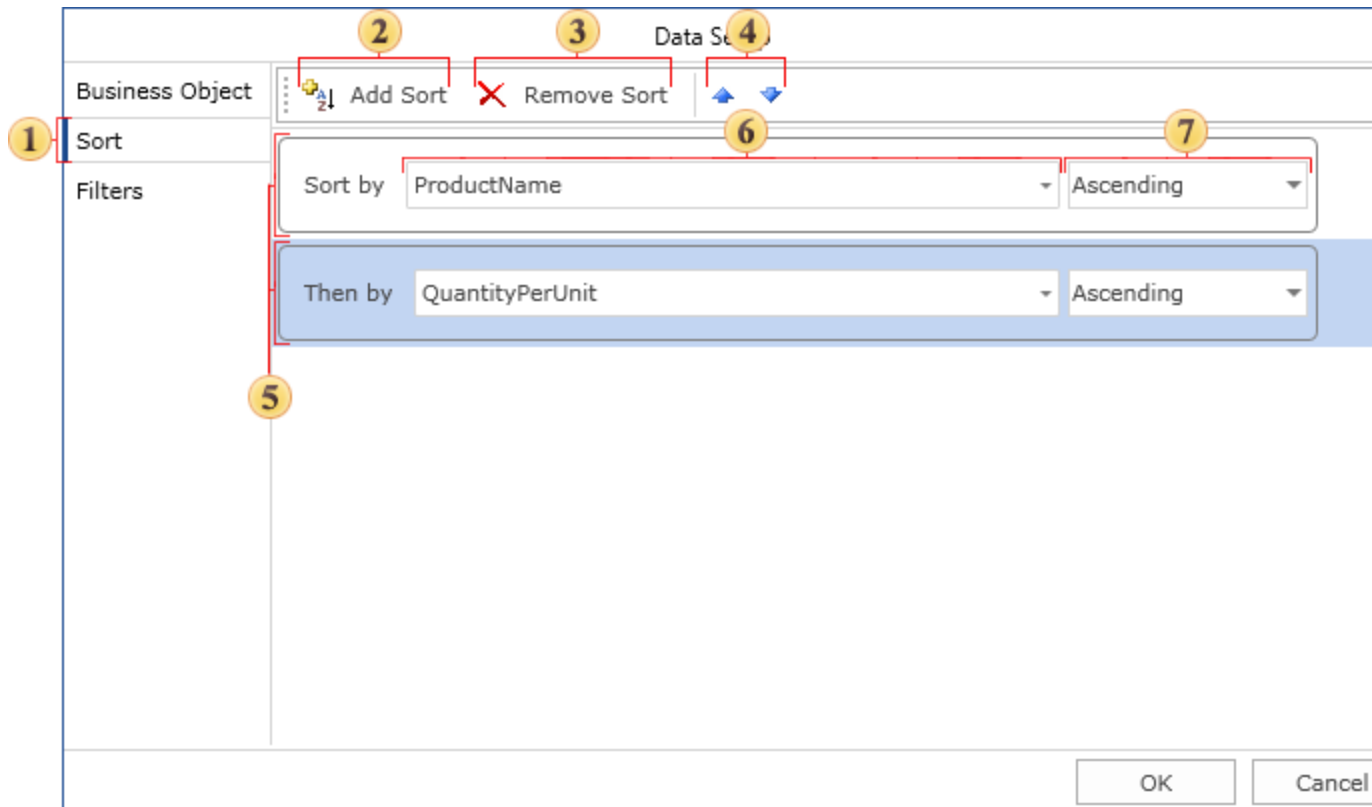
① The Sort bookmark;
② The button to add a new level of sorting;
③ The button to remove the selected level of sorting;
④ Move the selected level of sorting upwards;
⑤ Move the selected level of sorting downwards;
⑥ Level of sorting;
⑦ The column or expression which are used for sorting;
⑧ The button 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.
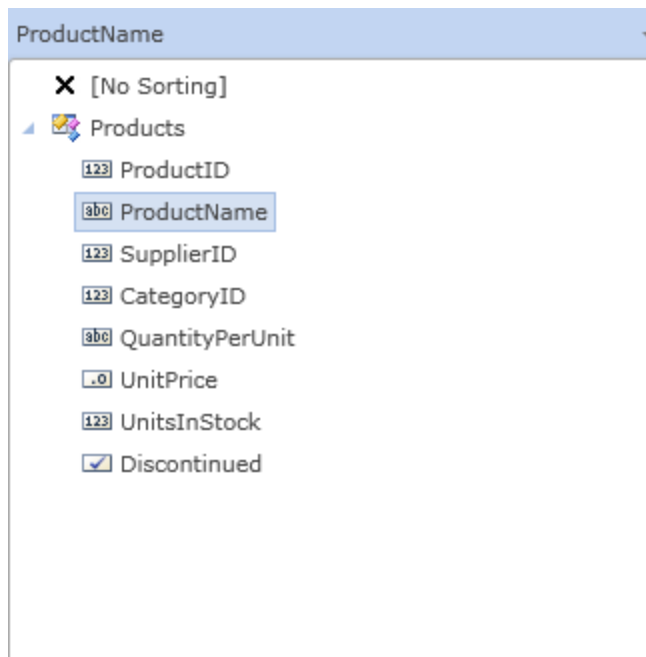
3.14.12 Data Filtering

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 filtering is used. Data filtering 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.

FilterOn	☑	
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 filtering 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:

    ✔ **Value.** The condition is set using the wizard;

    ✔ **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;

❸ 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;

❹ 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;

❺ 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	✔	✔	✔	✔	✔	If the first value is not

to						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.
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.

3.14.13 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.

3.14.14 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**.

3.14.15 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.
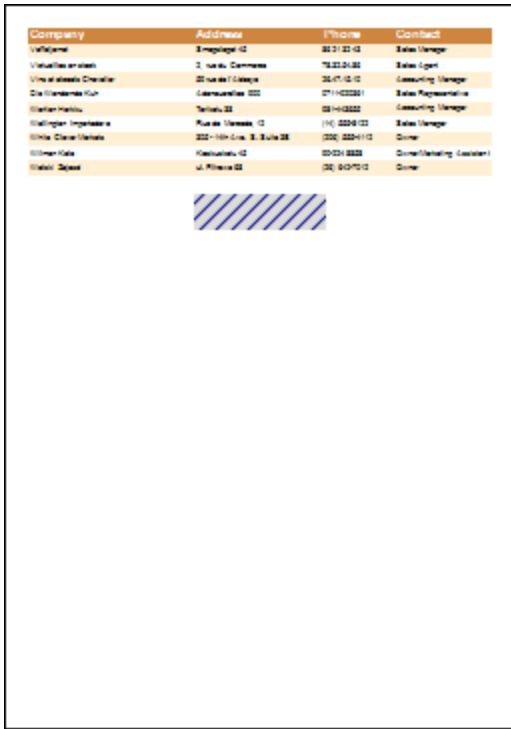
3.14.16 PrintOnAllPages Property

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**.

3.14.17 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.
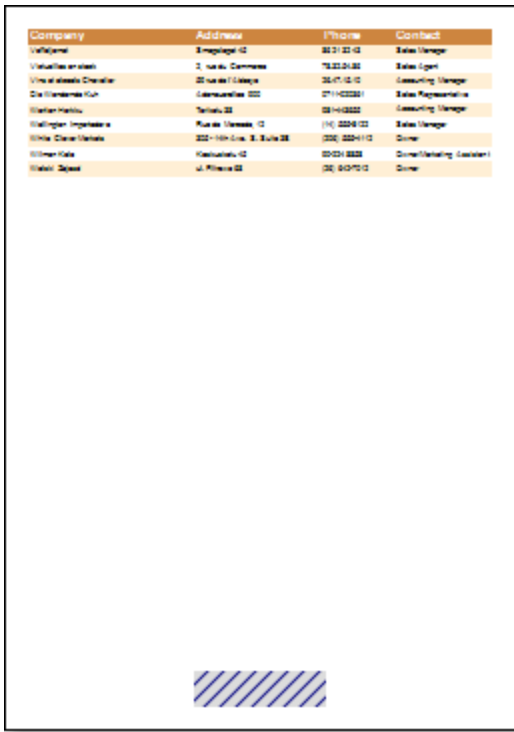
The picture above shows data footer output after data.

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**.

3.14.18 Drag and Drop From Dictionary

The report designer supports a way of dragging components, including the data dictionary. You can drag and drop data sources, columns, variables, functions, and more. You can create a list simply by dragging the data source from the dictionary in the report template. The picture below shows an example of dragging the data source Order Details from the Dictionary on the report page.
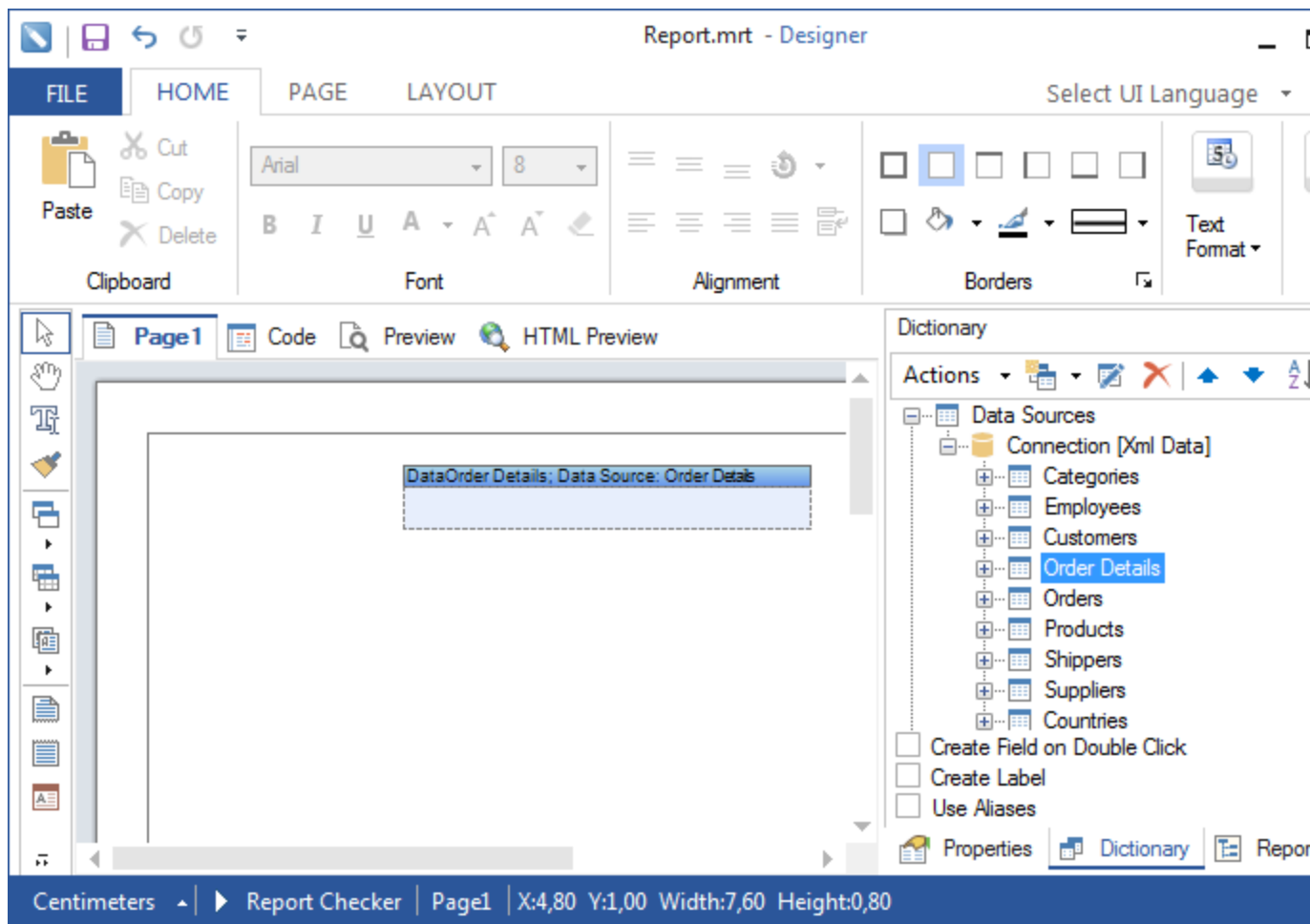
After you release the left mouse button, you will see a dialog box Data, in which you should set the parameters of a new report template. Below is a Data dialog:
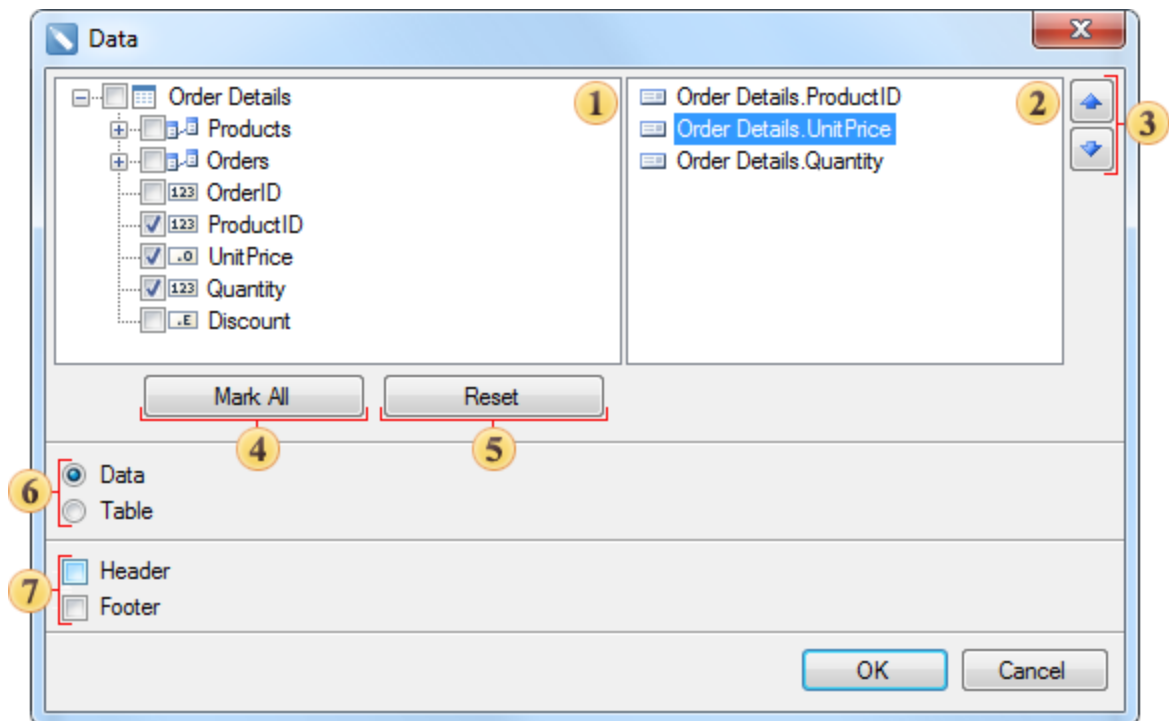
1 This panel displays the columns which contain the data source and the connection between sources. If you need to select the column, references which will be present in the text components on the data band.

2 This panel displays the selected data columns and their order. The order (top-down) on this panel is the order of arrangement of text components on the data band from left to right.

3 These buttons are used to move the selected columns on the panel **2**, thus changing the order of text components on the data band.

4 The button **Mark All**. When clicking it, all columns (a checkbox is set to true) on the panel are selected.

5 The button **Reset**. When clicking, it sets the selection parameters by default (checkbox is set to false), i.e. no column are selected.

6 Selects a container for data: data band and a table.

7 If you want to add bands Header and/or Footer into the report template, you should set the appropriate option.

3.14.19 Check Box

For displaying Boolean values, you can use the **Check Box** component. Various styles can be applied to it. The picture below shows the available styles of check boxes:
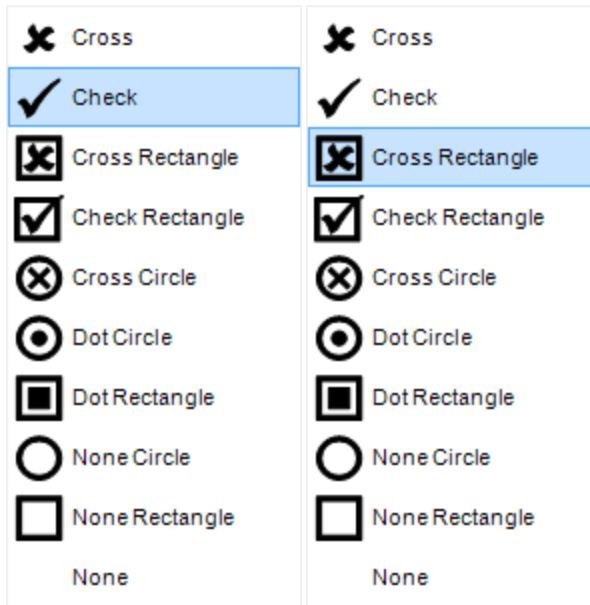
You can set a checkbox style to each Boolean value. To do this, select values of the Style property for True (Check style for **True**) and style values for False (Check style for **False**). You can also change the type of values.
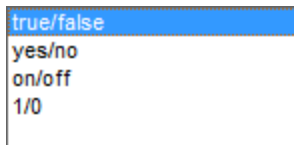
selecting the necessary type in the property field **Values**.

3.15   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:

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.

3.15.1 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.

Master Componen [DataBand1]   [...]

The selection can be made in the **Data** band editor window.
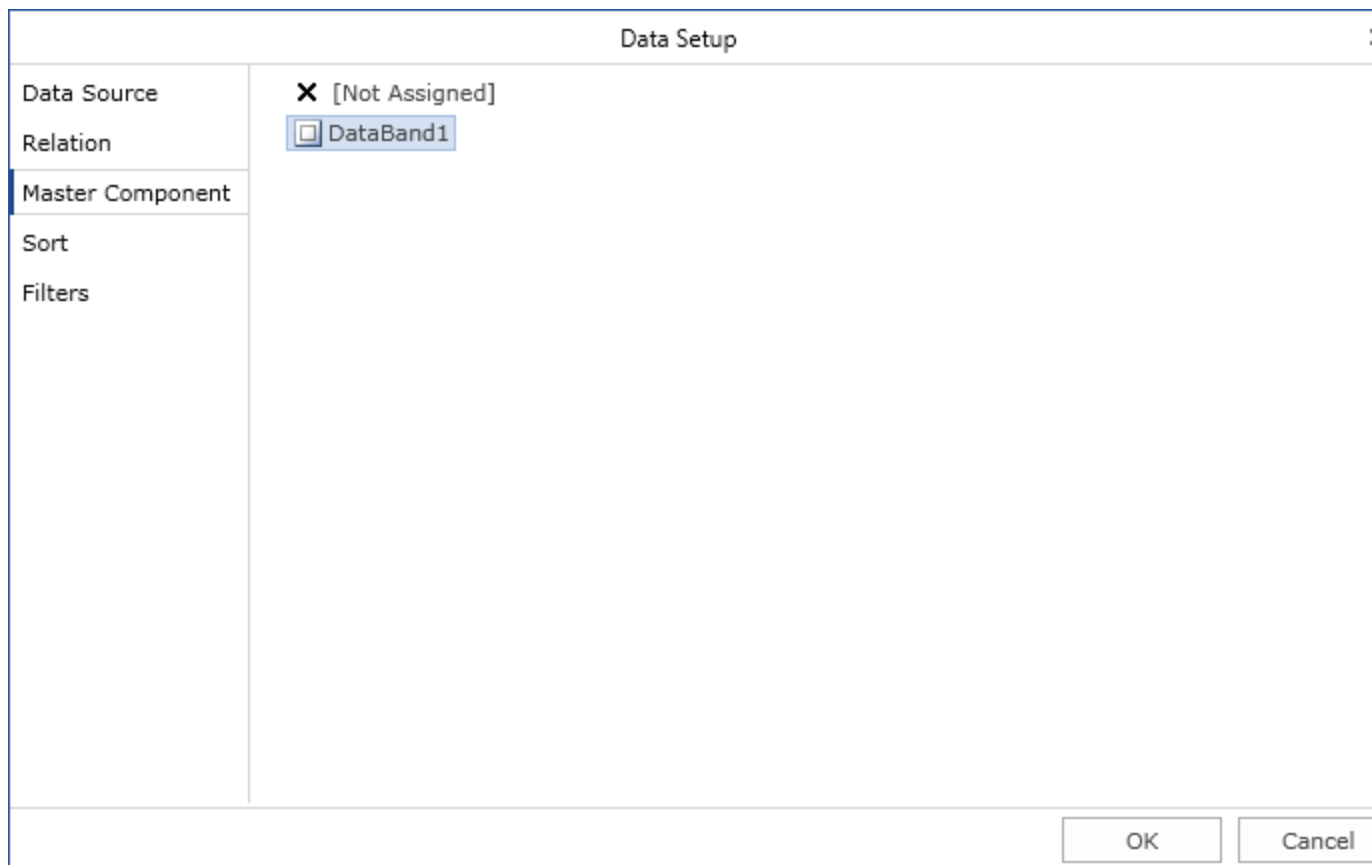
After filling the **MasterComponent** property two bands will be bound to each other. When printing one row of the Master band, all rows of the Detail band will be output. The Detail band will not be printed itself but only in relation to the Master band.

3.15.2 DataRelation Property

After filling the **MasterComponent** property it is necessary to fill the **DataRelation** 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.

Data Source		New Relation
Relation	✕ [Not Assigned]	
Master Component	⊟ 📦 Demo	
Sort	📄 Categories	
Filters	📄 Suppliers	
		OK Cancel
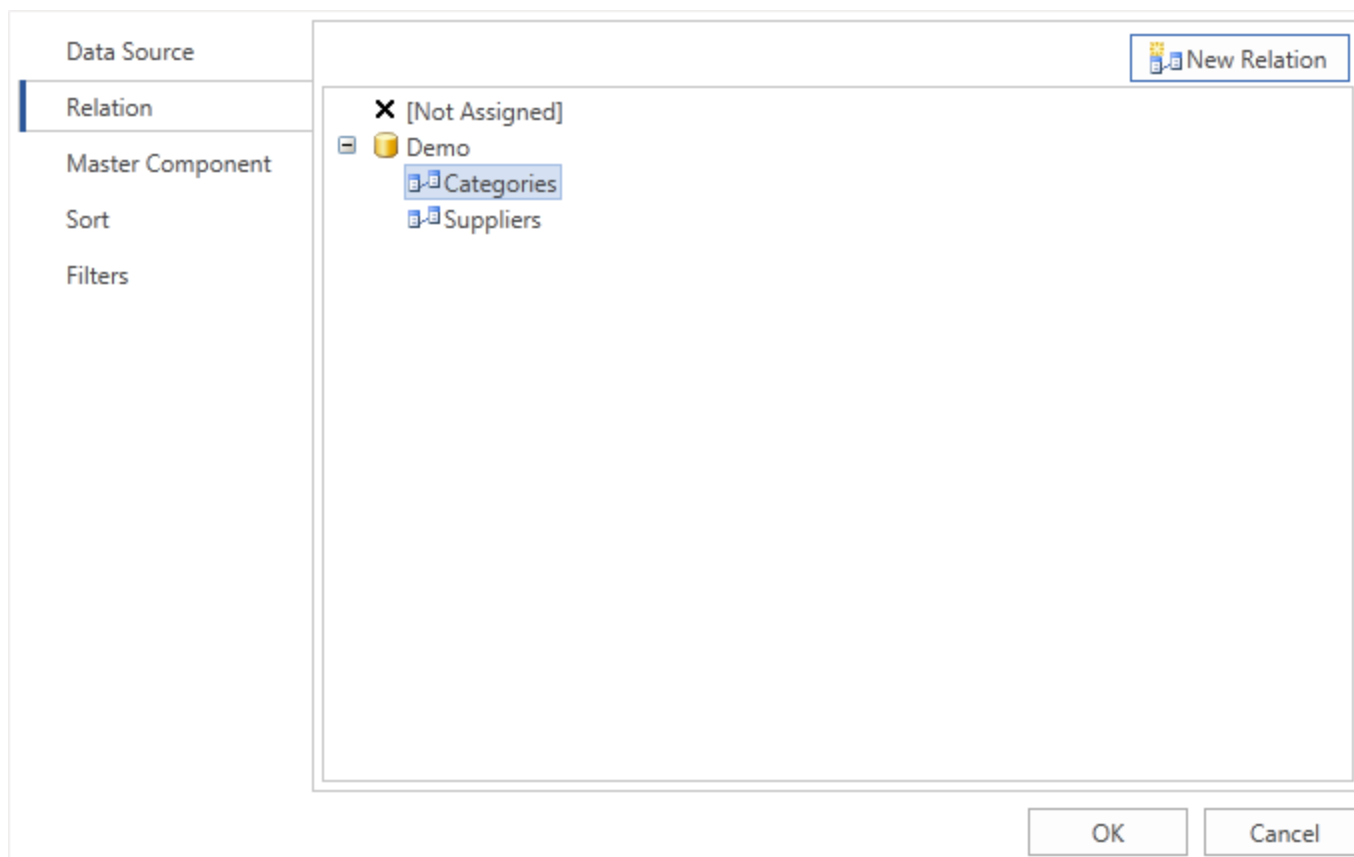
Selection is done between relations which were created between Master and Detail data

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.

3.15.2.1 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:
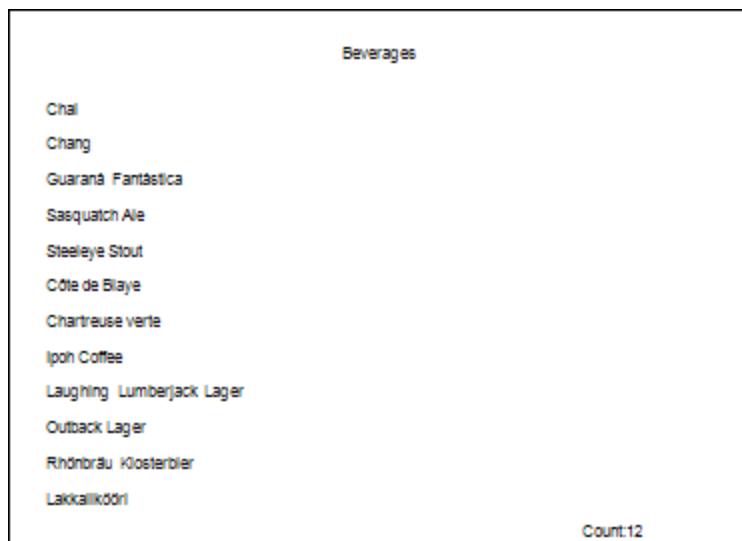
Categories

	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:
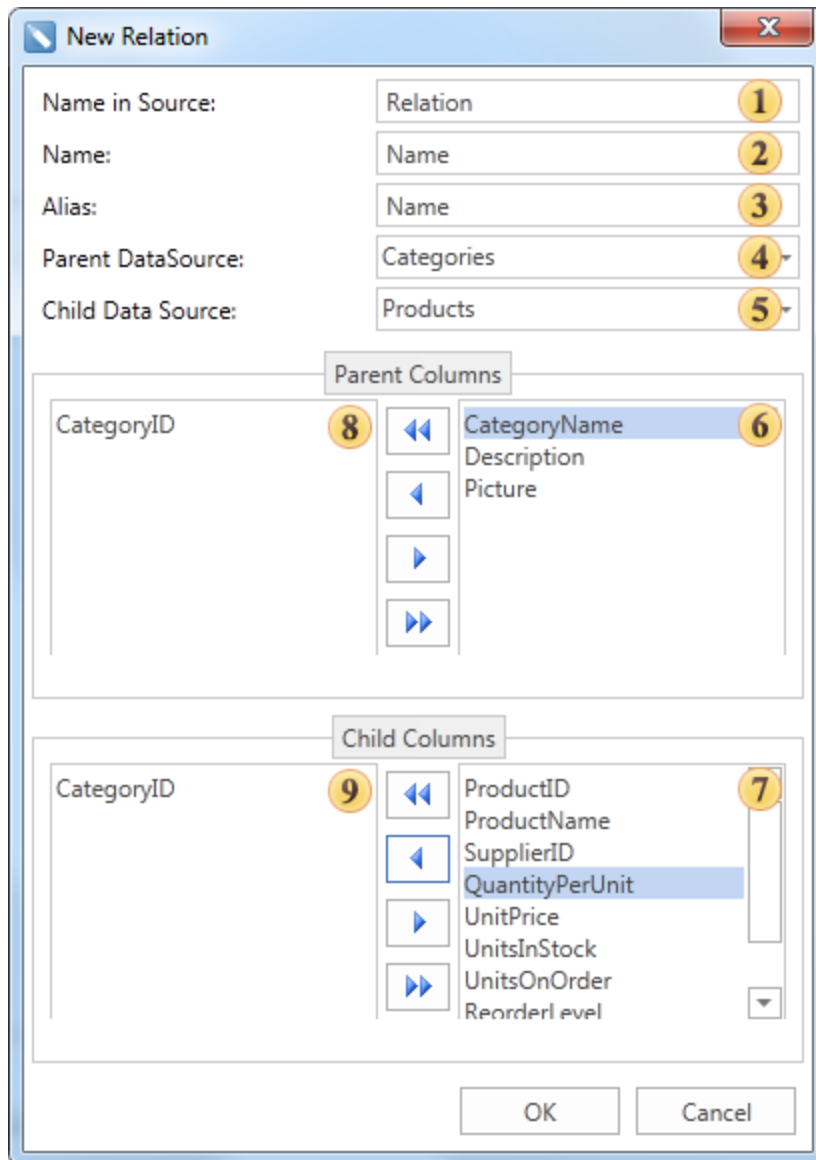
As can be seen on the picture above, nine fields, which define the relation parameters:

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 ◀◀

2 ◀

3 ▶

4 ▶▶

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.

3.15.3 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.
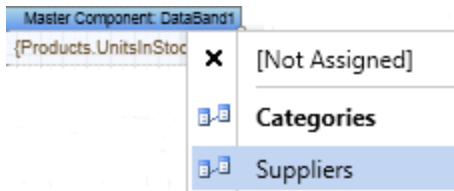
Then this report will be ready for rendering. One **Master** band may have more than one **Detail** band. In other words two, three or four **Detail** bands may refer to it. And each of them may have their own Detail bands. There are no limitations on number of nesting levels in the **Master-Detail** reports.

❗ **Notice:** Number of nesting levels in the Master-Detail reports is unlimited.

3.15.4 KeepDetails 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 **KeepDetails** property. By default, this property is set to **false**.
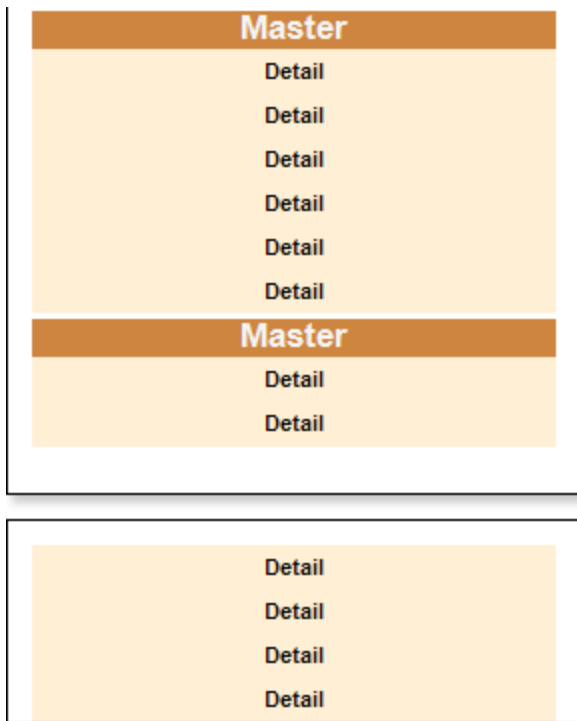
The picture above shows a report in what a part of Details is located on one page, while the other part of details has been moved to the next page. If property is set to **true**, then the report generator will try to place the **Master** and **Detail** records on one page. If the report generator cannot do it, the **Master** and **Details** together will be moved to the next page.
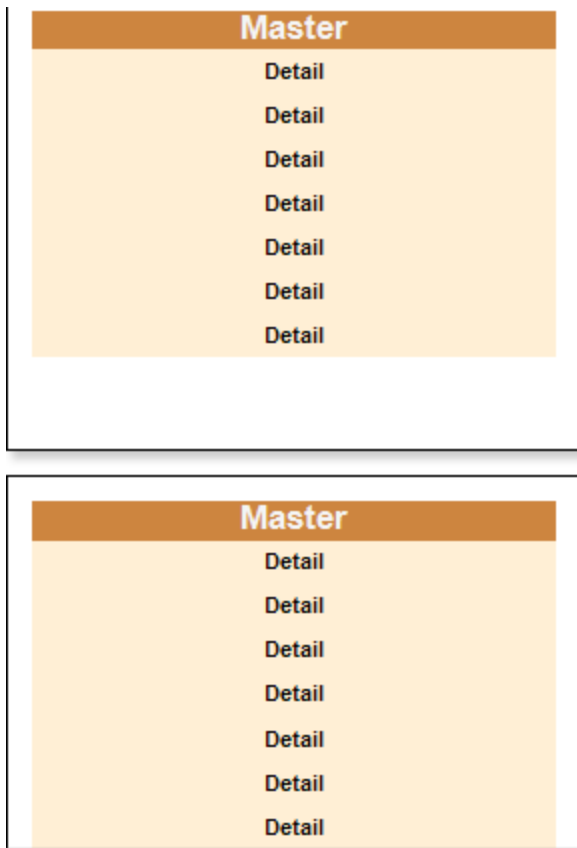
The picture above shows an example of a report with the **KeepDetails** property of the **Master** set to **true**. If it is not possible to put them together, then the data will be forcibly broken and displayed on different pages. In this case, if the **Master** component has many **Detail** records and take a significant part on the page, and the **KeepDetails** property is set to **true**, then there may be a large empty space at the bottom of each page.

3.15.5 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.

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:

3.15.6 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.

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.

3.15.7 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.
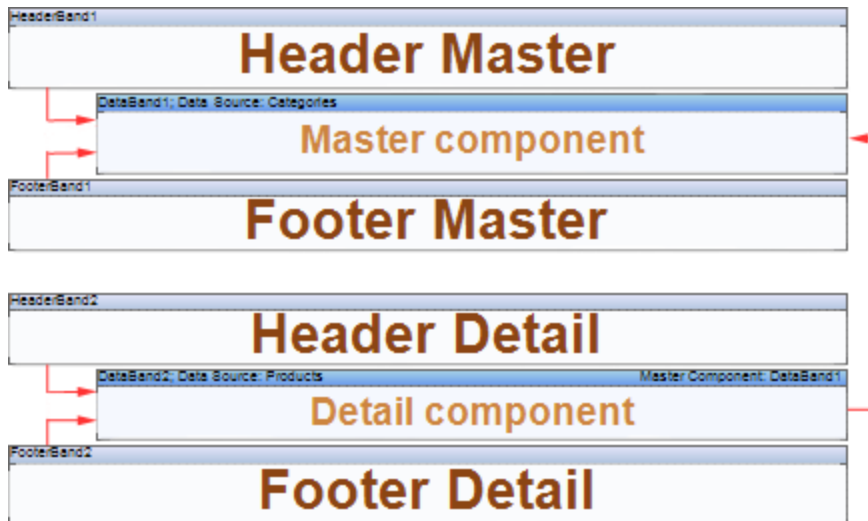
The picture above shows a structure of a **Master-Detail** reports with two **DataBand** bands.

3.15.8 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:

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**.

3.16 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:

3.16.1 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 Group Header band.

❗ **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.

3.16.2 Group Header band

The Group header is created using the **Group Header** band, the basic band for rendering reports that use grouping. It is impossible to generate grouped reports without using a **Group Header** band.

The **Group Header** 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 **Group Header** band designer or the **Condition** property of the band.

❗ **Note:** The Header band is always output before the **Group Header** band, regardless of where bands may be positioned on a page in the designer.
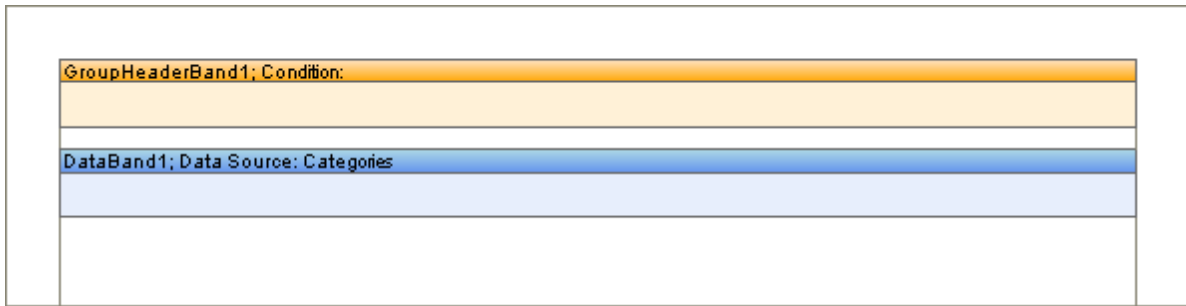
When rendering a report the report generator binds the group header to the specified Data band. The **Group Header** band is positioned on a page before the **Data** band that outputs data rows. The **Group Header** band always belongs to a specific **Data** band, usually the first **Data** band positioned under the **Group Header** 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.

3.16.3   Group Footer band

The **Group Footer** 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 **Group Footer** band belongs to the **Group Header** band associated with it, and will not be output if there is no associated **Group Header** band.

❗ **Note:** The **Group Footer** band is always output before the Footer band regardless of where bands may be positioned on a page.
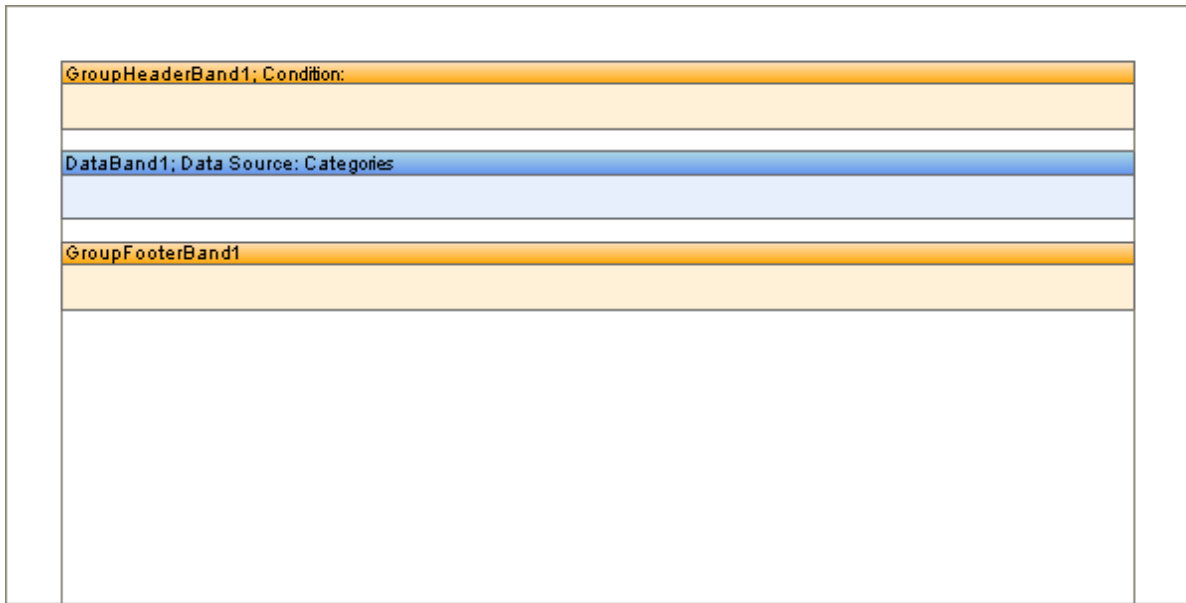
The **Group Footer** band is used to output information specific to each group. For example, if you want to output the number of rows in a group, it is enough to put a text component on the **Group Footer** band and assign it the following expression:

{Count()}

3.16.4 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**.

> **None**. The data will be displayed in order they are put in the data source.

> **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:

> **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:

3.16.5 GroupFooter

It is enough to place a text component with an aggregate function in a **Group Footer** to output footer by group. Also, the footer of a group may be placed in a **Group Header** band. 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 **Group Footer** band.

After rendering, it is possible to see that in the footer of each group calculation by number of rows is done.

3.16.6 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.

This can be avoided using the **KeepGroupTogether** property of the **Group Header** band. 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.

Work with this property may lead to empty space on page, if groups contain a large number of rows.

3.16.7 KeepGroupHeaderTogether Property

The **Group Header** band 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.

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.

By default the **KeepHeaderGroupTogether** property is set to **true**.

3.16.8 KeepGroupFooterTogether Property

The **Group Footer** Band 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:
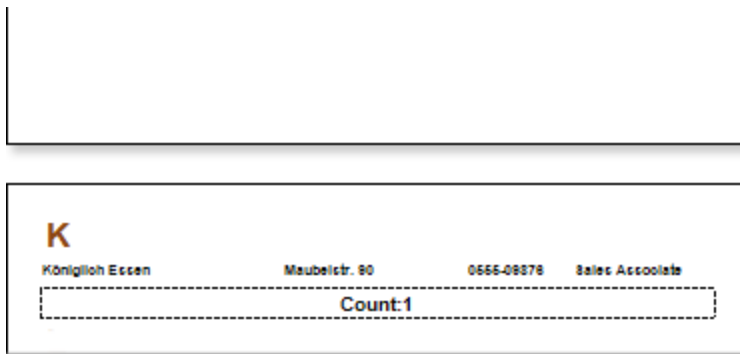
By default this property is set to **true**.

3.16.9 Events and Group Header band

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.

3.16.10 Group Without Group Header

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 **Group Header** band, but if you do not want it to display it can be hidden by setting the height of the **Group Header** band to **0**

which will cause the report to be rendered successfully but the **Group Header** band will not appear in the output.
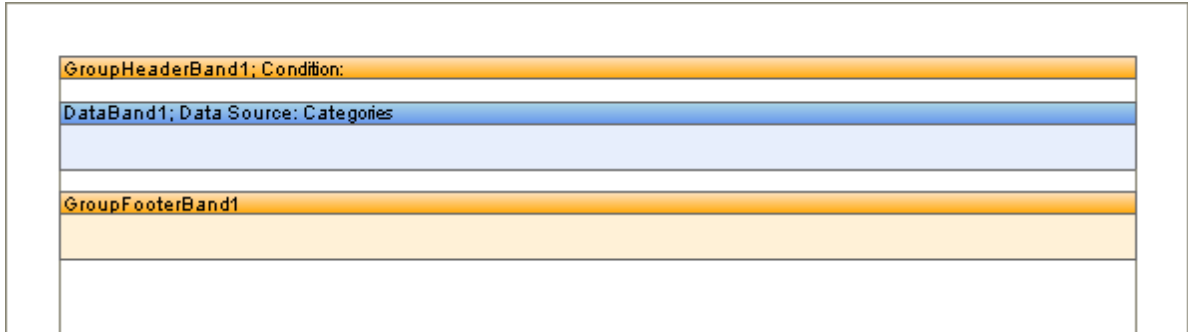
3.16.11 Nested Groups

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 **Group Header** bands before the **Data** band and ideally the same number of **Group Footer** bands immediately after it:
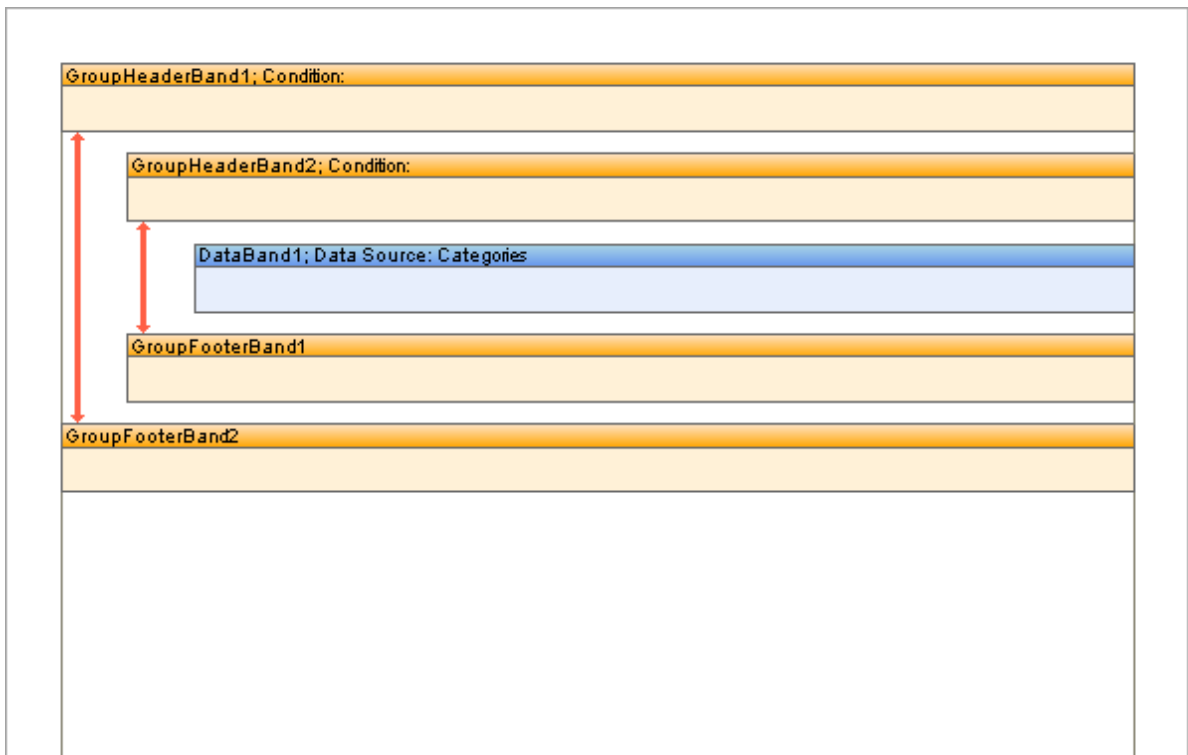
Although it is possible to leave out unwanted **Group Footers** it is recommended that you always place equal numbers of **Group Header** and **Group Footer** bands on a report to avoid unexpected results. If the number of **Group Footer** bands is greater than the number of **Group Header** bands then the outer ones will be used and the inner bands ignored. If the number of **Group Footer** bands is less than the number of **Group Header** bands, then the **Group Header** 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 **Group Header** band you must specify the grouping criteria. When rendering the report the **Group Header** 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 **Group Footer** bands on a report page it is important remember that the last **Group Footer** band is always associated with the first **Group Header** band.

3.16.12 Groups Without Group Footer

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 **Group Footer**, 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 **Group Headers** and **Group Footers** in pairs.

❗ **Important:** To render reports with grouping you should always use Group Headers and Group Footers in pairs to avoid the possibility of unexpected results.

If you do not want the **Group Footer** 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.

3.16.13 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	Rue Ords, 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).

3.16.14 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.

3.16.15 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:

A text component with the GroupLine system variable can be placed in the Group

Header band band, and in the Group Footer band band.

3.16.16 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 **Group Header** is placed on a page above the **Data** band, which outputs data rows. The **Group Header** band always refers to a specific **Data** band. Typically, the band is the first **Data** band, which is placed below the **Group Header** band. To render a report with the grouping, the **Data** band is required. The **Group Footer** band is placed below the **Data** band. It is meant that very **Data** band, with what the **Group Header** band is bound. Each **Group Footer** band, refers to a certain **Group Header** band. The **Group Footer** band will not be output if there is no the **Group Header** band.

The picture above shows a combination of **Group Header** band and **Group Footer** band bands with **Data bands** in a **Master-Detail** report.

3.17   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: **Page Header**, **Page Footer**, and **Empty Data**.

3.17.1 Page Header Band

The Page Header band is used to output information such as page numbers, dates, and company information at the top of a page. The Page Header band is output at the top of every page of the report. An unlimited number of Page Header bands can be placed on a page.

❗ Note: The number of Page Header bands that can be placed on a page is effectively unlimited other than by available space.
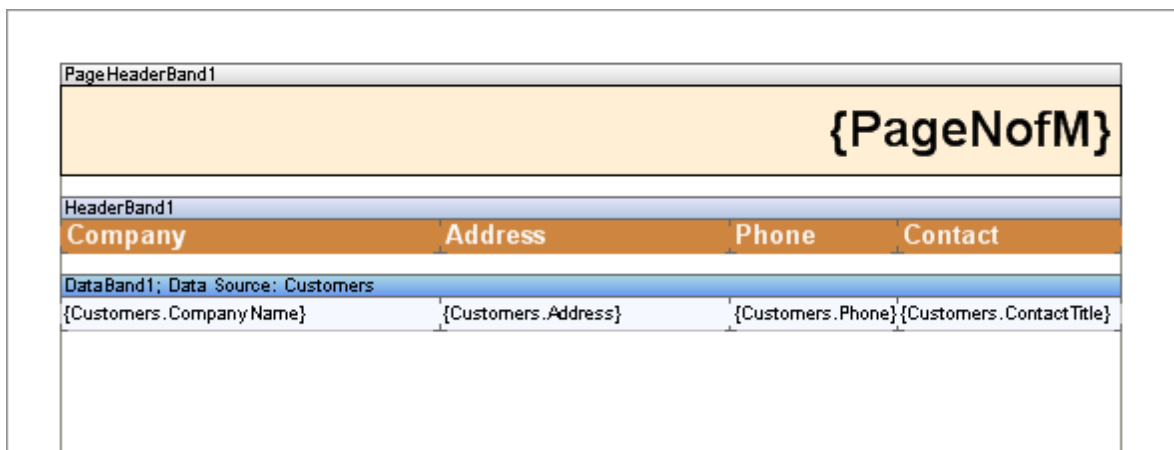
Example

Create a new report and drop three bands on a page: a Page Header band for the current page number and number of pages in the report, a Data band to output data and a Header band band to output data column headers. Drop a text component on the Page Header 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.

3.17.2 Page Footer Band

The Page Footer band is used to output information such as page numbers, dates, and company information at the bottom of a page. The Page Footer band is output at the bottom of every page of the report. An unlimited number of Page Footer bands can be placed on a page.

❗ **Note:** The number of Page Footer 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 Page Footer band for the current page number and number of pages in the report, a Data band to output data and a Header band band to output data column headers. Drop a text component on the Page Footer 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:

HeaderBand 1			
Company	Address	Phone	Contact

DataBand 1; DataSource :Customers			
{Customers .CompanyName}	{Customers .Address}	{Customers .Phone}	{Customers .ContactTitle}

Page FooterBand 1
{PageNofM}

Now run the report, and you will see that the page number is printed at the bottom of each page.

3.17.3 PrintOnEvenOddPages Property

The **PrintOnEvenOddPages** property is used to print headers and footers on even/ odd pages, for **Page Header** bands and **Page Footer** bands.

The picture above shows a sample of a report with the **PrintOnEvenOddPages** property of the **Page Header** band set to **EvenPage**.

The picture above shows a sample of a report with the **PrintOnEvenOddPages**

property of the **Page Header** band 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.

3.18 Report Bands

There are two report bands in Stimulsoft Reports: the **Report Title** and the **Reports Summary** bands. The Report Title band is output in the beginning of a report and the Report Summary band is output in the end of a report. The number o **Report Title** and **Report Summary** bands on a page is unlimited. The **Report Title** and the **Report Summary** bands can be output more than one time and can be used on each page.

3.18.1 Report Title band

One of the ways to display the report header is the way of using the **Report Title** band. The report header will be output only once in the beginning of a report. The **Report Title** band is placed after the **Page Header** band, and before the **Header** band. The number of **Report Title** bands on a page is unlimited.
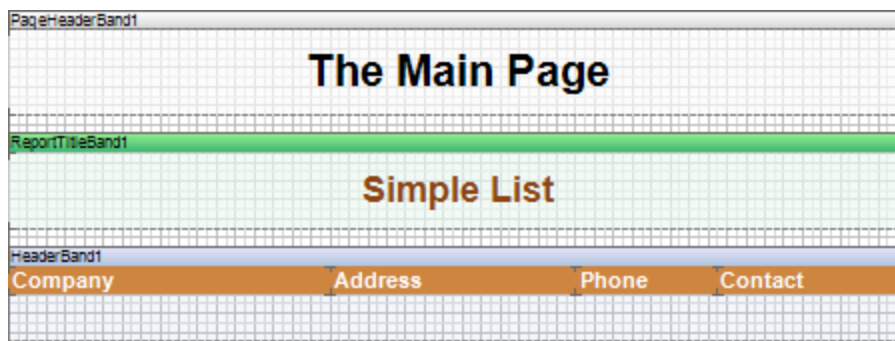
On the picture above shows how bands can be placed on a page. Here one can see top-down the **Page Header**, **Report Title**, and **Header** bands.

3.18.2 Report Summary band

A report summary can be output using the **Report Summary** band. The number of

Report Summary bands placed in a report is unlimited. This band is output on each page as many times as there are pages.
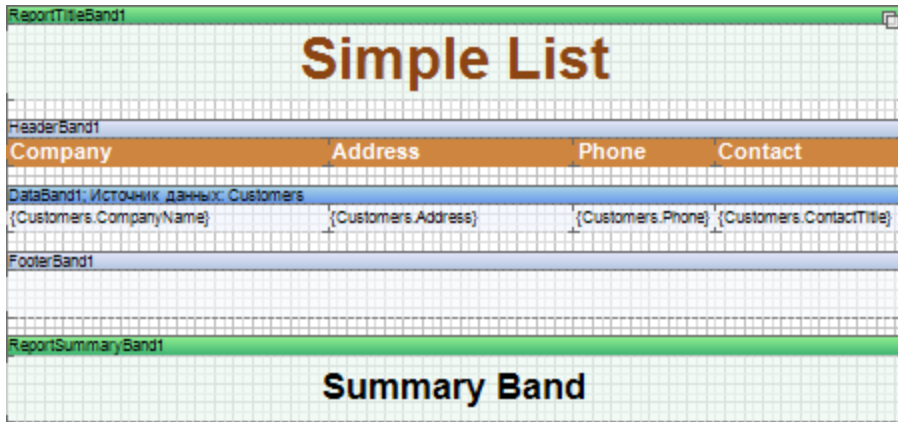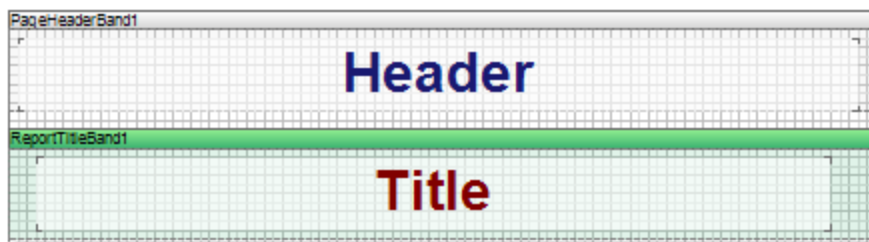

This band is used to output report summary.

On the picture above shows how bands can be placed on a page. Here one can see the top-down order of bands:
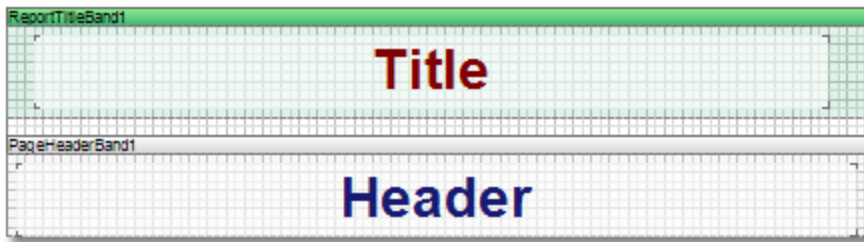- ✓ The **Report Title** band;
- ✓ The **Header** band;
- ✓ The **Data** band;
- ✓ The **Footer** band;
- ✓ The **Report Summary** band.

3.18.3 ReportTitleBand Property

By default, the **Page Header** band is placed above the **Report Title** band:

but it is also possible to output the **Report Title** band before the **Page Header** band:

By default this property is set to **false**. Set the **TitleBeforeHeader** property to **true** and the **Report Title** band will be output before the **Page Header** band.

3.18.4 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.
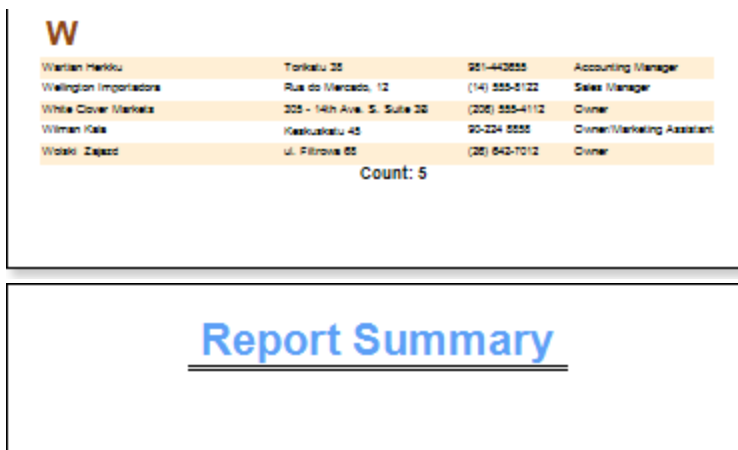
To avoid such unpleasant incidents the **Report Summary** band has the **KeepReportSummaryTogether** property.
If the **KeepReportSummaryTogether** property is set to **true**, then minimum one data row will be printed with the report summary. Thus it is necessary to take into account that after the data row is transferred free space may remain on a fist page. Therefore, one should take this into account when working with this property.
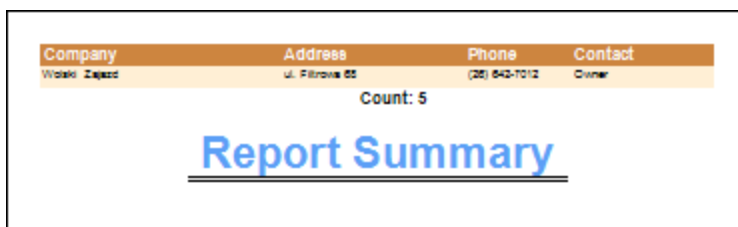
The default value of the property is set to **true**.

3.18.5 Print At Bottom 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.

But it is necessary that the report summary should be placed on the bottom of the page. The **Report Summary** band 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
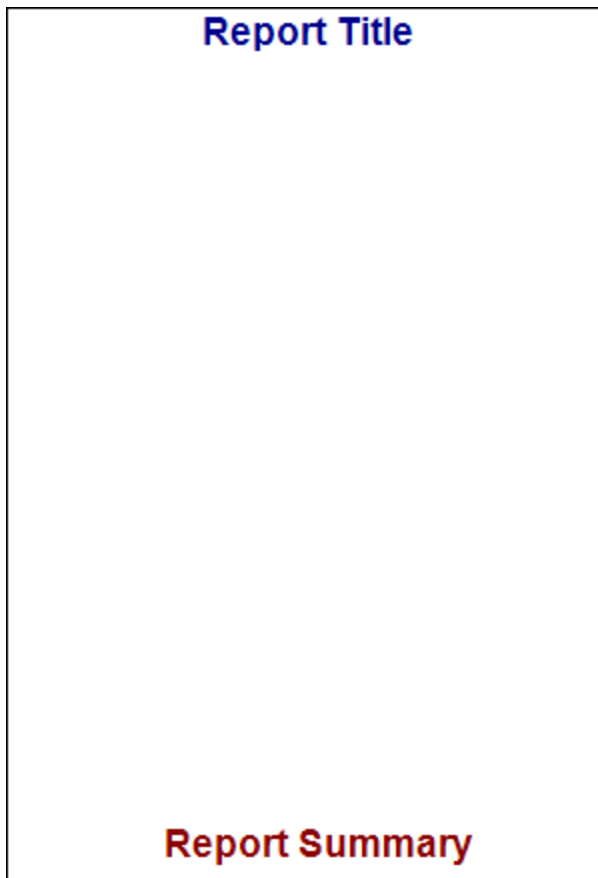
bottom of the page.

3.18.6 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 **Report Title** band, and the **Report Summary** band.

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 Report Summary band band is also set.

3.19 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.

3.19.1   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	☐

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.
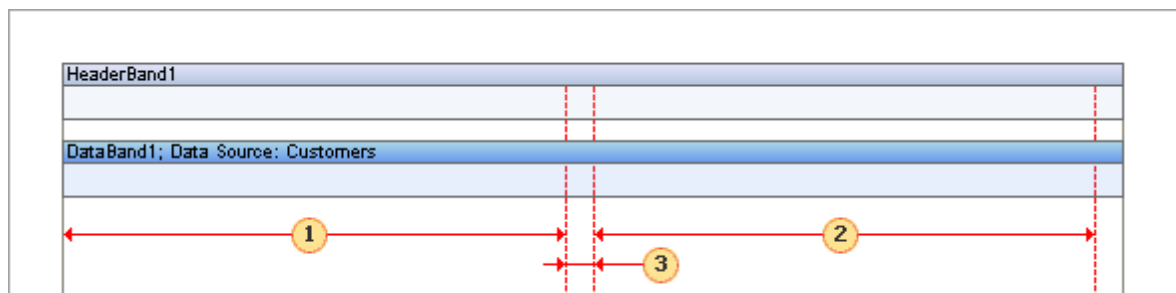
1	The first column width
2	The second column width
3	The space between columns

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.

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.

3.19.2 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.
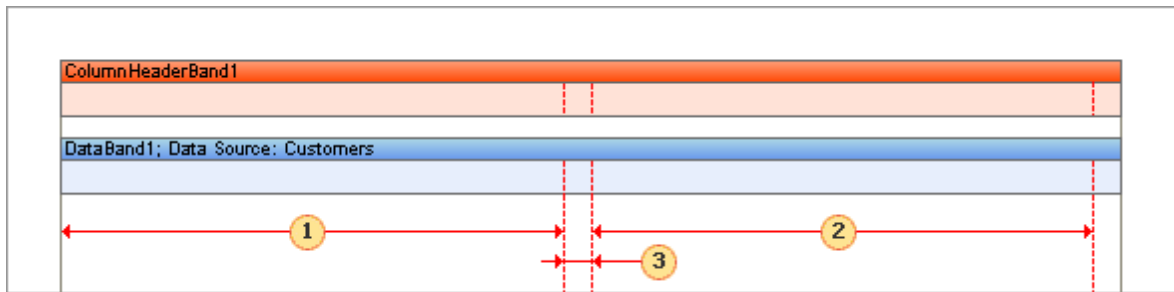
1️⃣ The first column width
2️⃣ The second column width
3️⃣ The space between columns

❗ **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**.

3.19.2.1 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.

**Multi Column Bands
Across then Down**

❗ **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.

3.19.2.2 DownThenAcross Mode

The **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 the **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 Taqueña	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	

3.19.2.3 Minimal Number of Rows 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:

3.19.2.4 Column Header Band

The Header band is normally used to output data headers, but there is also a special **Column Header** band. The Header band is output once before the Data band and contains only one set of data. The **Column Header** 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 **Column Header** 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 **Column Header** band. Put two bands on a page: A **Column Header** 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 the **DownThenAcross** mode.

Place a text component on the **Column Header** 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.
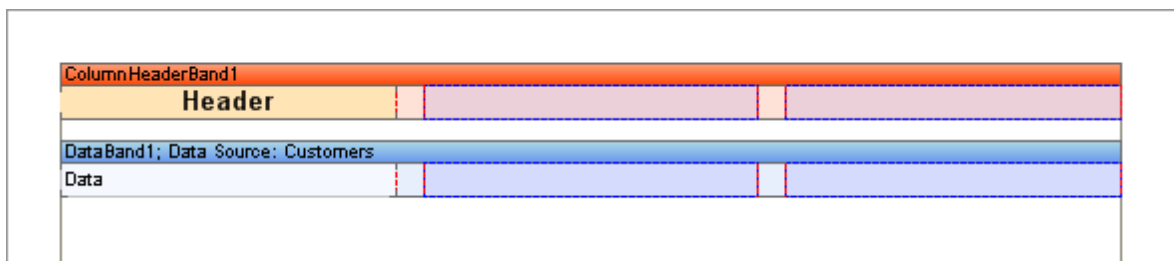
Now run the report and you will see that the word "Header" is shown over every column. You need only create a single column header and it will be automatically printed on each column.

3.19.2.4.1 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.
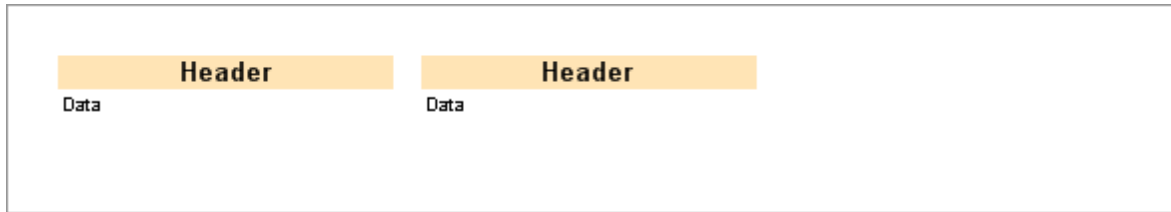
If you want to ensure that the same number of column headers are shown as the number of columns on a page without considering the number of strings available you can use the **PrintIfEmpty** property of the **Column Header** band. If you set this property to true, then one header will be output for each 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.

3.19.2.5 Column Footer Band

The Footer band is normally used to output data footers, but there is also a special **Column Footer** band. The Footer band is output once after the Data band and contains only one set of data. The **Column Footer** 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 **Column Footer** band. Put two bands on a page: A **Column Footer** 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 **Column Footer** 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.
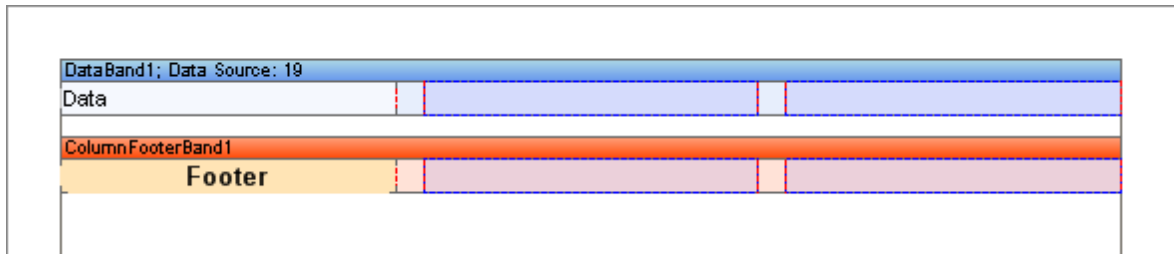
Now run the report and you will see that the word "Footer" is shown under every column. You need only create a single column footer and it will be automatically printed on each column.
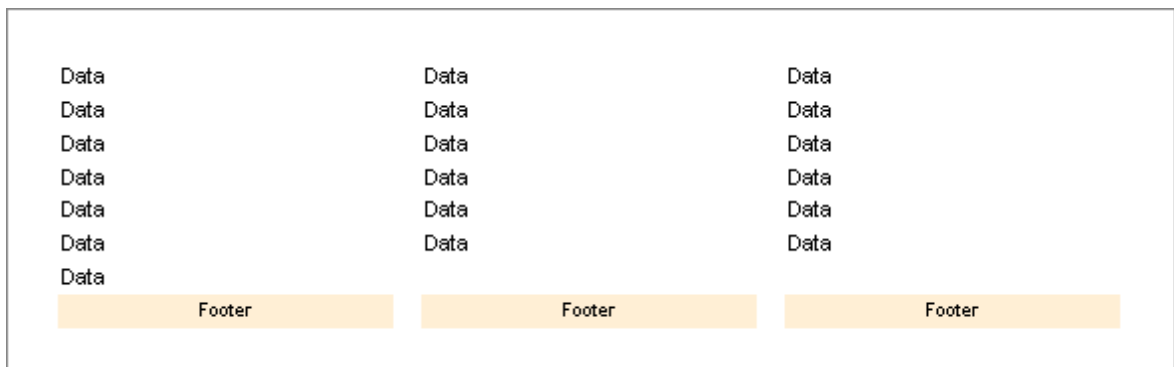
3.19.2.5.1 PrintIfEmpty Property

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 **Column Footer** 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.

3.19.2.6 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, **Column Header** bands, **Column Footer** bands and just one **Data** band at a random order.
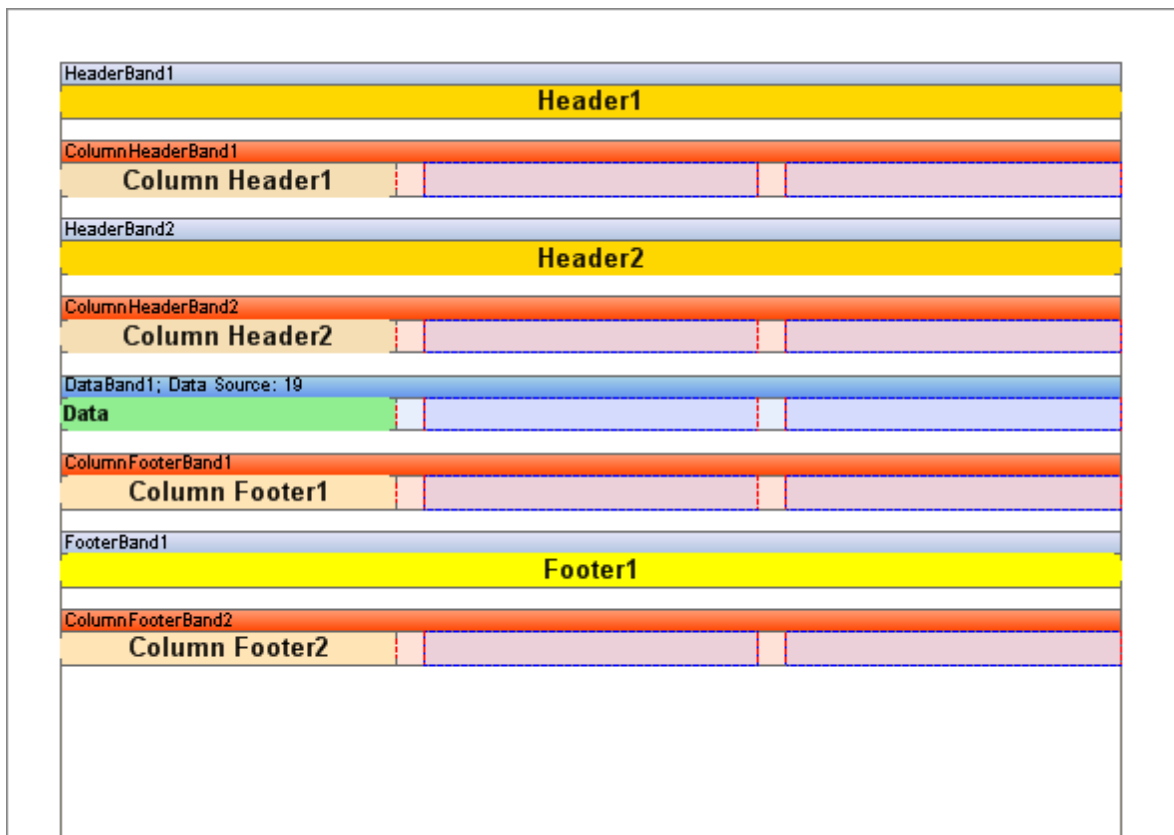
There are two modes used to output columns which will affect the output, and these will be reviewed in the following topics.

3.19.2.6.1 AcrossThenDow n Column Mode

In the **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 Column Footers 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.

3.19.2.6.2 Dow nThenAcross Column Mode

This mode is similar to the **AcrossThenDown** 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 Column Footer bands are output and the Footer bands are ignored.

3.20 Page and Column Break

Sometimes it is necessary at some moment to start rendering a report on a new page. This phenomenon in Stimulsoft Reports is called **page break**. Page break can be performed using the following properties: **NewPageBefore**, **NewPageAfter**,**NewColumnBefore**, **NewColumnAfter**. These features provide the ability to generate a new page/column before or after a certain band. This feature is similar to the page break in Microsoft Word.

3.20.1 Page Break

NewPageBefore property
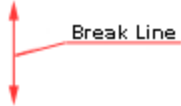
To break and insert a new page before a certain band you can use the **NewPageBefore** property. If the property is set to **false** for the band, then the report generator reaching this band will output it after the previous band without generating a new page. The picture below shows the **Footer** band that is output immediately after the **DataBand**:

If the **NewPageBefore** property is set to **true**, then the report generator at the time of the rendering a certain band, will make a gap (so that the band will be output on a new page), and on the previous page data output will be finished, despite the availability of free space on the page. The picture below shows, the **Footer** band which the **NewPageBefore** property is set to **true**:
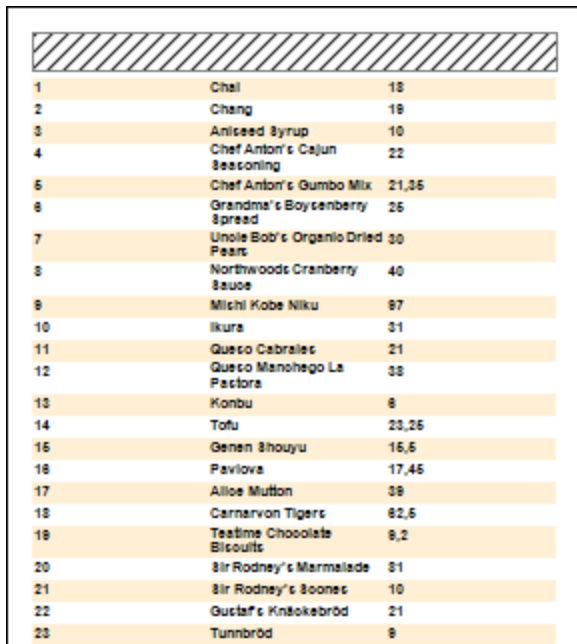
It is necessary to consider that the new page first displays all service bands (Page Header Band, Page Footer Band, Header Band). Also, when rendering a new page, the report generator will take into account the value of the following properties:**Break if Less Than** and **Skip First**.
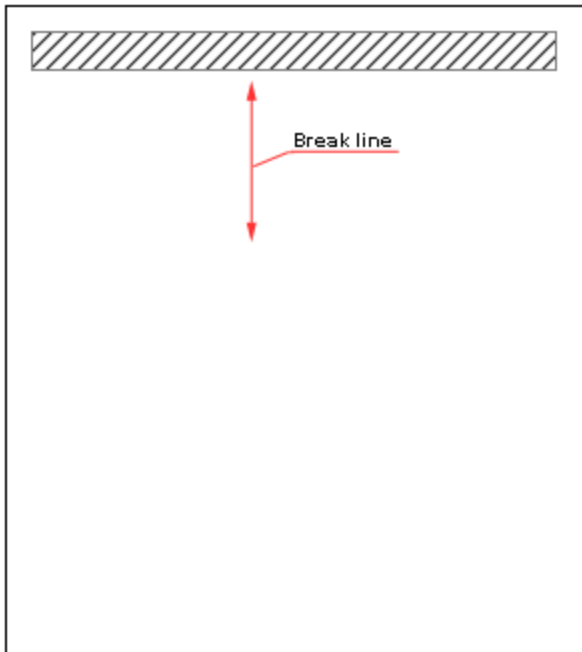
NewPageAfter property

Also, you can create a break and insert a page after a certain band. This can be done with the **NewPageAfter** property. If this property is set to **false** for the band, then the report generator when comes to render it will not do the gap, and immediately after it the other bands will be built. The picture below shows, the Header band that is output before the Data band:

If the **NewPageAfter** property is set to **true**, then the report generator will render the band, which property will generate the new page. The next band, will be output on a new page. The picture below shows, the Header band which the **NewPageAfter** property is set to **true**:

Break line

3.20.2 Column Break

At the time of break one can only insert not only new pages but new columns. This can be done using the **NewColumnBefore** and **New Column After** properties. The logic of inserting new columns is the same as for the pages.

NewColumnBefore

To break and insert a column before a certain band you can use the **NewPageBefore** property. If the property is set to **false** for the band, then the report generator reaching this band will output it after the previous band without generating a new column.

In order to make the break, set the **NewColumnBefore** property to **true**. In this case, the report generator at the time of rendering the band, will output a new column and add it before this band. The picture below shows the **Data** band with the**NewColumnBefore** property set to **true**.

In this case, it is necessary to consider that the new first column displays all service bands (Page Header Band, Page Footer Band, Header Band). Also, the construction of a new column, the report generator will take into account the value of the following properties: **Break if Less Than** and **Skip First**.

NewColumnAfter property

Also, you may need to make a break and insert a new column after a certain band. This can be done with the **New Column After** property. If the **NewColumnAfter** property is set to **false**, then all the bands will be displayed one after another.

In order to insert a new column the **NewColumnAfter** property should be set to **true**, after rendering the band, the report generator output a new column after this band. The picture below shows the Data band with the **NewColumnAfter** property set to **true**.

3.20.3 Break if Less Than Property

The **Break if Less Than** property can be any number value from 0 to 100. The value of this property affects where it will generate a new page or column. At the time of the report creation, report generator measures the amount of free space on the page as a percentage. If the entire page is empty, it is equivalent to 100 per cent, if the page is full - 0 percent. It should be considered that by default the **Break if Less Than** property is set to 0, which means that this option is disabled. Therefore, if the value of this property is 0, the report writer compares the percentage of free space on the page with the specified property value. As a result, the report generator will generate a new page or column, if the free space on the page is less than a predetermined value. The property value is a key value, i.e **Break if Less Than** property cannot be empty, it must take a value.

3.20.4 Skip First Property

The **Skip First** property works only with the **NewPageBefore** and **NewColumnBefore** property. With this property, the first entry from the database will be output on the page and then the page break will be executed, i.e. the first item will be skipped. For this, the **Skip First** property should be set to **true**. If it is set to **false**, a blank page (column) will be generated right after the band.

3.21 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}

3.21.1 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 **Page Header** band. System variable can be used in any text component. The text component can be placed on any page band.

3.21.2 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**.

3.21.3 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.

3.21.4   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**.

3.21.5 Sequentially Numbered Pages

Sequential numbering (numbering without taking into account the **ResetPageNumber** property) set the **SystemVariables**:

➤ {PageNumberThrough} - **PageNumberThrough**, displays the page number;

➤ {TotalPageCountThrough} - **TotalPageCountThrough**, displays the total number of pages of the rendered report;

➤ {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.

3.22 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.

3.22.1 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.
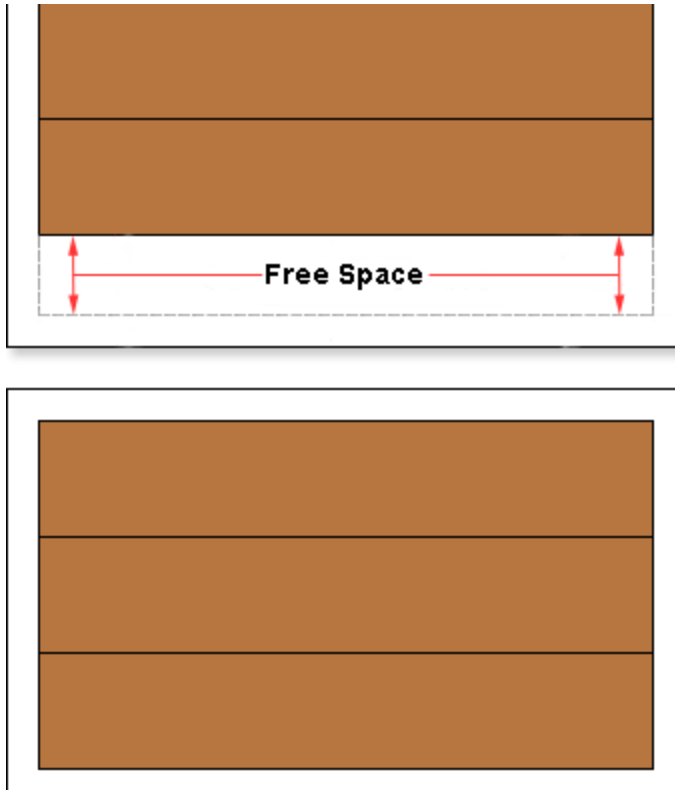
In this case, free space available remained on the first page of the report, because the band could not fit entirely and was moved to with the report engine to the next page. If to set the **CanBreak** property to **true**, then this will be "broken. The picture below shows how the of the third band is broken.
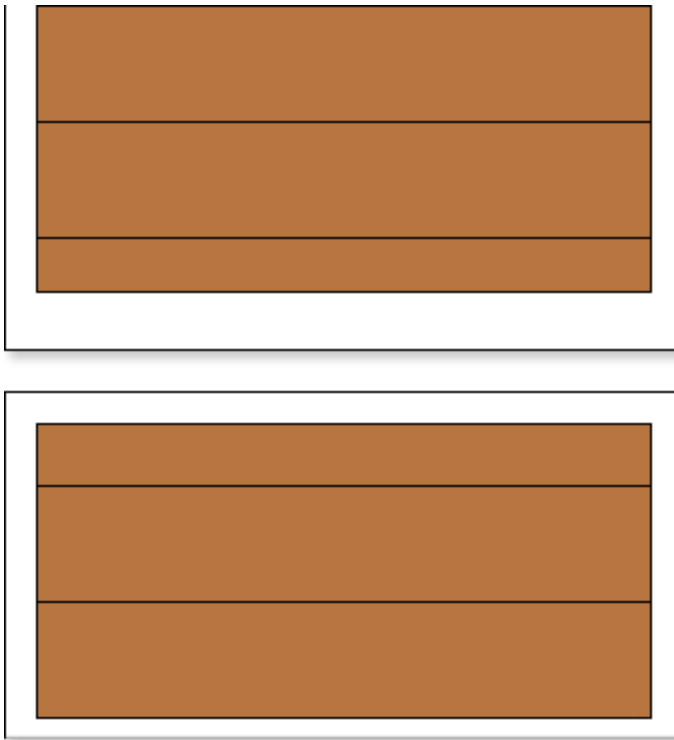
In this case we see that the third band could not fit, so it was broken: one part was left on the first page, and the second was moved to the next page, respectively. So all the space of the page was used. It should also take into account that the band may not fit within a single page. If the **CanBreak** 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**.

3.22.2 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.

3.22.3 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.
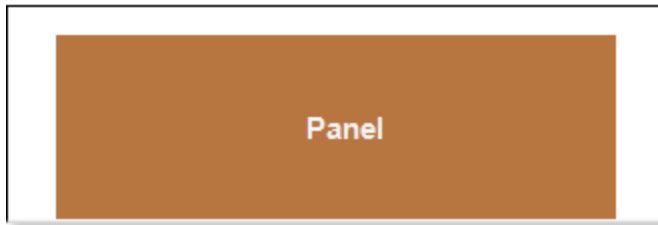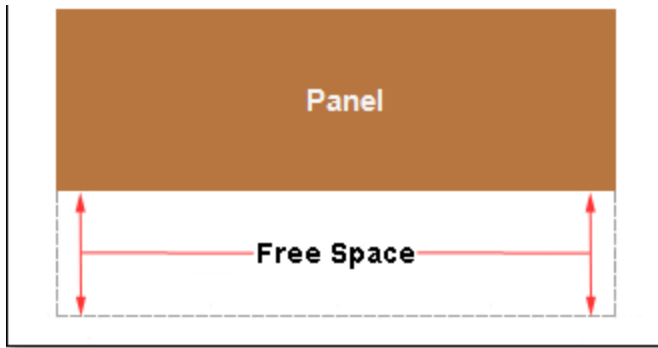
As shown in the picture above, the **Panel** was moved to another page, and free blank space remained on the previous page. If the **CanBreak** property is set to **true**, then the report may look like on the picture below:
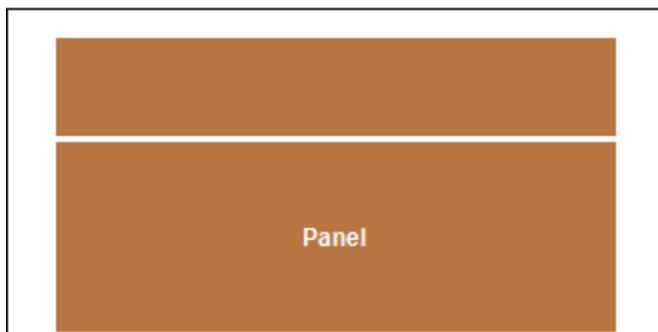
As shown in the picture above, the **Panel** was broken, i.e. a part of it remained on the first page, and the other was moved to the next page. It should also take into account that the panel may not fit a single page. If to set the **CanBreak** property to **false**, then it

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**.

3.22.4 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**.

3.22.5 Breaking Images

In some cases the **Image** does not fit one page. So the image will be moved to the next page.
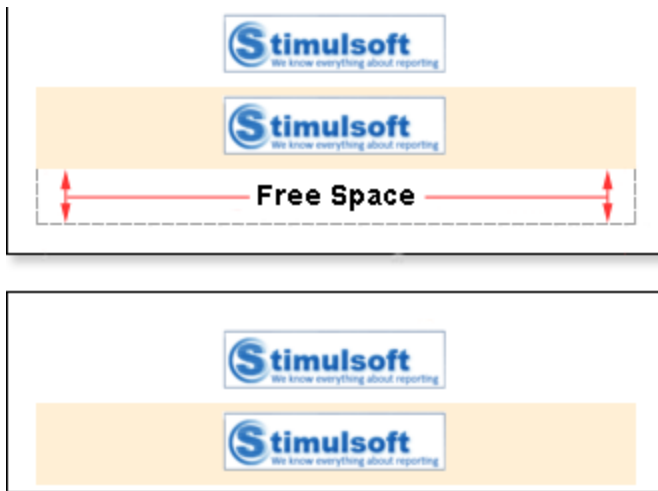
As you can see on the picture above, free space remained on the first page. To avoid this, set the **CanBreak** property to **true**. And then, the **Image** component will be broken, as seen on the picture below:

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**.

3.22.6 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.

3.22.7 Breaking and Page Bands

There is no possibility for the **Page Header** and **PageFooter** bands 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).

3.23 Hierarchical Band

The **Hierarchical** band is used to display report data as a tree. The picture below shows an example of a hierarchical report:

3.23.1 Data Output

In order to obtain a structured list in a report as a tree, you must follow these steps:

✓ Specify the **DataSource** for the **Hierarchical** band using, for example, the **DataSource** property:

Data Source   [Data.Employees]   [...]

✓ 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;

✓ 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;

✓ 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);

✓ 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

3.23.2 KeyDataColumn Property

The **Hierarchical** band 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 **Hierarchical** band of the data source, which entries will be keys for creating a report. For example, if the **Employees** data source is specified to the **Hierarchical** band, 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.

3.23.3 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 **Hierarchical** band 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 **Hierarchical** band, 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:
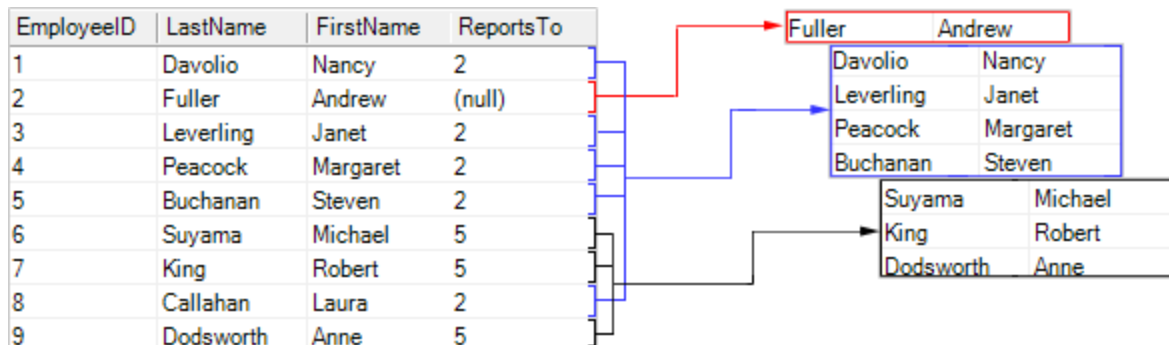
3.23.4 ParentValue Property

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:

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:

3.23.5   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:

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 **Hierarchical** band, 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.

3.24   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 Group Header 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

3.24.1 Multi Line Header

The **Child** band is a band that is a continuation of the band, after which it is placed.
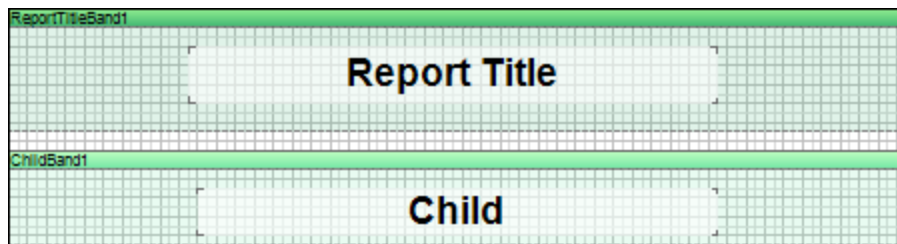
In the picture above shows the **Child** band is placed after the **Report Title** band, respectively, it is a continuation of this **Report Title** band. There are no limitations on

the number of **Child** bands placed on a page.
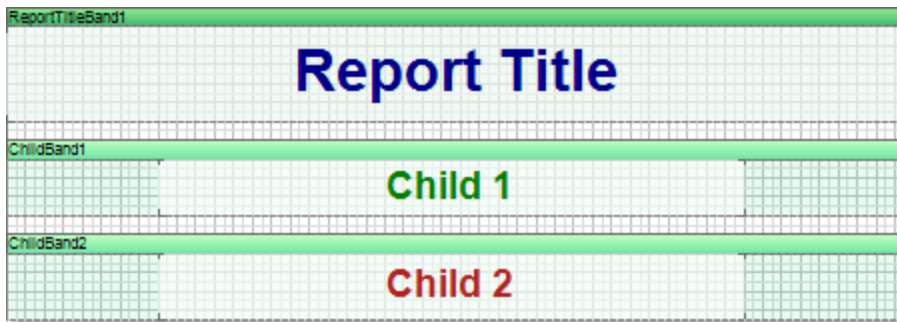
The picture above shows two **Child** band, which are a continuation of the **Report Title** band. Suppose there is a report with the report title that consists of a few lines. If the text is placed on the **Report Title** band, then visually it may look not entirely correct:
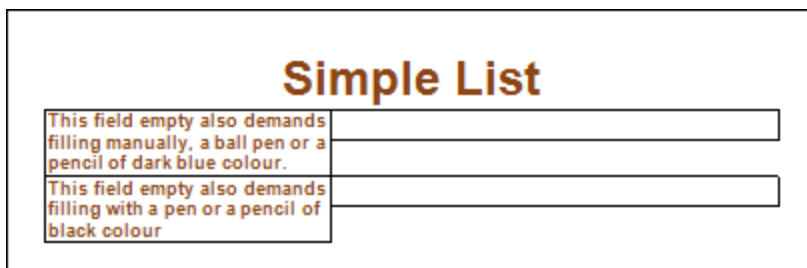
Even when using the **GrowToHeight** property, then visually it cannot be convenient:

Therefore, in some cases, the title of the report is better represent with the **Child** band:
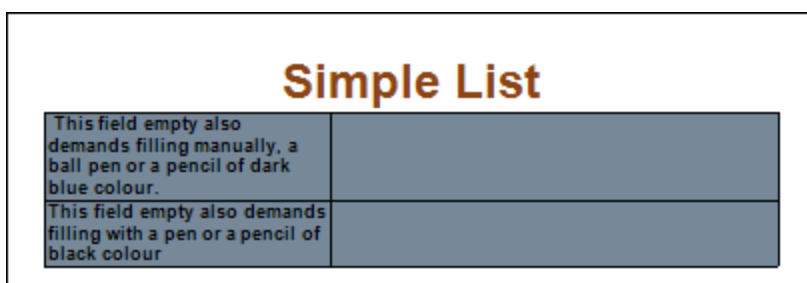
The picture below shows the report title located in the **ReportTitle** band and two **Child** band.
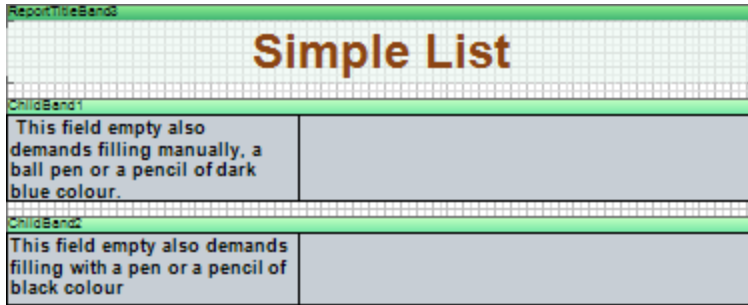
3.24.2   Child Band and Data

How to output two bands on one data row? You can use the **Child** band. Create a new report. Put the **Data** band on a page. Put the **Child** band under the **Data** band.
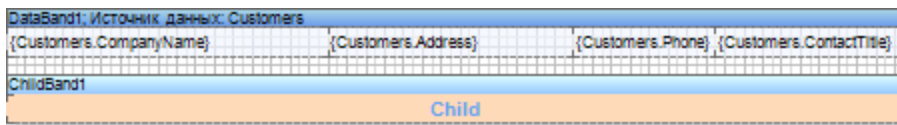
Run a report for execution. As you can see, the **Child** band was printed as many times as the **Data** band. I.e. the **Child** band is a continuation of the **Data** band. But at the same time it remained to be a band, with all its properties.

The **Child** band can be used not only with the **Data** band. It can be placed after any band on a page. For example, after the **Header** band or after the **Group Header** band.

❗ The **Child** band can be used in association with any band.

3.24.3   KeepChildTogether Property

For example, add the **Child** band to the **Data** band, as the result a data row and an empty row (**Child** band 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 **Child** band, for example **Country.**
The picture below shows that instead of empty space, the country name will be output.

Centro comercial Moctezuma	Sierras de Granada 9993	(5) 555-3392	Marketing Manager
Mexico			
Chop-suey Chinese	Hauptstr. 29	0452-076545	Owner
Switzerland			
Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647	Sales Associate

Company	Address	Phone	Contact
Brazil			
Consolidated Holdings	Berkeley Gardens 12 Brewery	(171) 555-2282	Sales Representative
UK			
Drachenblut Delikatessen	Walserweg 21	0241-039123	Order Administrator
Germany			

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 **Child** band has the **KeepChildTogether** property.

By default the property is set to **true**.

3.25   Empty Band

The **Empty Data** 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 **Empty Data** band:

Adding an **Empty Data** 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 **Empty Data** 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:

❗ **Note:** This band is not working on the Panel and Sub-Report.

3.25.1 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.

3.26 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.

3.26.1 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.

On the table below Text properties for watermark are described.

Properties	Description
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.

Properties	Description
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.

3.26.2 Overlay Band

The **Overlay** band is used to output text, images, primitives and other data.

The **Overlay** band 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 **Overlay** band 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 **Overlay** band 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.**

3.26.2.1 Vertical Alignment Property

The **VerticalAlignment** property is used to define the place of the "watermark" inscription which is output using the **Overlay** band. This property may have three values:

▶ **Top**. The **Overlay** band will be output on the top of a rendered report before the page header and the page header.
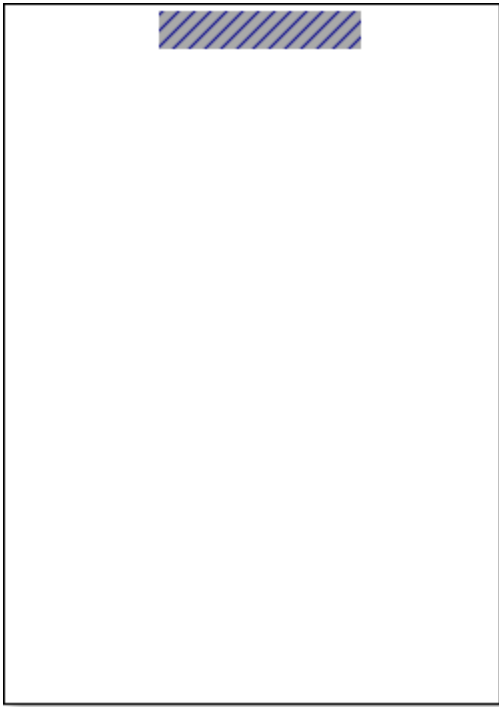
❯ **Center**. The **Overlay** band will be output on the center of a rendered report and in front of data placed on the page.

❯ **Bottom**. The **Overlay** band will be output on the bottom of a page of a report and

after the page footer.
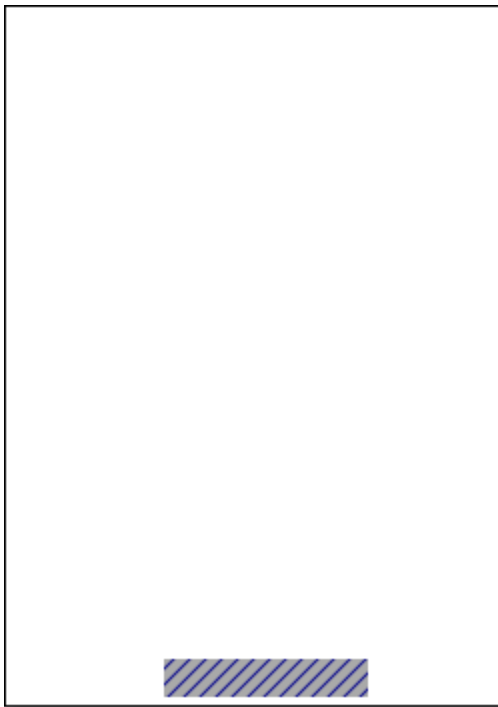
3.26.3 Direct Allocation on Page

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.

3.27 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.
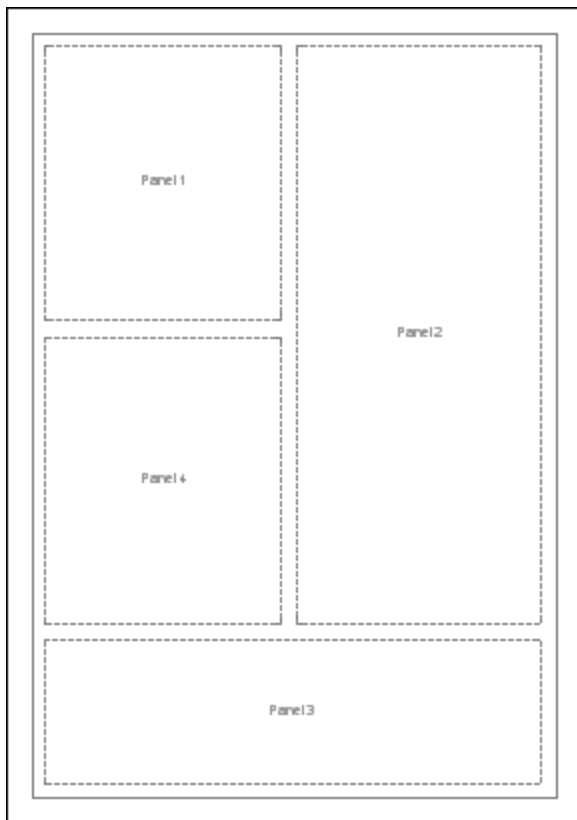
3.27.1 Placing Bands on Panel

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
Report Title	No
Report Summary	No
Page Header	No
Page Footer	No
Group Header	Yes
Group Footer	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.

3.27.2 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.

3.27.2.1 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.
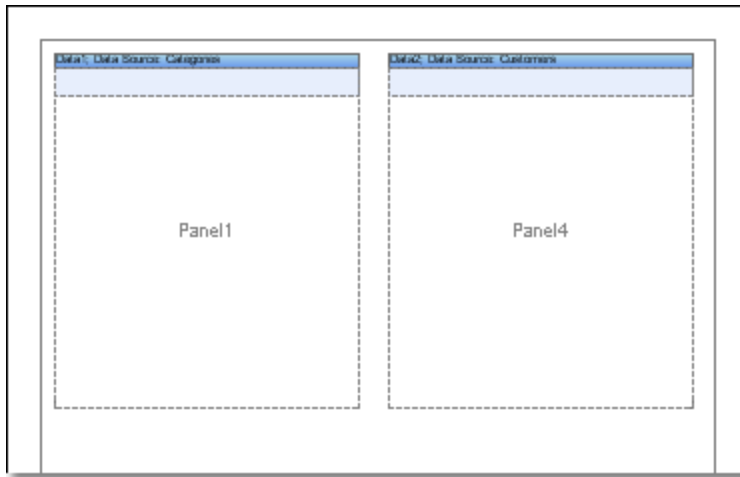
❗ **Notice:** Number of panels on one page is unlimited.

3.27.2.2 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".
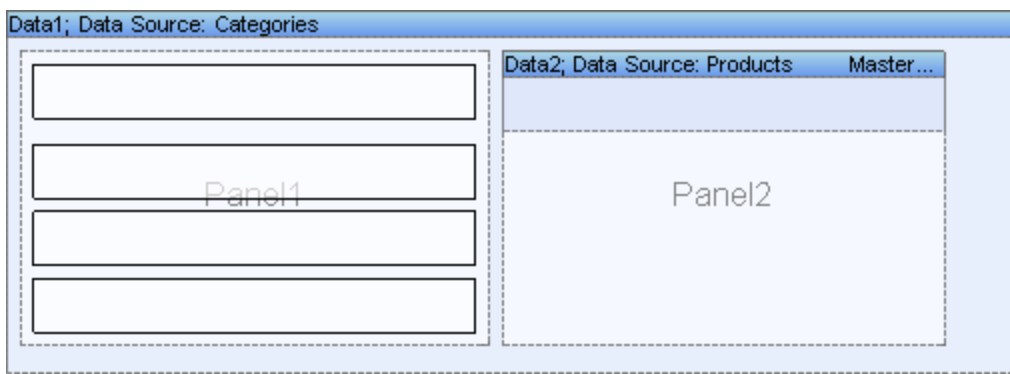
3.27.2.3 Placing Panels on Panel

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.
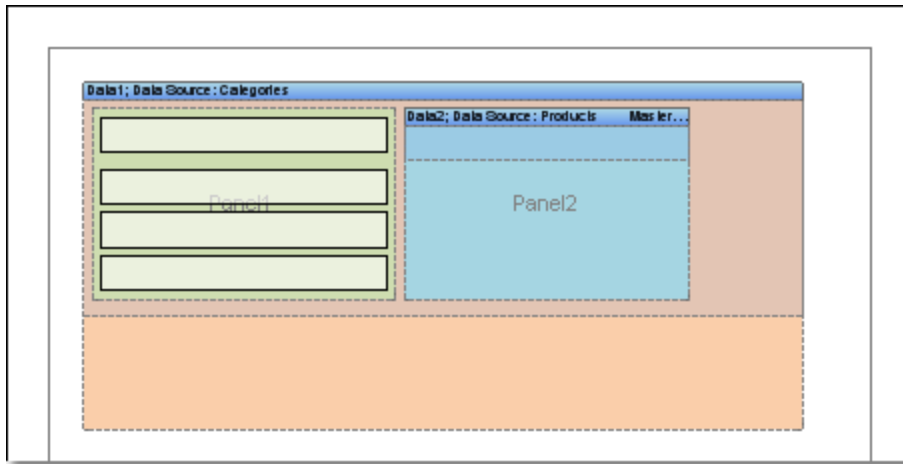
3.27.3 Side-by-Side Reports

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.
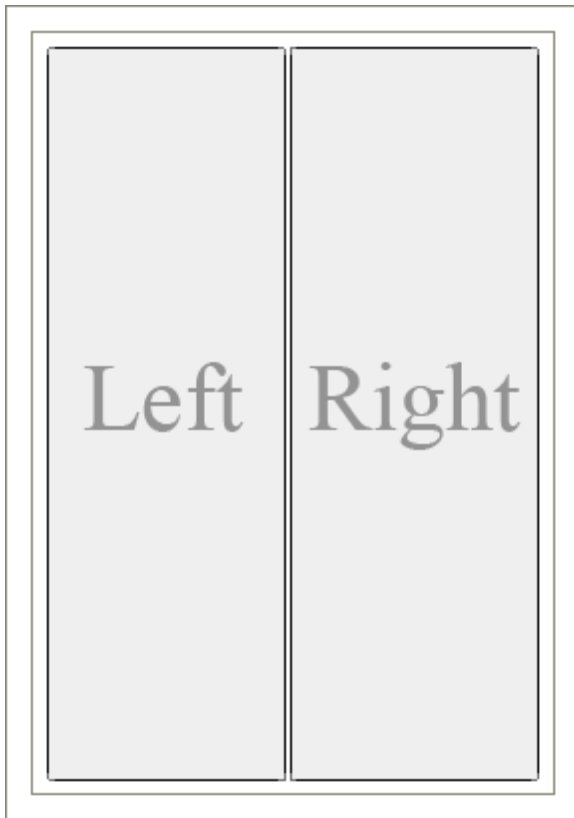
How to build a **Side-by-Side** report. Put two containers on a page. Set the **DockStyle** property of one component to **Left**. Set the **DockStyle** property of the second component to **Right**. Docking component is necessary to take all space on a page by the height. In cases it should not be done. 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.

3.27.4 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.

3.27.5 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.
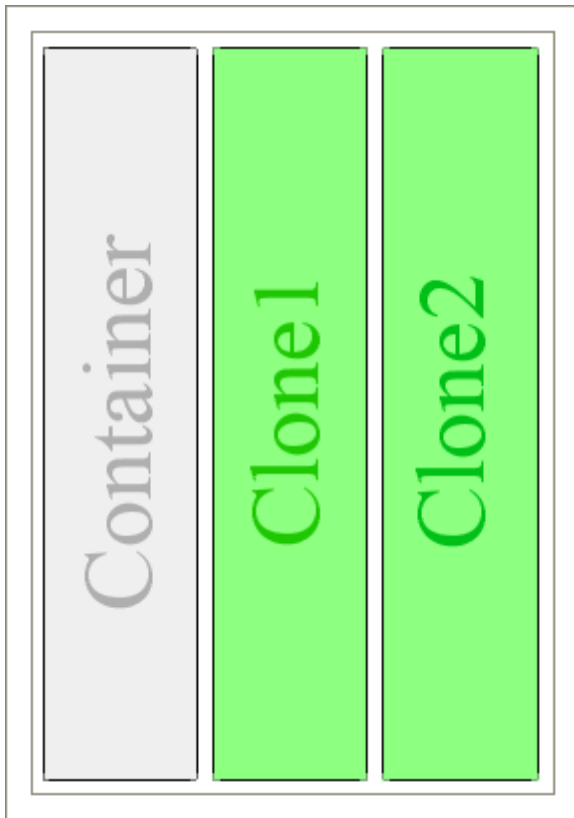
Run a report. The panel will be rendered first. The list will be output on the left side of a page. Then the list will be continued to output on the place where the **Clone** component is placed. The **Clone** component clones all bands of the panel. Using the **Clone** component it is possible to render 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.

3.28 Cross-Tab

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.

3.28.1 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.

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.
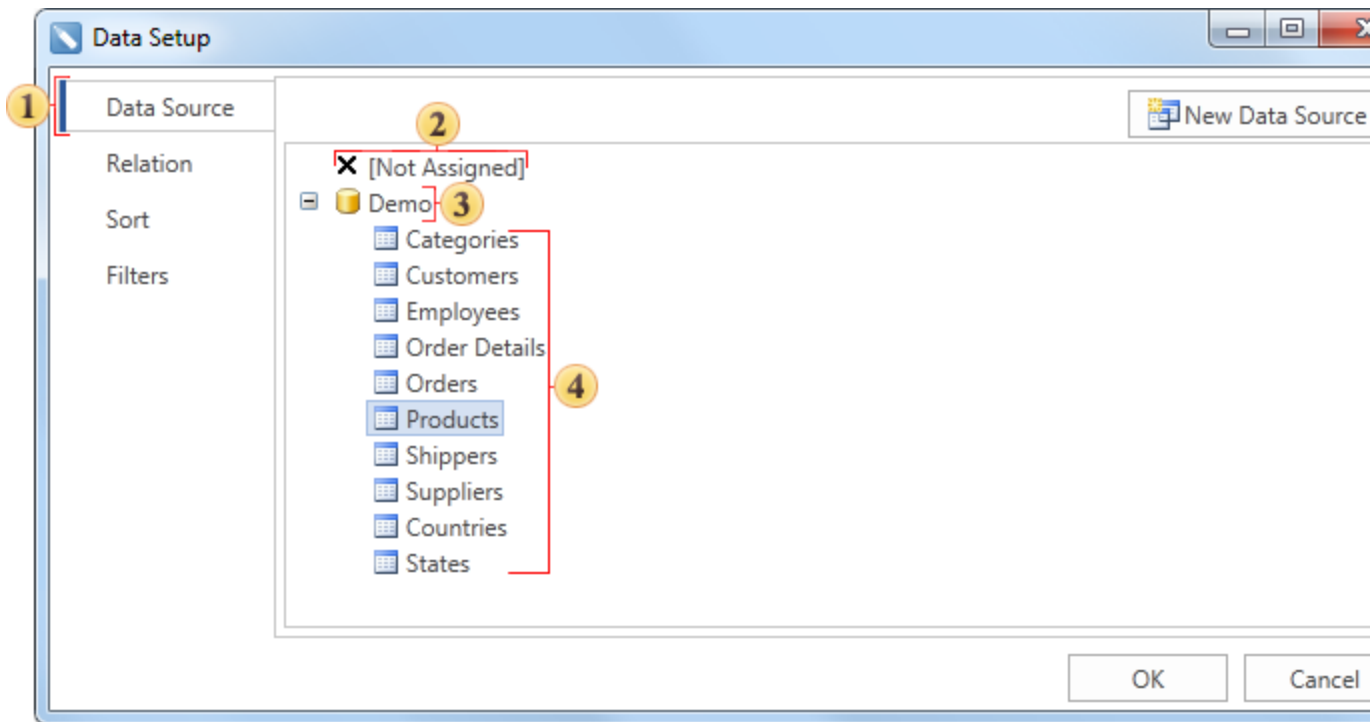
① The tab to select the data source;
② Select this node if you do not need to specify the data source;
③ The "Demo" data category;
④ The "Demo" data source category.

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.
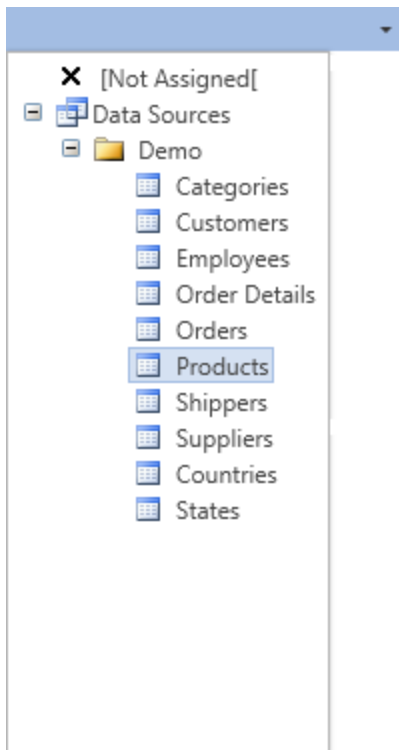
3.28.2 Cross Table Items

After selecting the data source you need to specify the following items: columns, rows, and cells for summation.

3.28.2.1 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:

Columns:

CategoryName

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:

3.28.2.2 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

Grouping is done only by its values for one row:

Rows:

Country

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:

Rows:

Country
City

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

3.28.2.3 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.

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:

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.

3.28.3 Cross-Tab Editor

When you create or edit the **Cross-Tab** component, a special editor will be called when editing the component. The editor tabs - **Data**, **Cross-Tab**, **Styles** - contain the configuration settings of the Cross-Tab component. In addition, the settings and parameters are grouped on each tab.
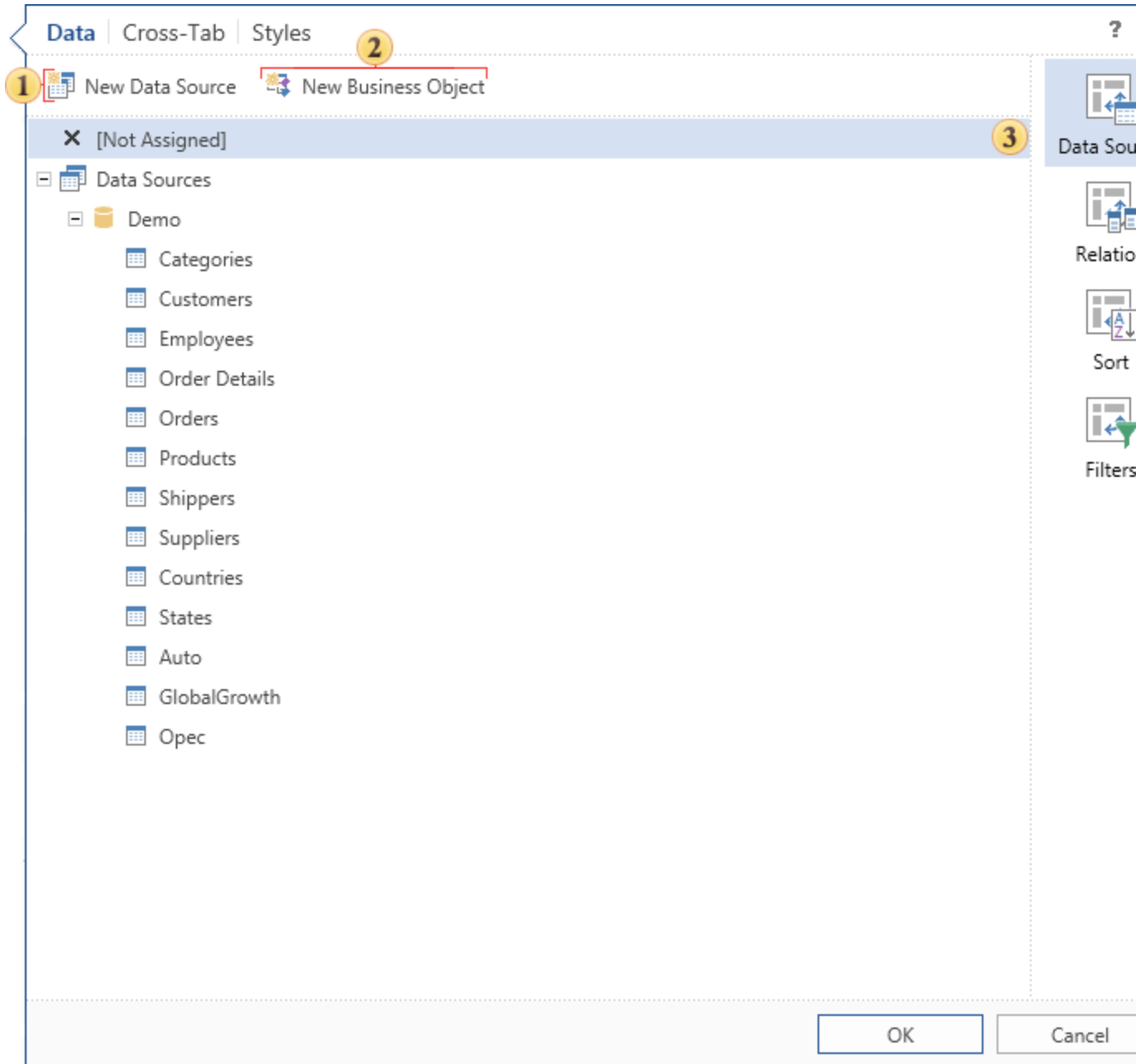
1️⃣ The **New Data Source** button. Calls the window to create a new data source.

2️⃣ The **New Business Object** button. Calls the window to create a new Business Object.

3️⃣ In this field you can find settings and parameters. The picture above shows the selected group **Data Source**. The filed shows all available data sources. Select the data source that will be used when creating the cross-tab.

4️⃣ The list of parameters and settings for the active tab.

As seen from the picture above, in the Data tab, and all settings are divided into the

following groups:

> **Data Source**

In this group, you can select the data source for the cross-tab. In addition, there are buttons to create a new data source and new Business Object.

> **Relation**

In this group, you can set the relation between the selected sources. There is also a new button **New Relation**, when clicked, it calls the create new relation window.

> **Sort**

In this group, you can set the sorting parameters. You need to set the data column by which sorting will be done and the direction of sorting.

> **Filters**

In this group, filtering parameters are determined. A new filter is added and filtering criteria through the expression or value is specified.

3.28.3.1 Cross-Tab Tab

The Cross-Tab tab defines the structure of the Cross-Tab component. It specifies the data column for the rows, columns, total cells:

1 The data source that will be used to build a cross-tab.
2 A list of data columns that will form the cross-tab row.
3 The button to delete the selected item from the field **Rows**, **Columns**, **Summary**.
4 If more than one element was added into the fields of the cross-tab (rows, columns, summary) then the buttons will be available to move the selected item in the list.
5 The reverse button between the columns and rows. Every press of the button changes the contents of the string field on the contents of the column field.
6 The list of data columns that will form the cross-tab column.
7 The list of data columns that will create the summery of a cross-tab.

⑧ Displays the preview of the cross-tab.

3.28.3.2 Styles Tab

The final stage in the creation of a cross-tab is to define its style:

① The button to call the style designer.
② The list of predefined styles available by default. If you need your own style, you need

to call the style designer and create a new one. In order to select the style you need, you simply select it. In this case the preview pane will show the structure of a cross-tab with the style preview.

❸ The Preview panel. A red box around the cell indicates that the cell is selected.

3.28.4 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.

3.28.5 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**

3.28.6 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).
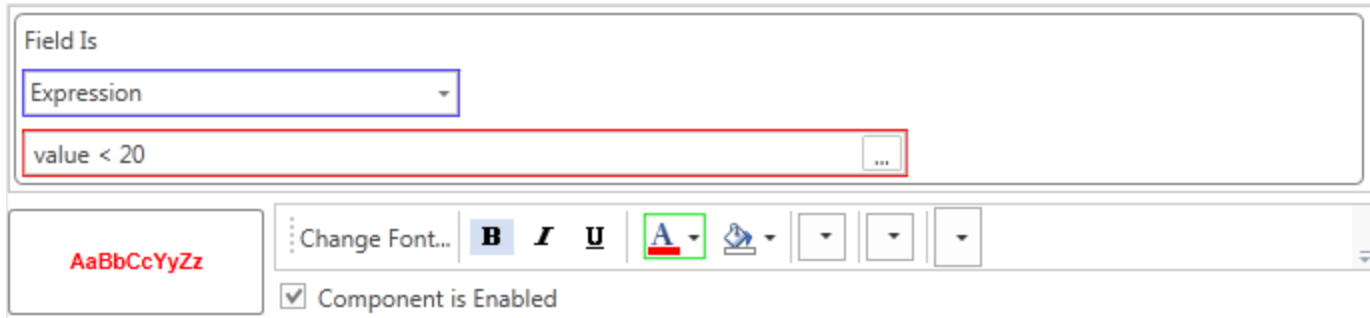
Change the value of the Field Is field on the Expression (marked with blue). Specify the required expression (marked with red):

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

3.28.7 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:

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**:

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.

3.28.8 Processing Values for Summary

The **Cross-Tab** 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

3.28.9 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**:

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**:

By default, the **WordWrap** property of cross-table components is set to **false**.

3.29 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.
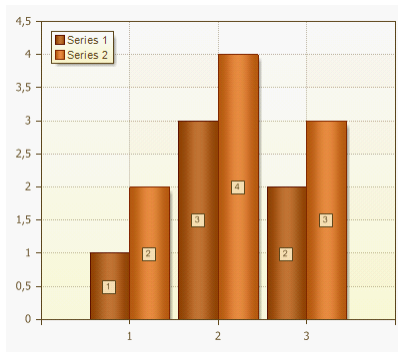
Stacked Column

Stacked column charts show the relationship of individual items to the whole, comparing the contribution of each value to a total across categories.
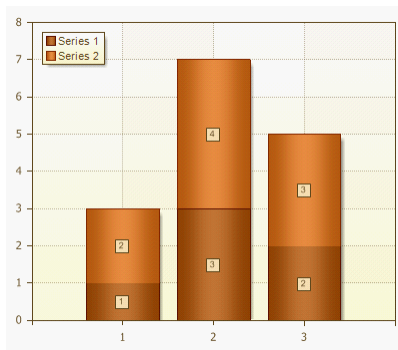
Full-Stacked Column

Full-Stacked column allows comparing the percentage of each value.
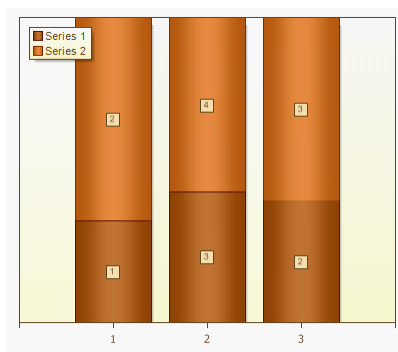
Bar Area

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.
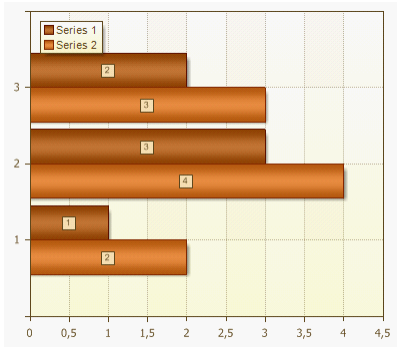
Stacked Bar

Stacked bar charts show the relationship of individual items to the whole.
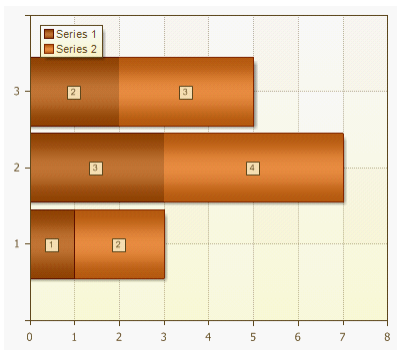
Full-Stacked Bar

This type of charts allows comparing percentage of each value with the total inside the category.
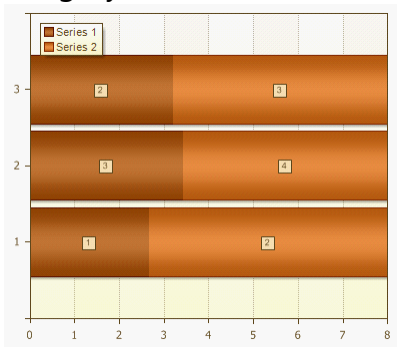
Pie Area

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.
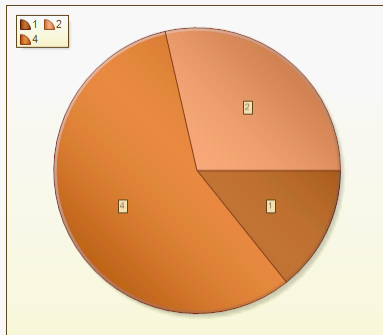
Doughnut

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.
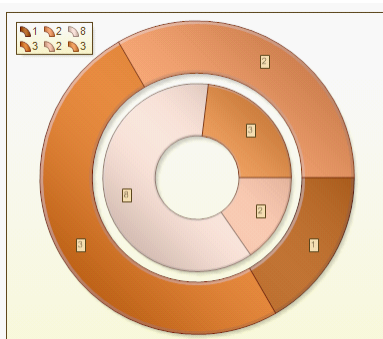
Line Area

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.
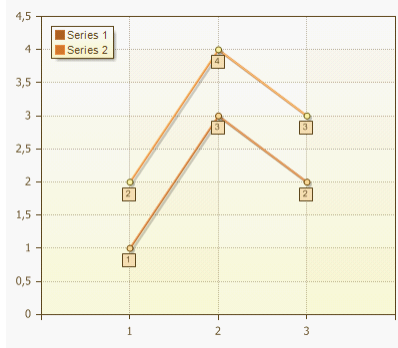
Stepped Line
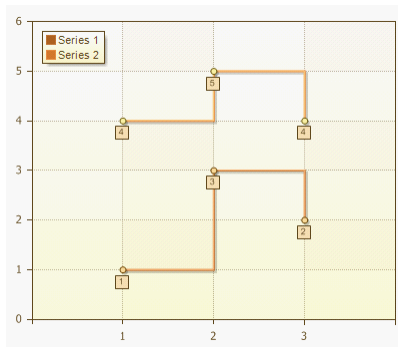
Stacked Line

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.
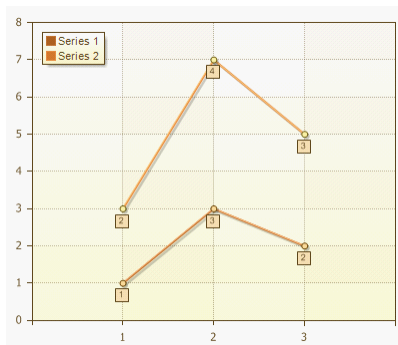
Full-Stacked Line
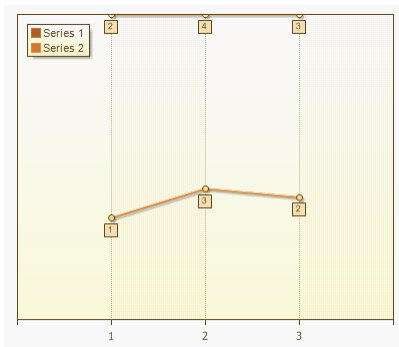
Spline Area
Spline
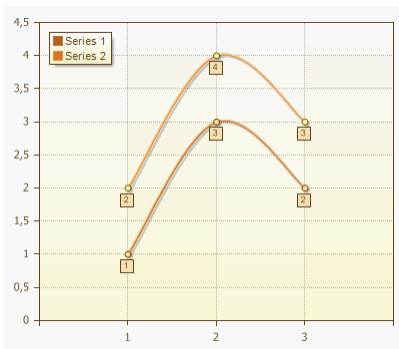
Stacked Spline
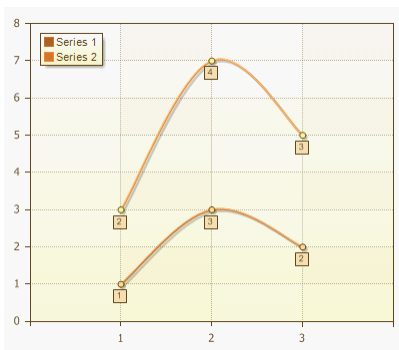
Full-Stacked Spline
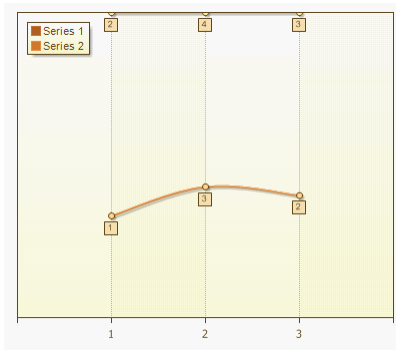
Area

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.
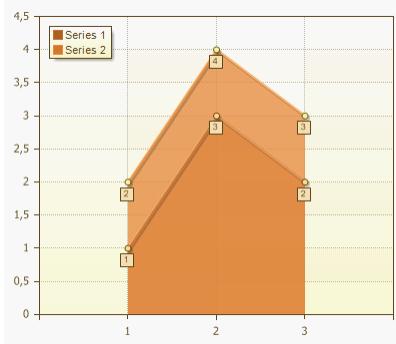
Stepped Area
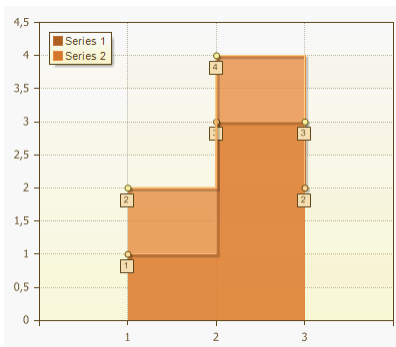
Stacked Area

Stacked area charts display the trend of the contribution of each value over time or categories.
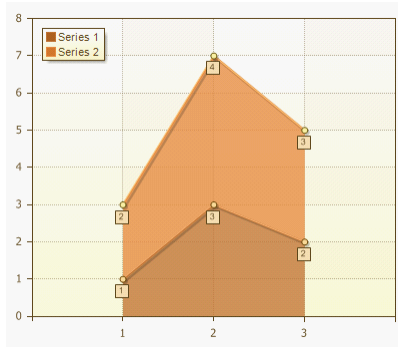
Full-Stacked Area

Full-Stacked Area charts display the trend of the percentage each value contributes over time or categories.
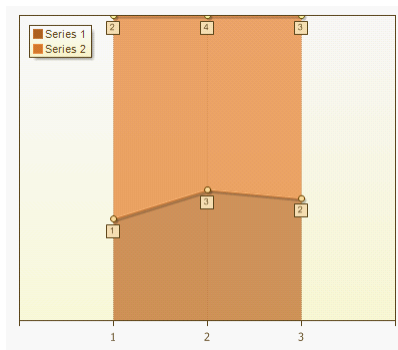
Spline Area

Spline Area
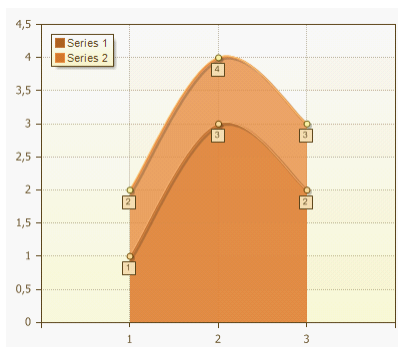
Stacked Spline Area
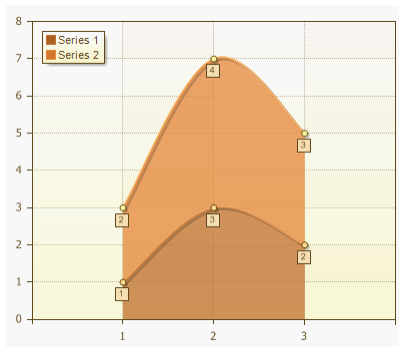
Full-Stacked Spline Area
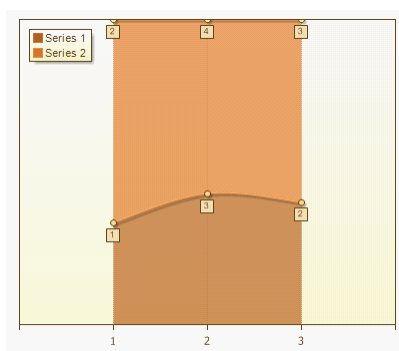
Range

The chart type **Range** can be used to display the interval of values per unit of time or period of time. To build such a diagram you should have start and end values.

Range
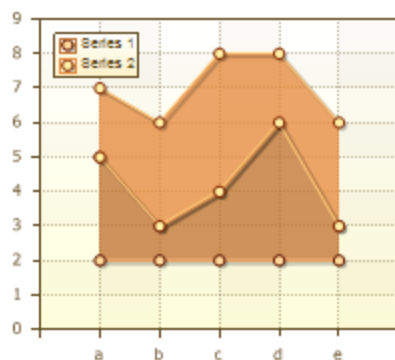
Stepped Range
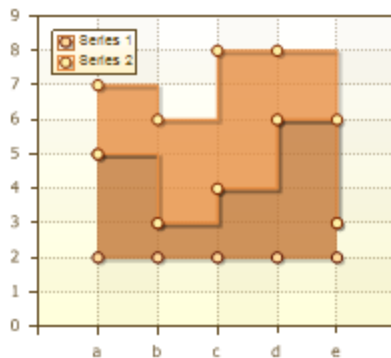
Gantt Area

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.
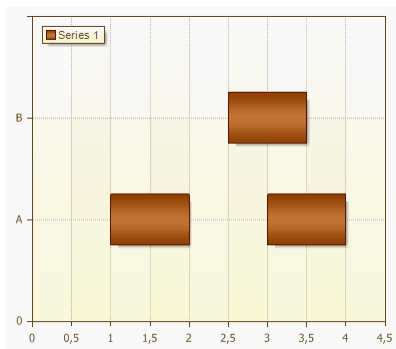
Scatter Area

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.
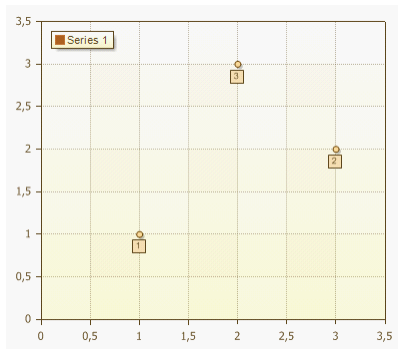
Scatter Line

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.
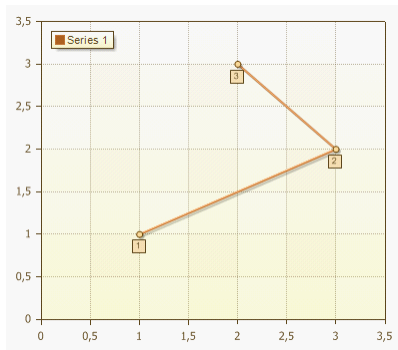
Scatter Spline

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.
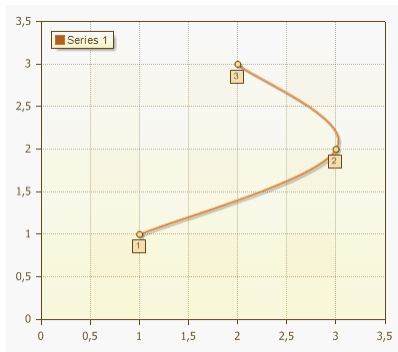
3.29.1 Charts Editor

When you add the component Chart in the report template, the chart editor is called. This editor is used to create the chart: defining the types of rows, data sources, styles, and other settings. A chart can be created using the wizard or manually. Below is a diagram editor.
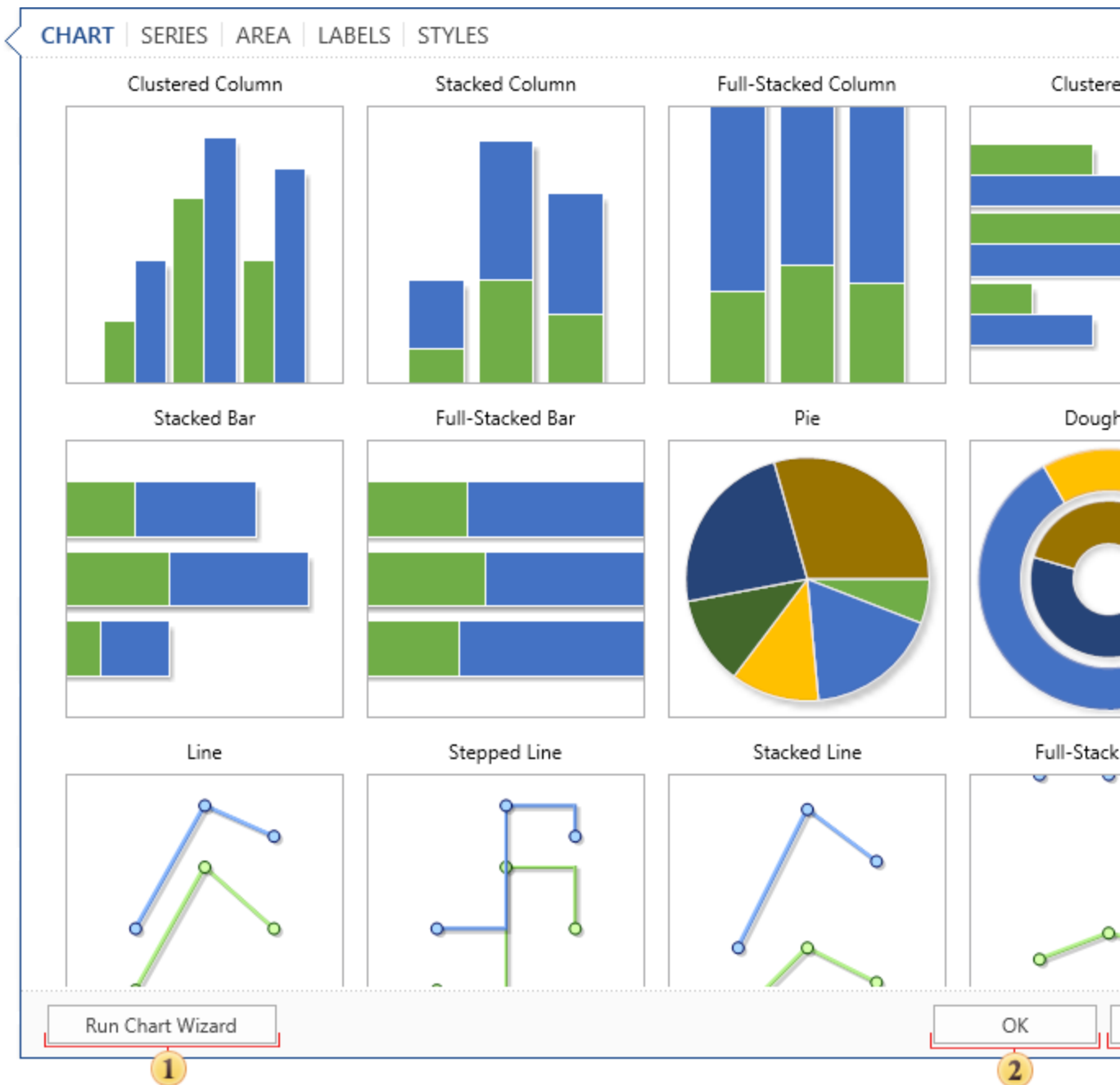
① The button **Run Chart Wizard**.
② When you press this button, a chart of a certain type with the specified parameters is created.
③ Pressing this button cancels the creation of a chart but the component remains is the report template.

As can be seen from the picture above, the chart editor contains the following tabs:
> **Chart**. Defined the Chart type;
> **Series**. Defines the parameters of the series;
> **Area**. Sets areas with axes;

> **Labels**. Sets chart labels;
> **Styles**. Sets the style for the chart.

3.29.1.1 Tab Chart

The tab **Chart** defines the parameters relating to the diagrams. These parameters are grouped depending on the selected group on the property panel.

① The **Preview window**. This panel displays the chart and immediately previews changes made in real time.

② All chart parameters are grouped. A list of these groups is represented on this panel. When a group

is selected, the Properties panel will display the parameters of the selected group:

> ❯ The group **Common**. Contains common settings such as a data source for the chart, the vertical/ horizontal alignment, rotation angle and others.

> ❯ The group **Legend**. Contains settings for the legend such as enabling/disabling it, alignment options, direction, etc.

> ❯ The group **Title**. Contains settings for the title of the chart such as text, alignment options, etc.

> ❯ The group **Constant Line**. Contains settings for constant lines. Moreover, in this parameter group involves adding a constant line in the chart.

> ❯ The group **Strips**. Contains settings to control strips in charts. You can add a new strip here.

> ❯ The group **Table**. Contains settings to display values as a table.

It should be noted that in some groups you can add elements to the chart. In this tab, this note concerns groups **Constant Lines** and **Strips**.
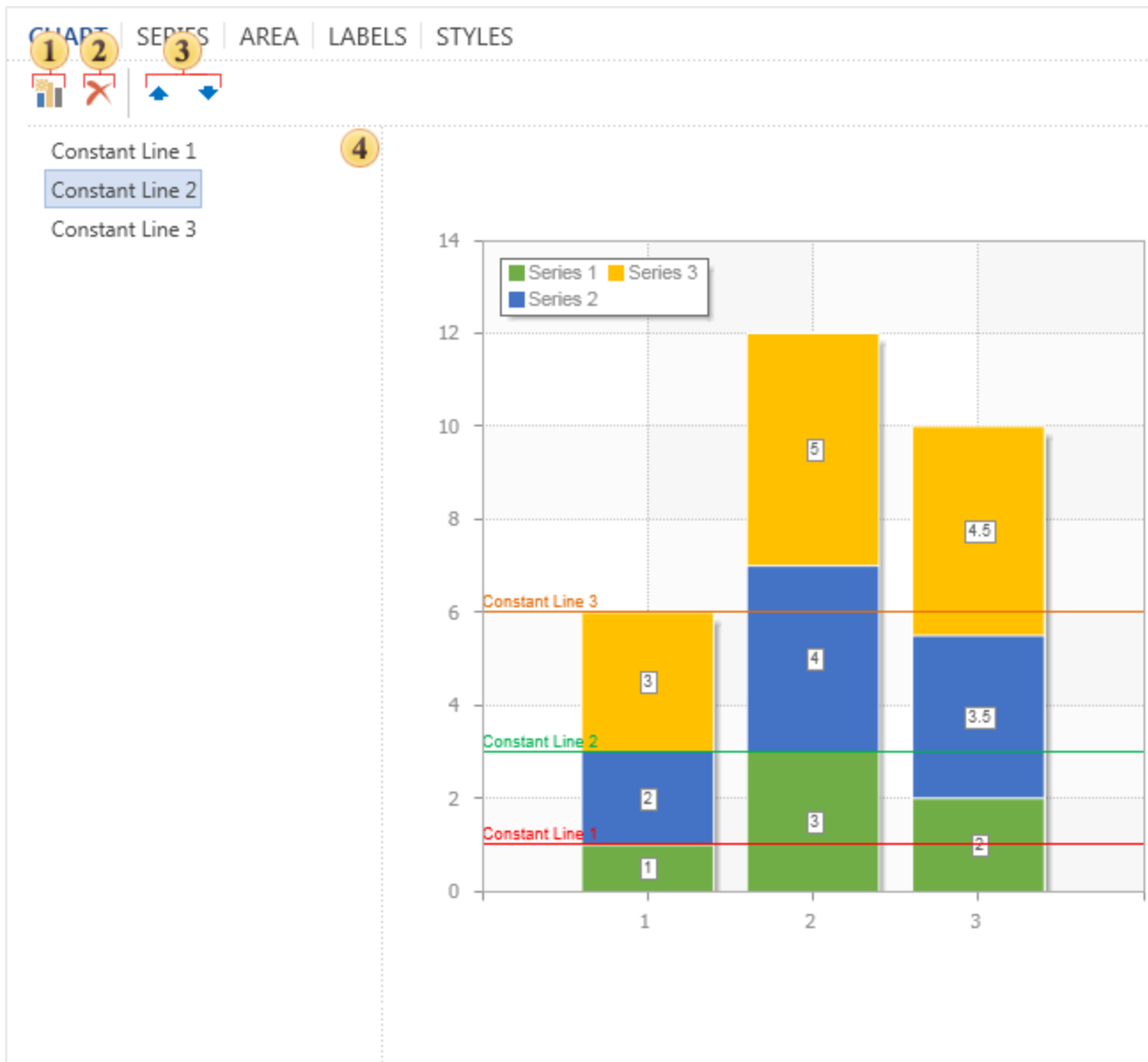
1 The button is used to add the constant line.
2 The button is used to erase the selected line.
3 The buttons move the selected item in the list on the panel **4**.
4 The panel with the list of items.

3.29.1.2 Tab Series

Series of the chart component are the main element of the diagram. Series are important to visualize data. It should be understood that construction is not possible without series of the diagram.
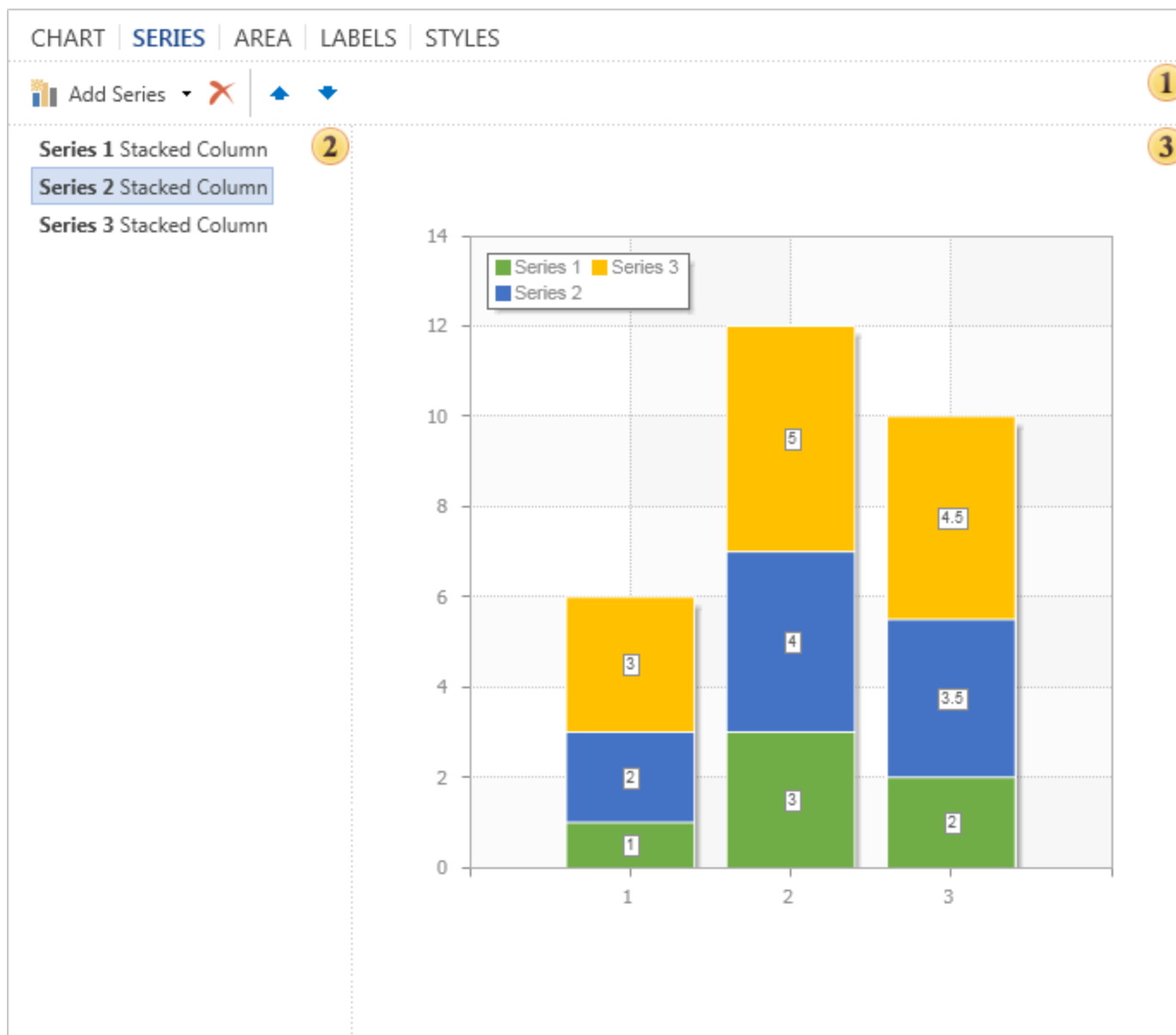
1 The toolbar contains the basic commands to control the chart series: adding series, deleting the selected one, moving the selected series in the list.

⚠ **NOTICE**: If the chart type is defined on the **Chart** tab, in the menu of adding rows, only series of this type will only be available, and those that can interact with the type of a chart. If the chart type is not specified, the type of a chart will depend on the selected series.

2 The list of chart series. As can be seen from the picture, this chart contains three rows.
3 The preview panel. This panel displays the chart and immediately previews changes made in real time.
4 The list of group of parameters of the tab Series:

> The group **Common**. You can find settings for the selected series. Among them are data source, data, etc.

> The group **Conditions**. Here you can set parameters for the selected series.

> The group **Filter**. Parameters of filtering of the selected series can be set here.

> The group **Interaction**. Here you can setup interaction of the series.

> The group **TopN**. In this group you can set maximum or minimum values.

> The group **Series Labels**. This group of parameters are used to define position, rotation for series labels etc.

> ℹ **Information**: Various displaying modes of series labels can be applied in the chart series - Show Series Labels: **From Series** and Show Series Labels: **From Chart**. The mode is defined in the tab Labels in the group of properties Series Labels.
>
> > If the mode **Show Series Labels**: From Series is enabled, then on the tab Series, the group Series Labels you should define the type of labels. In this mode, you can specify a particular type of labels for each series.
>
> > If the mode **Show Series Labels**: From Chart is enabled then the type of series labels will be the same as selected in the tab Labels. For example, you have a chart with 10 series in it and labels should have the same style, i.e. be of the same type. In this case, on the tab Labels, you can define the type and in settings of each series you can specify the mode Labels From Chart.

3.29.1.3 Tab Area

The **Area** is a space that includes the basic chart items: rendered data series, axes, chart title and legend. The management of this space is carried out on the tab **Area**, in the editor **Diagram**.

1 The panel **Preview**. This panel displays the chart and immediately previews changes made in real time.

2 The list of parameters groups in the tab Area:

> The group **Common**. The group contains settings such as rotation, horizontal, vertical display, border color etc.

> The group **X Axis**. The group contains settings for the X axis.

> The group **Y Axis**. The group contains settings for the Y axis.

> The group **X Top Axis**. The group contains settings for the X top axis .

> The group **Right Y-Axis**. The group contains settings for the right Y axis.

> The group **Grid Lines Hor**. The group contains settings for horizontal lines.

> The group **Grid Lines Vert**. The group contains settings for vertical lines.

> The group **Grid Lines Hor Right**. The group contains settings for right horizontal lines.

> The group **Interlacing Hor**. The group contains settings of alternation of horizontal cells in the

chart area.

➤ The group **Interlacing Vert**. The group contains settings of alternation of vertical cells in the chart area.

3.29.1.4 Tab Labels

On this tab you can set the type of labels in the chart. The selected appearance of the title will be applied to all rows that have the mode **Show Series Labels: From Series** disabled.

> **ℹ Information**: You can use a variety of modes of display labels. - Headlines from the series or title of the chart.
>
> ➤ If the mode **Show Series Labels**: From Series is enabled, then on the tab Series, the group Series Labels you should define the type of labels. In this mode, you can specify a particular type of labels for each series.
>
> ➤ If the mode **Show Series Labels**: From Chart is enabled then the type of series labels will be the same as selected in the tab Labels. For example, you have a chart with 10 series in it and labels should have the same style, i.e. be of the same type. In this case, on the tab Labels, you can define the type and in settings of each series you can specify the mode Labels From Chart.

The picture below shows the tab Labels.

1 This panel displays a list of different types of labels.

2 The preview panel. This panel displays the chart and immediately previews changes made in real time.

3 The list of groups of parameters:

> The group **Common**. You can find settings such as Text before, text after, rotation etc.

> The group **Conditions**. Here you can set parameters for the selected series.

3.29.1.5 Tab Styles

You can completely change the design of charts, ranging from basic colors and ending with shadows, borders, and so on. You can do this in the tab **Styles**.

1 The button is used to call the style designer. In the designer, you can create a style for the chart and the collection of styles for other components.

2 In this panel you can see the list of styles that are available by default.

3 The panel **Preview**. This panel displays the chart and immediately previews changes made in real time.

4 The button **More Styles**. When you click it you will see the list of styles available by default.

⚠ **Notice**: If the **AllowApplyStyle** is enabled then the style will be applied. If you disable the **AllowApplyStyle** then the parameters of series will be considered.

3.29.1.6 Wizard

The Chart wizard provides an opportunity to create a chart in a few simple steps. To start the wizard, you should the button Chart Wizard in the chart editor. The wizard provides a step-by-step procedure to create a chart. By default, the first type (Clustered Column) is selected in the list.
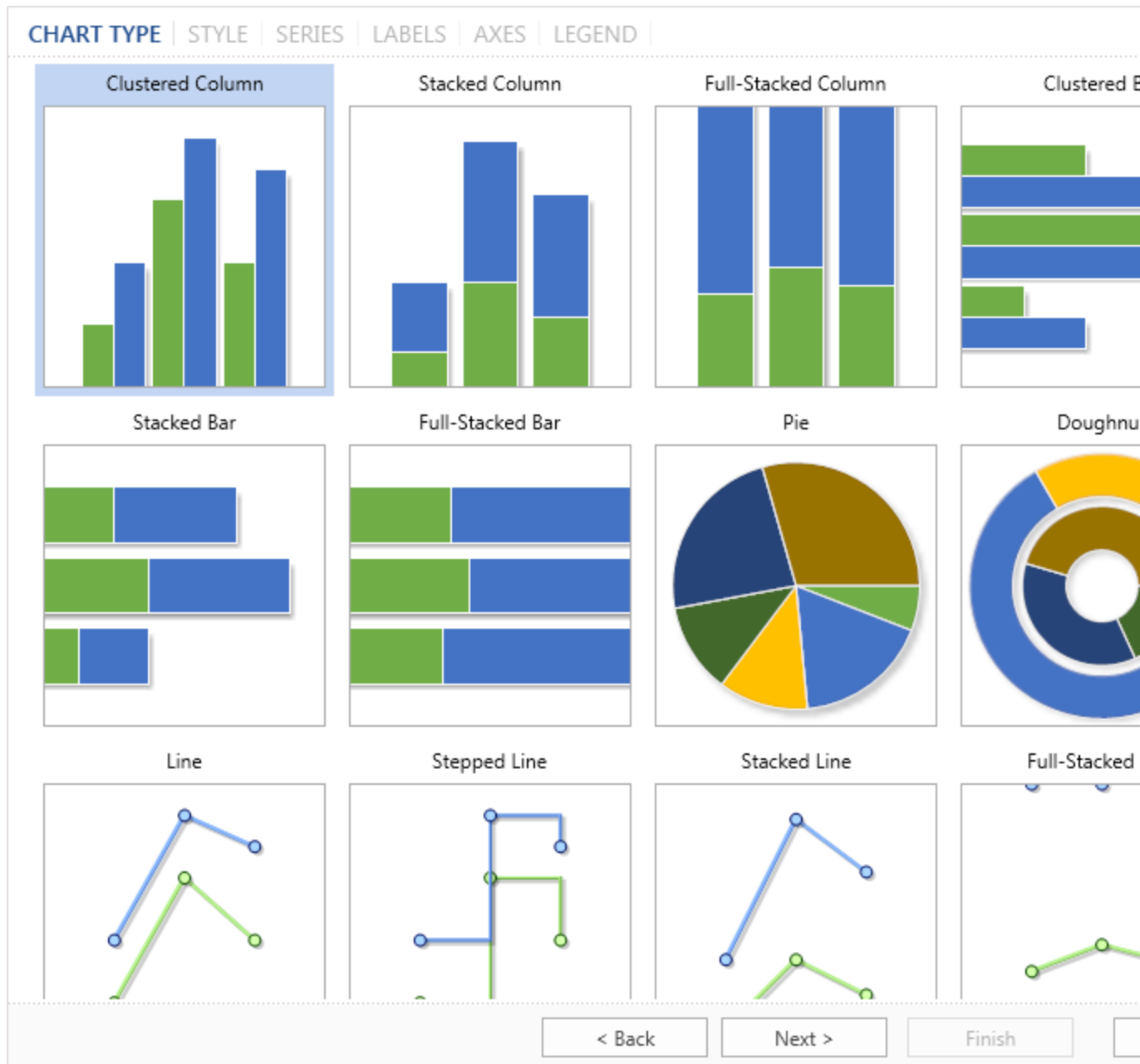
Information: In order to proceed to the next step, press the button **Next**. You should remember that you can always return to the previous step by clicking the button **Back**.

The chart component contains a collection of preset styles for the chart. Select one of them to create a chart. By default, the first style in the list is selected.

In the next step, you need to create a series of charts and specify their values.
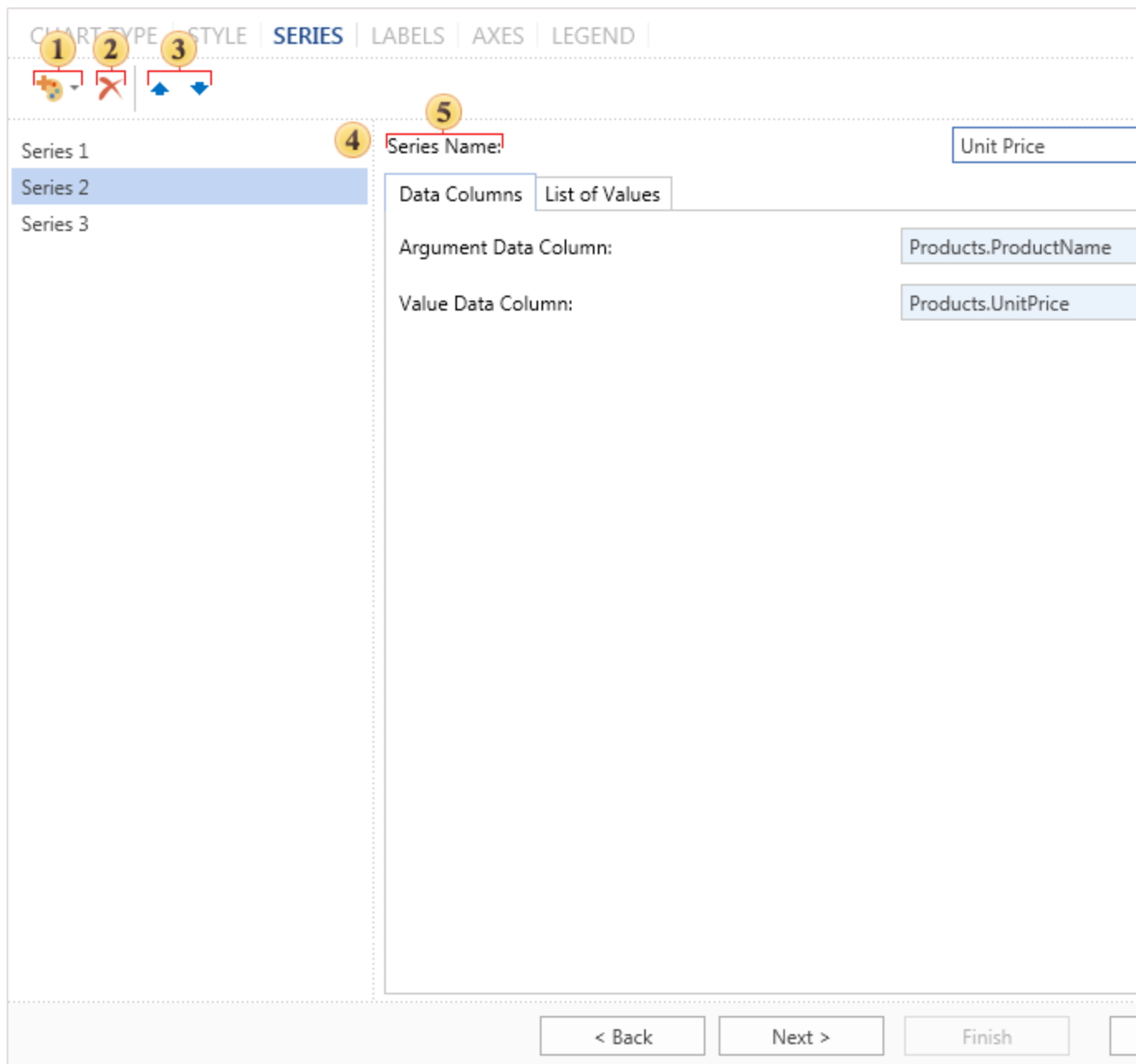
1 Clicking on this button a list of series opens. Depending on the particular type of chart, the list will have different types of series. To add a series to a chart you should select it in the list.

2 Deletes the selected series of a chart.

3 The buttons are used to move the selected number of series in the list of charts.

4 This panel displays a list of chart series.

5 In the field of this this option you can change the name of the series. By default, all series have the name as Series+"number".

6 In this panel you can set chart arguments and values. This panel has two tabs:

> The tab **Data Columns** you must specify the data columns for arguments and values. For example, the column of arguments contains entries A, B, C. The values column will contain entries: 23, 43, 56. In this case, the argument A will match the value 23, the argument B will match the value

43, and the argument C - the value 56.

❯ Besides data columns you can manually set the arguments and values. You can do this in the tab **List of Values**.

1 Add new block that consists of fields Argument and Value. You should know that in the added block the specified value will correspond to the argument in this block.

2 Remove the last inserted block of fields Value and Argument.

3 The list of arguments fields, i.e. in these fields arguments of a chart are specified. For example, the arguments A, B, C, D.

4 The list of values fields, i.e. in these fields the values of the chart are shown. For example, the values 1, 2, 3, 4.

> ℹ **Information**: It should be noted that for rendering the chart there must be at least one values, i.e. the value is required to be specified. Arguments, if they are not specified, they will be automatically created.

On the next step, it is necessary to define the look of labels in the chart. By default, labels are disabled.

① The list of labels for the chart, with examples of their placing on this type of a chart.
② Parameters of labels, their angle, the text before the header text after the header, etc.

> **ℹ Information**: You should know that when you create a chart manually, i.e. without using the wizard, you can specify label look as the entire chart and its our look for each row of the label. When you create a chart using the wizard, you can only define the general form of signatures for the whole diagram, i.e. one type for all series of the chart.

On the next step, it is necessary to define axes settings.

1 The panel **Preview**.

The most important settings are displayed on the axes. Moreover, this panel has tabs axis X and axis Y.
2 The parameter **Title**. This group of settings specifies the text of the axis title and its alignment.
3 The parameter **Ticks**. It is determined by the number of intermediate ticks and display mode - without labels, only the main, and all labels.
4 The group **Grid Lines**. This group defines the parameters of the grid line.
5 The group **Labels**. In this group you can specify the parameters of axis titles such as on/off, reverse, etc.

In the last step you need to define parameters of the chart legend. Legend is an area that displays the symbols of different data series in the chart.

1 The panel **Preview**.
2 The group **Title**. Here you can specify the title for the legend.
3 The group **Alignment**. Legend can be located in different places in the chart. In this group you can setup the vertical and horizontal alignment of the legend in the chart.
4 The group **Direction**. Entries in the legend can be placed in different directions. Here you can indicate the direction in the legend in this group.

5 The group **Marker**. The marker is an icon that helps you to visually recognize a series of charts. The number of markers corresponding to the number of rows. Setting markers is performed in this group of parameters.

6 The group **Spacing**. Increasing or decreasing the vertical and horizontal indentation in the legend is carried out with the help of these parameters. Also, in this group there is a parameter Visible. If this option is enabled the legend is displayed. If not - the legend is not displayed.

Click the button **Finish** and the chart will be created.

3.29.2 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:
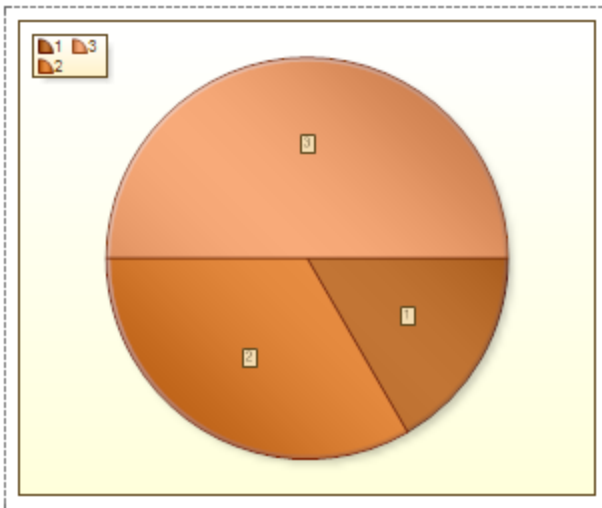
As can be seen from the picture, the series are arranged consecutively in a clockwise direction. In the Doughnut chart, the number of rows will match the number of rings. The picture below shows an example of a chart that has three rows:
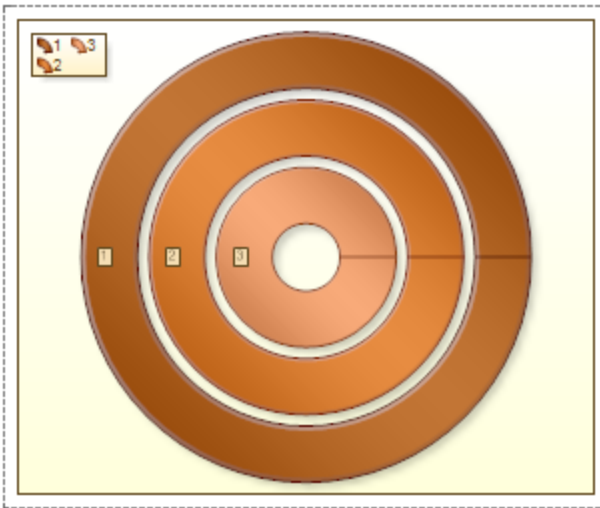
3.29.2.1 Doughnut

The **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:
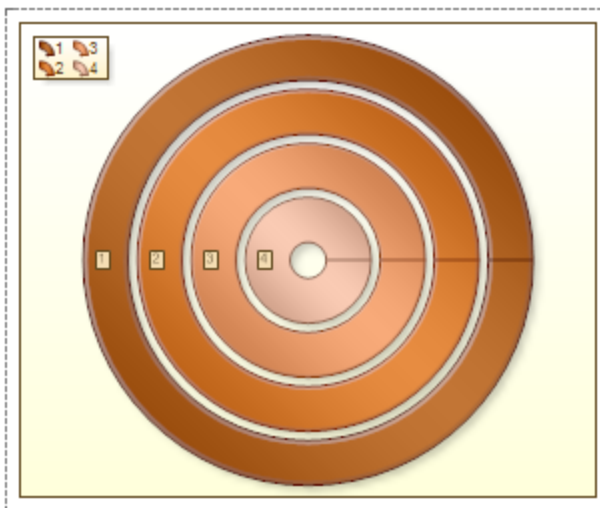
3.29.2.1.1 Series Labels

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:
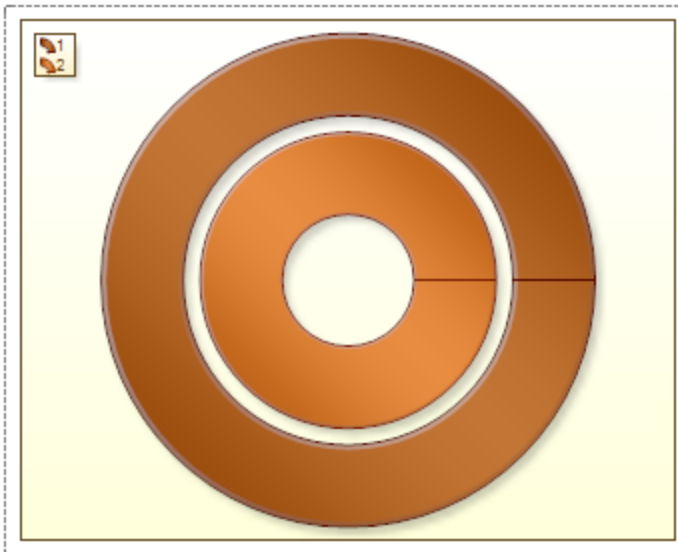
If the **Series Labels** property is set to **Center**, then labels are shown in the center of the chart ring. The picture below shows the doughnut with labels:
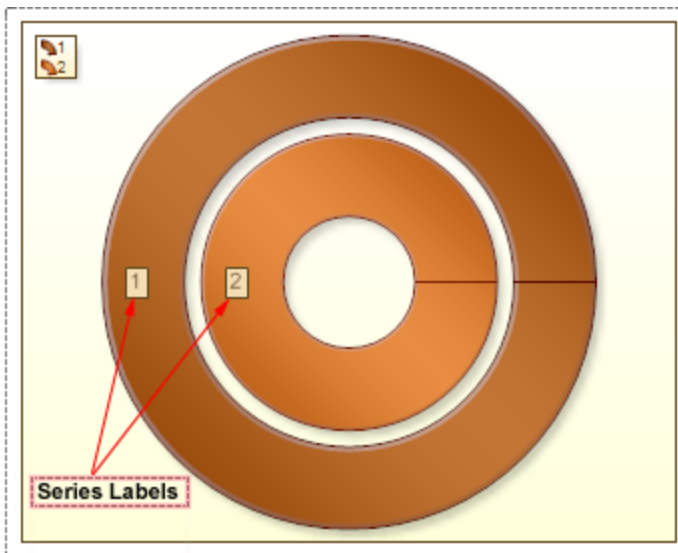
3.29.2.2 Pie

The **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.
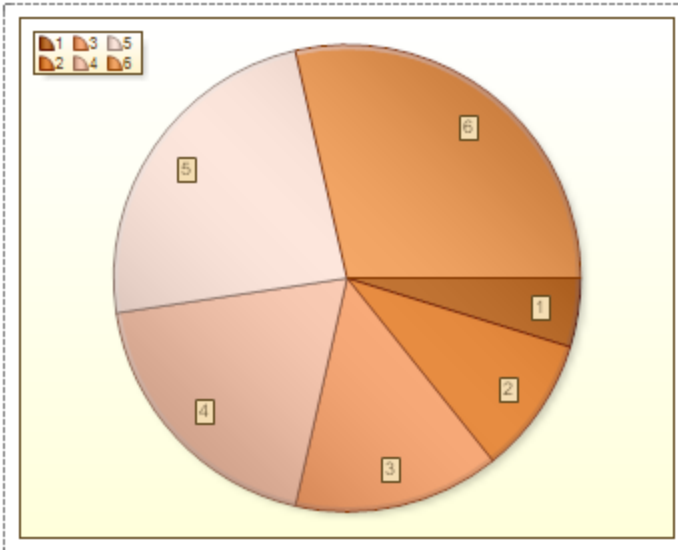
3.29.2.2.1 Series Labels

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.

❯ **None**. Series Labels are not shown. The picture below shows an example of a Pie chart with the **Series Labels** set to **None**:
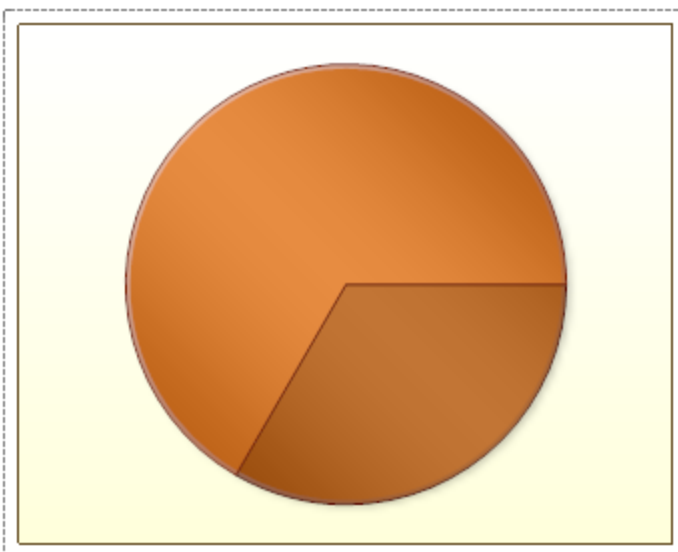
❯ **Inside End**. Series Labels are displayed inside the slice and far from the center. The

picture below shows an example of a Pie chart with the **Series Labels** set to **Inside End**:
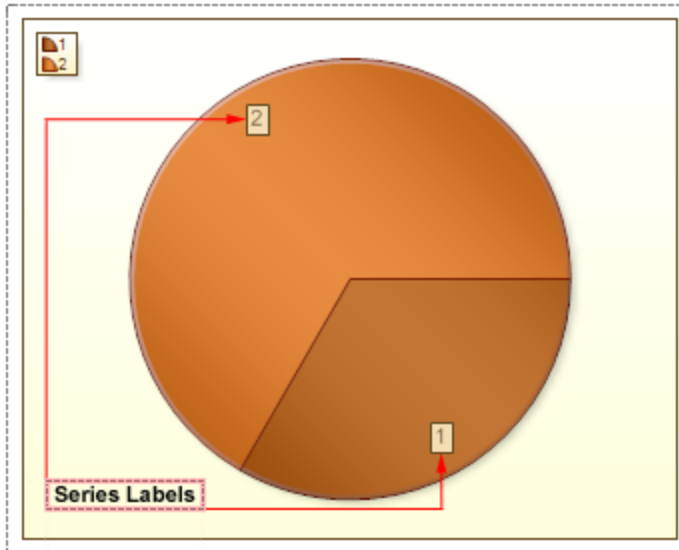
> **Center**. Series Labels are displayed in the center of the slice. The picture below shows an example of a Pie chart with the **Series Labels** set to **Center**:
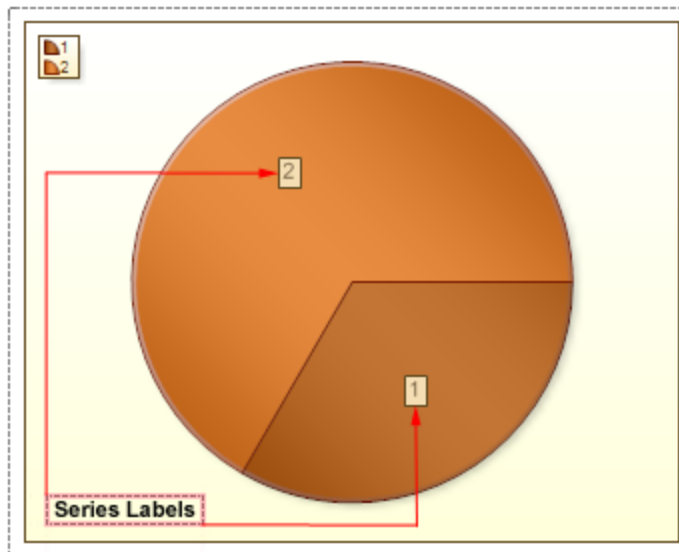
> **Outside**. Series Labels are displayed outside the chart, but in a Pie area. The picture below shows an example of a Pie chart with the **Series Labels** set to **Outside**:
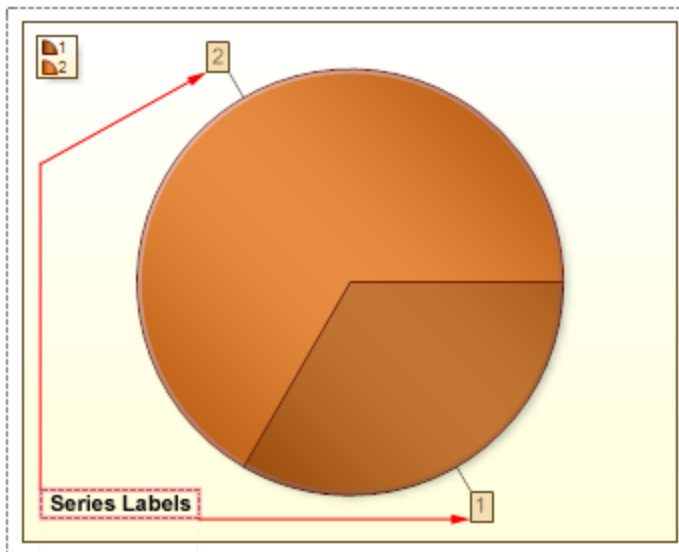
❯ **Two Columns**. Series Labels are displayed outside the chart in two columns: on the left and right of the chart. The picture below shows an example of a Pie chart with the **Series Labels** set to **Two Columns**:
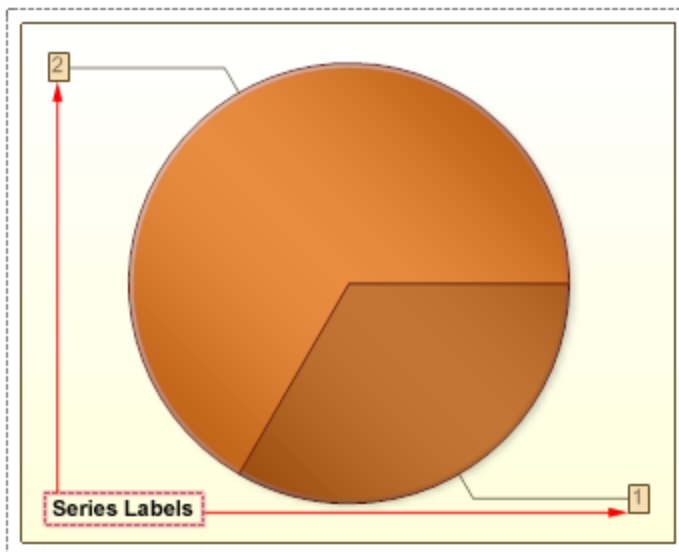
3.29.2.2.2 CutPieList Property

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:
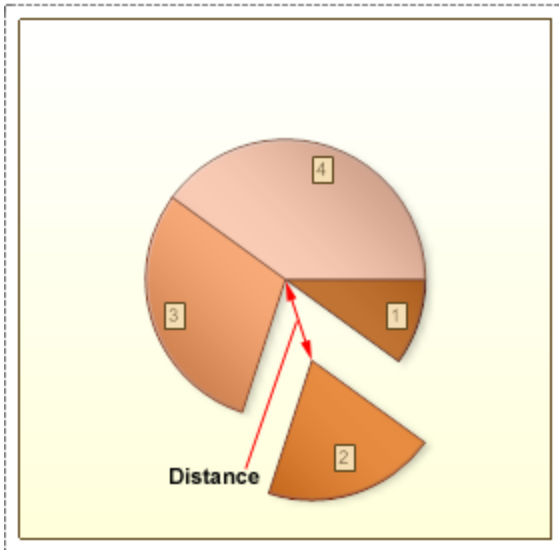
If the field of the **CutPieList** property is filled, and the field of the **Distance** property is not filled, then the segments will not be pulled out. If the field of the **Distance** property is filled, and the field **CutPieList** property is not filled, then all segments of this series will be 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:
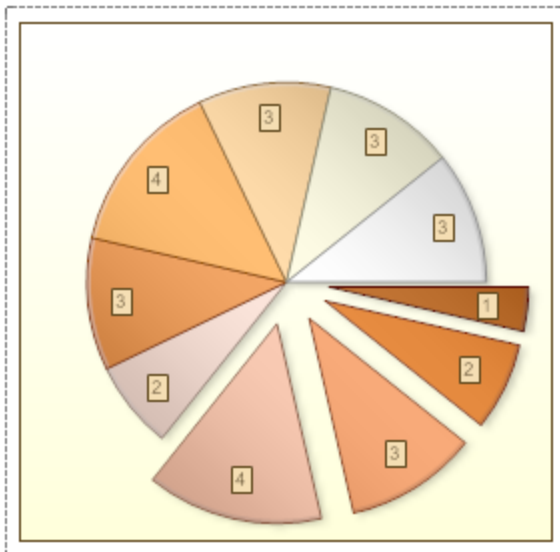
3.29.3 Legend

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.

3.29.3.1 Title Property

The **Title** property of the Legend allows setting the Legend title. The full path to this property is **Legend.Title**. If 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:
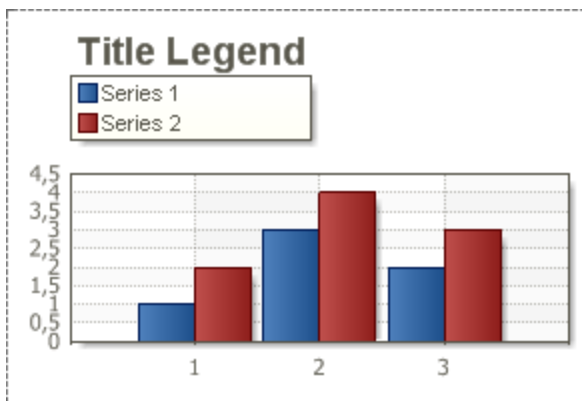
The **Title** property has the following properties:
✓ **TitleColor** - sets the Title color;
✓ **TitleFont** - sets the Title font size and font style.

3.29.3.2 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:
➤ **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**:
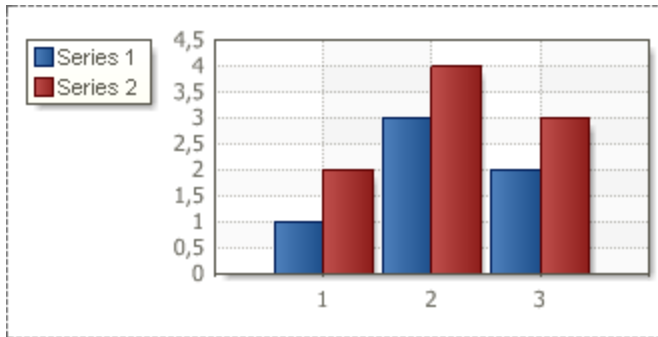
> **Left**. The legend will be placed inside the Chart area on the left. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Left**:
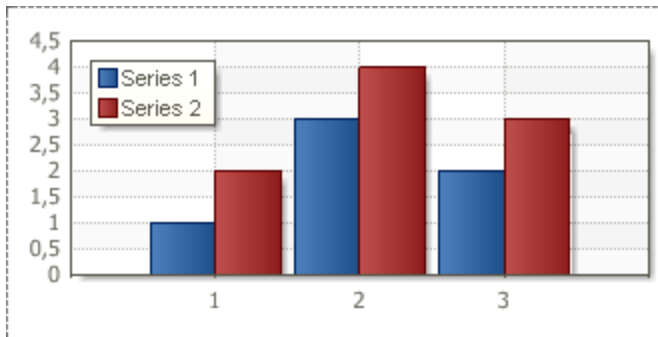
> **Center.** The legend will be placed inside the Chart area in the center. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Center**:
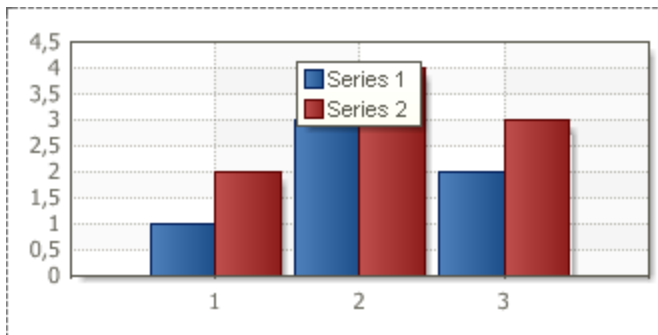
> **Right**. The legend will be placed inside the Chart area on the right. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Right**:
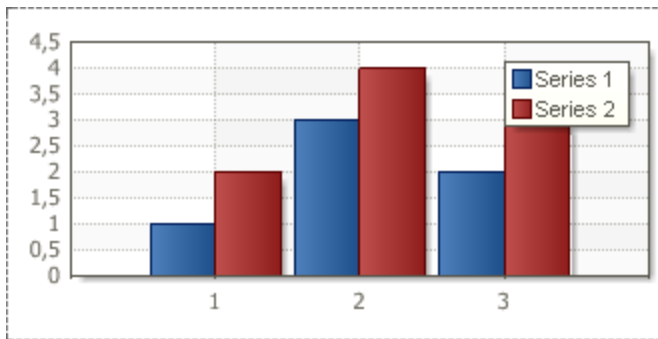
> **Right Out Side**. The legend will be placed out side the Chart area on the right. The picture below shows where the Legend will be placed if the **Horizontal Alignment** property is set to **Right Out Side**:
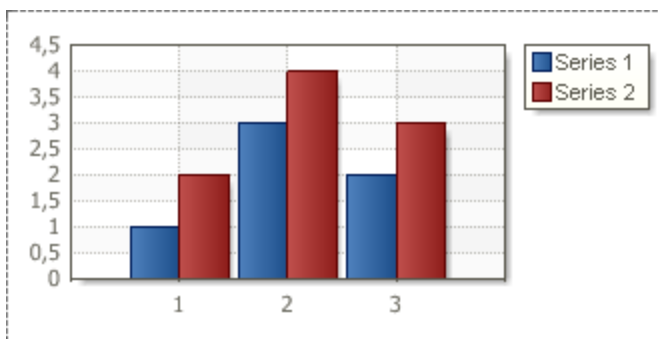
By default the **HorizontalAlignment** property is set to **Left**.

3.29.3.3 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:
> **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**:
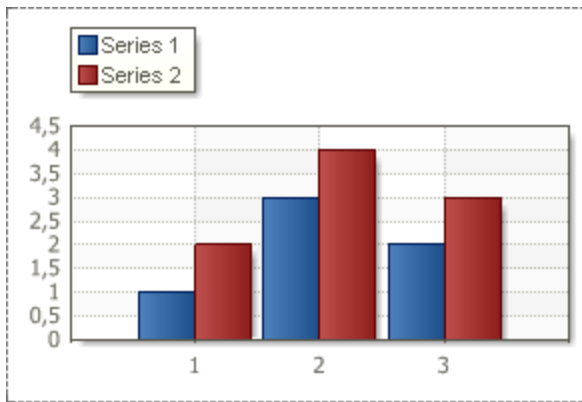
❯ **Top**. The legend will be placed inside the Chart area on the top. The picture below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Top**:
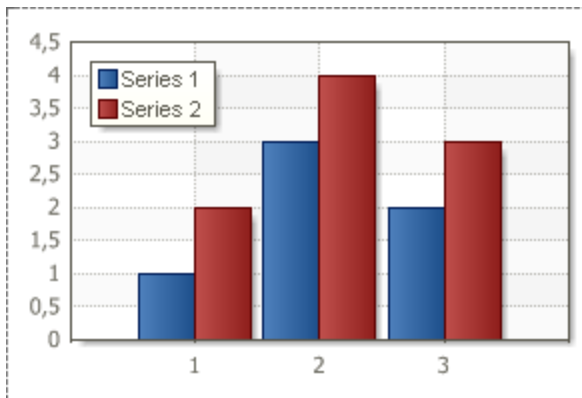
❯ **Center**. The legend will be placed inside the Chart area and vertically in the center. The picture below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Center**:
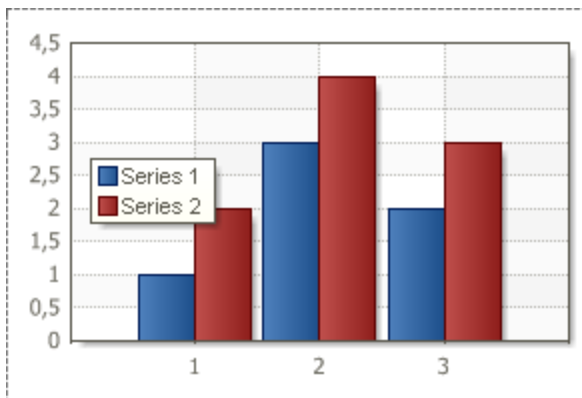
❯ **Bottom**. The legend will be placed inside the Chart area on the bottom. The picture

below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Bottom**:
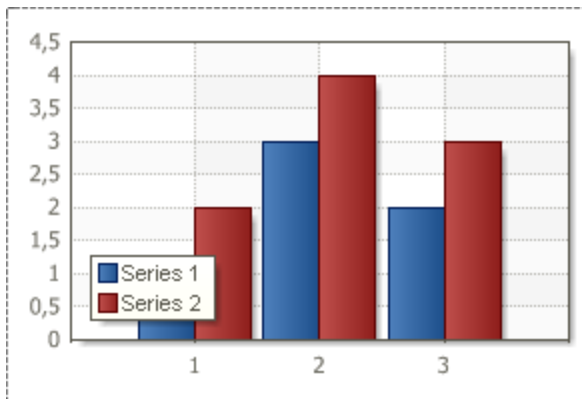
❯ **Bottom Out Side**. The legend will be placed under and outside the Chart area. The picture below shows where the Legend will be placed if the **Vertical Alignment** property is set to **Bottom Out Side**:
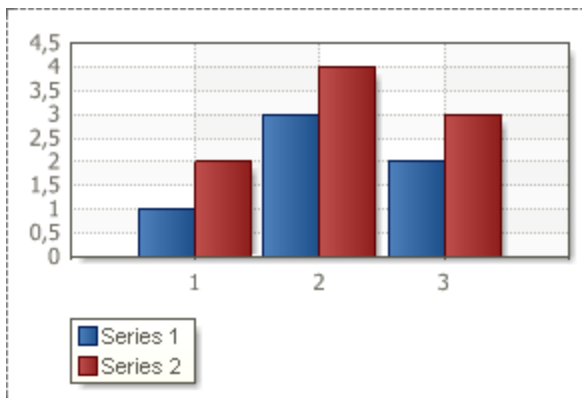
By default the **Vertical Alignment** property is set to **Top**.

3.29.3.4 Horizontal Spacing and Vertical Spacing 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.

3.29.3.5 Marker

The **Marker** is an icon that indicates the chart 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:
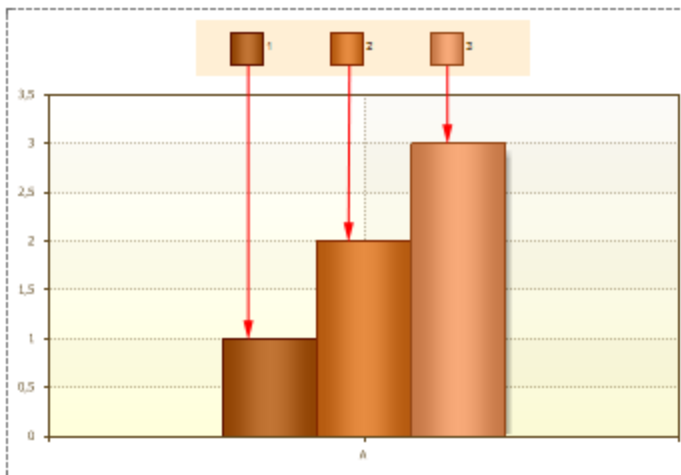
3.29.3.5.1 Direction Property

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:
> **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:

> **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**:

> **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**:

> **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**.

3.29.3.5.2 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.

3.29.3.5.3 Marker Alignment 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**.

The **MarkerVisible** property allows showing/hiding the legend markers. The full path to this property is **Legend.MarkerVisible.** If the **MarkerVisible** property is set to **true**, then markers are shown. The picture below shows a sample of the Legend which the **MarkerVisible** property is set to **true**:

If the **MarkerVisible** property is set to **false**, then the Legend markers are hidden. The picture below shows a sample of the Legend which the **MarkerVisible** property is set to **false**:

By default the **MarkerVisible** is set to **true**.

3.29.4 Axes Area

The **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:
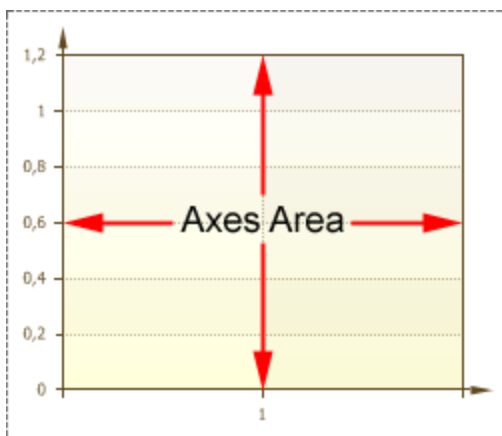
3.29.4.1 Axes

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.
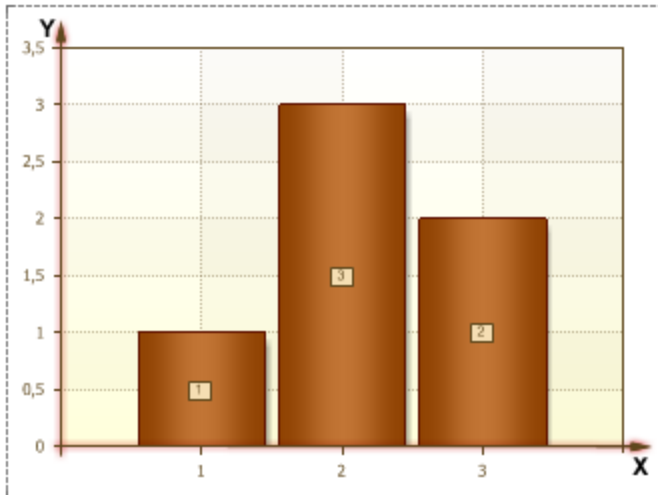
Besides, the **Axes Area** can contain top and central **X** axis, and right **Y** axis.
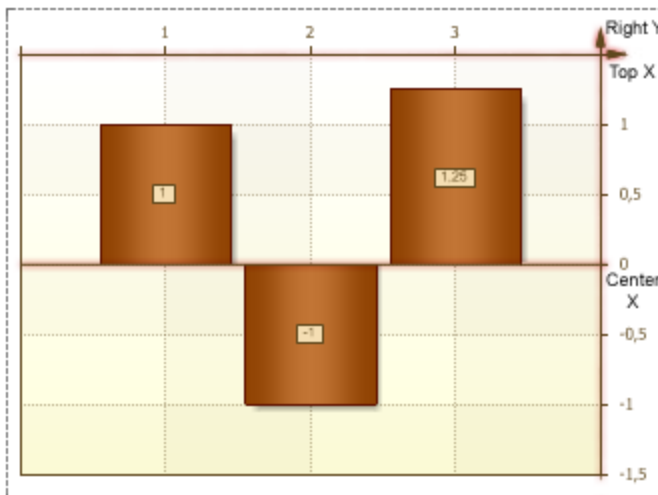
3.29.4.1.1 Arrow Style Property

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:
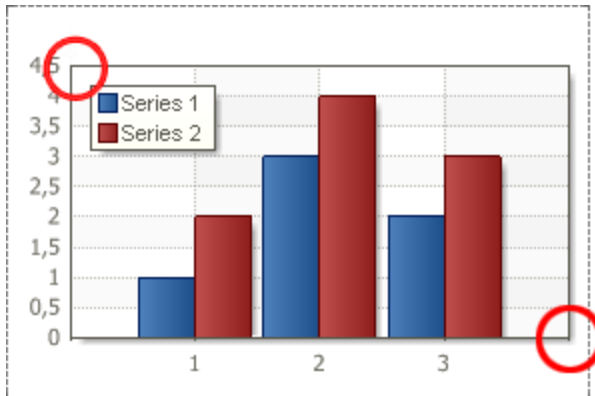
As you can see, if the **ArrowStyle** property is set to **None**, then **X Y** axes do not have style. The **ArrowStyle** property can be set to **Triangle.** In this case the arrow style will look like on the picture below:
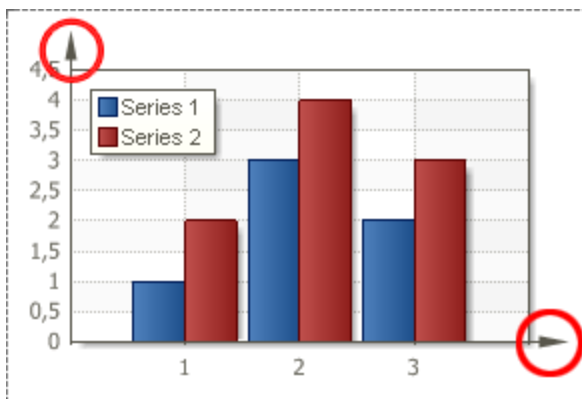
The **ArrowStyle** property can be set for each axis. Each axis may have its own values of the **Arrow Style** property. On the picture below different values of the **ArrowStyle** property of **X** and **Y** axes:
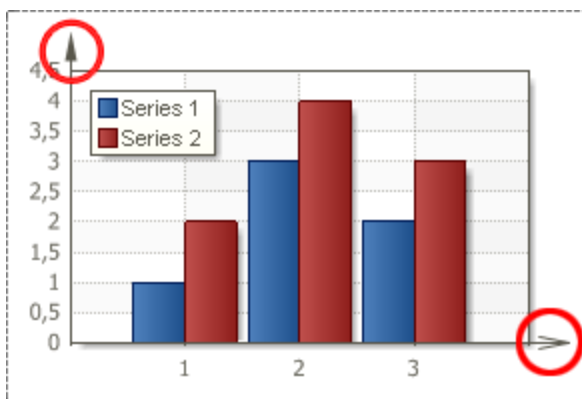
As seen from the picture above, the **ArrowStyle** property, of the **Y** axis is set to **Triangle.** And the **ArrowStyle** property, of the **X** axis is set to **Lines.**

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:
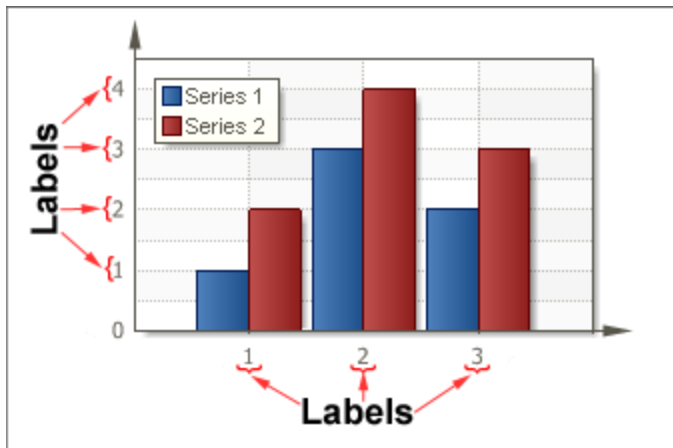
Also, Labels have a number of properties such as:
- ✓ **Angle** - sets an angle of inclination of labels;
- ✓ **Antialiasing** - sets smooth-edged type of labels;
- ✓ **Color** - sets the labels color;
- ✓ **Font** - sets the font type of labels;
- ✓ **Format** - changes the label format (numeric, percentage 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.

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:
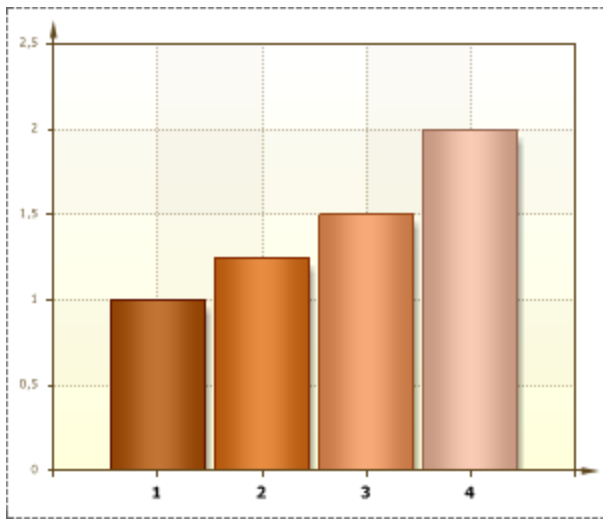
The value of this property can be negative and positive. If the value of the property is negative then Label is inclined clockwise. If the value of the property is positive then Label in inclined anticlockwise. The picture below shows the chart sample, which Angle property by the **X** axis is set to **50**:
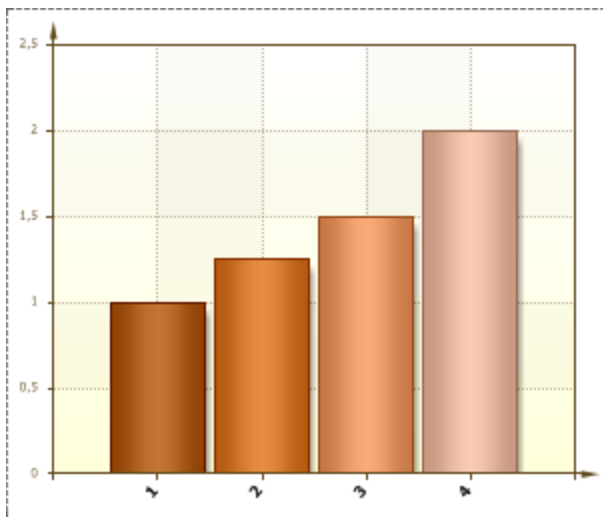
3.29.4.1.2.2 Format Property

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.

> **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**:
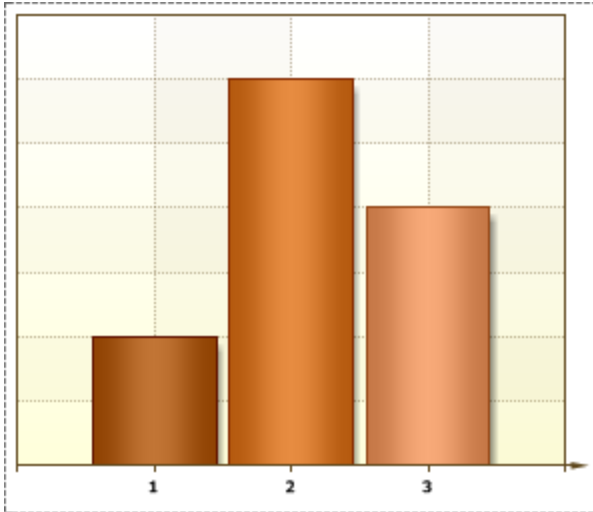
➤ **Currency.** The **C** value of the **Format** property is used to display Labels with a currency symbol. With the **C** value, it is possible to specify the number of decimal places that you want to use. The picture below shows a chart with the **Format** property of Series Labels set to **C**:
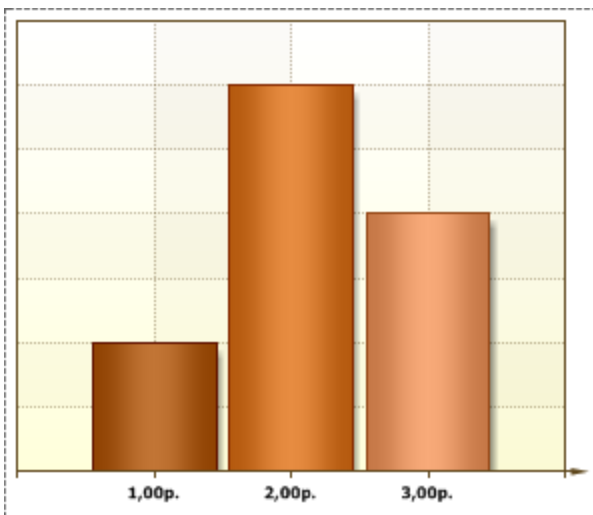
➤ **Percentage**. The **P** value of the **Format** property is used to display Labels with percent symbol. After the **P** value, it is possible to specify the number of decimal places that you want to use. The picture below shows a chart with the **Format** property of Series Labels set to **P**:
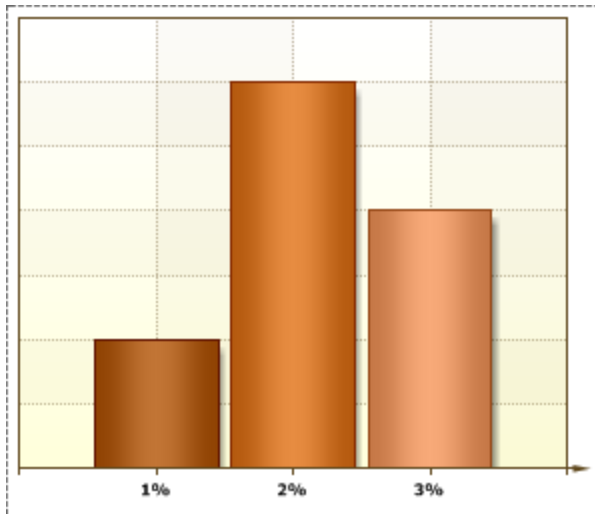
> Date formatting. The **MM/dd/yyyy**, **MMMM dd**, **yyyy MMMM** values of the **Format** property convert values of arguments to date. **MM/dd/yyyy** - the date is shown like "01.20.2010", **MMMM dd** - the date is shown like "September 29", **yyyy MMMM** - the date is shown like "2010 March". The picture below shows a chart and its **Format** property is set to **MM/dd/yyyy**:
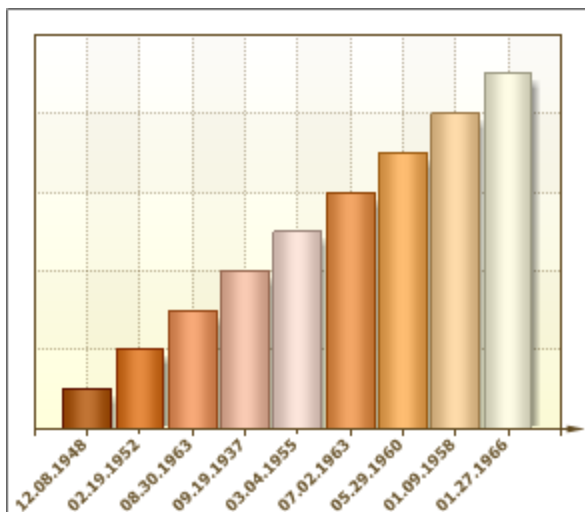
To reset the **Format** property of selected cells, and return to the default format, clear the Format by selecting empty field.

3.29.4.1.2.3 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**.

❯ **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:
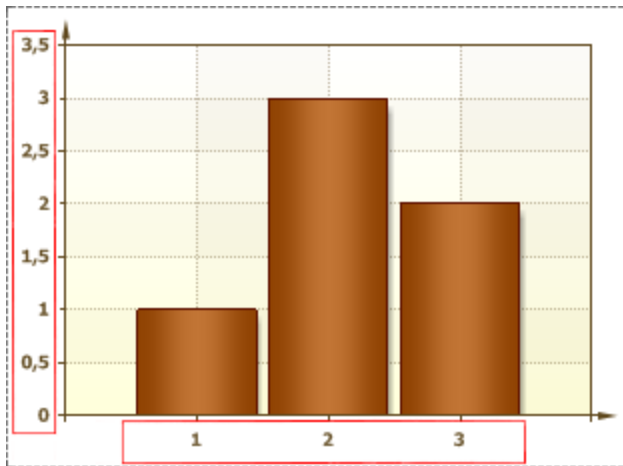
❯ **Two Lines**. In this case, labels are placed in two lines horizontally or vertically, depending on the X or Y axis, respectively. The picture below shows an example of a chart, with the **Placement** property set to **Two Lines** for of X and Y axes:
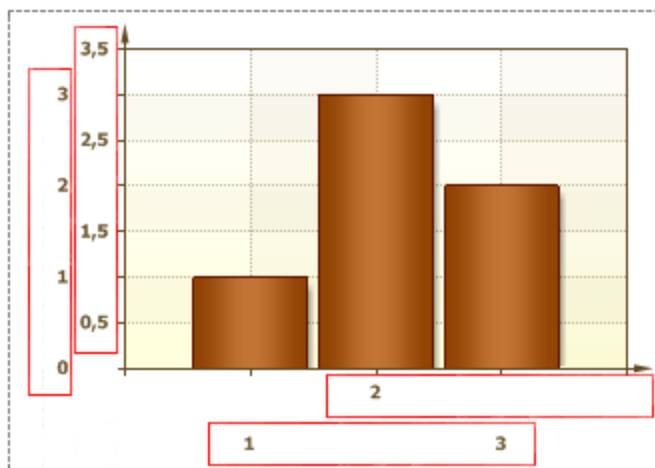
❯ **None**. In the case labels are not shown. The picture below shows an example of a chart, with the **Placement** property set to **None** for of X and Y axes:
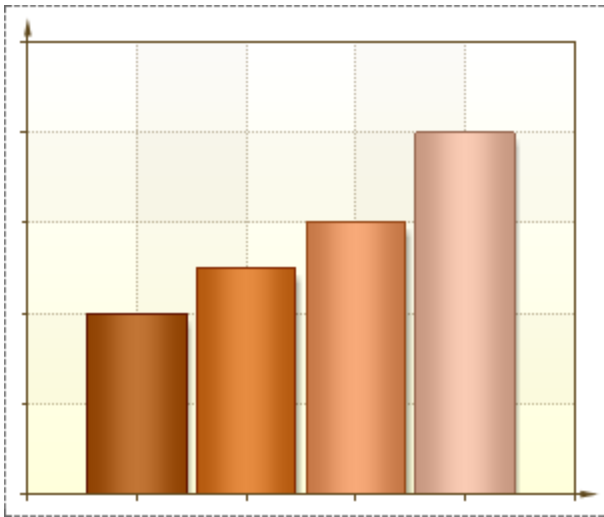
By default, the **Placement** property is set to **One Line**.

3.29.4.1.2.4 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**:
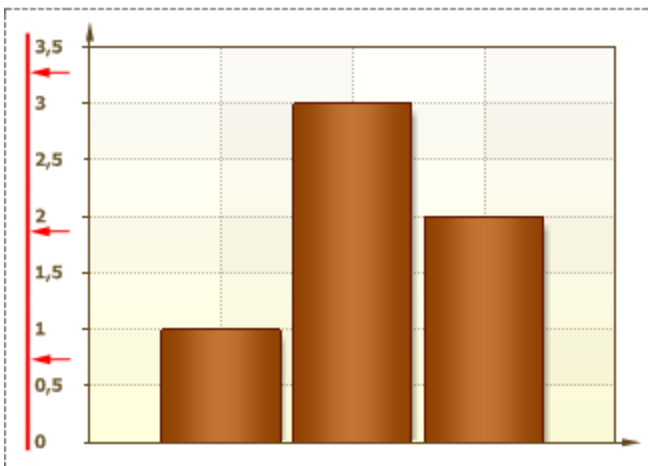
If the **TextAlignment** property set to **Right**, then the labels are aligned by the Y axis. The picture below shows an example of chart with the of **TextAlignment** property set to **Right**:
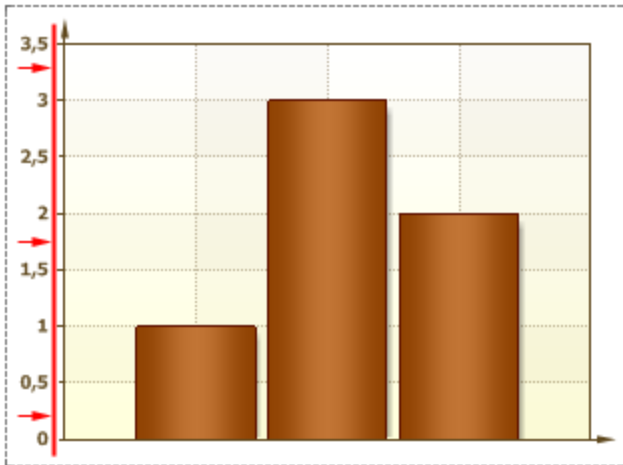
By default, the **TextAlignment** property is set to **Right**.

3.29.4.1.3 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:
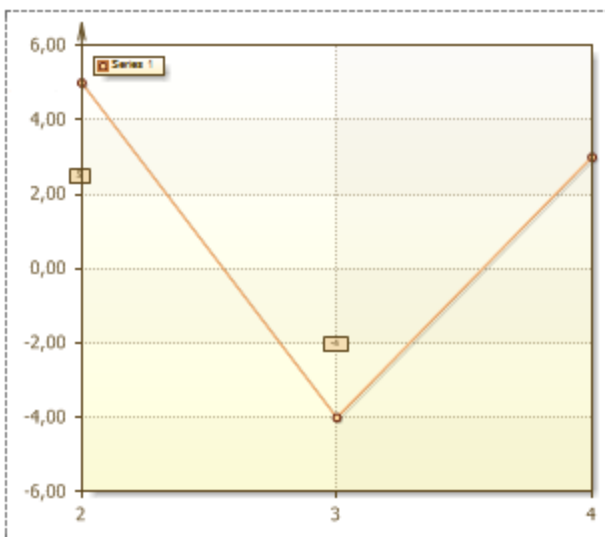
The Range consists of the values of three fields:
➤ **Auto**. If the Auto field is set to true, then a chart is shown entirely, i.e. the range of values will be calculated automatically. The picture below shows an example of it:
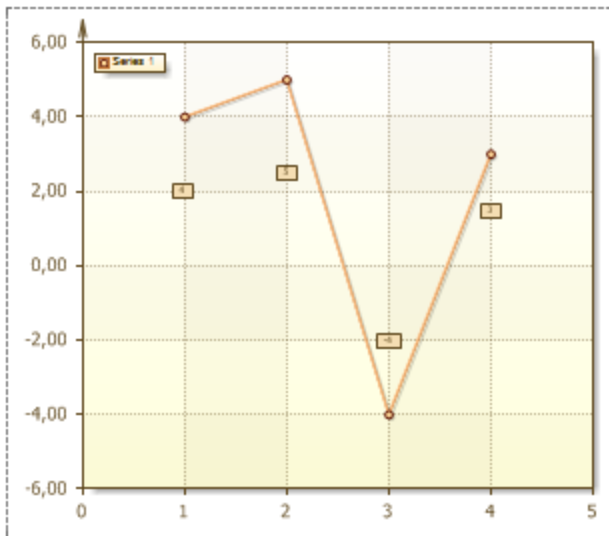
If the **Auto** field is set to **false**, then all values of the range which are specified in the **Minimum** and **Maximum** fields are considered. If the **Auto** field is set to **false**, and values the **Minimum** and **Maximum** fields are set to 0, then the chart will be shown entirely.

> **Minimum** - sets the beginning of the range.
> **Maximum** - sets the end of the range.

If the **Maximum** value is less then the **Minimum** value, then the chart will be displayed entirely.

3.29.4.1.4 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):
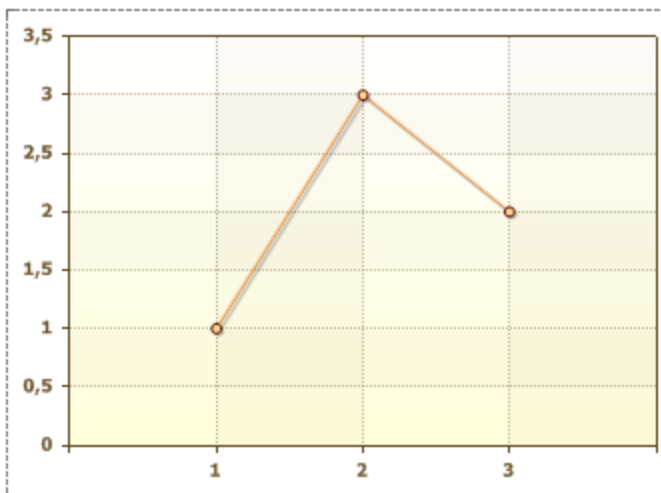
If the **Visible** property is to set the **false**, then X and Y axes will not be shown. The picture below shows this:
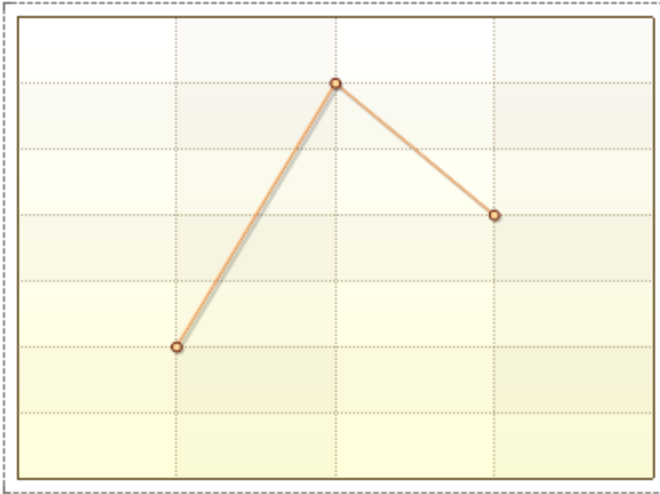
The **Visible** property has the X axis and the Y axis. It is possible to hide/show axes separately. Also, this property is used to display the top X axis and right Y axis. By default, for the axes, the property is set to **false**. The picture below shows an example of a chart, to display the top X axis and the right Y axis:
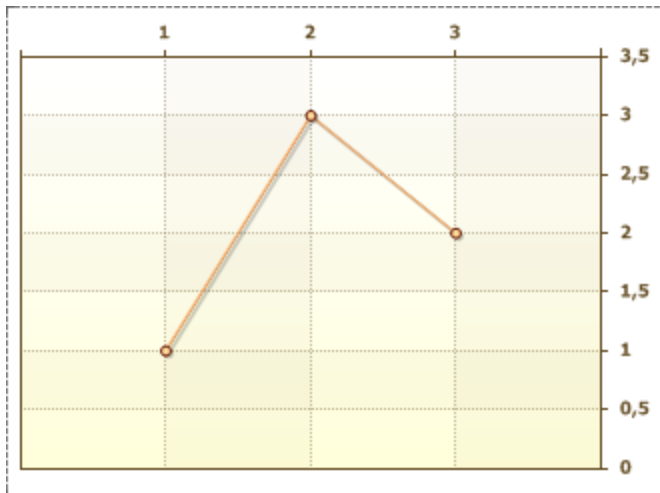
The **Visible** property has the top X axis and the right Y axis. It is possible a combination, for example, the top X axis and the left Y axis or the X axis and right Y axis or any other combinations.

By default the **Visible** property is set to **true**.

3.29.4.1.5 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:
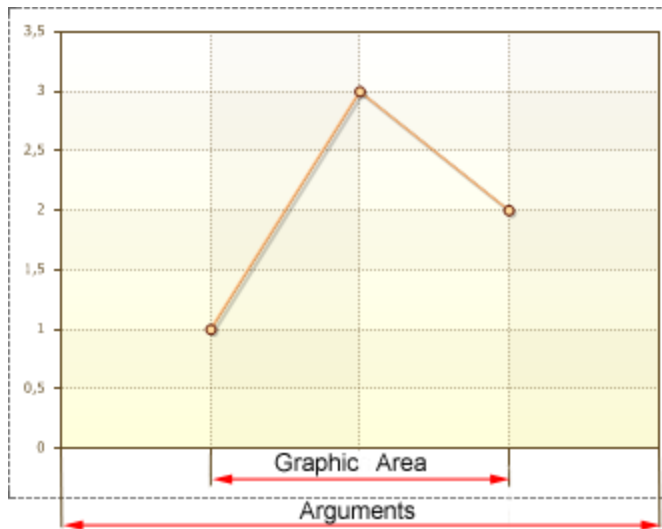
If the **Start from Zero** property to set **false**, then the Range of the chart area will be shown. The picture below shows an example of a chart with the **Start from Zero** property set to **false** for the X axis:
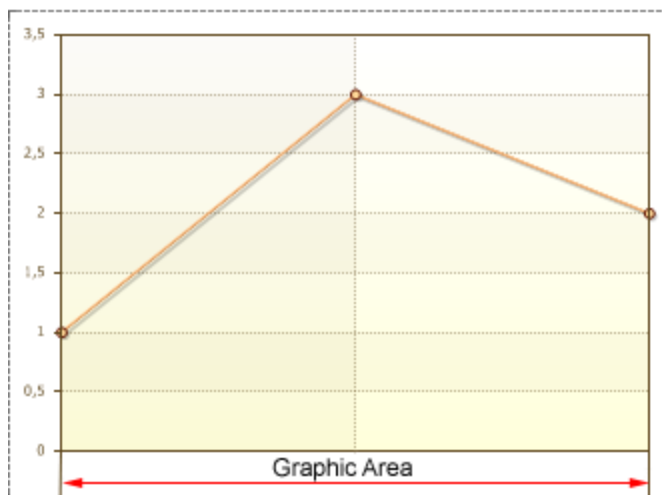
3.29.4.1.6 Ticks

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 shows a chart with ticks:
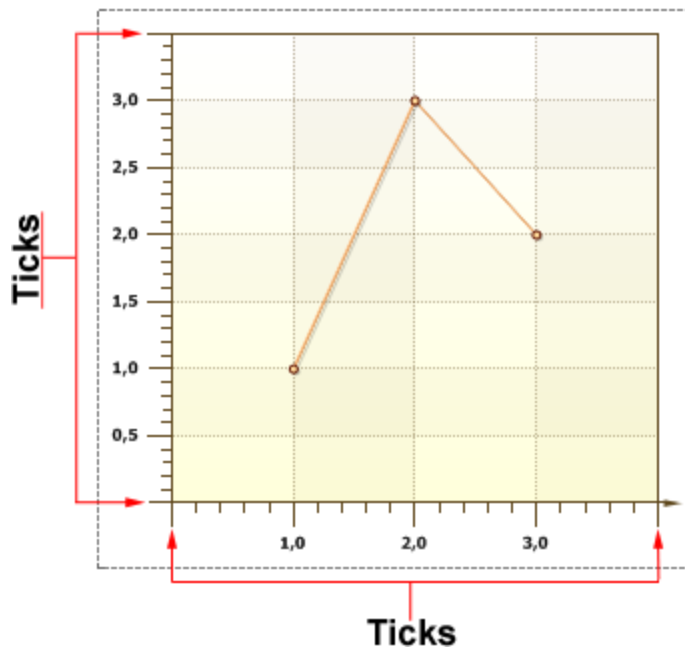
Ticks have the following properties:

✓ **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.

3.29.4.1.6.1 Minor

Minor ticks show the proportion of a single axis segment. **Minors ticks** have the following properties: **MinorCount**, **MinorLength**, **MinorVisible**.

❯ **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:
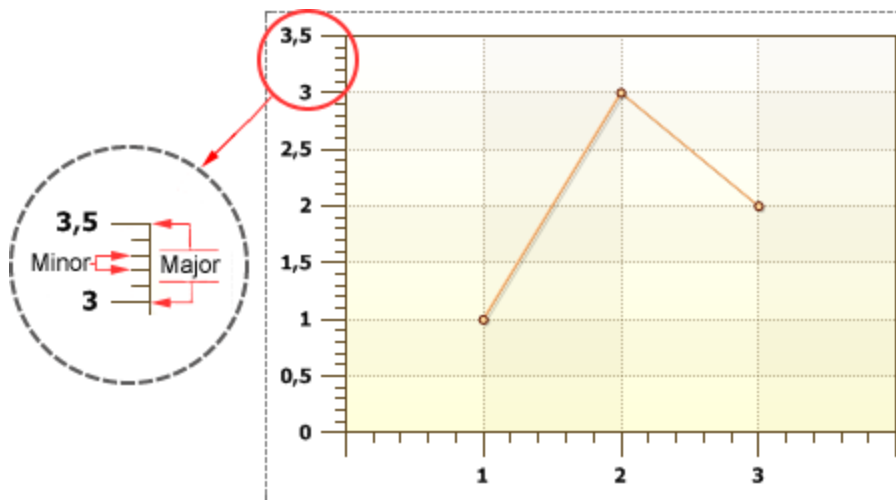
> **Minor Length** is used to change the length of Minor ticks. The value of this property can be any positive number greater than 0, the field of this property can not be left blank. The length of Minor ticks can be longer than the length of Minor ticks.

> **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:
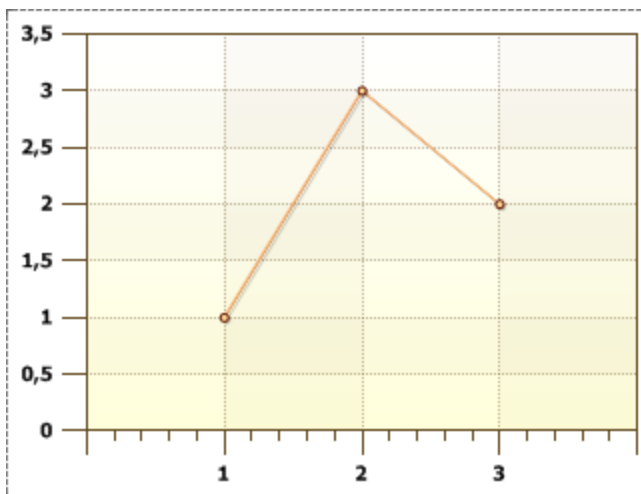
By default, the **Minor Visible** property is set to **false**.

3.29.4.1.6.2 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.
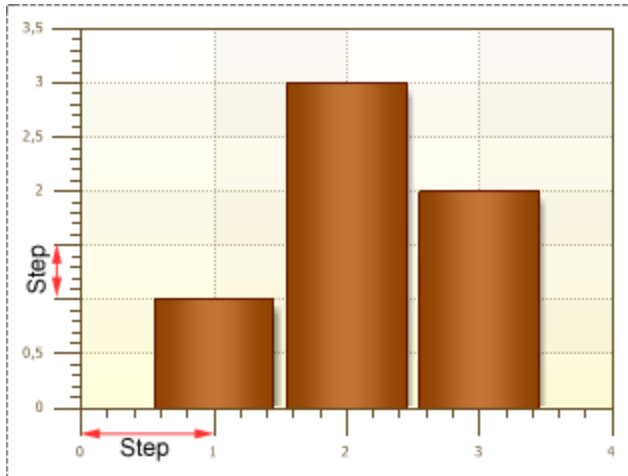
As one can see, if the value is 0, then the distance between two between neighbor Major ticks by the Y axis is **0.5**, and **1** by the X-axis. If to set the Step property to **Z** value, then the report generator will multiply **Z** value by the value of the unit interval. The result obtained is the 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**:
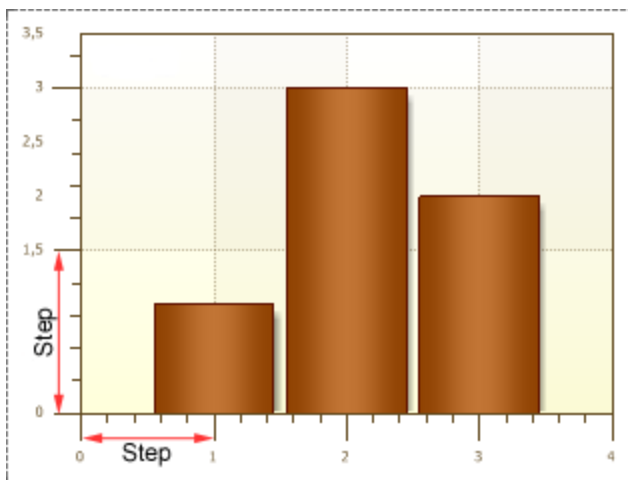
3.29.4.1.7 Title Property

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**":
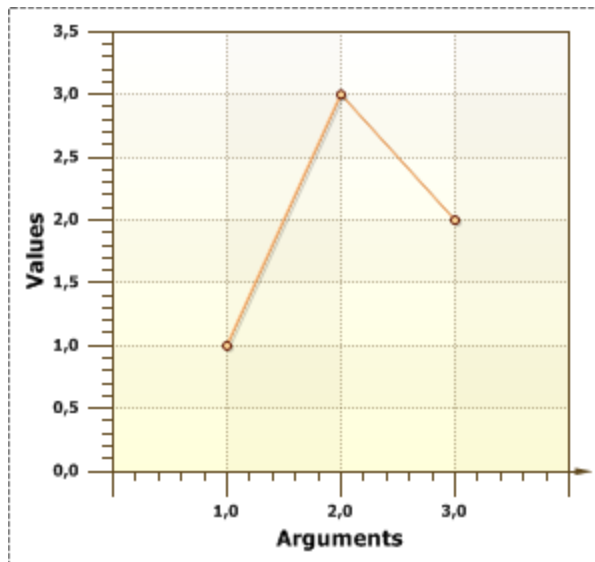
Also, the **Title** property has the following properties:

✓ **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.

3.29.4.1.7.1 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**.

❯ **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**:
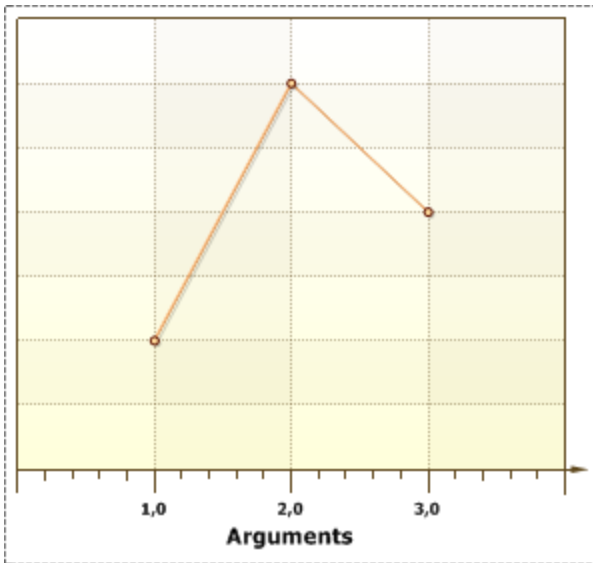
> **Far**. Aligns the title of the axis on the opposite side from origin of coordinates. The picture below shows an example of a chart, with the **Alignment** property of a title of the X axis set to **Far**:
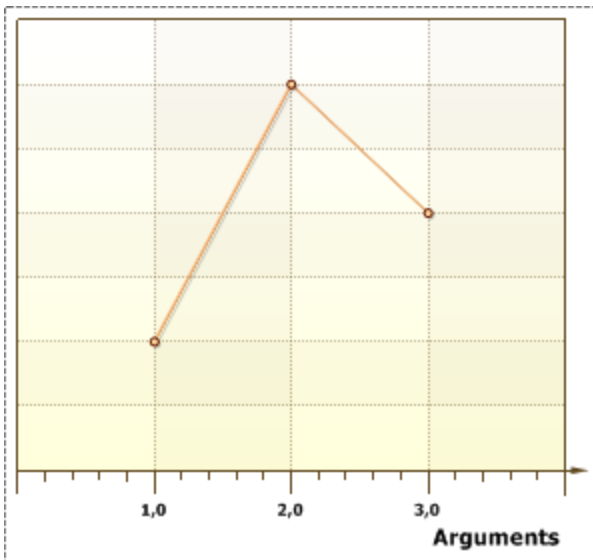
> **Near**. Aligns the title of the axis on the near the origin of coordinates. The picture below shows an example of a chart, with the **Alignment** property of a title of the X axis set to **Near**:
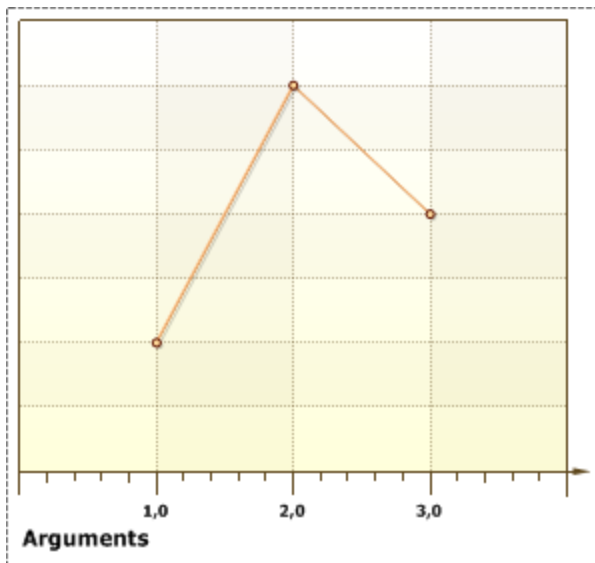
By default, the **Alignment** property of series is set to **Center**.

3.29.4.2 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.):
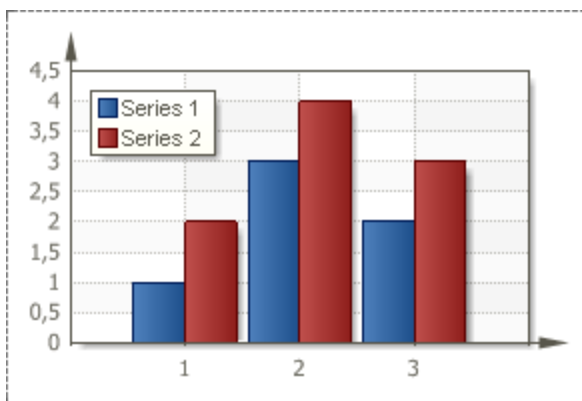
If the **Reverse Vertical** property is set to **true**, then the chart will appear in the opposite direction vertically. The picture below shows an example of a chart, with the **Reverse Vertical** property is set to **true** (As one can see, the values of the x-axis have downright direction.):
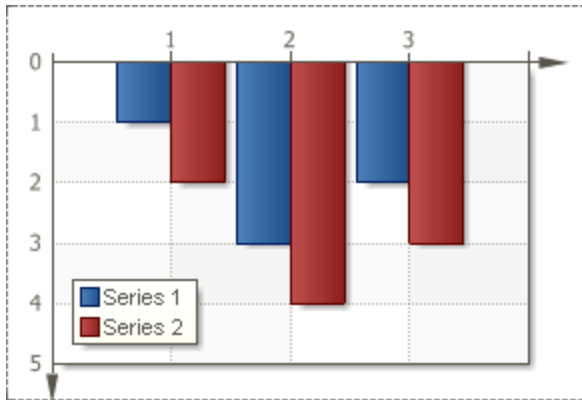
By default, the **Reverse Vertical** property is set to **false**.

3.29.4.3 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.):
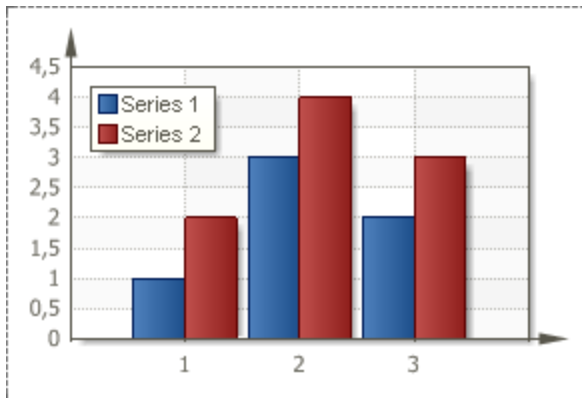
If the **Reverse Horizontal** property is set to **true**, then the chart will appear in the opposite direction horizontally. The picture below shows an example of a chart, with the Reverse Horizontal property is set to true (As one can see, the values of the x-axis have right to left direction.):
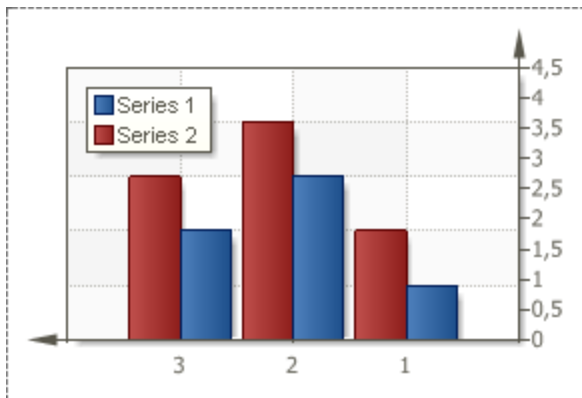
By default, the **Reverse Horizontal** property is set to **false**.

3.29.4.4 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:
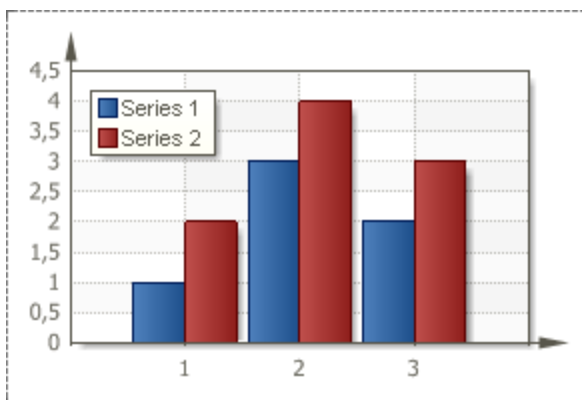
If the **Color Each** property is set to true, then each value of X axis has its own color. The picture below shows an example of a chart with the **Color Each** property set to **true** for two series:
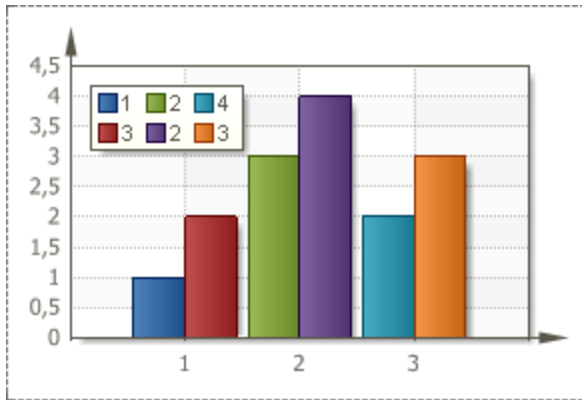
3.29.5 Series

The **Series** type depends on the chart type. They are divided into series, placed on doughnut charts, and placed in the axis area.

3.29.5.1 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:

> To set the column data from the dictionary;

> To specify an expression;

> Manually specify values for the series as a list, through the '**;**' separator.

3.29.5.1.1 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:
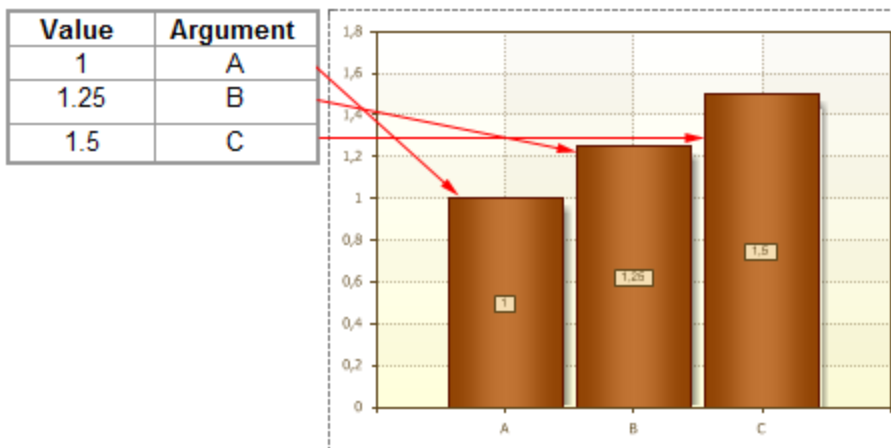
3.29.5.1.2 Expressions

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:
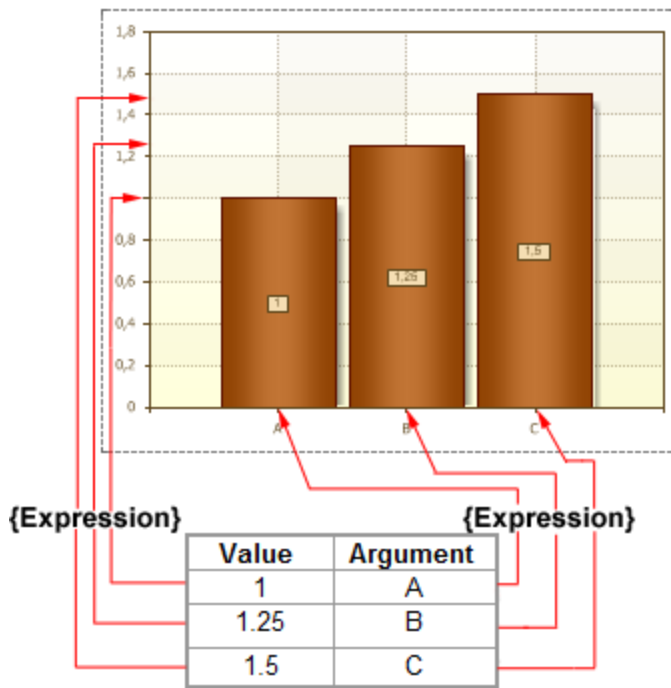
3.29.5.1.3 List of Values Property

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:
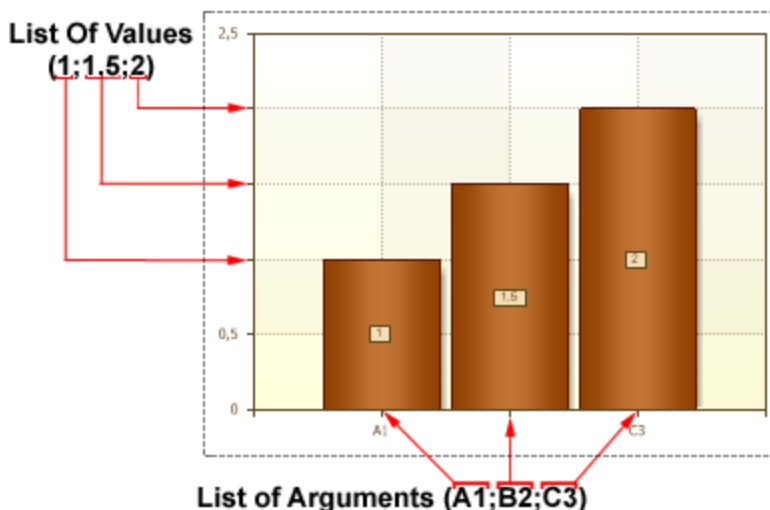
3.29.5.2 AutoSeries

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**:
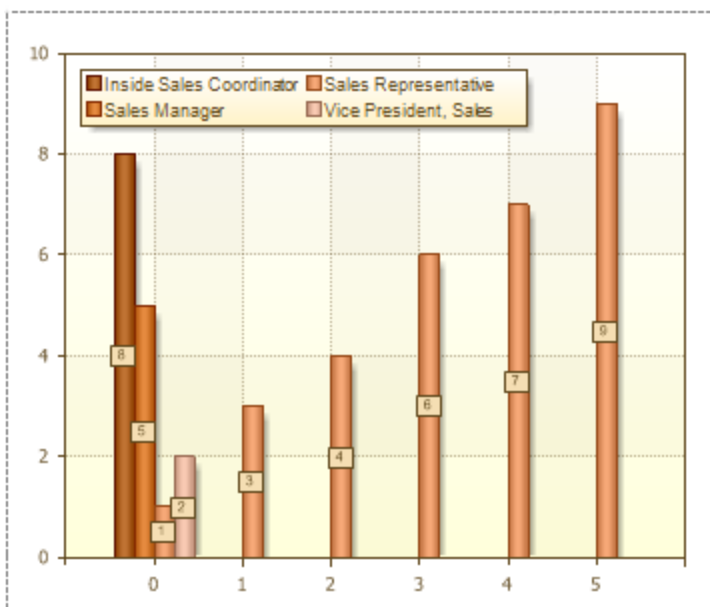
There are 4 rows on the picture above. The 1st, 2nd, 4th series have one value, and the 3rd series has 6 values. This means that the **Employees** data source in the **Title** column contains **9** lines, and 6 lines have identical values (records), and the remaining three are different. Values (records) 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**:
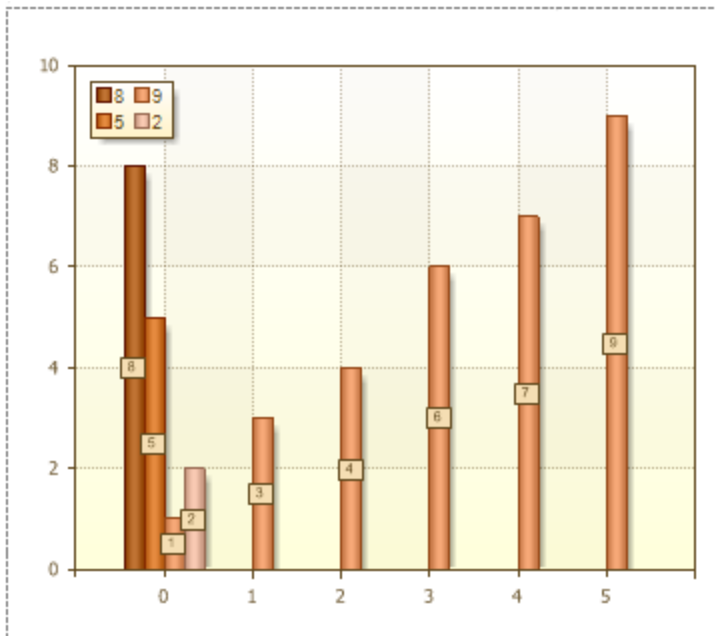
As seen from the picture above, the series labels are changed. As the series labels, string values are taken from the columns of the data source that is listed in the **Auto Series Title Data Column** property, in this case, this is the **EmployeeID** column.

3.29.5.3 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.

3.29.5.4 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.

3.29.5.5 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.

➤ 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.

❯ The **Brush** property is used to change the type of filling and series color.

❯ 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.

❯ 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.

❯ 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.

❯ 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.

❯ 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.

❯ 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.

❯ 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.

3.29.5.6 Axis Y

For each row, you can choose left or right axis Y, which is about the plot. Attachment to the axis of the graph depends on the properties of a number of axis Y (Axis Y), depending on the value of this property and are binding. If this property is set to Left axis Y (Left Y Axis), it will bind to the left axis, and if the property is set to the right axis Y (Right Y Axis) - to the right. Typically, this feature is used when you want to display a chart of different types of series. Let us consider in more detail with an example. We construct a diagram that will contain data on global economic growth for 2006 and 2008. Data for the 2008th displayed as a histogram, and in 2006 as a line. Chart datum, in this case, leave the default, ie to the left axis Y. The figure below shows a diagram constructed:
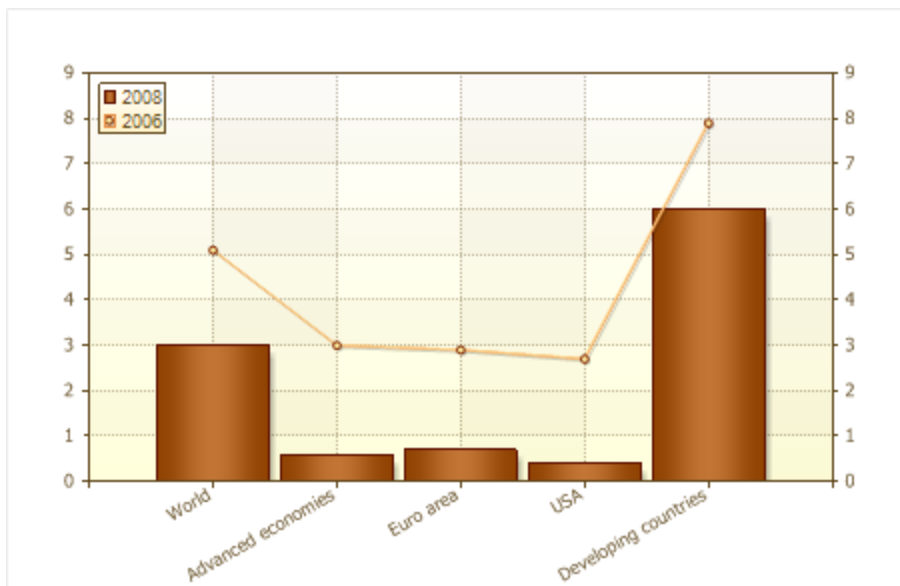
As can be seen from the picture, in general, global economic growth by region for 2006 was higher than in 2008. In this case, the report generator will generate the left Y-axis by choosing the maximum value of the columns of data in those rows that are tied to it, ie, from the column data in bar charts and line. And then, build graphs for the axis Y. If the right Y-axis is enabled, the value of this axis will be duplicated on the left axis Y. Now change the example slightly, we establish a number of anchor line (Line) to the right Y-axis and construct a graph. The picture below shows a diagram with reference to the right and left axis Y, different series:
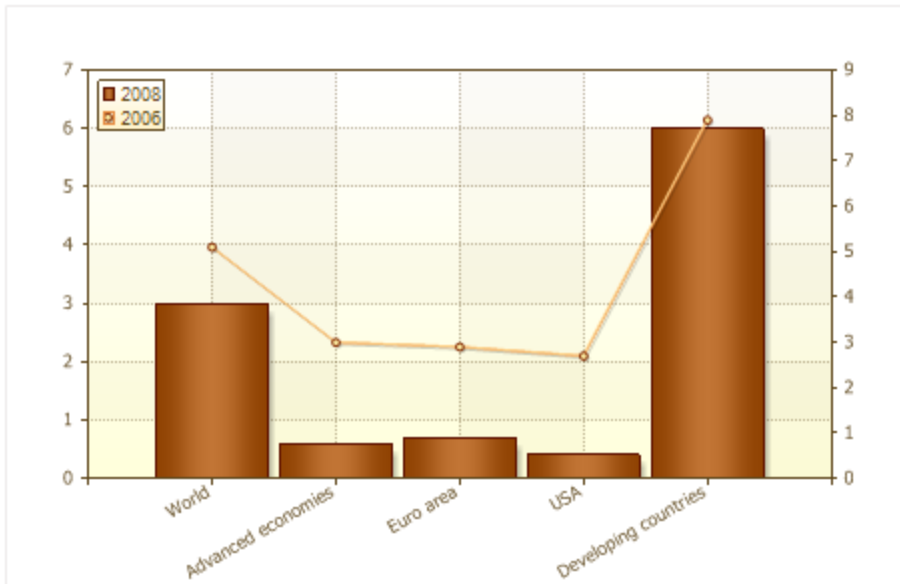
As can be seen from the picture, the value and dynamics of global economic growth have not changed. But the values of the left and right Y-axis are not identical. In this case, a report generator built on the left Y-axis maximum value from a column of data series that is tied to the left axis, ie by the maximum value from the histogram and the right axis Y - by the maximum value at the line. It is also worth noting that you can specify a different axis, and for the series of the same type. The picture below shows two diagrams (on the left - both series are tied to the left axis Y, on the right - first row to the left axis, the second - to the right):
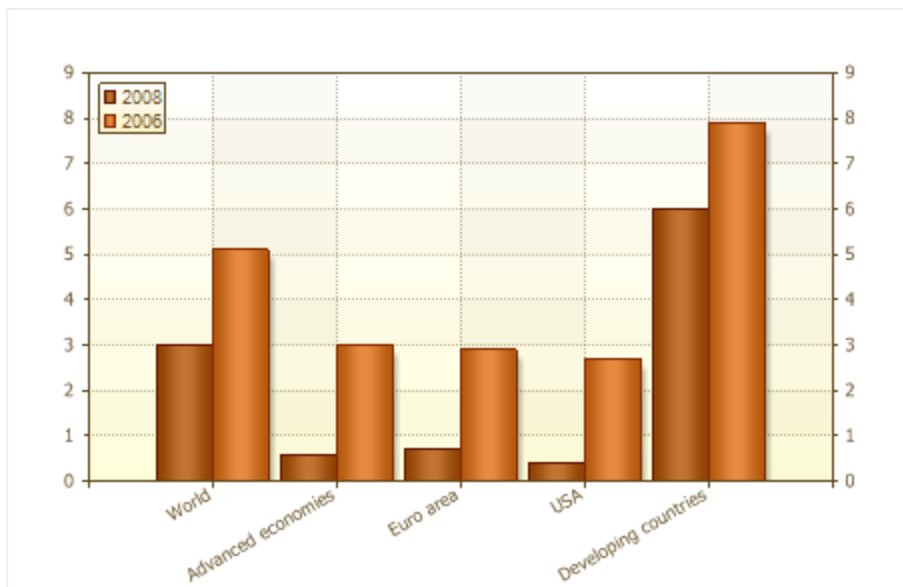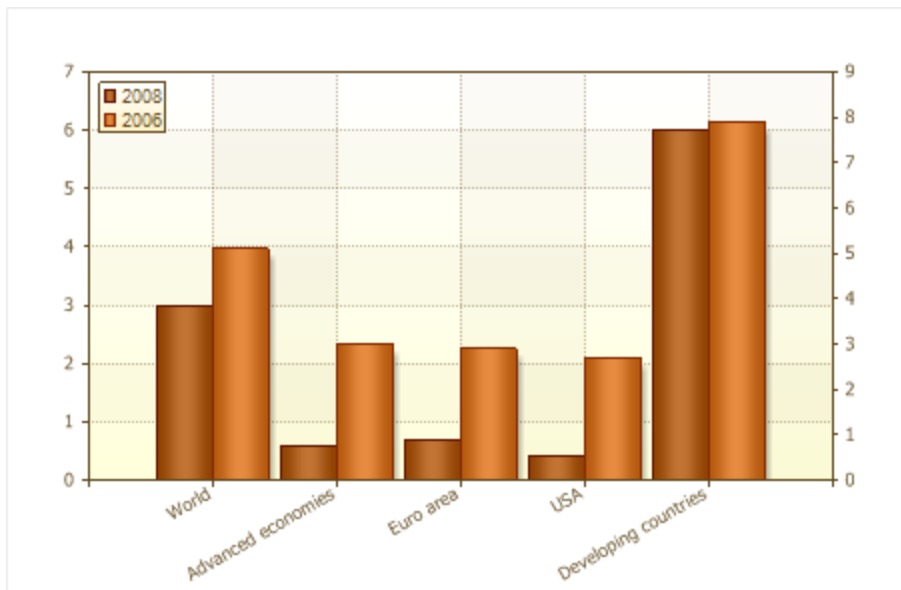
As can be seen on the diagram, where the binding is to a single axis, it is better visible the dynamics of growth (or loss), but at the same time, if the values of one series would be great, and the second is considerably small, should be used to bind to different axes. This will enable even the smallest value to visualize. Also, it should be understood that the rows of stacked rows of binding to different axes Y is incorrect, because This contradicts the method of charting the accumulation.

3.29.5.7 Top N

Using the group of properties Top N you may highlight the maximum or minimum values in the chart, and the rest one group into a single value. Grouped value is a sum of all values that were not identified. Features offered by the group of properties Top N, can be applied in different cases: when the chart has many values but it is needed to allocate a certain amount of the maximum (minimum) ones or, for example, if you want the chart to display the difference between the maximum (minimum) values and set other values. Let's consider the properties of Top N in more detail.

1. The **Count** property provides the ability to determine the number of values that will be displayed and will not be subject of grouping. If this property is set to 2, then it means that the two maximum (minimum) values will be displayed, and the rest are grouped into a single value.
2. Depending on the value of **Mode** property will be allocated the maximum or minimum values. If the **Mode** property mode is set to **Top**, the maximum values will be highlighted, and if the property is set to Bottom - the minimum ones will be selected. If

the **Mode** property is set to **None**, then all the values in the fields of the properties **List of Value**, or **Value Data Column** will be displayed.

3. Specify the signature of the argument values grouped, you can use the properties of the Other Text. If the field is empty for this property, the signature of the argument have grouped the values will be absent.

4. Displaying or not hiding the grouped property value provides an opportunity to Show Other. If this property is set to true (true), then this value is shown in the diagram, and if the value lies in the (false) - a group the values are not displayed.

Consider the possibilities offered by a group of Top N properties as an example. There is a report that plotted on the population in some states of America. The picture below shows this diagram:
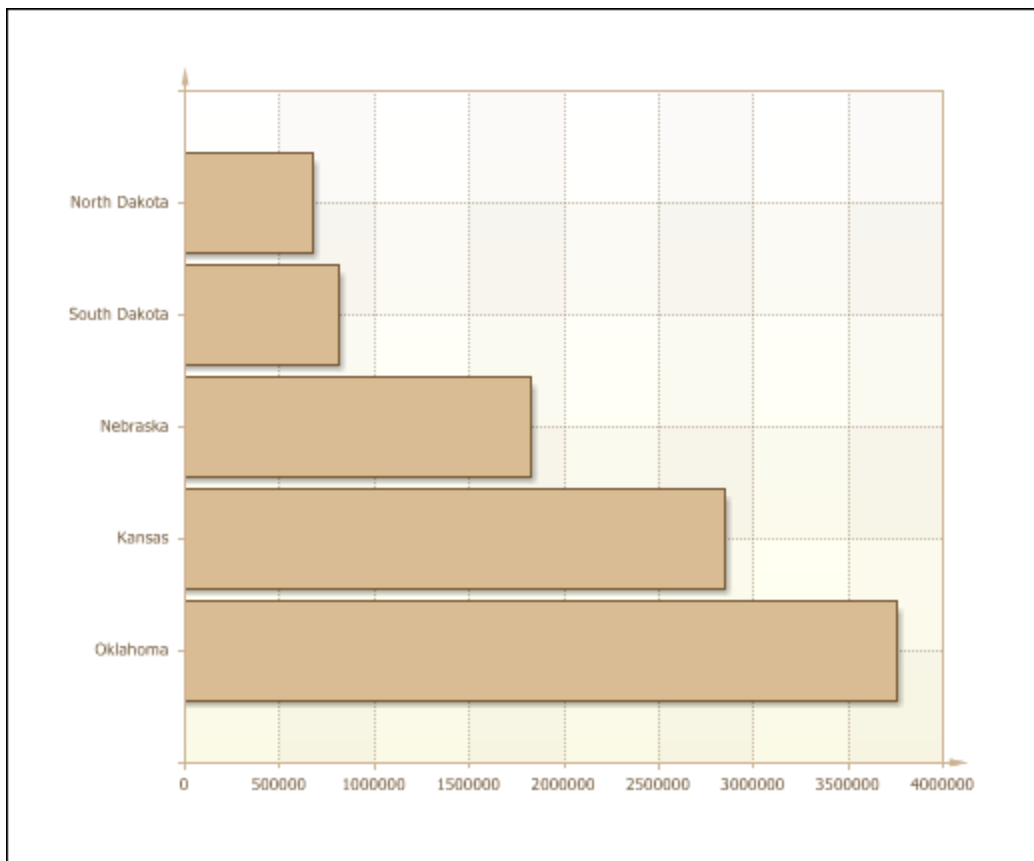
As you can see from the picture, the population of Oklahoma is the largest in the diagram. For example, to visually display the differences in the population of Oklahoma and the total population of other states in this diagram. Define the property values of Top N. Since it is necessary to allocate a single maximum value (population of Oklahoma), the number of property (Count) should be set to 1, and the **Mode** property -

is set to Top. If you want you can add a signature argument of the aggregate value. In this example, the property Other Text define to be the Other. Show Other property also must be set to true (true), as in this example, the goal is to visually display the differences between populations in Oklahoma and other states in this diagram. The picture below shows a diagram with the properties of the applied group Top N:
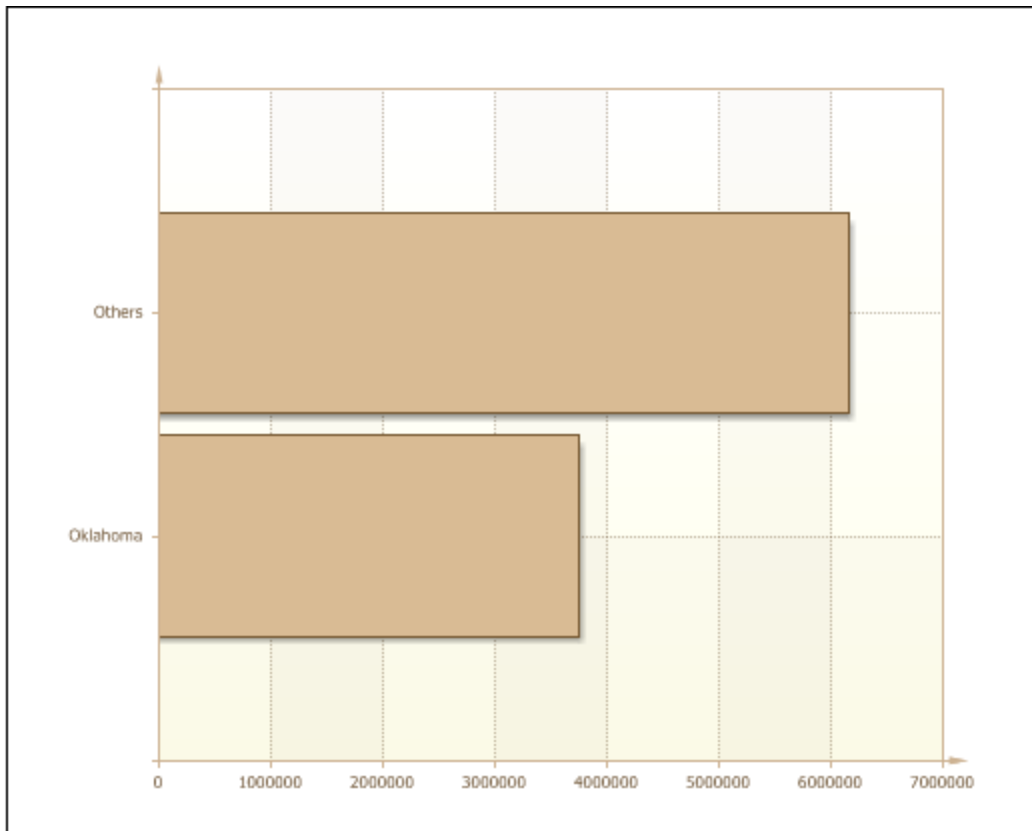
As can be seen from the picture, the other values were grouped into a single value with the signature of an argument Other. Out of the diagram shows that the total population exceeds the population of the four states of Oklahoma. Consider another example. There is a chart with a set of values, in this case the products and their prices. The picture below shows a diagram:
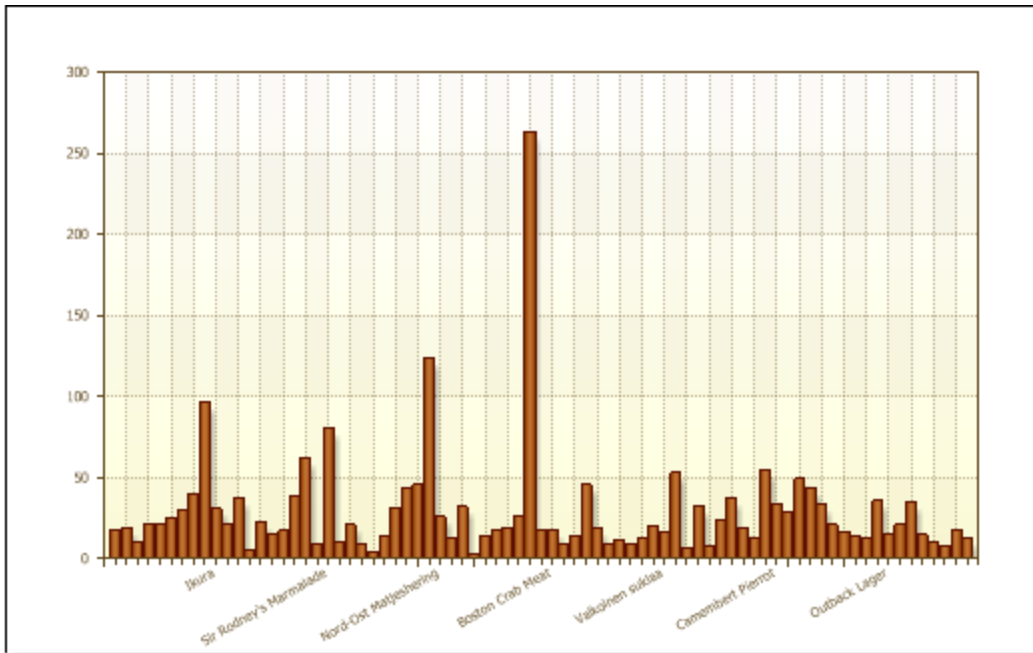
As the picture shows, visually, this picture is seen with difficulty, and select the maximum (minimum) value is problematic. In this example, we select 5 products to the most minimal prices. To do this, set the **Count** property in the value 5, the **Mode** property - is set to Bottom, Other Text property field is left blank, because the property is set to Show Other value **false**. The picture below shows a diagram with the properties of the applied group Top N:
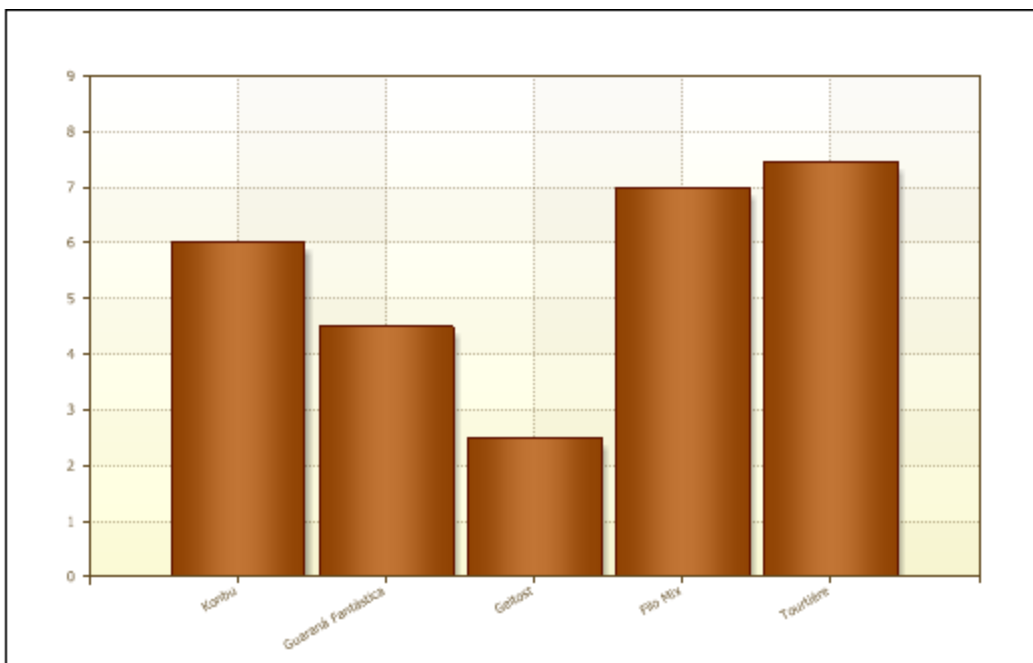
As can be seen from the picture, a kind of filtering is performed, ie Report Generator has identified five minimum values, and the rest grouped into a single value. Because the property found in the Show Other value lies (false), then grouped the value does not appear on this chart.

3.29.6 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:
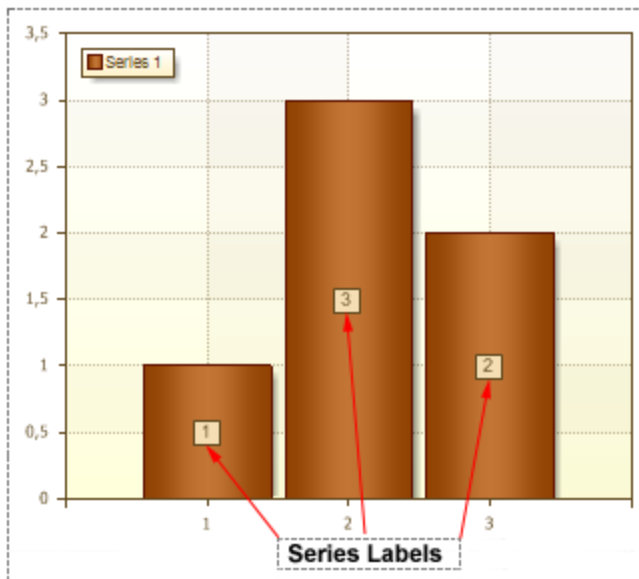
The **Series Labels** property is used to indicate position of series labels. The list of available options for this property depends on the type of chart. Also, the **Series Labels** property have some options that are used to change settings of Series Labels.

3.29.6.1 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.

3.29.6.1.1 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.
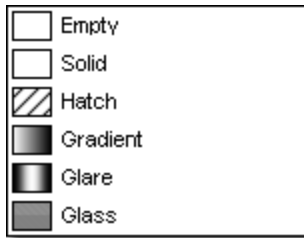
Six types of Brushes are available within Stimulsoft Reports:
- ✓ **Empty**
- ✓ **Solid**
- ✓ **Hatch**
- ✓ **Gradient**
- ✓ **Glare**
- ✓ **Glass**

Below are representations of the results all six Brush types:
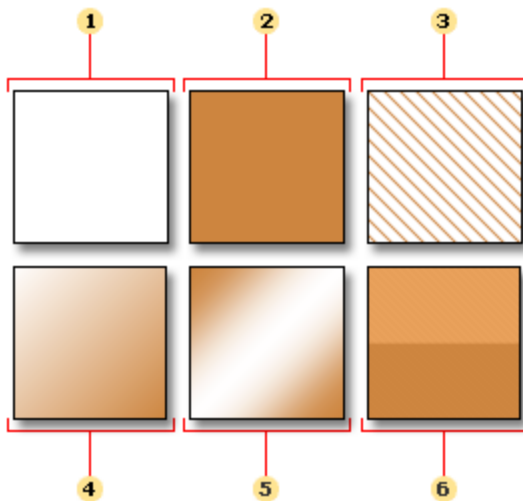
1 Empty. The background of a Series Label is transparent.

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 selected texture 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 angle can be 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**:
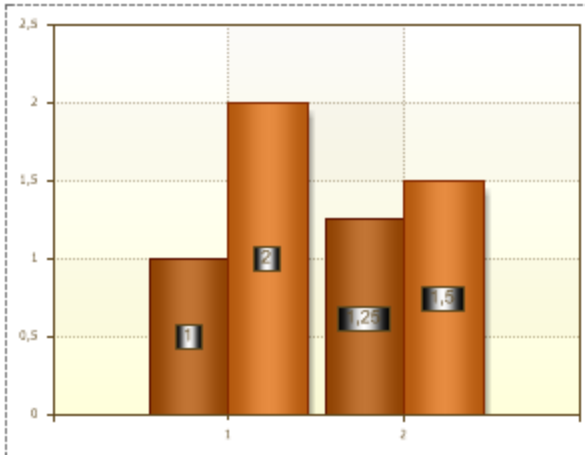
3.29.6.1.2 Font Property

The font for Series Labels can be set using the **Font** property within the Object Inspector.

Selecting font
Series Labels within a report can be output using different fonts. Three examples fonts are shown below:

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
<u>AaBbCcDd</u>
~~AaBbCcDd~~

The picture below shows a chart with text set to **Arial, Bold** style**,** font size - **12:**
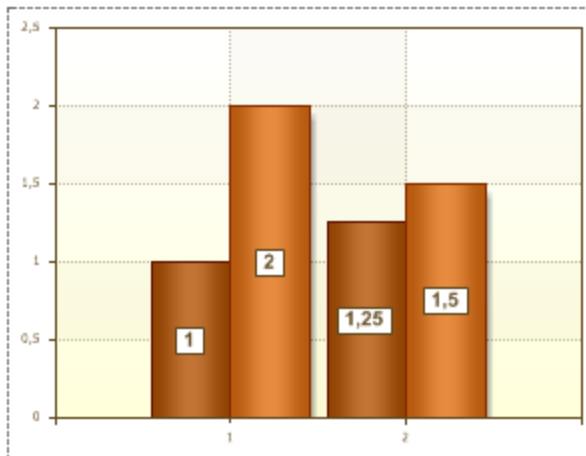
3.29.6.1.3 LabelColor Property

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**:
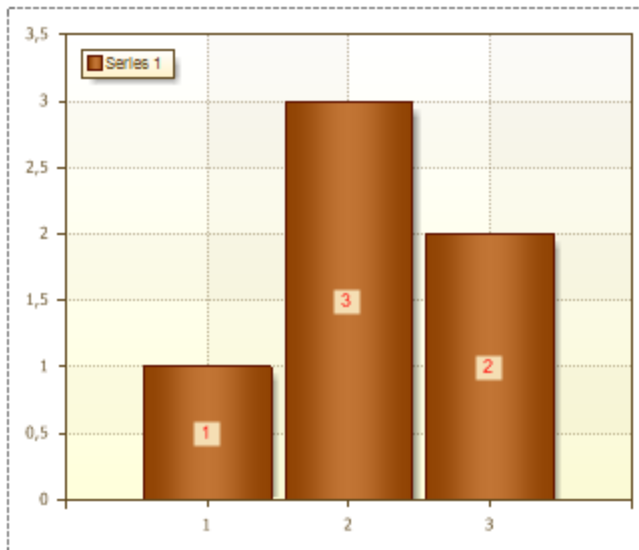
3.29.6.1.4 UseSeriesColor Property

The **UseSeriesColor** property is used to make the border color and the series label color match to the color of the series. If the **UseSeriesColor** 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 **UseSeriesColor** property set to **false**:
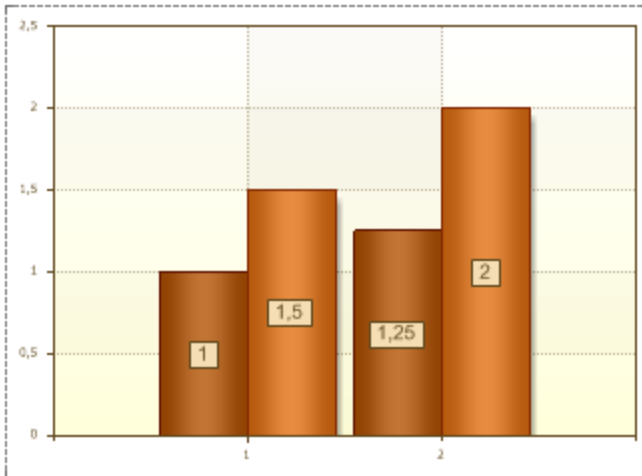
If the **UseSeriesColor** property is set to **true**, then the border color and series labels color will match to the color of series. The picture below shows an example of a chart, with the **UseSeriesColor** property set to **true**:
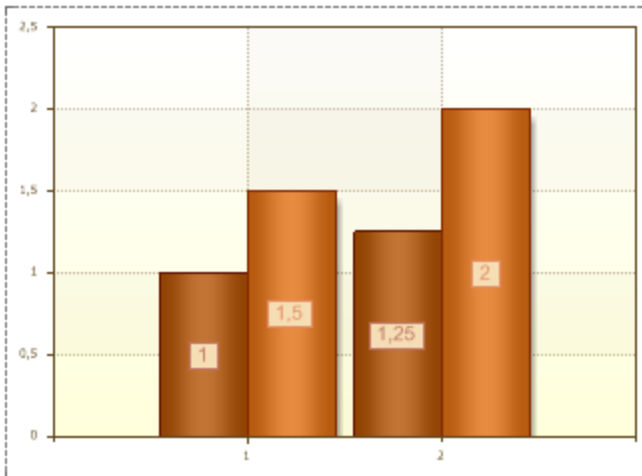
3.29.6.1.5 Angle Property

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**:
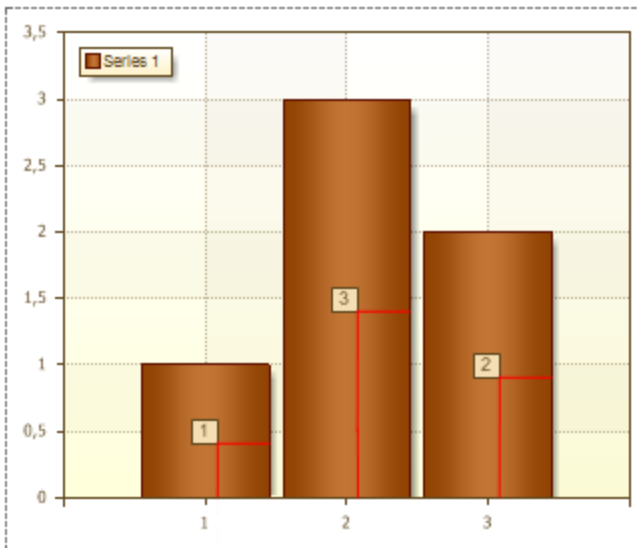
The value of the property can be negative and positive. If a value of the property is negative then Series Label is inclined anticlockwise. If the value of the property is positive then Label in inclined clockwise. The picture below shows a chart sample, which the **Angle** property for Series Labels is set to **45**:
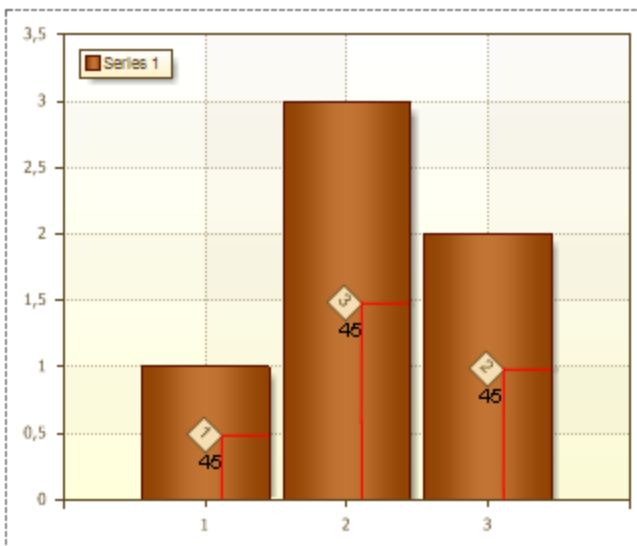
3.29.6.1.6 Draw Border Property

The **DrawBorder** property allows showing/hiding a border of Series Labels. It has two values: **true** and **false**. If the **DrawBorder** is set to **true**, then the border is shown. The picture below shows a chart with borders around Series Labels (the borders are red):
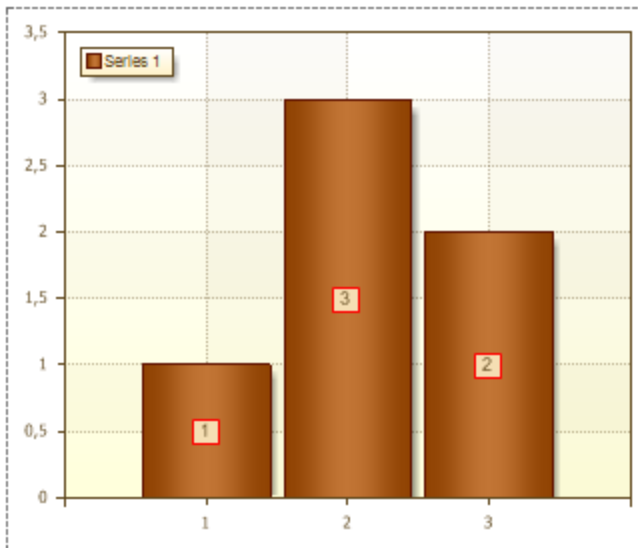
If the **DrawBorder** is set to **false**, then the border is hidden. The picture below shows a chart without borders around Series Labels:
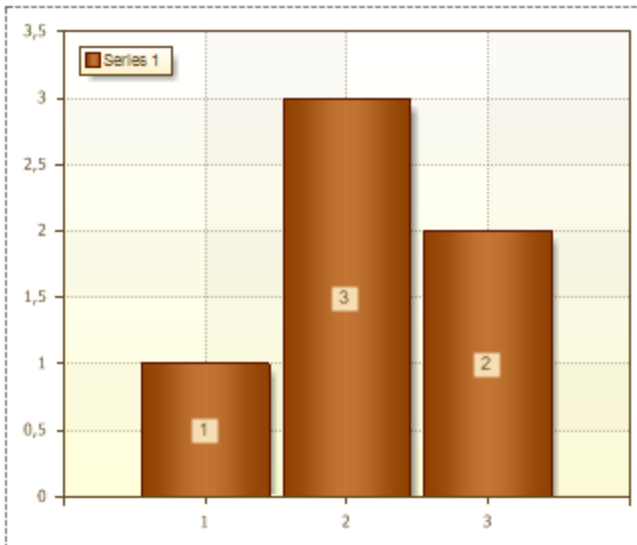
3.29.6.1.7 BorderColor Property

The **BorderColor** property is used to change the border color of Series Labels. The picture below shows a chart which Series Labels borders are blue:
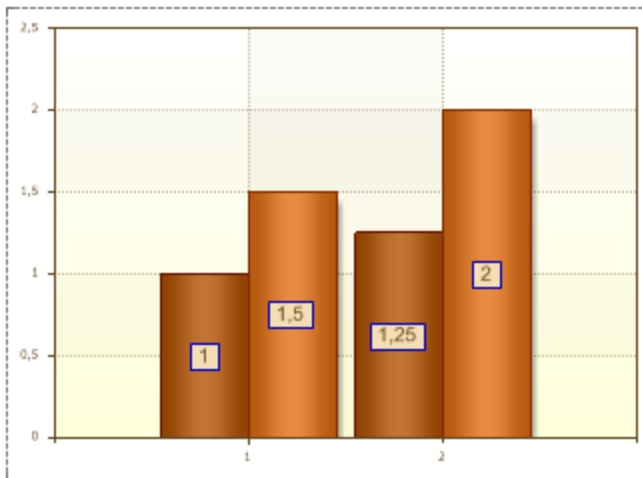
3.29.6.1.8 Antialiasing Property

The **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**:
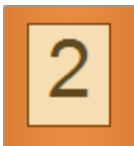
3.29.6.2 Format Property

The **Format** property is used to format the contents of Series Labels. This property has multiple values.

➤ **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**:
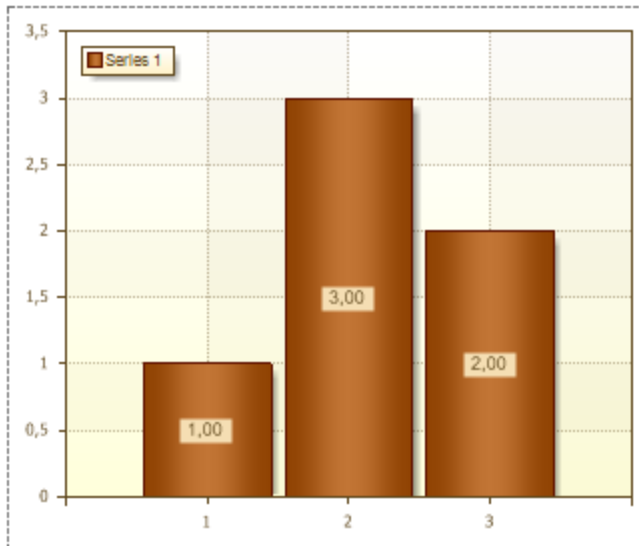
❯ **Currency.** The **C** value of the **Format** property is used to display Series Labels with a currency symbol. After the **C** value, it is possible to specify the number of decimal places that you want to use. The picture below shows a chart with the **Format** property of Series Labels set to **C**:
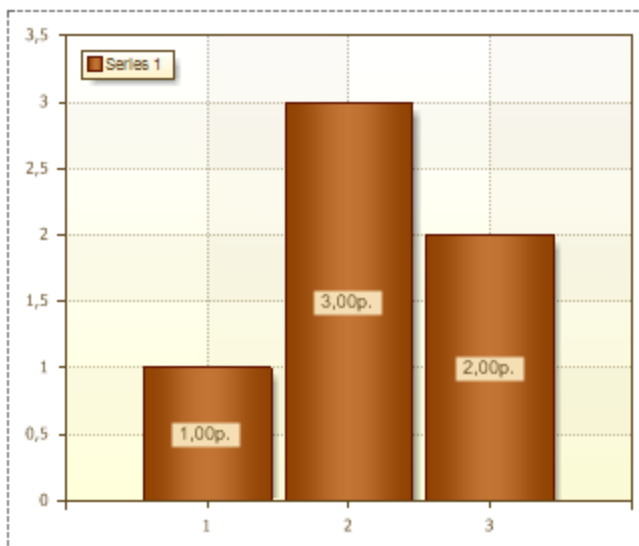
❯ **Percentage**. The **P** value of the **Format** property is used to display Series Labels with percent symbol. After the **P** value, it is possible to specify the number of decimal places that you want to use. The picture below shows a chart with the **Format** property of

Series Labels set to **P**:
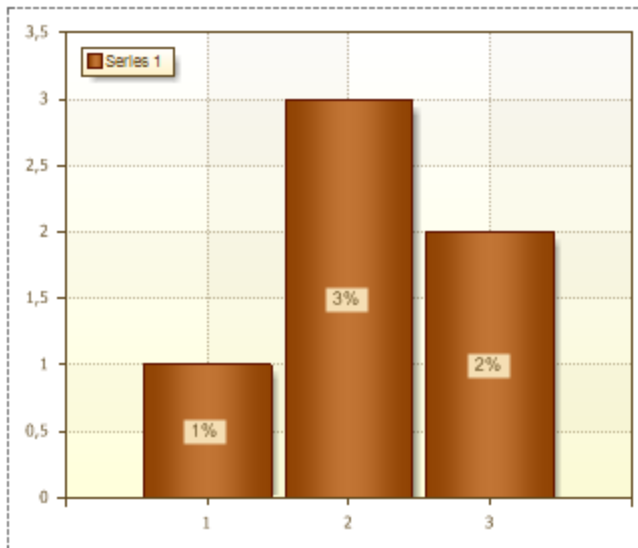
> **Date**. The **MM/dd/yyyy**, **MMMM dd**, **yyyy MMMM** values of the **Format** property convert values of arguments to date. **MM/dd/yyyy** - the date is shown like "01.20.2010", **MMMM dd** - the date is shown like "September 29", **yyyy MMMM** - the date is shown like "2010 March". The picture below shows a chart and with the **Format** property set to **MM/dd/yyyy**
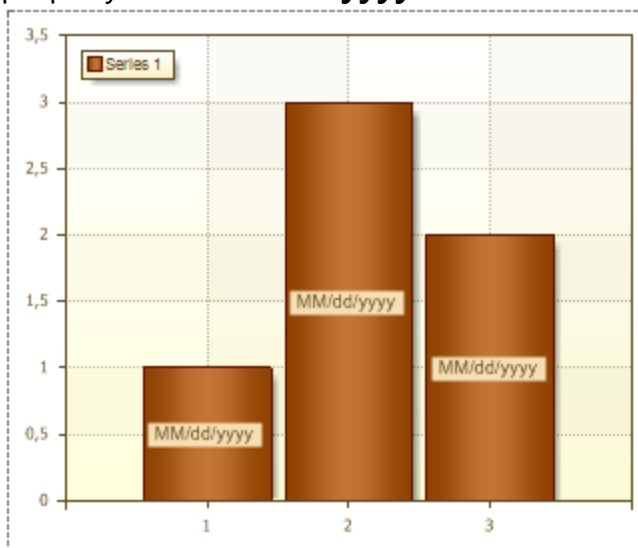
To reset the **Format** property of selected cells, and return to the default format, clear the Format by selecting empty field.

3.29.6.3 ValueType Property

The **ValueType** 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**.

❯ **Value**. The Series Labels are series values. The picture below shows an example of a chart with the **Value Type** property set to **Value**:
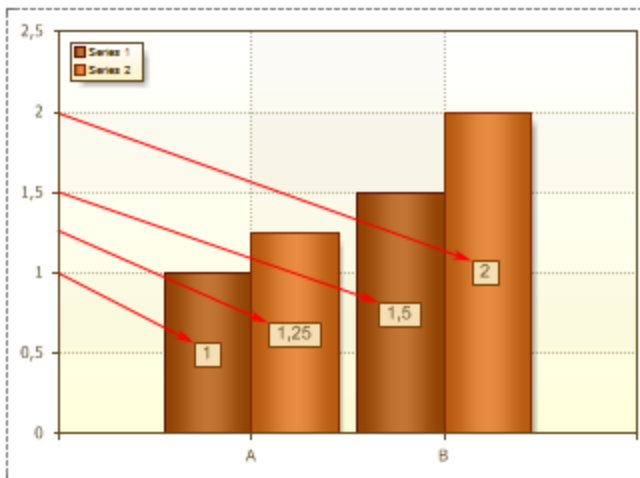
❯ **Series Title**. The Series Labels are records in the **Title** field in the **Series Editor**. The picture below shows an example of a chart with the **Value Type** property set to **Series Title**:
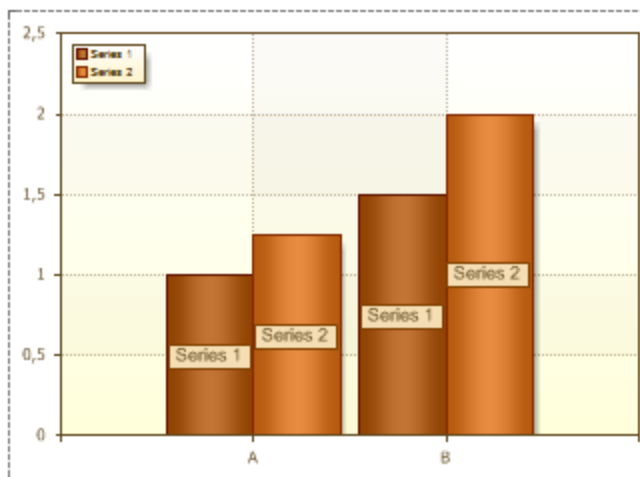
❯ **Argument**. The Series Labels are the arguments. The picture below shows an

example of a chart with the **Value Type** property set to **Argument**:
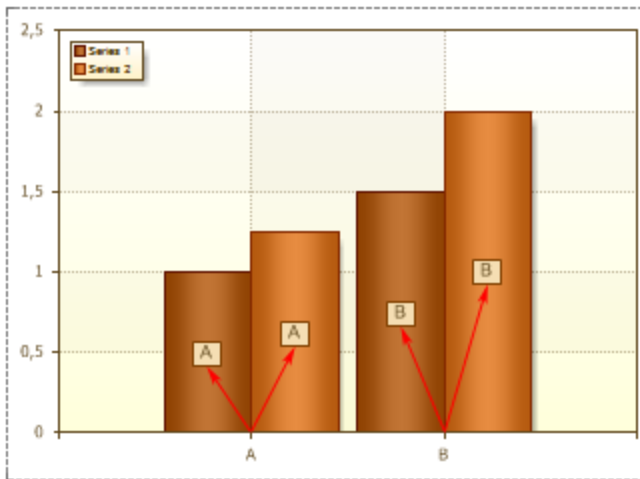
> **Value - Argument**. The Series Labels are **Values** and **Arguments** of series. The picture below shows an example of a chart with the **Value Type** property set to **Value - Argument**:
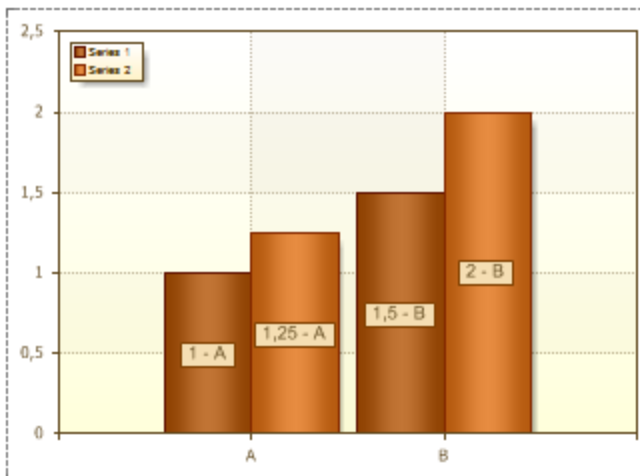
> **Argument - Value**. The Series Labels are **Arguments** and **Values** of series. The picture below shows an example of a chart with the **Value Type** property set to **Argument - Value**:
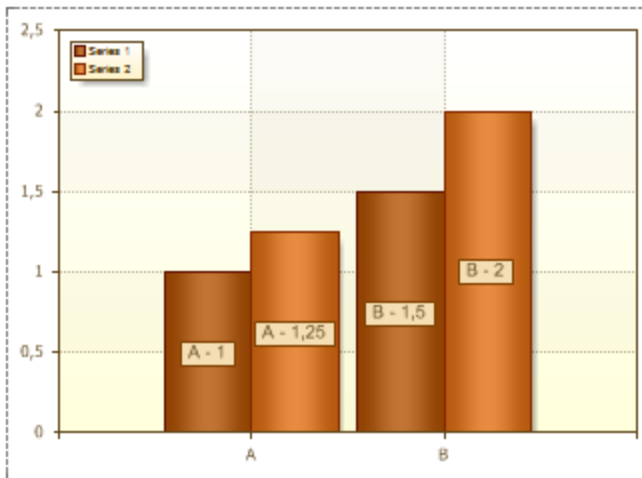
❯ **Series Title - Value**. The Series Labels are **Series Titles** and **Values**. The picture below shows an example of a chart with the **Value Type** property set to **Series Title - Value**:
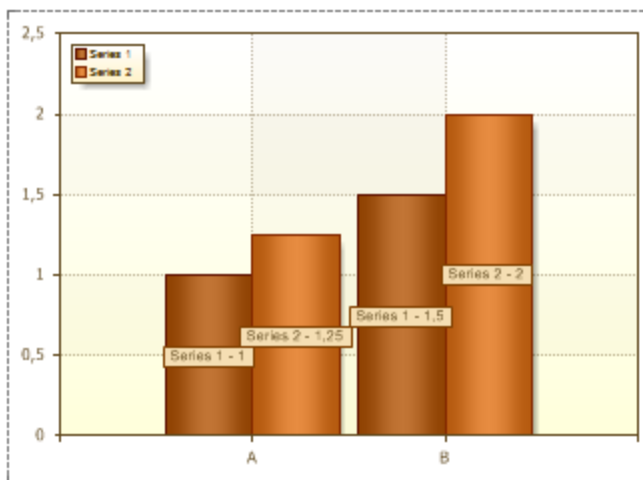
❯ **Series Title - Argument**. The Series Labels are **Series Titles** and **Arguments**. The picture below shows an example of a chart with the **Value Type** property set to **Series Title - Argument**:
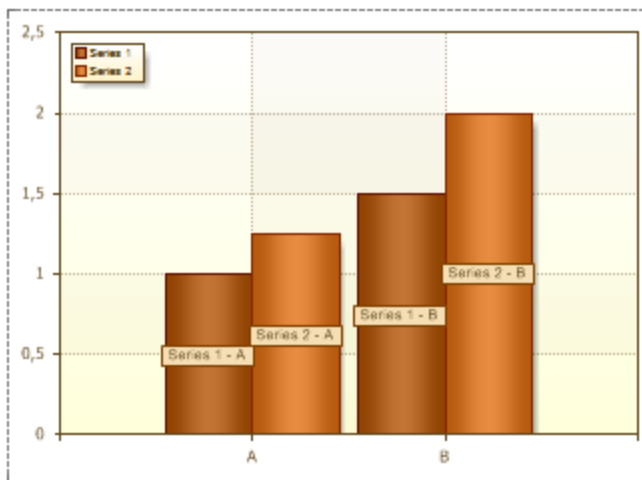
3.29.6.4 ValueType Separator

The **ValueTypeSeparator** property is used to change the type of values separator in the series labels. By default, the **ValueTypeSeparator** property is set to **'-'**. Any character or group of characters typed in the field of the **ValueTypeSeparator** property, will be the delimiter (including the **'space'**). If the field is unfilled, then the separator is a **'space'**.

3.29.6.5 PreventIntersection Property

The **PreventIntersection** property is used to avoid overlapping between series labels and with the borders of rendered values and axes. By default, the **PreventIntersection** 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 **PreventIntersection** property set to **false**:
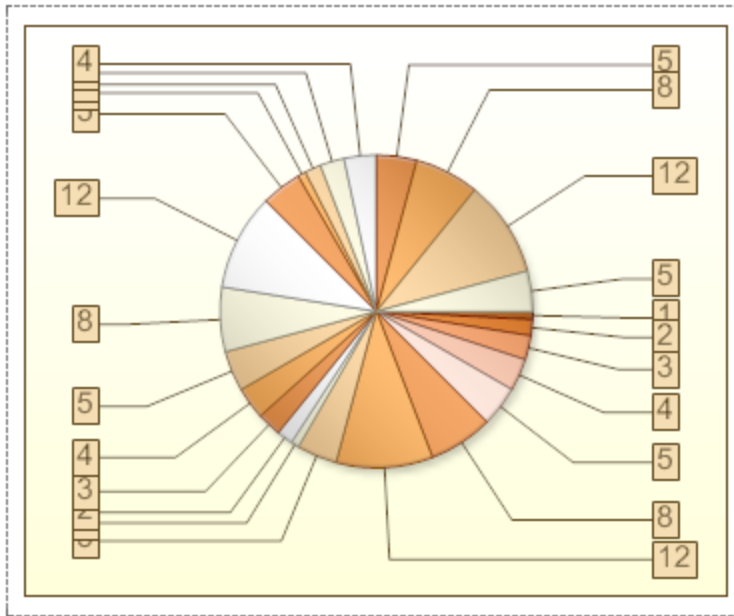
If the **PreventIntersection** property is set to **true**, then the series labels will not overlap. The picture below shows an example of a chart, with the **PreventIntersection** property set to **true**:
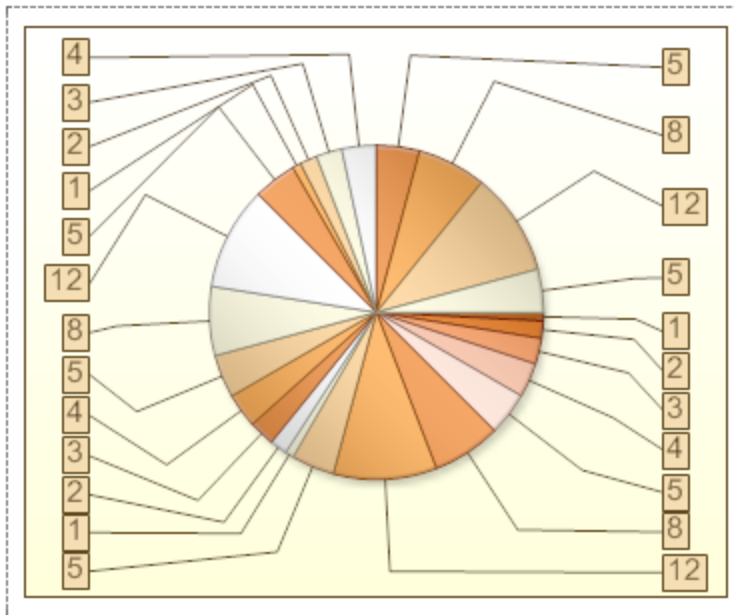
3.29.6.6 ShowOnZeroValues Property

Sometimes, when designing charts, 0 values of series can be met. Series labels of zero values can be displayed. The **ShowOnZeroValues** property is used to show/hide these

series labels. If the **ShowOnZeroValues** 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 **ShowOnZeroValues** property is set to **false**:
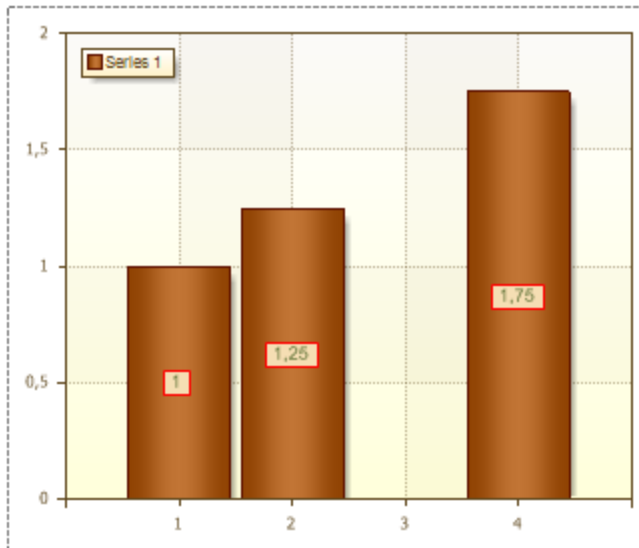
In this chart the 3rd argument is 0, and the series labels is not displayed. If the **ShowOnZeroValues** property is set to **true**, then series labels of zero values will be shown.The picture below shows an example of a chart with a zero value and the the **ShowOnZeroValues** property is set to **true**:
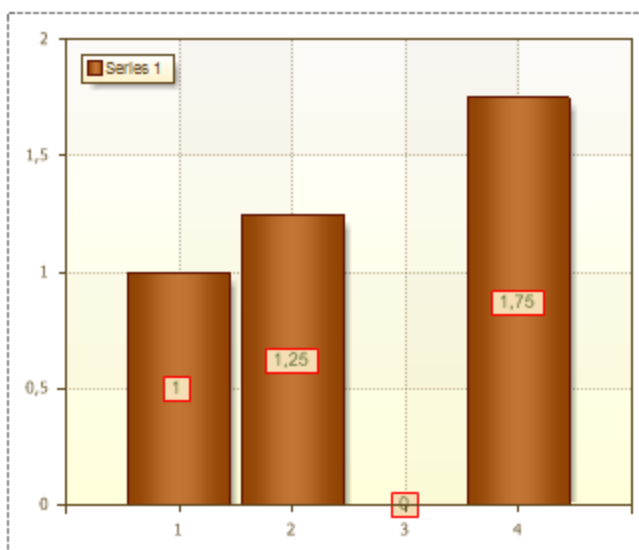
As can be seen from this picture, the 3rd argument is 0, and its title was shown.

3.29.6.7 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**:
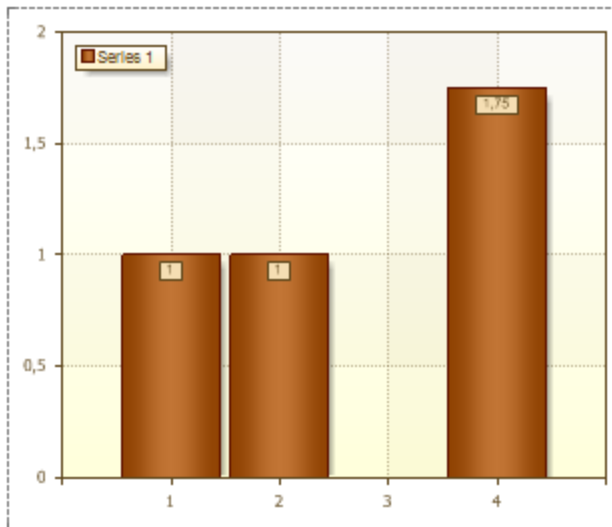
If the **Step** property is set to **2**, then Series Labels will be shown as it is shown on picture below:
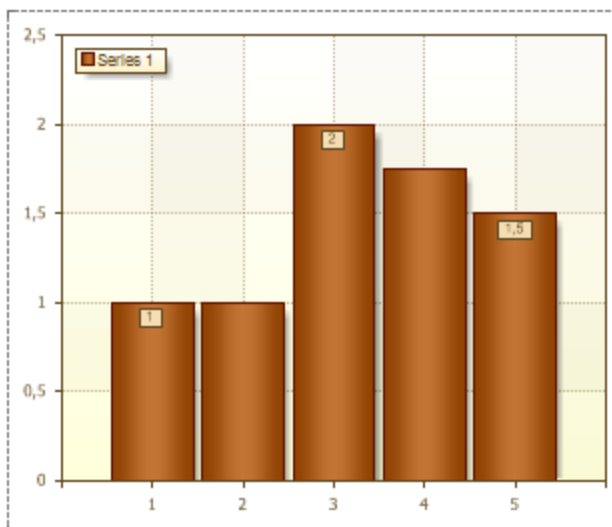
The value **1** of the **Step** property indicates that Series Labels will be shown for each value of Series.

3.29.6.8 TextBefore and TextAfter Properties

The **TextBefore** and **TextAfter** properties allow 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):
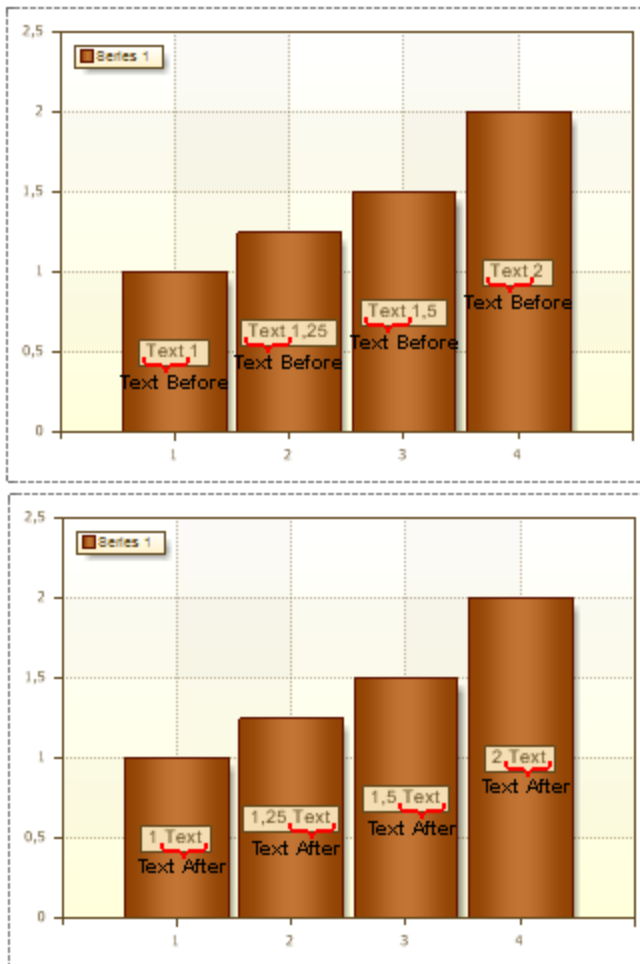
3.29.6.9 Visible Property

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:
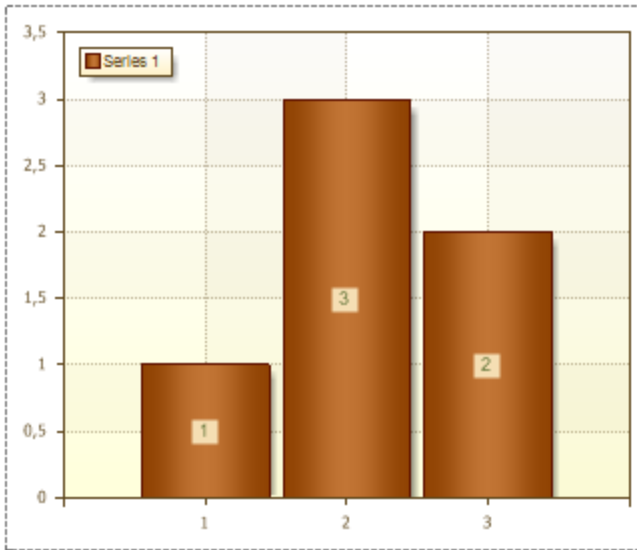
If the **Visible** property is set to **false**, then Series Labels are not displayed. The picture below shows a chart with hidden Series Labels:
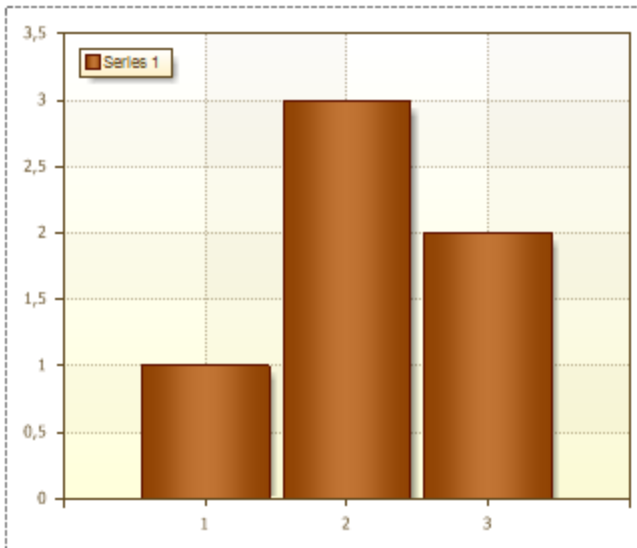
By default, the **Visible** property is set to **true**.

3.29.6.10 Marker

The **Marker** is an icon that is shown near the Series Labels. It is possible to change height and width of the **Marker**. The **Marker** takes the color of Series. The picture below shows a chart with **Markers**:

3.29.6.10.1 MarkerVisible Property

If to set the **MarkerVisible** property to true then the **Marker** is shown. By default, the **MarkerVisible** property is set to **false** and Markers are not visible. The picture below shows a chart with the **MarkerVisible** property set to **false**:
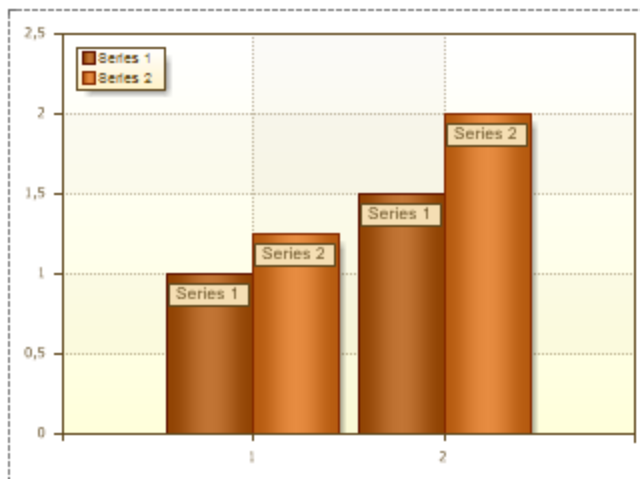
The picture below shows a chart with the **MarkerVisible** property set to **true**:
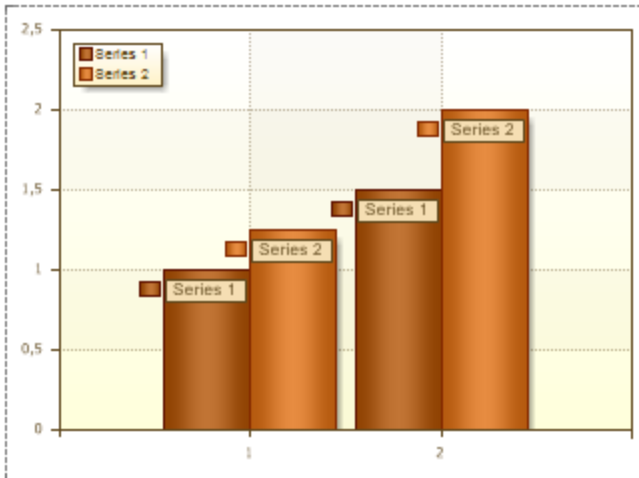
3.29.6.10.2 MarkerSize Property

It is possible to change height and width of a **Marker**. The **MarkerSize** property is used for this. It is possible to change **Height** and **Width** of a Marker. Marker Height and Width are set in pixels. If both values are more than **0**, then the Marker is shown.

3.29.6.10.3 MarkerAlignment Property

The **MarkerAlignment** property allows aligning a marker on the left or right of Series Labels. If the **MarkerAlignment** property is set to **Right**, then the marker is aligned to the left of Series Labels. The picture below shows the Markers aligned left:
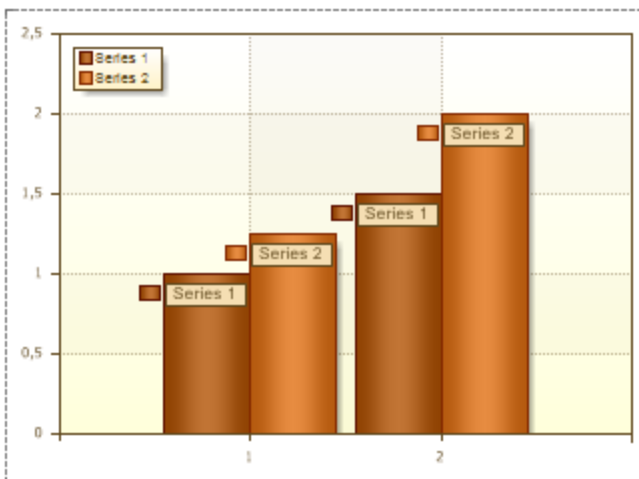
If the **MarkerAlignment** property is set to **Right**, then the marker is aligned to the

right of Series Labels. The picture below shows the Markers aligned right:
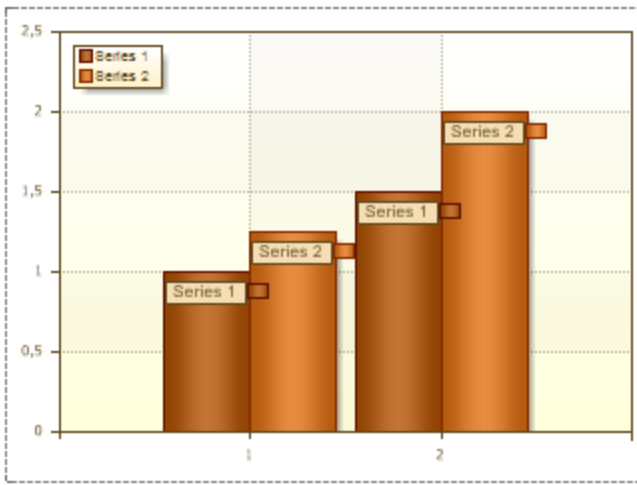
By default, the **MarkerAlignment** property is set to **Left**.

3.29.7 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:
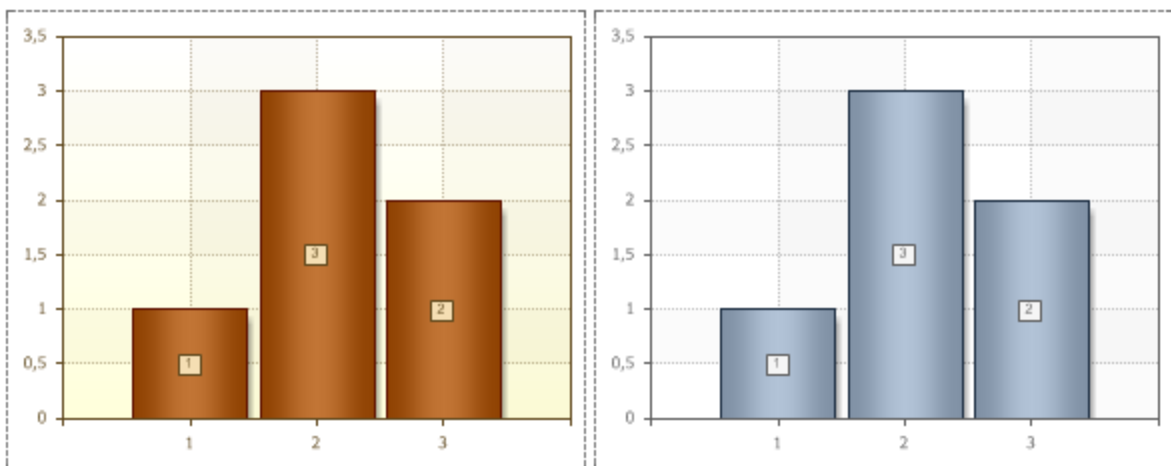
3.29.7.1 AllowApplyStyle Property

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.

3.30 Maps

Watch our [videos how to create maps in the report designer](#).

The Map component represents a tool to visualize data with reference to geographical location. With the help of maps, you can display various statistics for a particular region or in the world. For example, you can display the sales of any product for each state in the US or, for example, for each European country. The map may be placed directly on the page or other components like panels, bands, clones etc. Data for maps may be filled manually or obtained from the data source. The Map component can be setup in the component editor. In order to add the Map component in the report you should go to the Infographics menu on the toolbox or Insert tab:
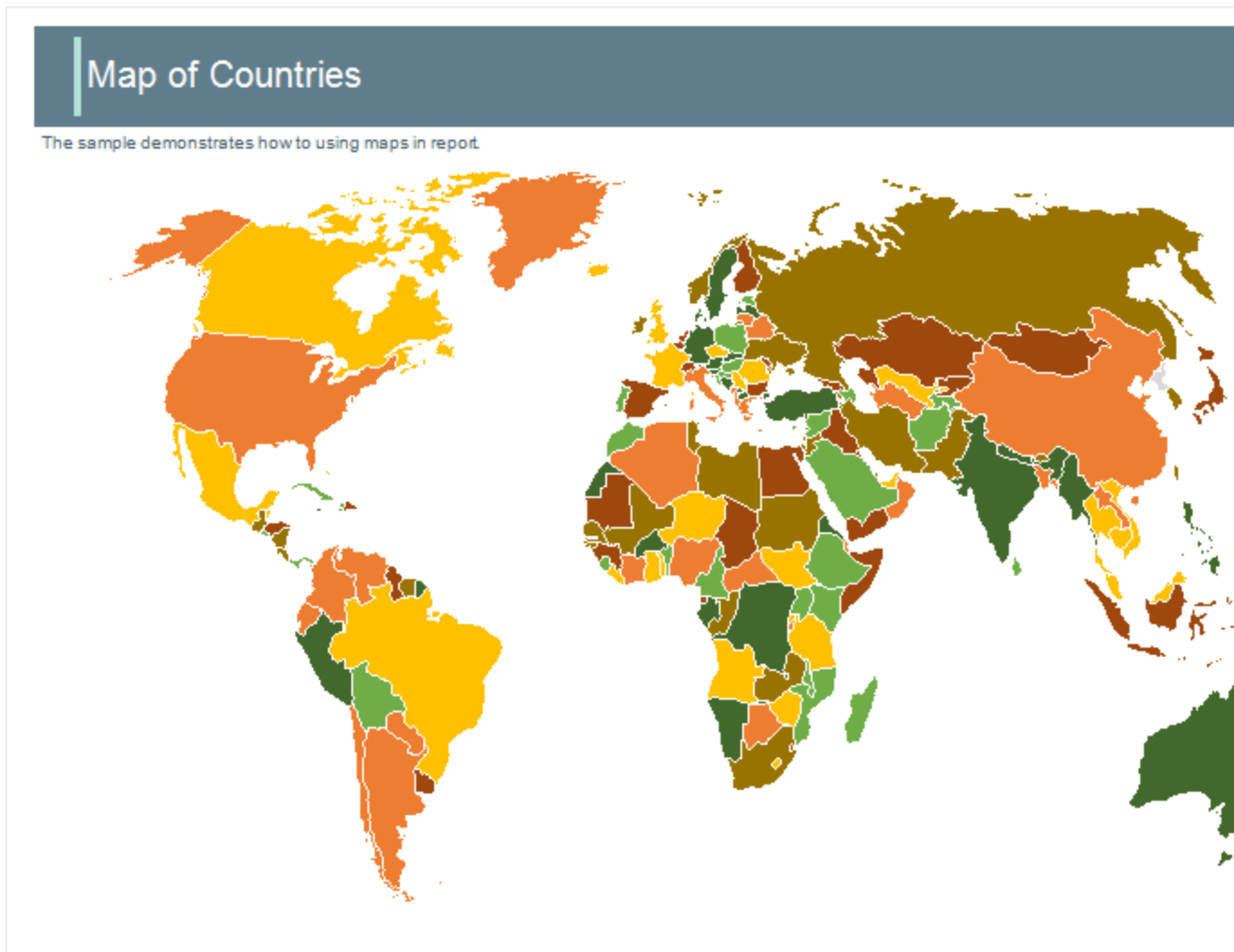
> Map Editor;
> Map Types;
> Map Keys;
> Data for Maps.

3.30.1 Map Editor

Setting the map can be done in the **Map** component. In order to call the **Map Editor** you should double-click the component in the report template or select the Design item from the context menu of the component. The map editor will be called. It has the following tabs:

> The **Map** tab
On this tab, you can change the look of a future map. You can select the global or

regional map. In this case, regions are grouped by continents. Depending on the type, the map will contain a variety of options:
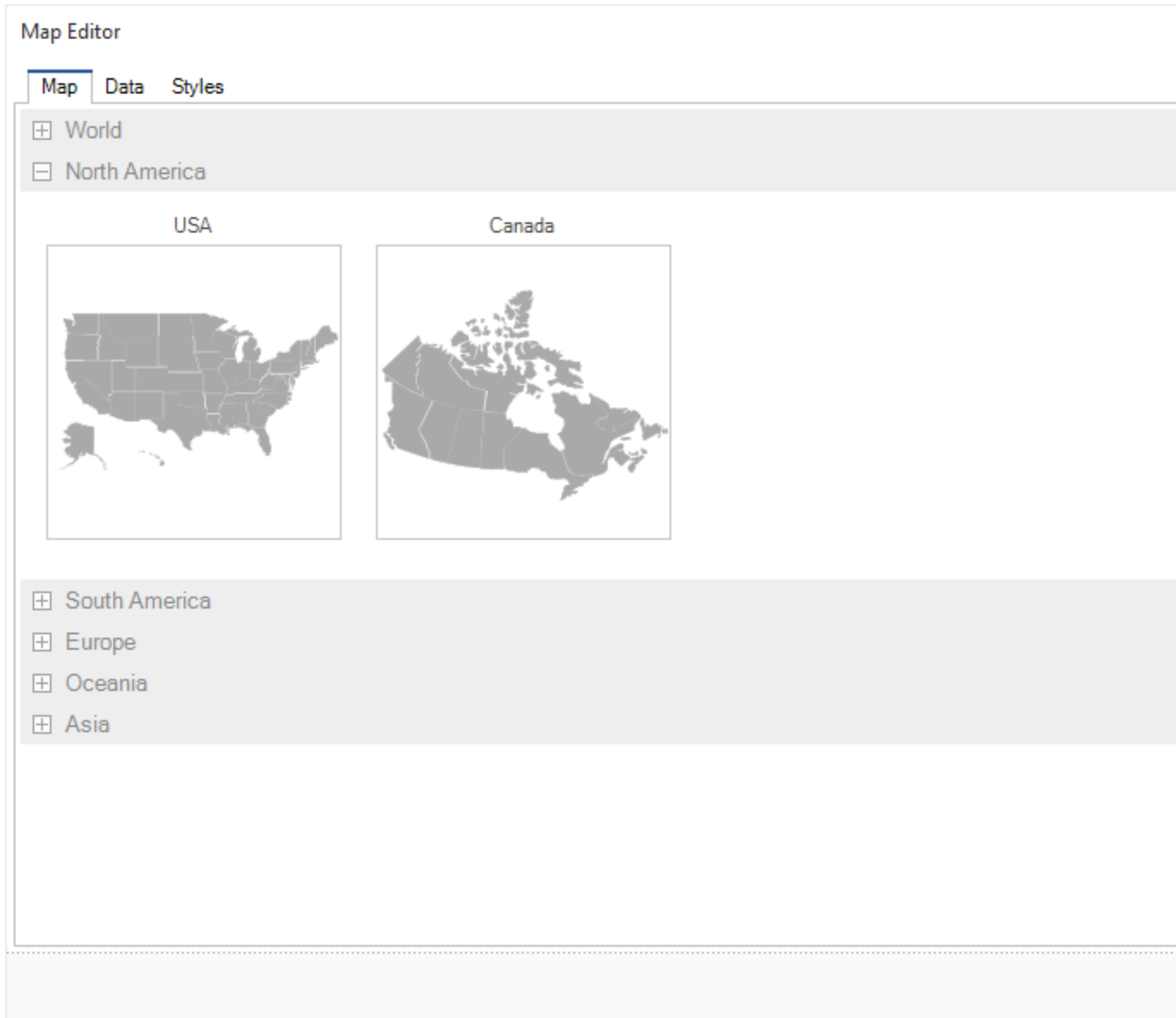
> The **Data** tab

On this tab you can set the map type, the data for the map and map parameters. Data can be entered manually or derived from a data source.

> The **Style** tab
On this tab you can set the map style. There are some preset styles and custom styles in style designer:
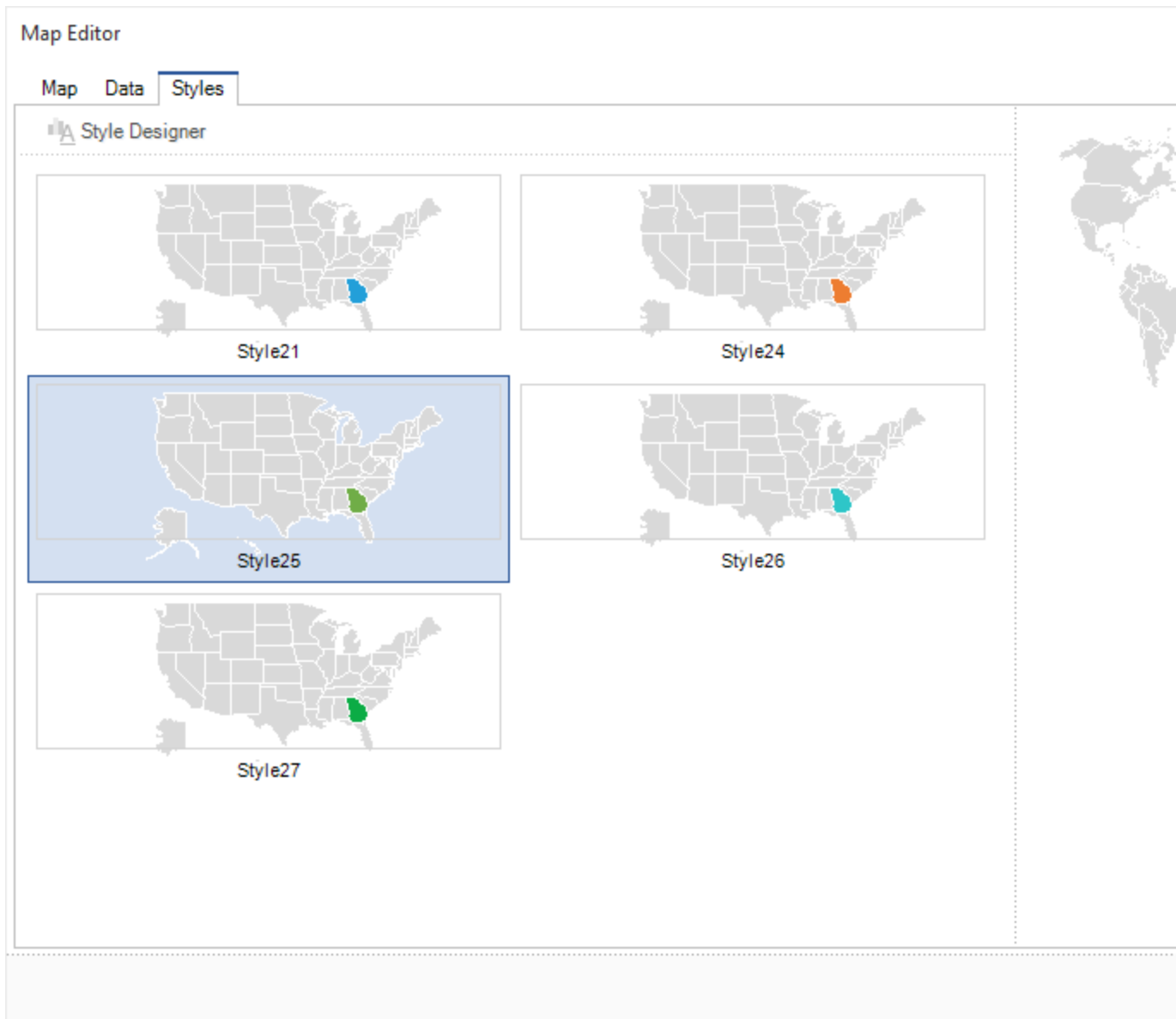
3.30.2 Map Types

Maps can be with a group, heatmap and heatmap with group. You can change the type of a map in the map editor on the Data tab selecting the **Map Type** parameter:

✖ **None**

In this case, the map, regardless of the style or color values of each key, will be filled with the base color. The map will remain to be split on items:

Group

In this case, it is necessary to fill the Group column with data or specify data columns indication in the Group field. Each group of map elements will be drawn with one of the color map style or color that is defined for each entry:
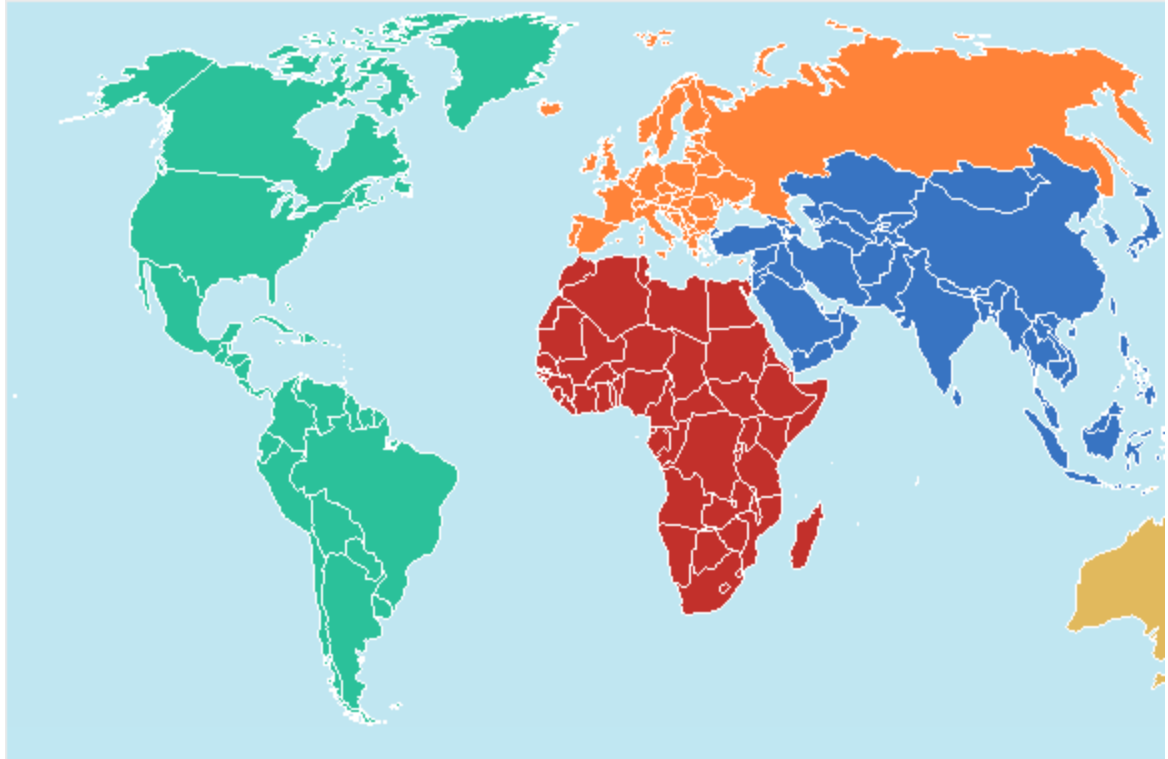
However, if the number of colors in the map style is less than the number of groups of elements of the map, colors can be repeated:
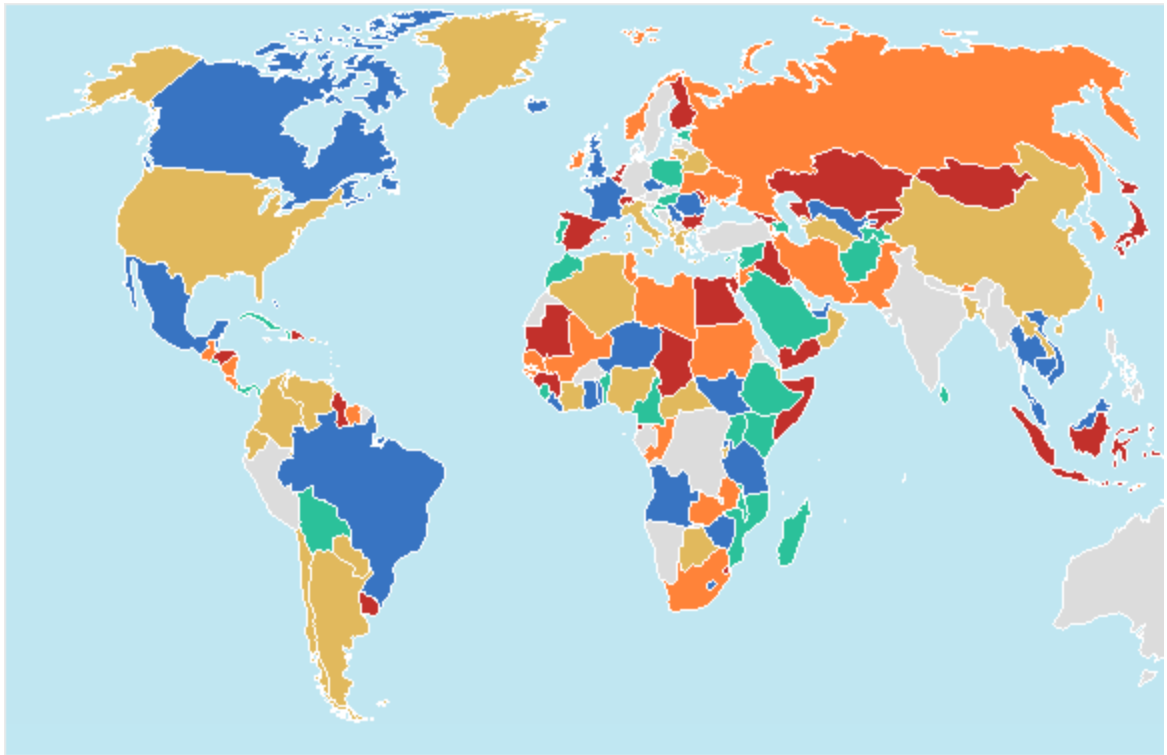
Heatmap

The heatmap provides an opportunity to graphically display values of map items. Therefore, it is necessary to specify the value of map items. Also, in the style of the map, it is necessary to determine the color of the heatmap. The reporting tool, at the time of the report creation process, will check all entries or all the values in the data column. The engine will determine the maximum value and assign to it the first color of the heatmap, and the minimum value will get the second color of the heatmap style. Then, for each value in the list, depending on how close the maximum or minimum values are, shades for values will be created. The shade will be applied to the map item.

⚠ **Notice**: When using a heatmap, the values of groups and elements of the map will not be grouped. If you want to display the heat map by the grouped values, it is necessary to specify the type of map as the heatmap with the group.
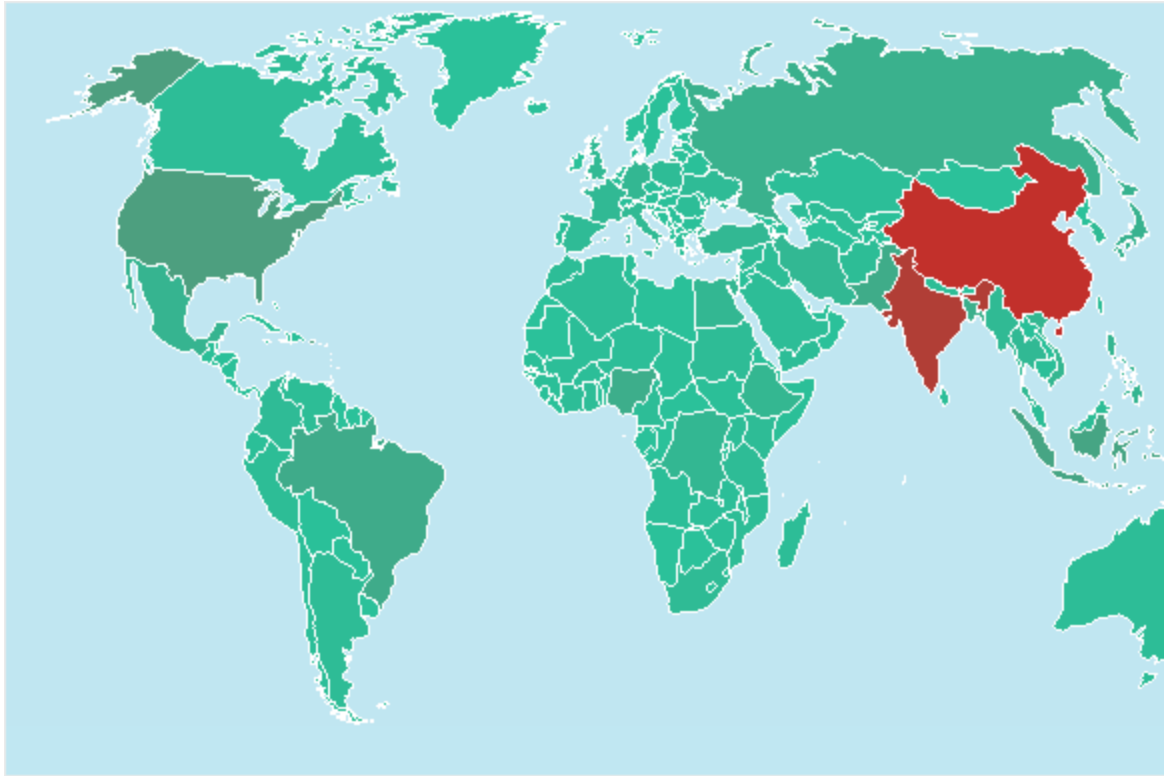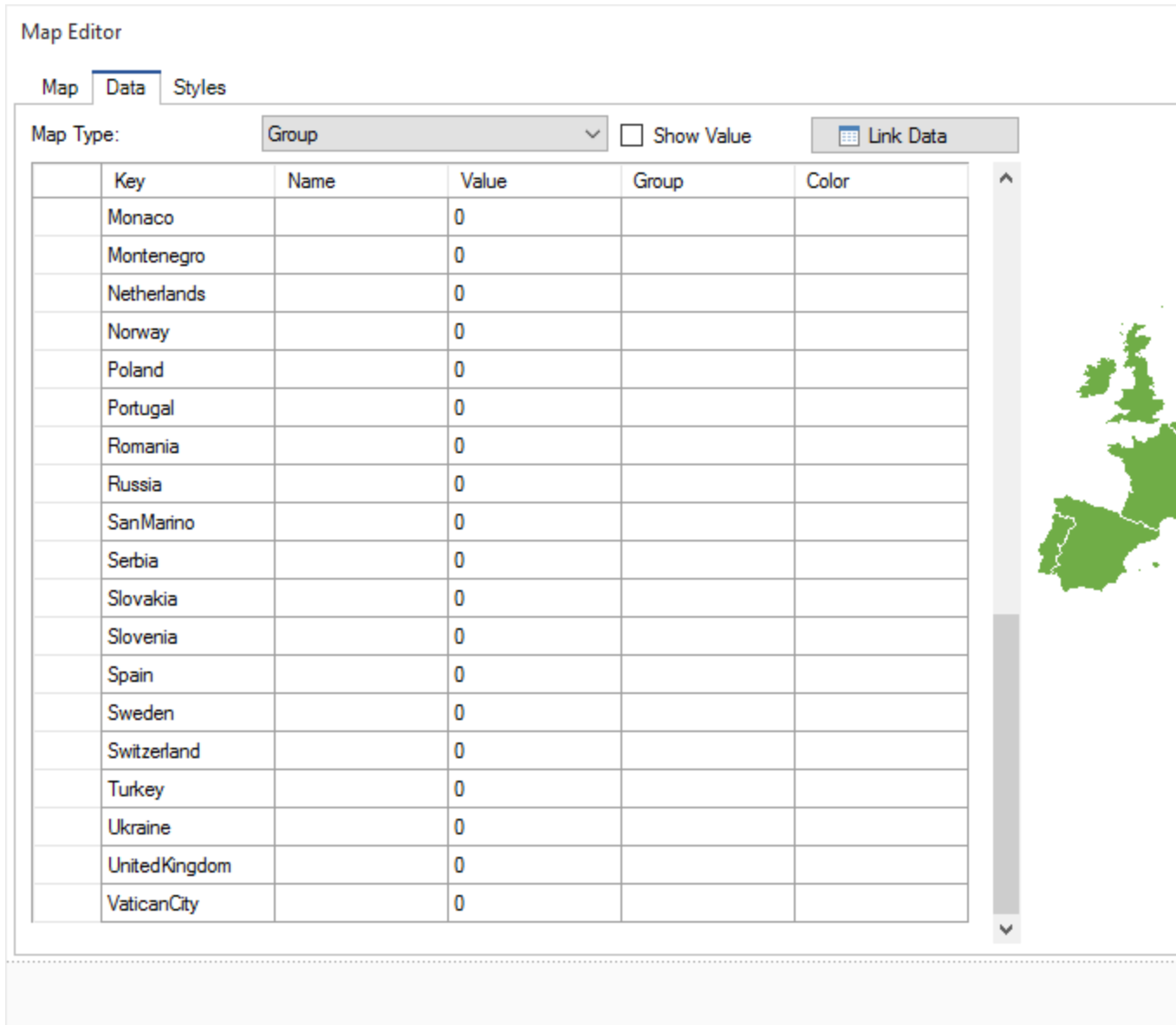
Heatmap with Group

In order to group the map elements and apply the heatmap to these groups, you should specify the type of a map as the Heatmap with Group. In this case, the map elements are grouped, and then the heatmap will be rendered. At the same time, it is necessary to fill the Group column or specify the data column in the Group field.

3.30.3 Map Keys

A map of any type is a graphic object consisting of one or more elements. Each element has its name which is called the **key**. Each type of the map uses its own keys. For example, a world map has the keys like the names of countries. For the US maps, the keys are names of states. For the EU map the keys are names of the European Union countries. In order to change the value or color of any map element, you must specify the key. Particularly, it is taken into account when the data for the map will be obtained from the data source. In this case, entries in the data column (indicated in the **Key** field of a map) must be identical to particular of map keys. In other words, in the world map, country names should match the entries in the data column.

In order to get the map keys, select the type of a map and click **Save** in the **Data** tab of the map editor on the preview panel:

Map Editor

| Map | Data | Styles |

| Map Type: | Group | ☑ Show Value | ☐ Link Data |

Key	Name	Value	Group	Color
Monaco		0		
Montenegro		0		
Netherlands		0		
Norway		0		
Poland		0		
Portugal		0		
Romania		0		
Russia		0		
SanMarino		0		
Serbia		0		
Slovakia		0		
Slovenia		0		
Spain		0		
Sweden		0		
Switzerland		0		
Turkey		0		
Ukraine		0		
UnitedKingdom		0		
VaticanCity		0		

Then choose the path to save the JSON file and confirm saving. This JSON file will contain the keys of the selected map view. Now they can be integrated into your data storage.

3.30.4 Data for Maps

The Map component provides an opportunity to visualize the data with reference to

geographical location. The data for maps can be specified manually and by passing from a data source. Consider both of these methods in detail.

✖ Manual

To enter the values manually, you should call the map editor, go to the **Data** tab and fill the cells in the table below:

Key	Name	Value	Group	Color
Alabama	AL	99	1	#FF0000
Alaska	AK	87	2	#FF7400
Arizona	AZ	50	3	#CD0074
Arkansas	AR	134	4	#009999
California	CA	127	5	#1D7373
Colorado	CO	74	6	#BF7130
Connecticut	CT	89	7	#FF7373
Delaware	DE	90	8	#33CCCC
Florida	FL	62	9	#FFB273
Georgia	GA	36	6	#BF7130

In order to draw the map, simply add the Map component to the report and specify its view, because the basic **Key** column is filled by default. In this case, the map carries only geographical information and will be drawn in one color. In order the map be informative, you should complete other columns:

❯ The **Name** column. This column contains the name of the element. For example, the USA map contains the full name of the states as keys. In the Name column, you can specify any text that will be displayed when you hover the cursor in the rendered report. This column is not required to be filled, and, if the text is not specified, then when hovering the element in the report, the key name will be shown.

❯ The **Value** column. This column contains a value for a particular map element. The value can be any number. The value will also be displayed in the rendered report when you hover the cursor, if the **Show Value** is enabled.

❯ The **Group** column. The values of this column are relevant when the map type is a map with the group, or a heatmap with the group. In this case, the group keys are specified. If you want to group some objects, you need be sure that their keys completely match. In this case, the map elements in the rendered report will be painted in one color. There will also be summed values of the group elements. The

result will be displayed in the rendered report with the Total prefix, when you hover over any element of the group.

❯ The **Color** column specifies the color of the map elements in the report. Color is defined by the #XXXXXX template. If the value in this column is not specified, the map element will be colored in a color map preset or custom style. If the color is specified, and the style is set for the map, the specified color will be applied to the map element.

Once the table is full, you can render a report. Also, entries can be stored in the JSON file, and can be used in reports in the future. To save the data, you should click the **Save** button in the map editor on the **Data** tab in the preview panel:
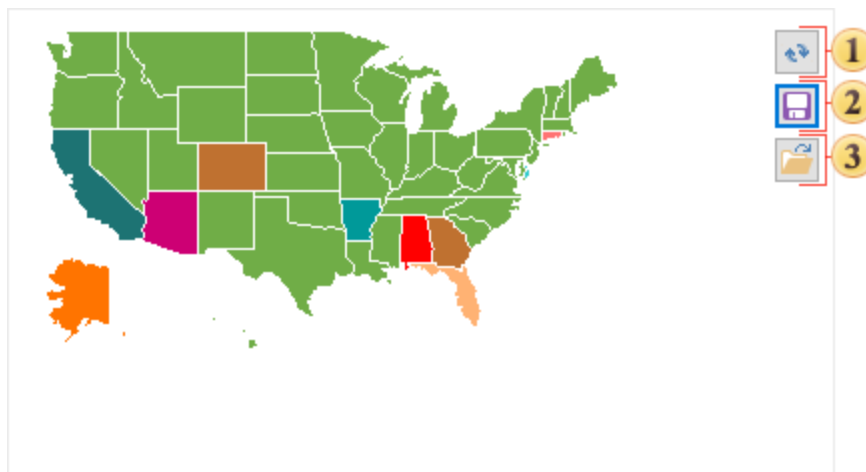
1 The button to update the map in the preview panel.
2 The **Save** button calls the menu in which you must specify the path to save the JSON file.
3 The **Open** button calls a menu where you can select a previously saved JSON file with the map data.

Data from Data Source

In addition to manual data input, the data map can be obtained from the data source. To do this click the Linked Data button in the map editor, in the Data tab:
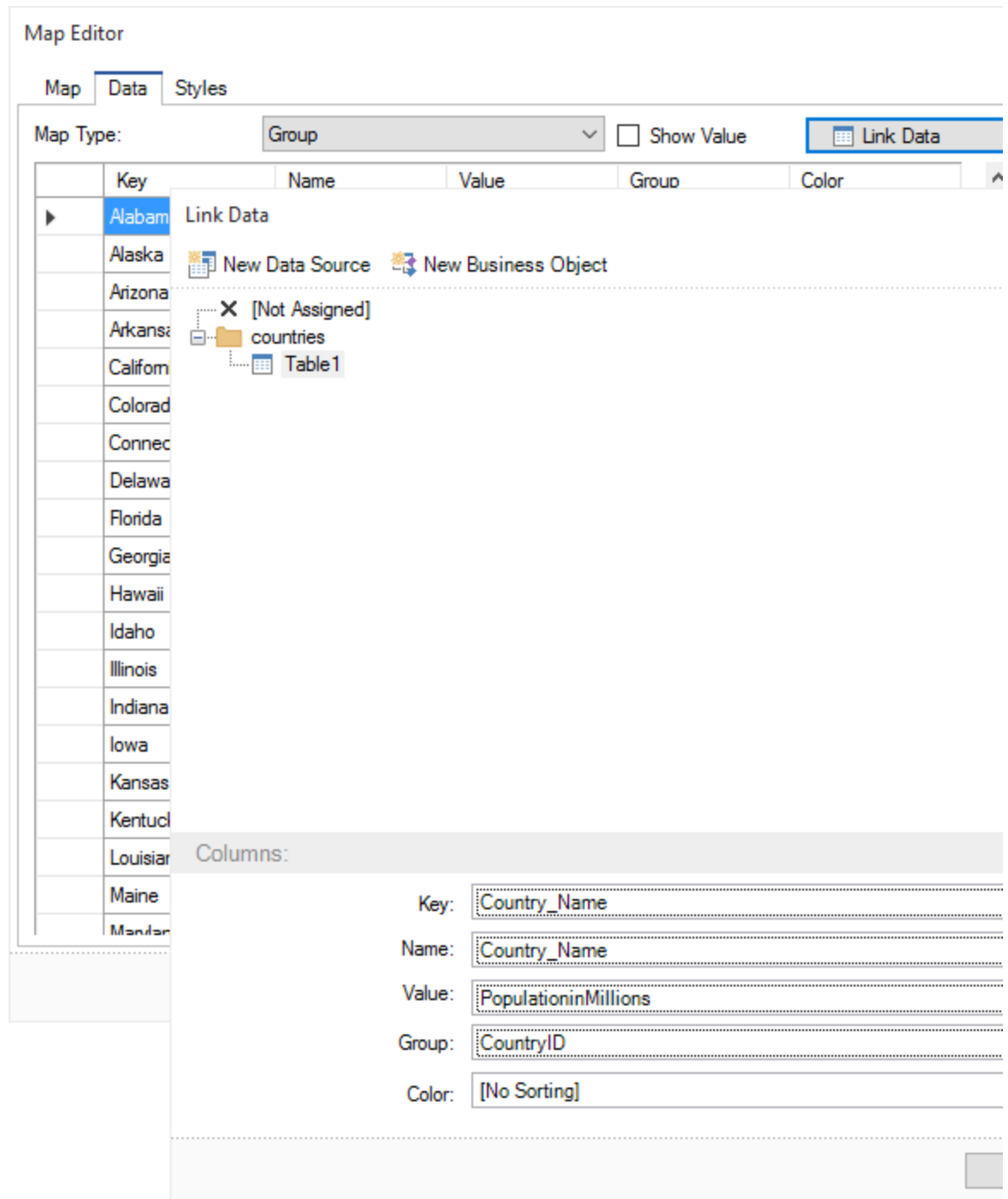
In the **Link Data** menu you can select the data source and specify the column from the data source for map fields:

> The **Key** field indicates the data column that contains entries identical to map keys of a specific type.

> The **Name** field specifies a column with the names of map elements.

> The **Value** field indicates a column with values for map elements.

❯ The **Group** field indicates the data column with keys for the group. In this field, you should specify the data column, if the map type is defined with the group, or a heatmap with the group. Grouped map elements in the rendered report will be painted in one color.

❯ The **Color** field specifies data column with a set of colors for the map elements. If the data column is not specified in this field, the map element will be colored in a preset or custom color of the map style. If the column contains data (the map style is set) then the color from the data columns will be applied to the map element.

3.31   Text Quality

The StiText component and components, inherited from it, have the **TextQuality** property. This property allows selecting/displaying the quality of the text. The property may have one of three values:

❯ **Standard.**
❯ **Typographic.**
❯ **Wysiwyg.**

In the **Standard** and **Typographic** modes, text displaying is performed using a **GDI +** system library. The difference between these modes is that in the **Typographic** mode, a text is output with antialiasing and looks fine, but the rendering is slow. In the **Wysiwyg** mode a text is displaying using the GDI system library. The text in this mode may not look as beautiful as in the other two modes.

Why do we need GDI, if GDI + exists and it is more beautiful and easy to use? To answer this question, lets turn to the definition of the **WYSIWYG**.

WYSIWYG (acronym for "What You See Is What You Get") is a way of editing, in which the content in the process of editing looks very similar to the final output. With regard to the reporting tool, this means that the report should look the same when editing a template, and viewing the finished report printed on paper. However, in practice, it is not so simple. Many methods can display a text in different ways on different monitors and in different ways to print it on different printers. This is particularly evident in a large text: when viewing in the preview with different zoom modes and printing, line breaks can be located in different places. This occurs due to many reasons. In the GDI + system library, most of these problems have been solved, but not all, and sometimes inaccurate displaying still occur. To solve the remaining problems one need full control of the text output. GDI + does not provide such control. Therefore, the **Wysiwyg** mode was added. In this mode a text is output using the GDI. GDI methods allow you to

control the output of each character of a text. This can eliminate almost all the problems. Thus, the **Wysiwyg** mode displays the text not as pretty as the other two methods, but more accurately.

There is another difference between these two modes: as a text in each mode is displayed in different ways, then the measurement of length of a line is done in different ways. For example, we have three text boxes with the "Test string" text; set the **TextQuality** to **Standard** for the first text box, to **Typographic** for the second one, and to **Wysiwyg** for the third. Set the **AutoWidth** property to true for all the text boxes. In the design mode of the report we get the following:
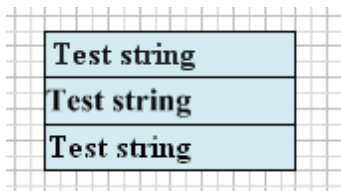
By sight the difference between these lines is not visible. However, after rendering, the width of the text boxed will be calculated depending on the width of the text, and we will immediately see the difference between the modes:
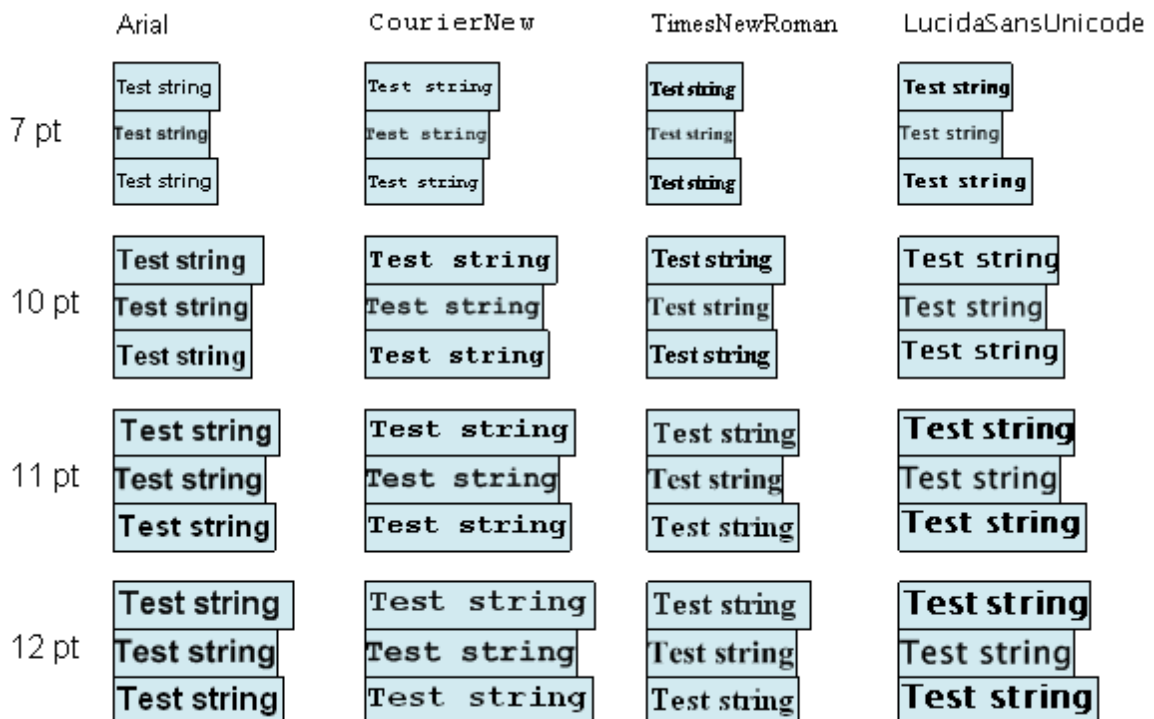
In the above picture it is clearly seen that for different types and sizes of fonts completely different results are obtained. This must be taken into account, for example, if you are going to use the Cross-Tab component. In this component the table columns widths are fit depending on text, and, in different modes, the width of the table can be changed.

In the above picture clearly shows that for different types and sizes of fonts are obtained completely different results. This must be taken into account, for example, if you're going to use the component CrossTab: this component width of the table columns to fit text, and different modes the width of the table can pretty much change.

Also, as practice shows, WYSIWYG in these applications are often not working properly. For example, your report in EXCEL in edit mode and in print preview may look different. Even more differences you will see if in edit mode will begin to change the page scale from 50% to 200%: at 100% scale text can be placed in a cell at 50% did not reach the cell edge, and at 200% the last word can be transferred to the next line. Another example - a multi-line text: with different scale is not always correct calculated line spacing, and height of text in a cell can vary. At one level in the cell can not fit all the text strings, ie truncate the text. At another level the same text can be compressed, and the bottom of the cell will remain blank. Even a team of Excel "Autofit row height" may give unpredictable results, especially in small fonts.

Therefore, when you export reports in MS-Office, we recommend using some of the techniques described below. Recommendations can be divided into two parts: general guidelines for preparing reports and recommendations for each export.

General recommendations on export reports in MS-Office are to design a report template:
Try whenever possible to keep the gap between the end of the line and the edge of textbox, in which case the problem should not arise;
It follows from the preceding paragraph: Do not use unnecessarily property AutoWidth, as the size of textbox in this case is calculated without gap;
pick a value for the text TextQuality, to a line of text to receive the most long and this will increase the likelihood that the text after export will appear normally.

Recommendations for the export of reports in MS-Word
When exporting to MS-Word Use the following trick: for each line of text font is installed seal. The value of the density of the font is measured in units of twips and stored in a static property StiOptions.Export.Rtf.SpaceBetweenCharacters

(StiOptions.Export.Word2007.SpaceBetweenCharacters). By default, the property is set to -2. On the eye, this quantity of text compression is not noticeable, but in most cases it is enough. If necessary, this value can be changed. Zero value of the property corresponds to the normal font, positive values correspond to the sparse font.

Recommendations for exporting reports to MS-Excel
When exporting to MS-Excel use the following trick: for all the problem textbox is recommended to set the right / or left border of textbox. Table cells in Excel do not have borders, so the border will be considered only when rendering the textbox as garantiroovanny gap. Border textbox sets the Margins, the value specified in hundredths of an inch. For most cases it is sufficient to establish the right boundary is equal to 1 one hundredth inch (written in the property 0, 1, 0, 0).

3.32 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.
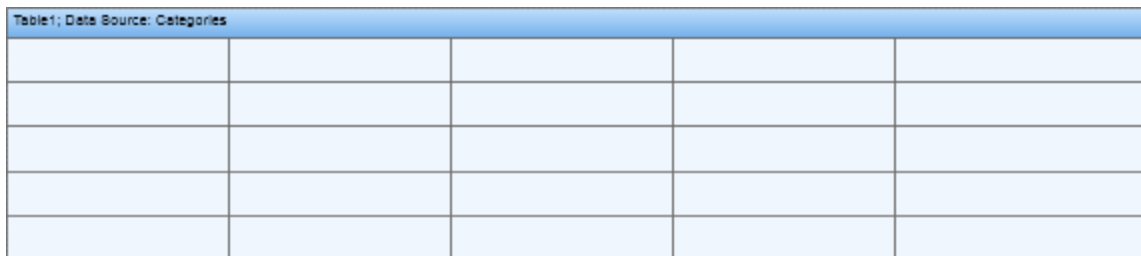
Table1; Data Source: Categories

❗ This component is designed to simplify the work in the designer. When the report is rendered, the table is converted into a set of bands and text components. If you need more flexibility, we recommend you avoid the use of tables in favor of bands, text and other components.

3.32.1 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.

Table2; Data Source: Not Assigned

On the picture below the table with 5 columns is shown.

Table2; Data Source: Not Assigned

3.32.2 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.

Table2; Data Source: Not Assigned

On the picture below the table with 5 rows is shown.

Table2; Data Source: Not Assigned

3.32.3 DataSource Property

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.
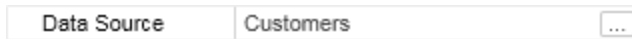
The Table editor allows selecting data source.
A data source can be selected by clicking the first tab of the editor. All data sources are grouped in categories. Each category corresponds to one connection with data in the report data dictionary. The picture below shows the Table editor.
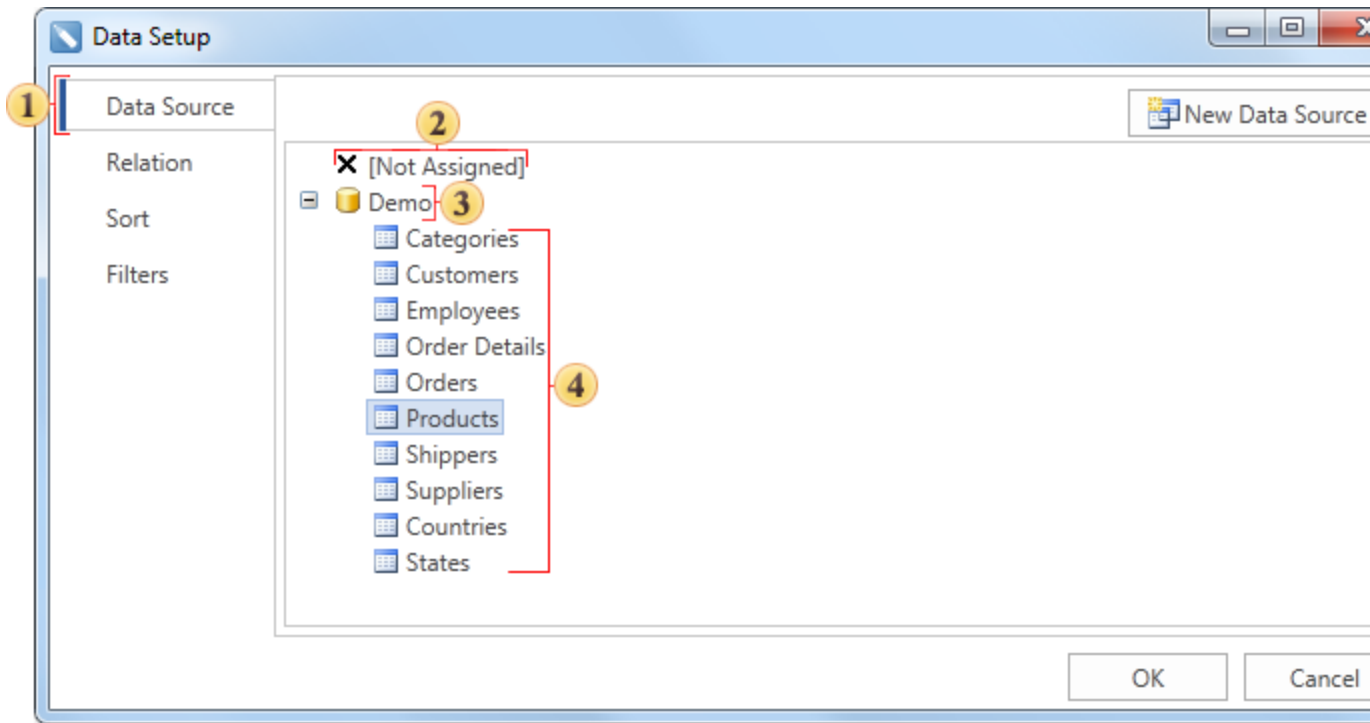
1 The tab to select the data source;
2 Select this node if you do not need to specify the data source;
3 The "Demo" data category;
4 The "Demo" data source category.

The data source can be also selected using the quick access buttons.
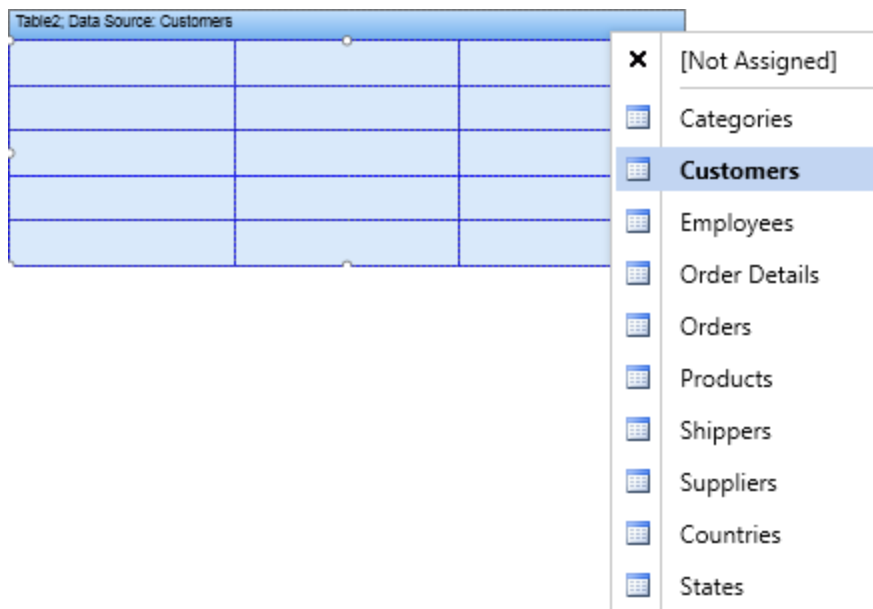
3.32.4 MasterComponent Property

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.

The second way is to set the Master table in the Table designer.
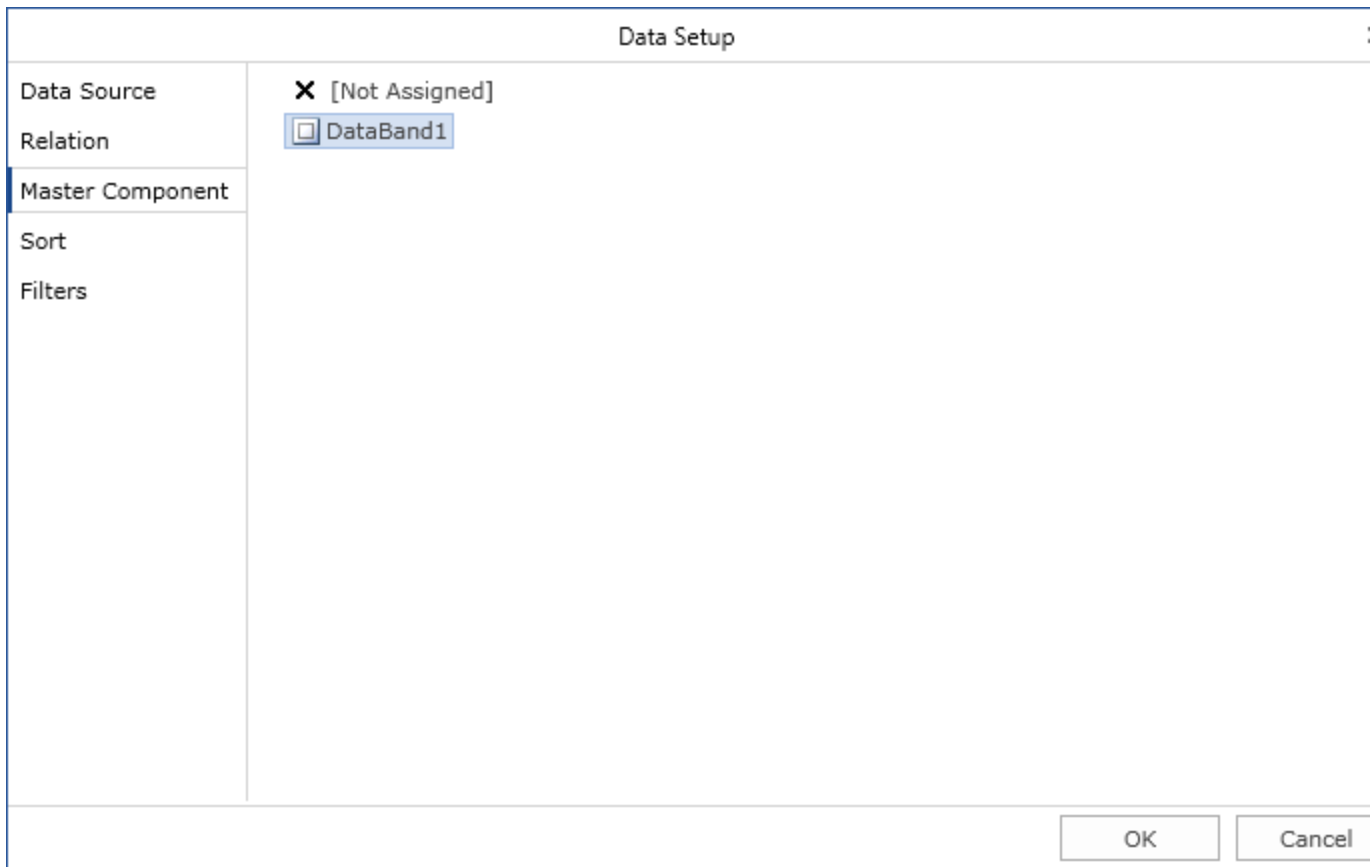
After filling the **MasterComponent** component two tables will be related to each other. When printing one data row from the Master data source (and, correspondingly, printing the Master table), the printing of appropriate rows from the Detail data source occurs (and, correspondingly, printing the Detail 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.

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

3.32.5 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.
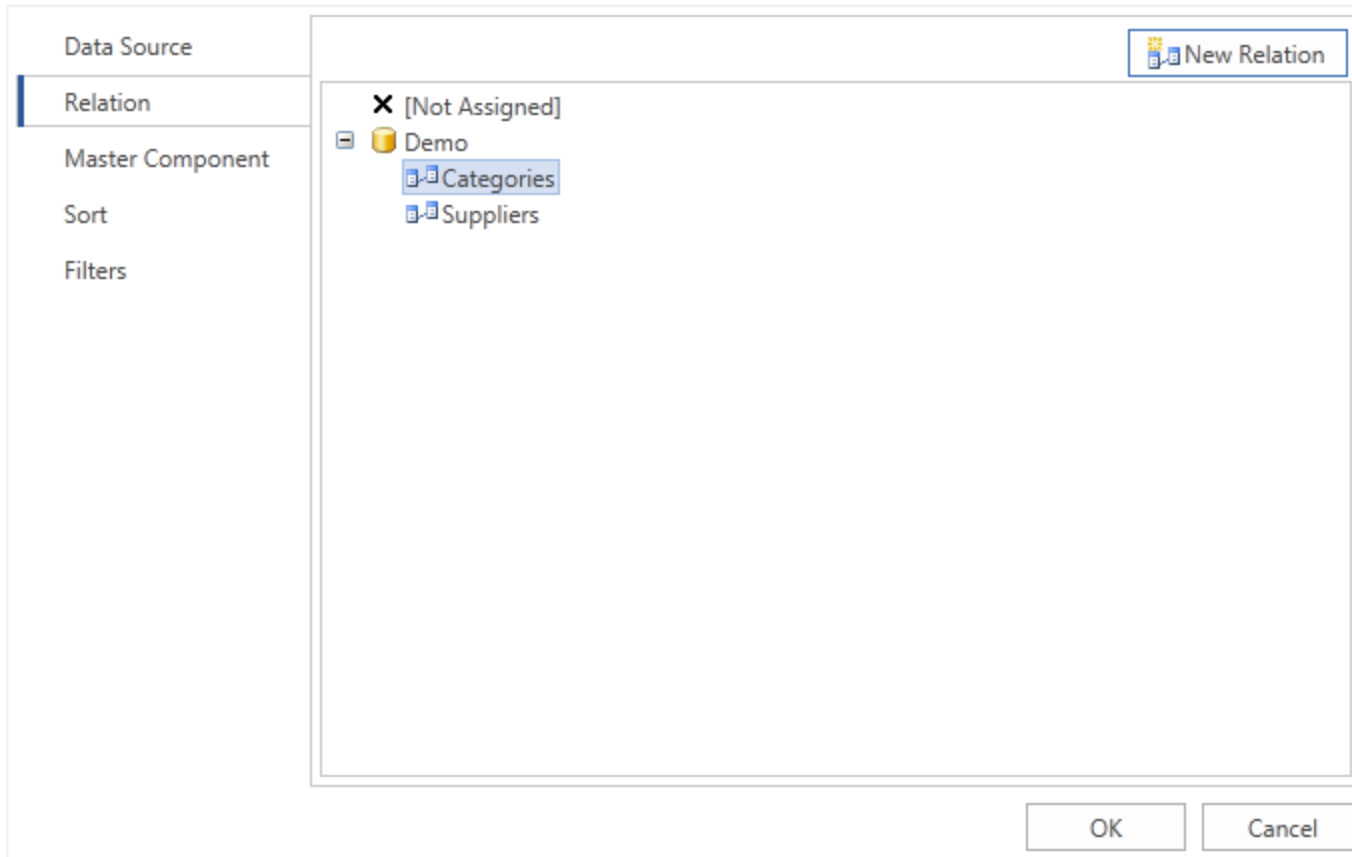
The selection is done between relations which are created between Master and Detail data sources and in what the Detail data source is the child data source.

3.32.6 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.

3.32.7 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.

GroupHeaderBand1; Condition: {Products.Categories.CategoryName}
{Products.Categories.CategoryName}

TableProducts; Data Source: Products			
{Products.ProductName}			{Products.ProductID}
{Products.QuantityPe	{Products.UnitPrice}	{Products.UnitsInSto	{Products.SupplierID

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

3.32.8 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.
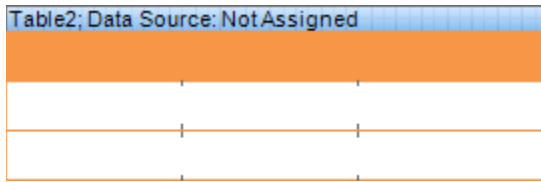
3.32.9 Table Footer

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.
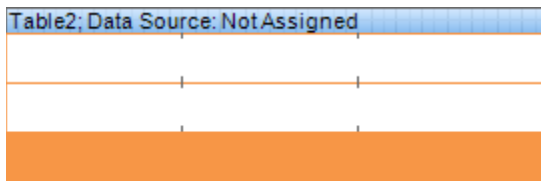
3.32.10 Cells Width Autochange

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.

3.32.10.1 AutoWidth Property

The **AutoWidth** property of a **Table** component indicates whether the reporting tool will fix the cells size after the report rendering.

➤ 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.
➤ 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.
➤ 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.

3.32.10.2 AutoWidthType Property

The **AutoWidthType** property of a table indicates how the reporting tool will fix cells width after report rendering.

> **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

> **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

> **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

3.32.11 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

3.32.12 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.

✔ Text is a cell will be output as a text. Cell settings are the same as the settings of a Text component;
✔ Image is a cell will be output as a text.Cell settings are the same as the settings of an Image component;
✔ 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;

✓ 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.
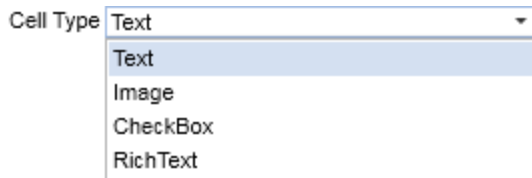
Also it is possible to indicate a cell style by clicking the quick access button of a cell.
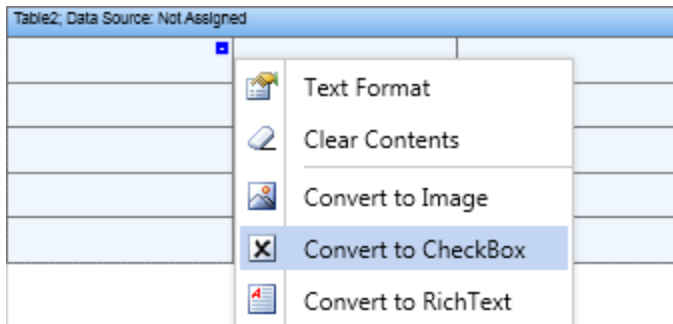
Or the context menu of a cell.
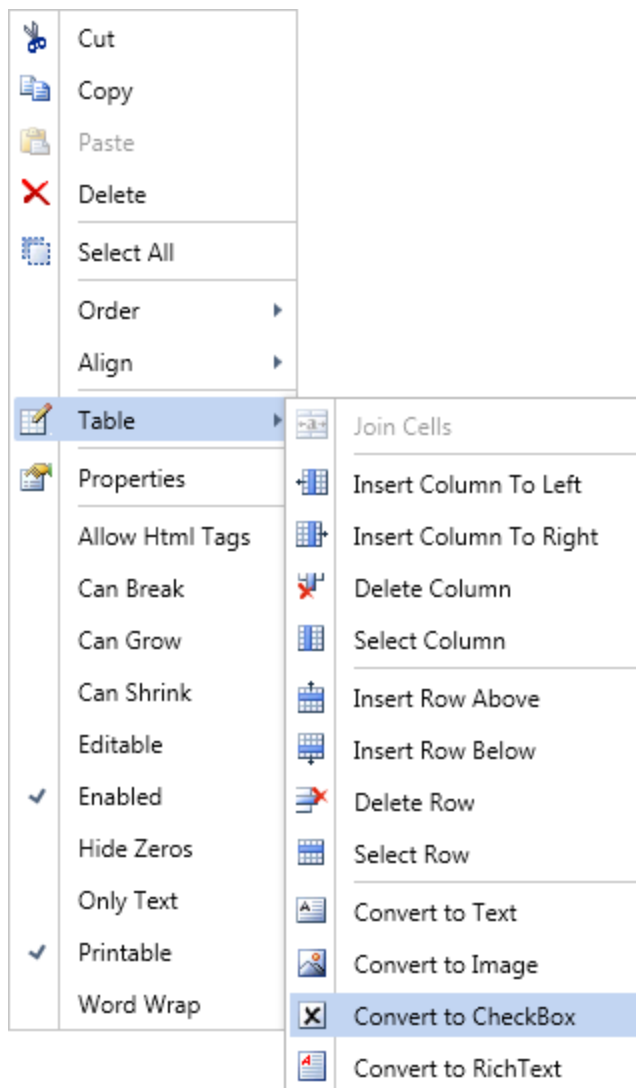
3.33 Primitives

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**:
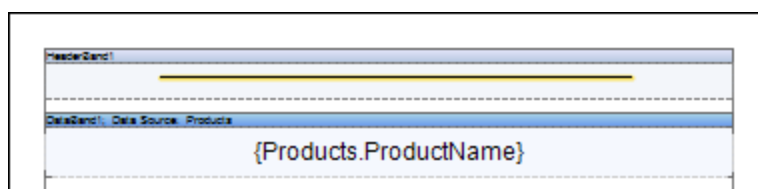
The **Shape** is a report component, which, depending on the type, shows this or that primitive. The **ShapeType** property is used to specify a primitive type. The picture below shows a list of values of the **ShapeType** property:
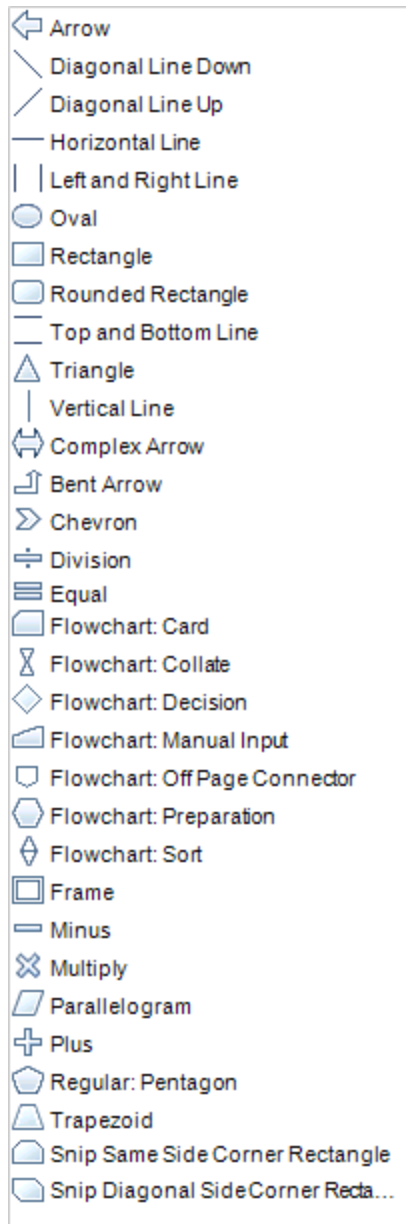
3.33.1 Cross-Primitives

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:
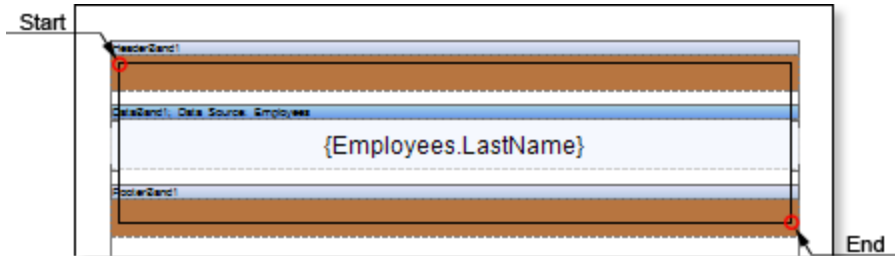
As can be seen in the picture, the start and end points of the **Rectangle** component are located on different bands: the start point is located in the **HeaderBand**, and the end point is in the **FooterBand**. When rendering the report, the report generator will render start and end points of the rectangle, and then it will render rectangle sides. The picture below shows an example of the rendered report pages with the **Rectangle** cross-primitive:
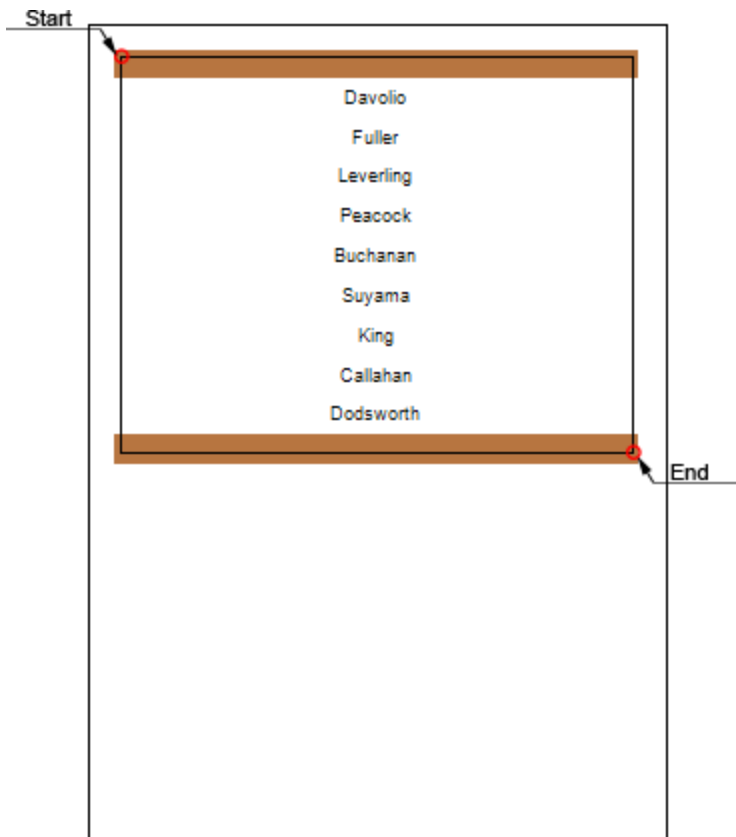
3.34 Sub-Reports

The **Sub-Report** component is used to display another report in the current report. In this case, the sub-report will be displayed in the current report only within the **Sub-Report** component. In other words, when you render a report with **Sub-Report** components, the report engine will build all the nested reports and place them in these components.
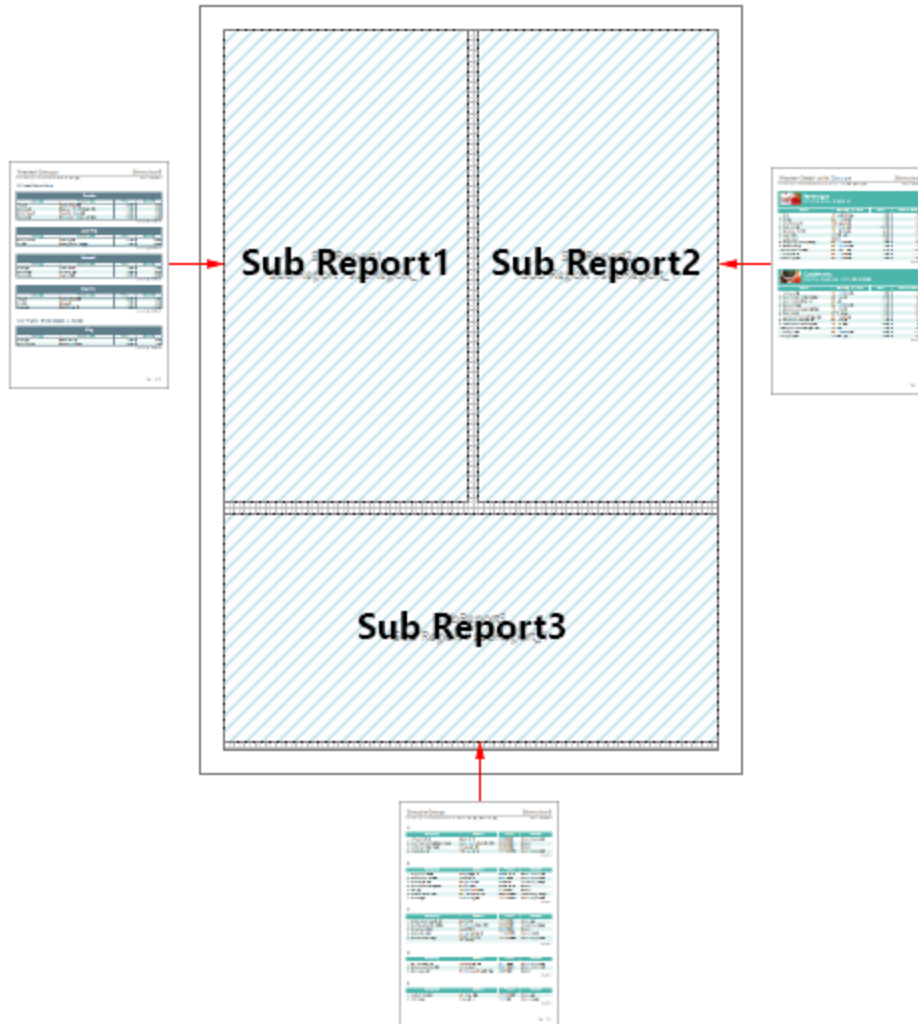
You can place sub-reports on:

❯ Bands;
❯ Pages;
❯ Panels;
❯ Any other components of the report that can be containers for sub-reports.

A report that will be displayed in the rendered report using the **Sub-Report** component can be obtained:
> From another page in the report template;
> From the file (*.mrt, *.mrz, *.mdc, *.mdz);
> By the hyperlink (*.mrt, *.mrz, *.mdc, *.mdz);
> From the report resources (*.mrt, *.mrz, *.mdc, *.mdz).

Information

You may place the **Sub-Report** component on another sub-report. So, the number of levels of nested reports is unlimited.

You can add sub-reports by:
> Selecting this component in the **Components** group in the **Toolbox** or in the **Insert** tab. In this case, a new page which is associated with this component will be automatically created in the report.
> Dragging the report from the resources to the report. In this case, a new page will not be created, and in the **Sub-Report** component, a link to the resource will be generated.

3.34.1 Editor

In the editor, you can specify the resource for the **Sub-Report** component and configure the settings.

To call the editor, double-click the **Sub-Report** component on the report page:
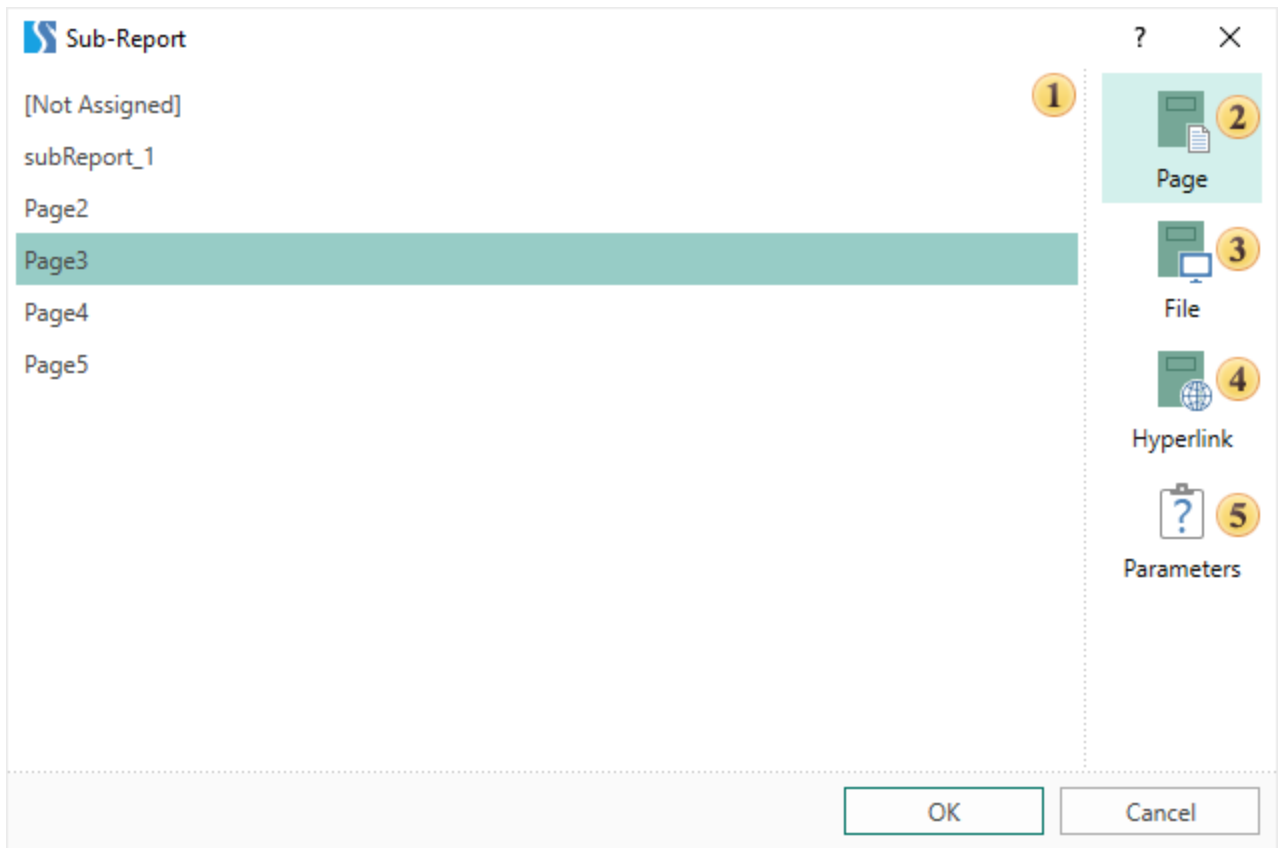
① Settings panel. If the **Page** tab is selected, the list of report template pages will be displayed on this panel. Any of these pages can be a resource for the component and, when rendering the report, it will be displayed on this component.

② The **Page** tab. In this tab, you can select the report template page that will be the resource for the **Sub-Report** component.

③ The **File** tab. In this tab, you can specify a path to the file (external report) that will be the resource for the **Sub-Report** component.
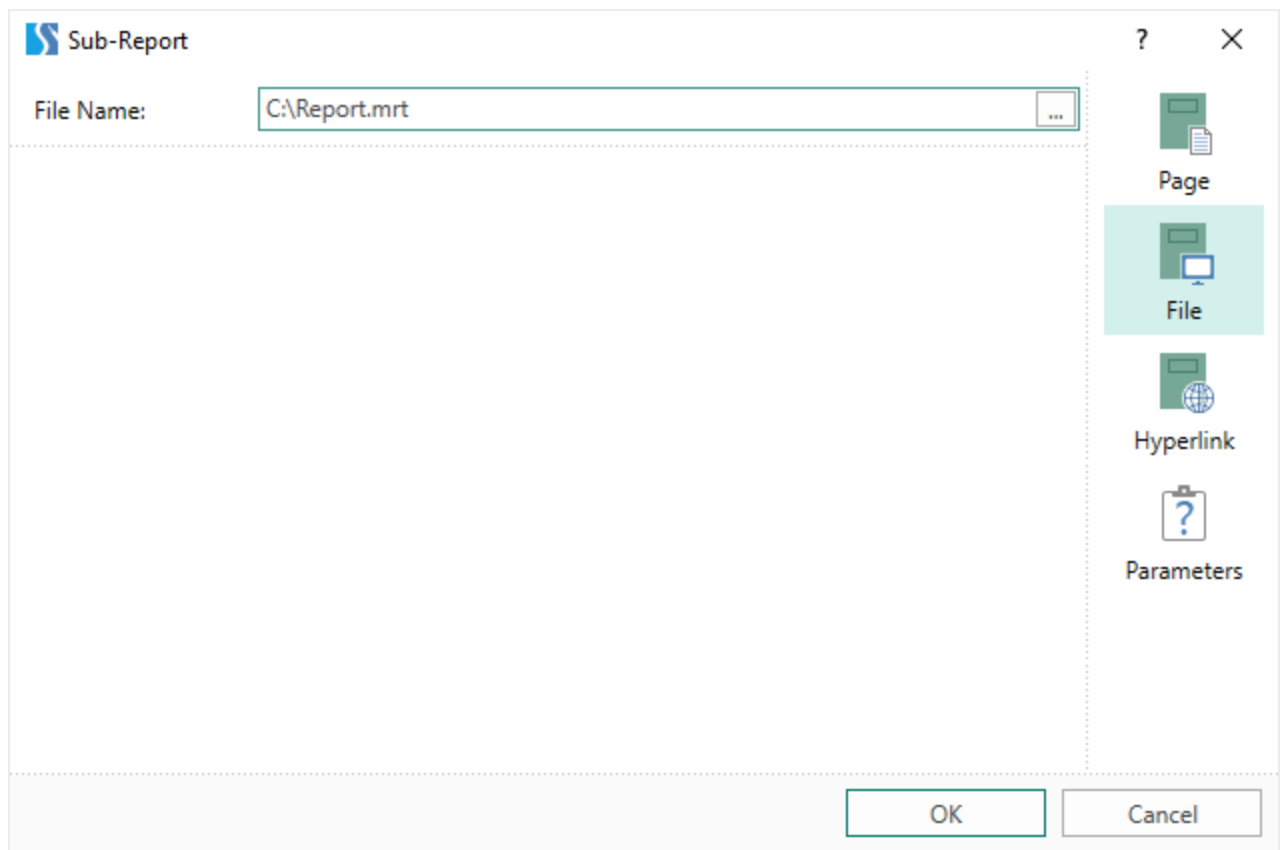
④ The **Hyperlink** tab. In this tab, you can specify a link to the external report or to the resource that will be the resource for the **Sub-Report** component.
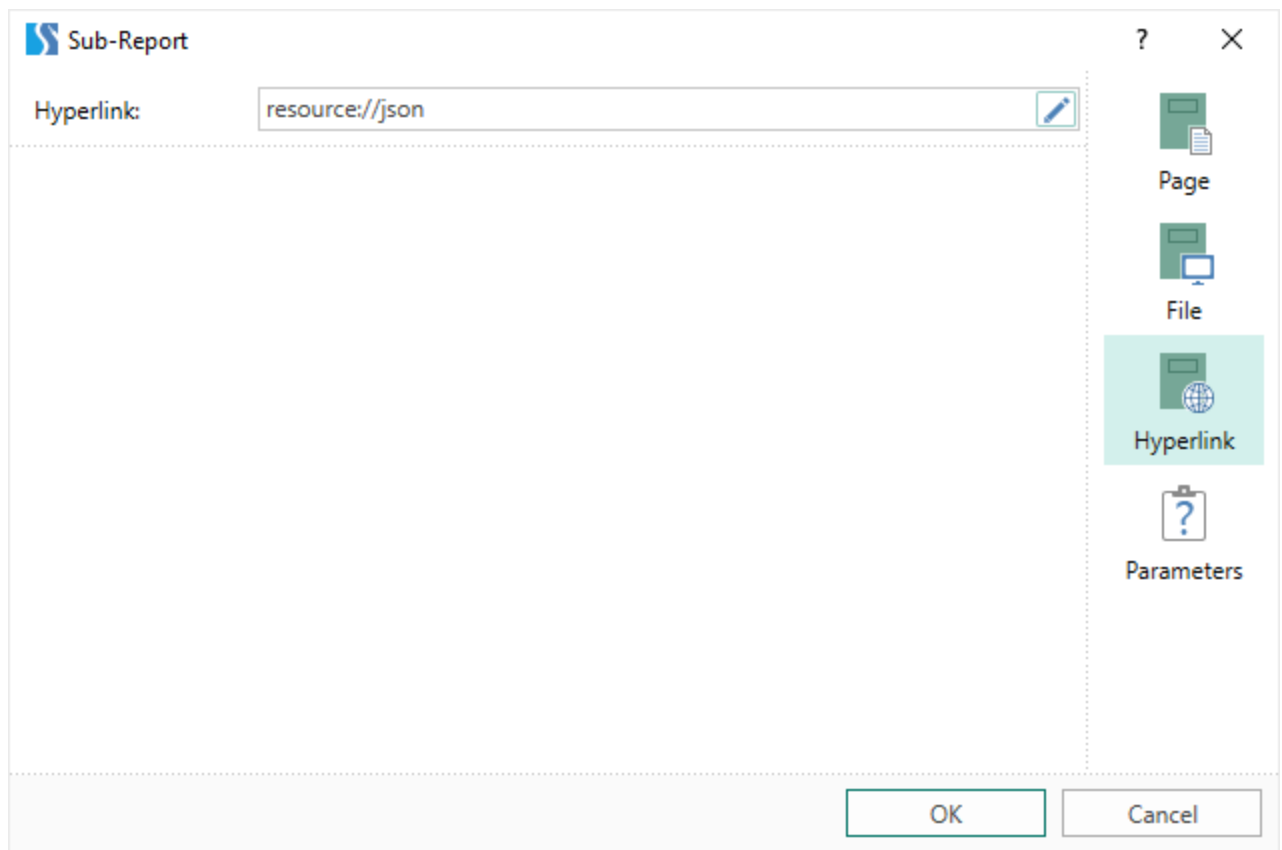
5 The **Parameters** tab. In this tab you can add and configure the settings that will be passed to the sub-report.
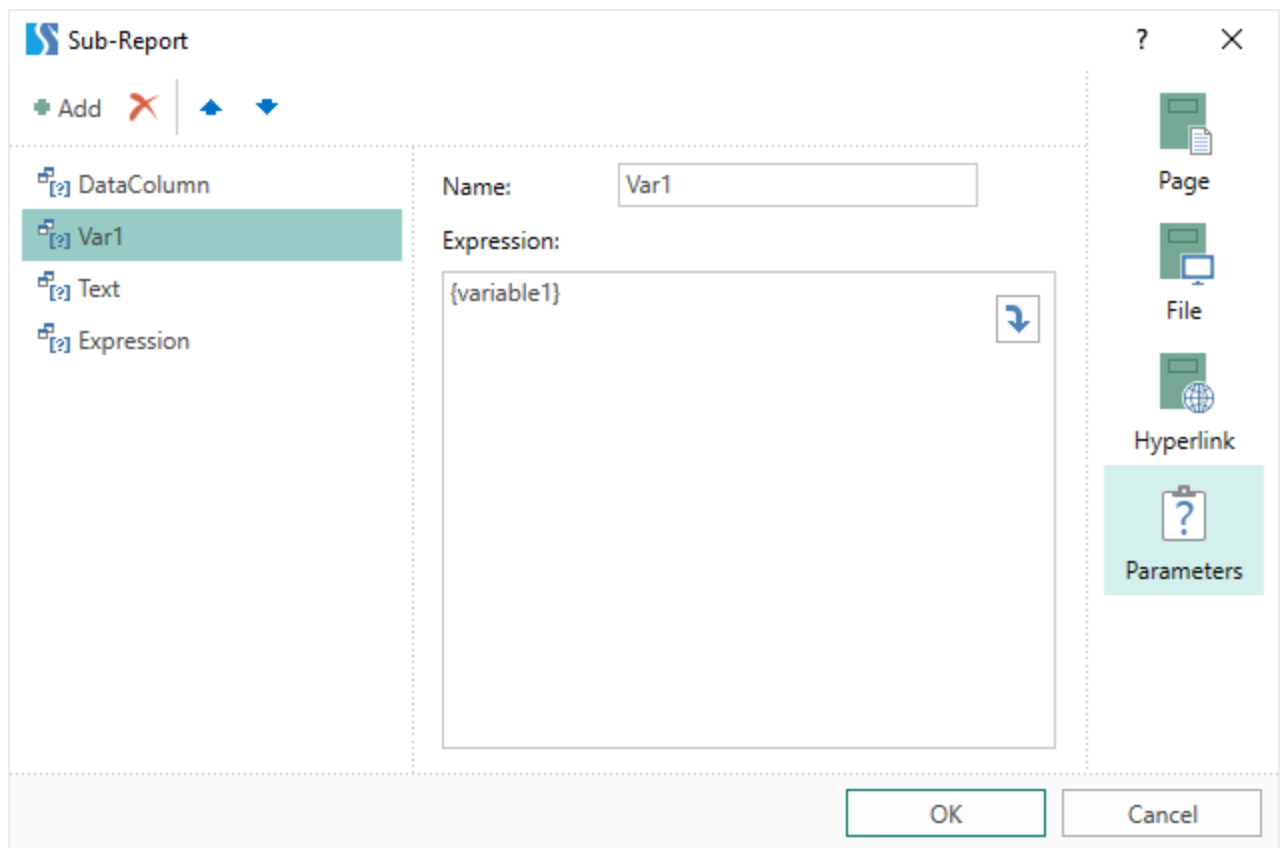
Parameters are usually used to filter data or transfer information from the main report to a sub-report. To add a parameter, you should:
➤ Call the editor of the sub-report;
➤ Go to the **Parameters** tab;
➤ Click the **Add** button;
➤ Specify the name of the parameter and its expression.

In the parameter expression, you can specify:
➤ The data column;
➤ Variable;
➤ Any other expression.

After that, you should go to the resource of the **Sub-Report** component (a page or another report) and specify this parameter, for example, in the filter expression.

3.34.2 Report sample with parameters

Let's create a report with products by category. The list of categories will be located in the main report, and the list of products will be located in the sub-report (on another

page in the same report template).

Step 1: Open the report designer;

Step 2: Connect the data;

Step 3: Add the **ReportTitleBand**, if required;

Step 4: Add a **Data** Band with a list of categories;
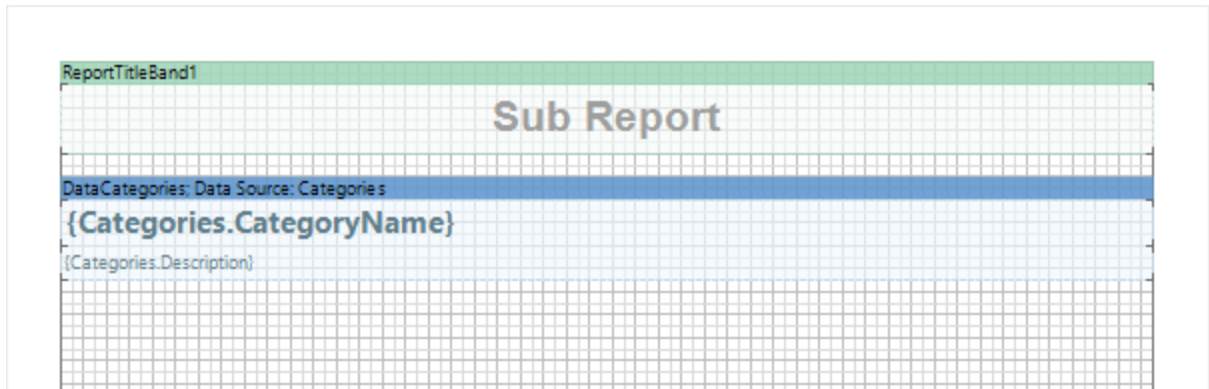
Step 5: Add the **Child** band;

Step 6: Place the **Sub-Report** component on this band. At the same time, the new page **subReport_1** will be added to the report template;
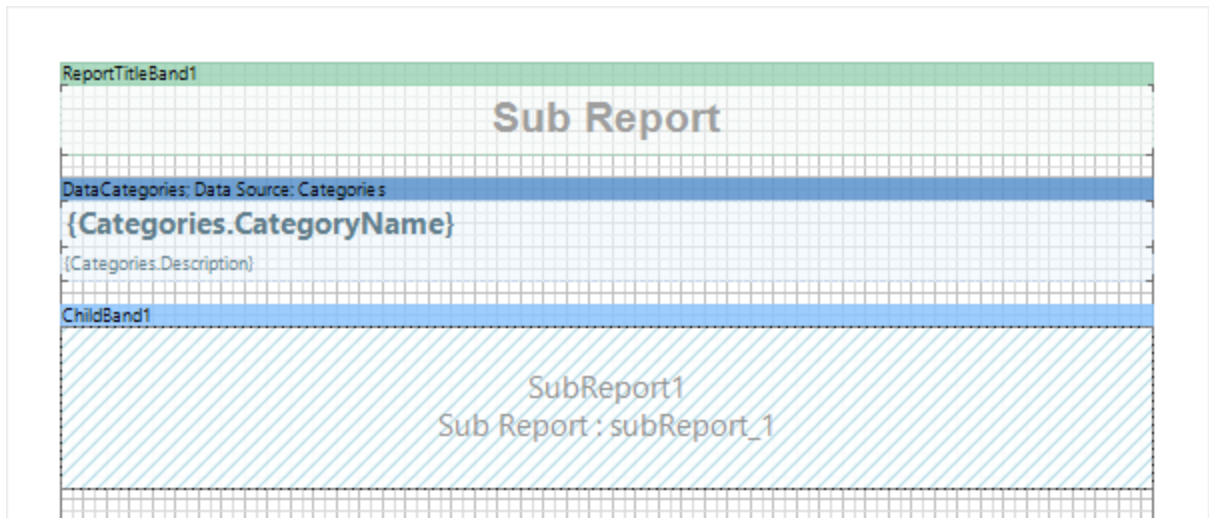
Step 7: Go to the new page of the report template and place a band with the list of products, titles and totals, if required;

Information

If you go to the **Preview**, then, for each category, the entire list of products will be displayed without considering to which category the products belong to. In order to display only products which belong to the category, you should add a parameter with category keys and transfer them to the sub-report.

Step 8: Go back to the page with the list of categories;

Step 9: Call the editor of the sub-report and go to the **Parameters** tab;

Step 10: Add a new parameter, specify a name and column **Categories.CategoryID** as an expression;
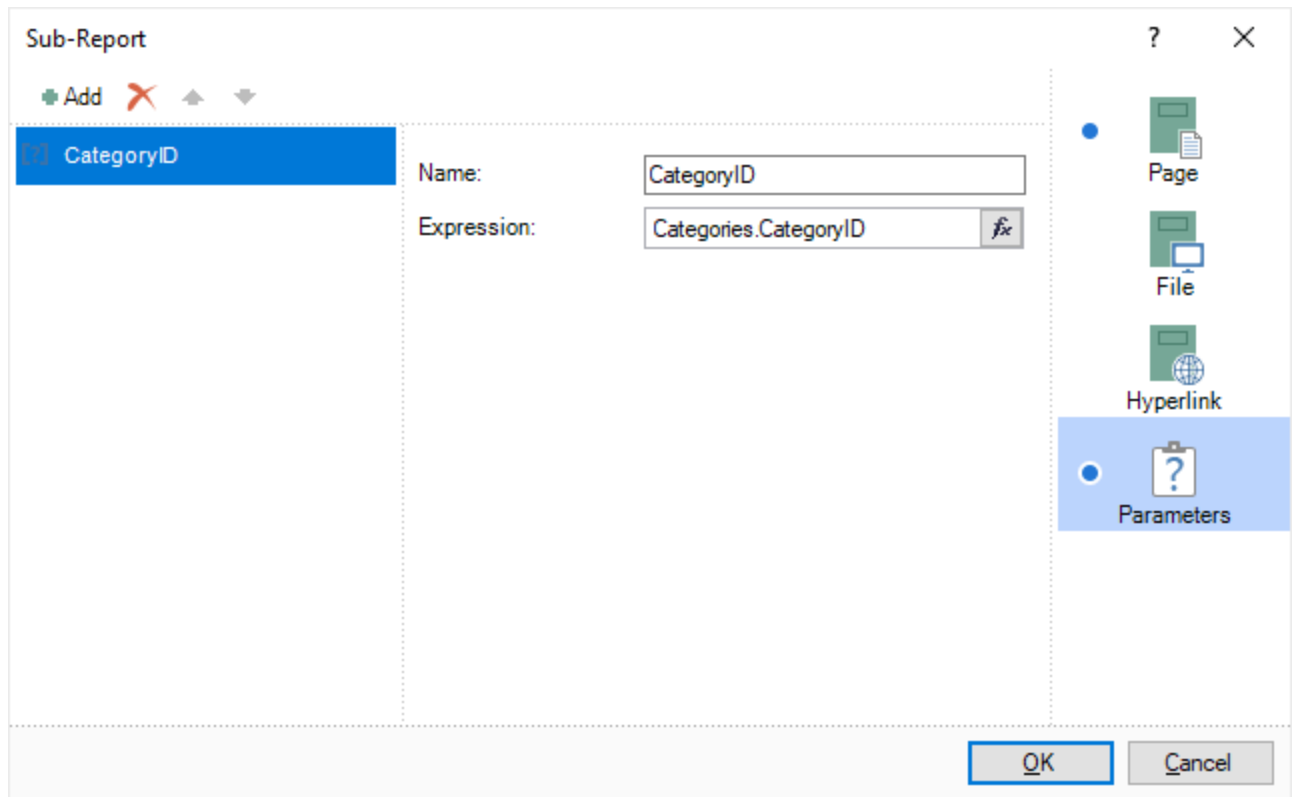
Step 11: Go back to the products page and specify the filter expression using this parameter **Products.CategoryID == (int)this["CategoryID"]**;
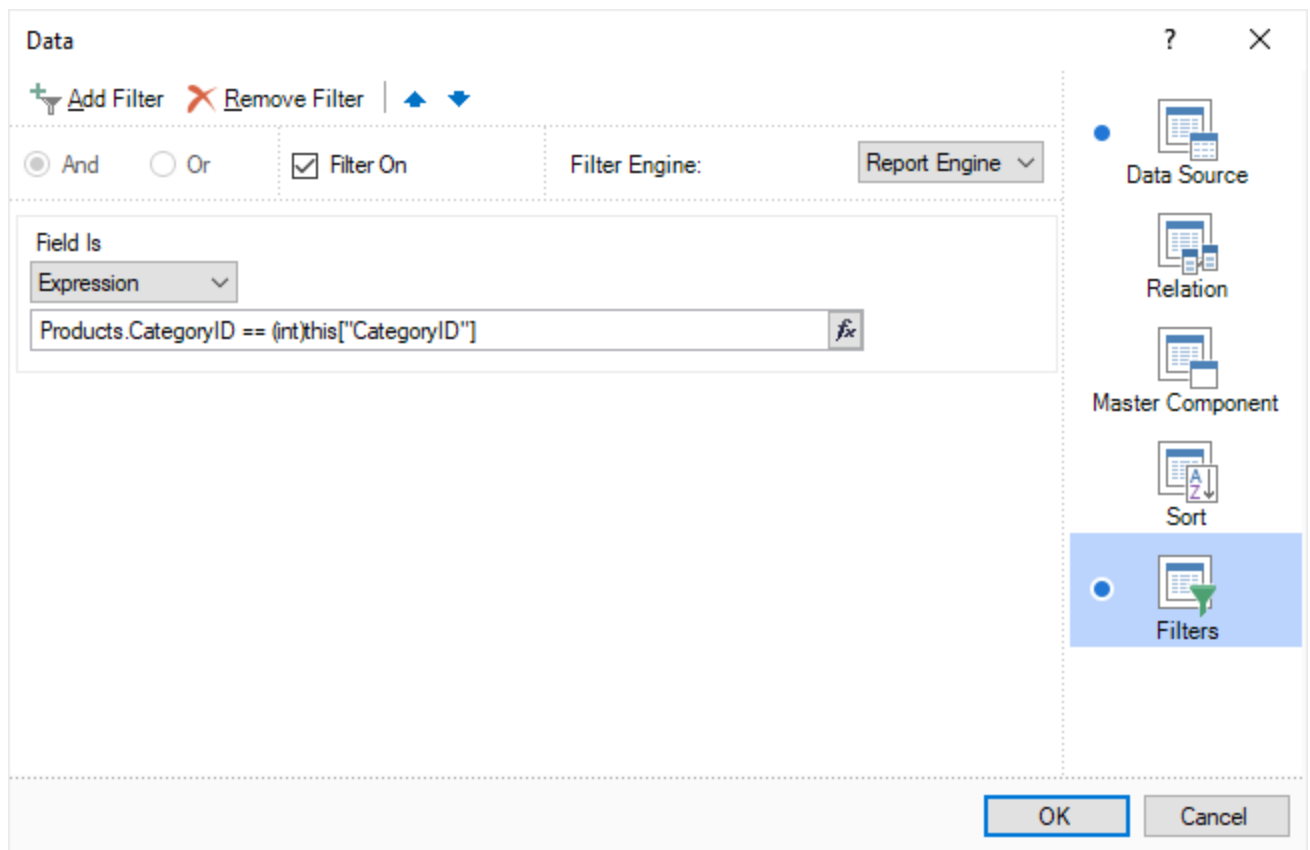
Step 12: Go to the Preview. A list of products will be displayed by categories.

Sub Report

Beverages

Soft drinks, coffees, teas, beers, and ales

Product Name	Unit Price	Units In Stock
Chai	$18.00	39
Chang	$19.00	17
Guaraná Fantástica	$4.50	20
Sasquatch Ale	$14.00	111
Steeleye Stout	$18.00	20
Côte de Blaye	$263.50	17
Chartreuse verte	$18.00	69
Ipoh Coffee	$46.00	17
Laughing Lumberjack Lager	$14.00	52
Outback Lager	$15.00	15
Rhönbräu Klosterbier	$7.75	125
Lakkalikööri	$18.00	57

Count: 12

Condiments

Sweet and savory sauces, relishes, spreads, and seasonings

Product Name	Unit Price	Units In Stock
Aniseed Syrup	$10.00	13
Chef Anton's Cajun Seasoning	$22.00	53
Chef Anton's Gumbo Mix	$21.35	0
Grandma's Boysenberry Spread	$25.00	120
Northwoods Cranberry Sauce	$40.00	6
Genen Shouyu	$15.50	39
Gula Malacca	$19.45	27
Sirop d'érable	$28.50	113
Vegie-spread	$43.90	24
Louisiana Fiery Hot Pepper Sauce	$21.05	76
Louisiana Hot Spiced Okra	$17.00	4
Original Frankfurter grüne Soße	$13.00	32

Count: 12

3.34.3 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:
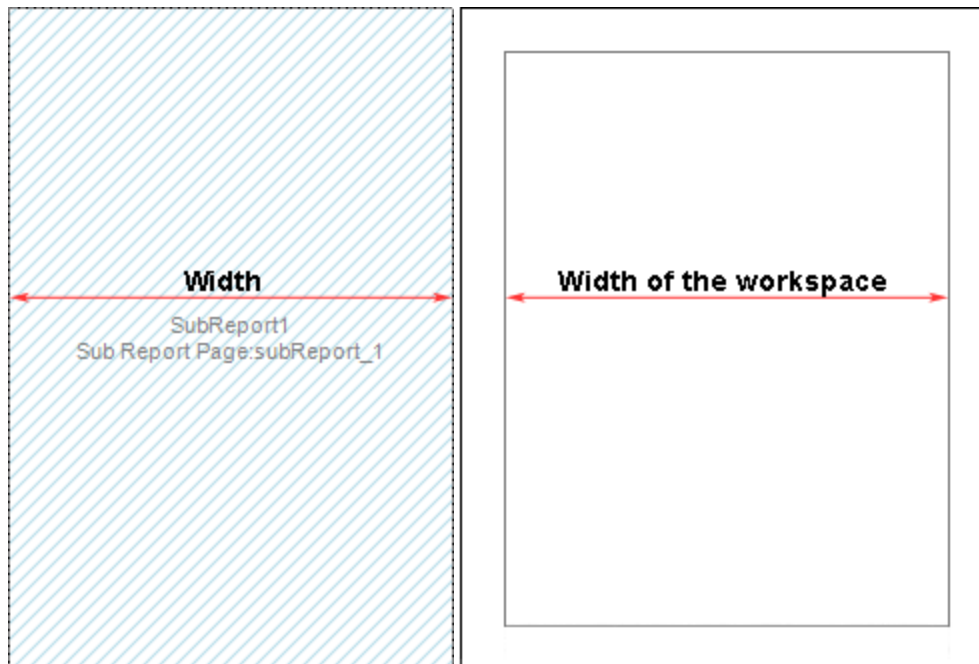
The **CanGrow** property of the **Sub-Report** component is always set to **true** but, when placing this component, it cannot be grown by height. So you should take into the account the height of the component on the nested page: it should not be higher than the **Sub-Report** component. When rendering 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:
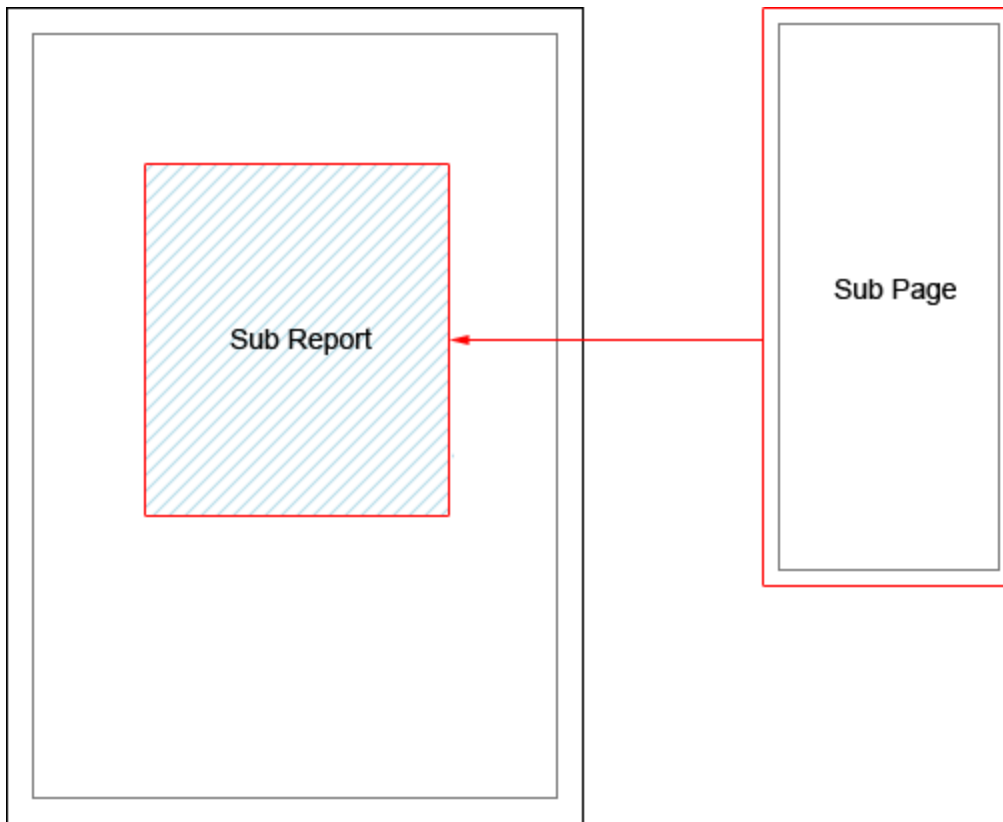
3.34.4 Sub-Reports on Data Band

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**:
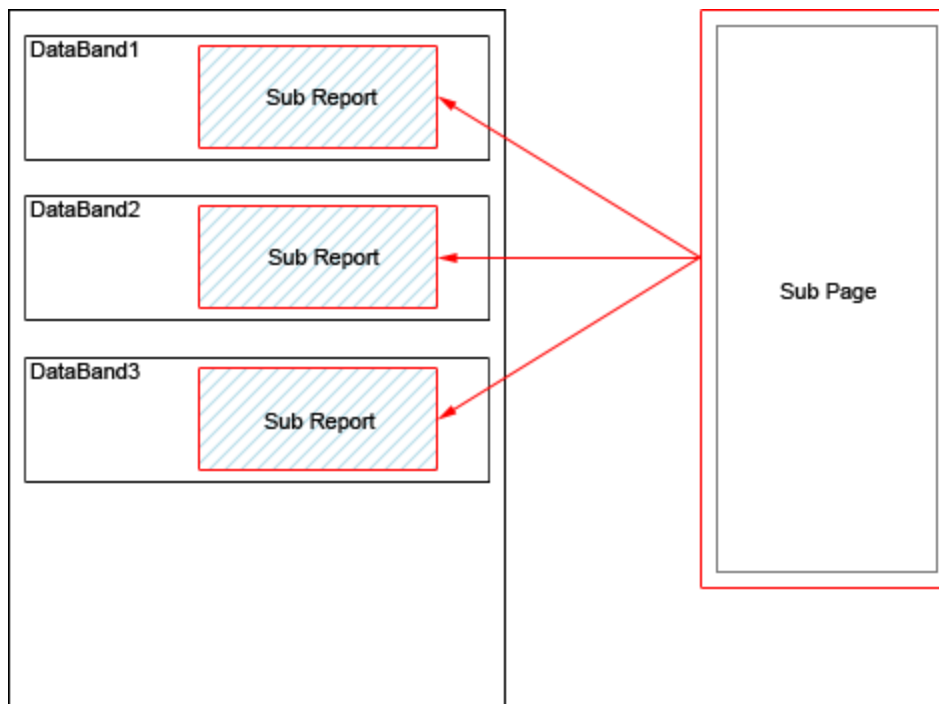
In this case the height of the component on the sub-report page of a report will be higher than the height of the **Sub-Report** component. So the **Sub Report** component is placed in the **DataBand** and rendered as the item of the **DataBand**, and, in this case, the **CanGrow** property works and the component can grow by height.

3.34.5 Master-Detail Reports and Sub-Reports

You can build a Master-Detail report using the Sub-Report component in several ways:
▶ [Pass parameters](#) from Master entries to Detail by filtering data;
▶ Using the Master Component property in the Data band.

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:
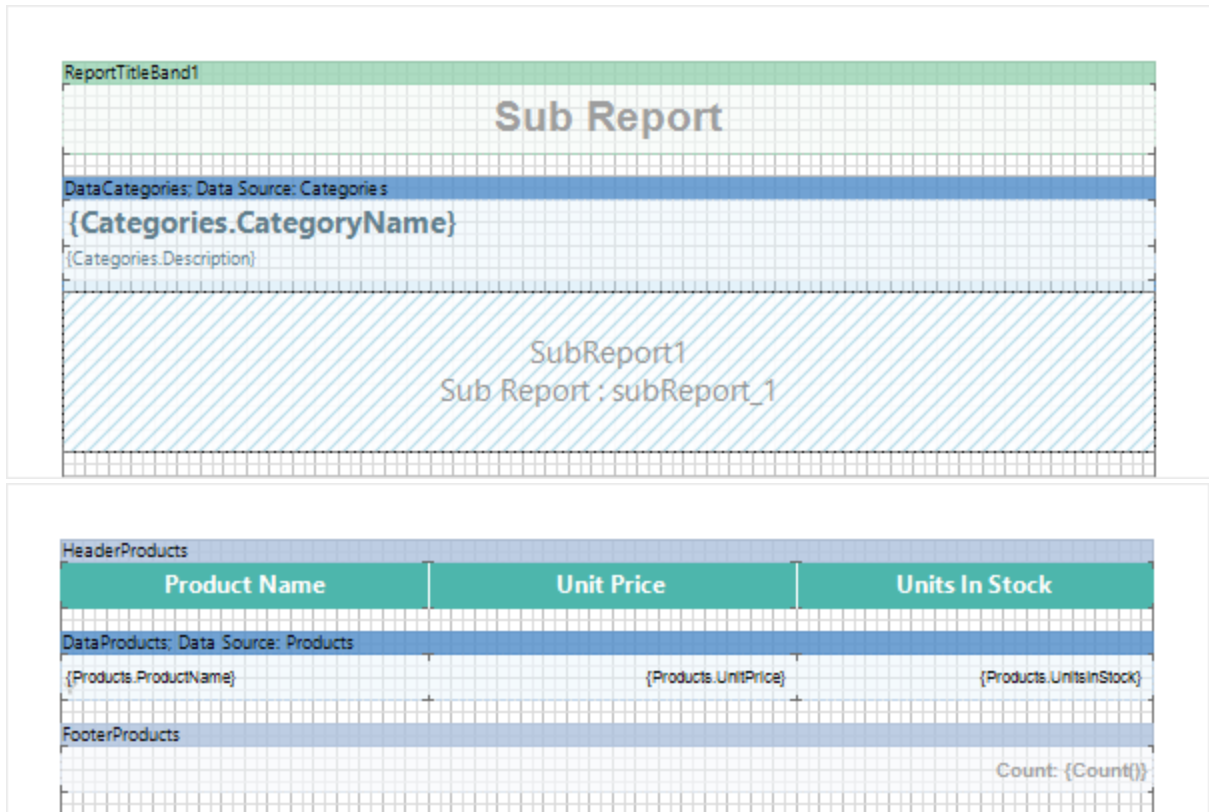
In this example the **DataBand1** can be defined as the **Master** for the **DataBand2** that is placed in the sub-report page of a report. For this you need to choose the **Master** component in the data settings. The picture below shows the sample of the **Data Setup** window:
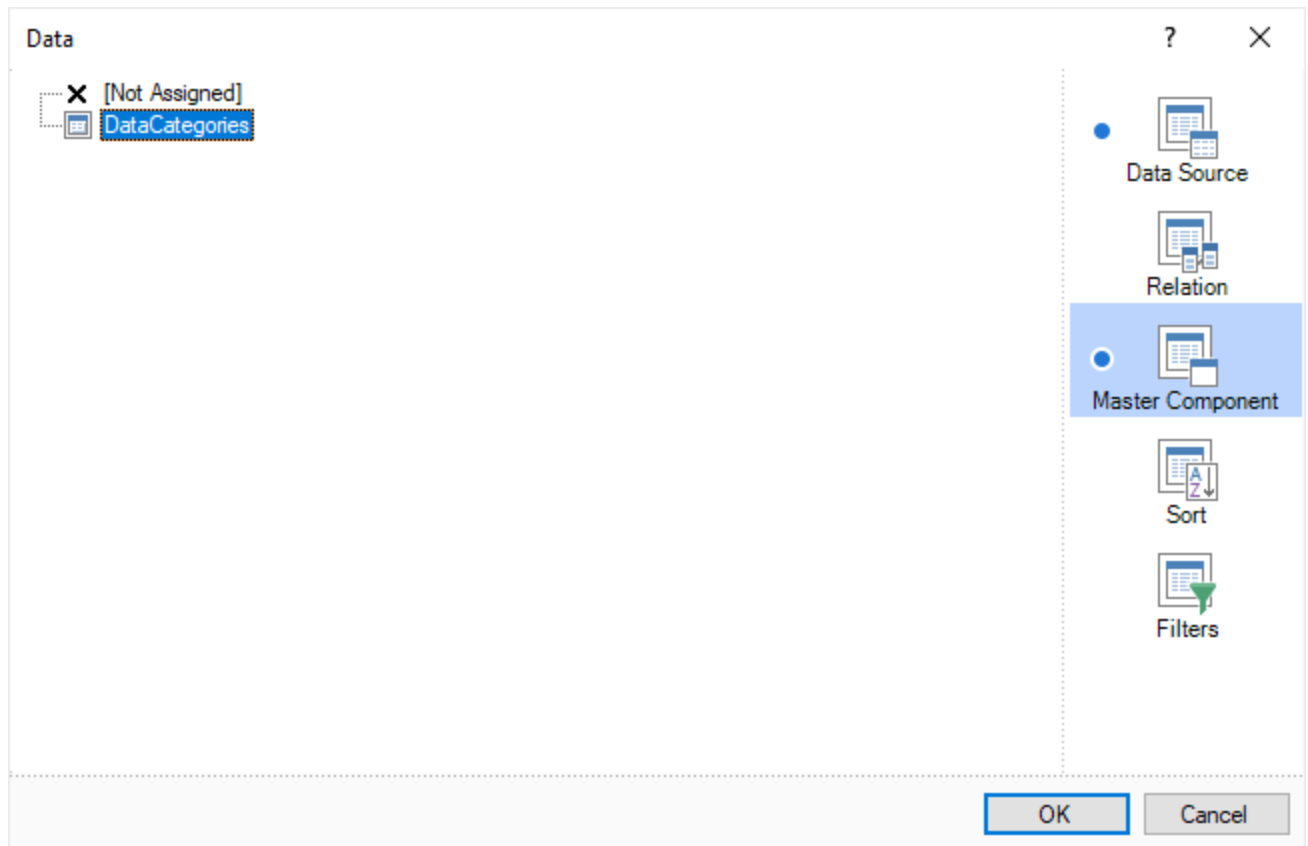
As you can see, the **DataBand1**, that is placed on the report page, is the **Master** in the **Master-Detail** report. If several **DataBands** are placed on the sub-report page then, when creating the **Master-Detail** report, the **Master** is either the **DataBand** in what the **Sub-Report** is placed or any other **DataBand**, placed in the sub-report page.

3.34.6   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:

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**:

3.35 Functions

The data dictionary is represented by the Functions category. This category contains the elements using which you can calculate a certain total or return the desired value. All elements of the Function category are divided into groups. The table below shows a list of functions and their brief description and examples.

View the Functions:

Function	Description	Sample

Date:		
{DateDiff(,)}	Calculates the distance between the specified dates ➤ Arguments should be of the DateTime type ➤ Returns the TimeSpan value	{DateDiff(DateSerial(2016,1,30),DateSerial(2016,1,1))} - the result is 29.00:00:00, that means 29 days. {DateDiff(DataSource.Column1,DataSource.Column2)} - the result will be calculated for each value in Column1
{DateSerial(,,)}	Specifies date. ➤ Arguments should be year, month, day ➤ Returns the DateTime value	{DateSerial(2016,1,30)} - the result is 1/30/2016 12:00:00 AM The function returns the DateTime value, but if you want to display only the date, you should apply text formatting to the text component
{Day()}	Shows a day from the specified date ➤ Arguments should be of the DateTime type ➤ Returns the long value	{Day(DateSerial(2016,1,30))} - the result is 30, since in arguments the January 30 2016 is specified {Day(DataSource.Column)} - the result will be calculated for each Column value
{DayOfWeek()}	Display a day of the week from a specified date in text form. ➤ In arguments specify: ① Date (the DateTime type) ② Culture (the string type) ③ The value true or false (the bool type), to display the result with a capital letter or with a lowercase	{DayOfWeek(DateSerial(2016,1,30))} - the result is Saturday. {DayOfWeek(DataSource.Column)} - for each value a day of the week will be calculated {DayOfWeek(DateSerial(2016,1,30),"de")} - the result

	④ The value true or false (the bool type), depending of which the system culture or designer localization will be used ➤ Returns the string type	will be Samstag, because the de culture is set. {DayOfWeek(DataSource.Column,"de")} - all results will correspond to the de culture {DayOfWeek(DateSerial(2016,1,30),"en", false)} - the day of the week will start from the lower-case letter, i.e. saturday. {DayOfWeek(DataSource.Column,"de", true)} - the days of the week will start with the capital letter, for example Samstag. {DayOfWeek(DateSerial(2016,1,30), false)} - the day of the week will be displayed on the culture that is used by the system. {DayOfWeek(DataSource.Column, true)} - the day of the week will be displayed in the culture that is used in the report designer.
{DayOfYear()}	Displays a day of the year ➤ Specifies the date in the argument (the DateTime type) ➤ Returns the long value	{DayOfYear(DateSerial(2016,2,14))} - the result is 45, since February 14 is the 45th day of a year. {DayOfYear(DataSource.Column)} - for each value of the Column the data of a year will be calculated.
{DaysInMonth()}	Displays the number of days in the month: ➤ In arguments specify:	{DaysInMonth(DateSerial(2016,2,1))} - the result will be 29, because 2016 is a

	❶ Date (the DateTime type) ❷ Year and month (the long type) ❯ Returns the long value	leap year and there are 29 days in February. {DaysInMonth(DataSource.Column)} - for each value the number of days in a month will be calculated. {DaysInMonth(2016,3)} - the result will be 31, since there are 31 days in March.
{DaysInYear()}	Displays the number of days in a year: ❯ Specifies the date in arguments (the DateTime type) or a year (the long type) ❯ Returns the long value	{DaysInYear(2016)} - the result will be 366 days, since 2016 is a leap year. {DaysInYear(DataSource.Column)} - for each value of the Column the number of days in a year will be calculated.
{Hour()}	Displays an hour: ❯ Specifies time in arguments (the DateTime type) ❯ Returns the long value	{Hour(DataSource.Column)} - an hour will be displayed from each value. For example, if time is 16:22:36, then the result is 16.
{Minute()}	Displays minutes: ❯ Specifies time in arguments (the DateTime type) ❯ Returns the long value	{Minute(DataSource.Column)} - minutes will be displayed from each value. For example, if time is 16:22:36, then the result is 22.
{Month()}	Displays months: ❯ Specifies time in arguments (the DateTime type) ❯ Returns the long value	{Month(DateSerial(2016,12,1))} - the result will be 12, as the date is set on December 1, 2016. {Month(DataSource.Column)} - for each value of the Column a month will be

		displayed.
{MonthName()}	Displays the month name of the specified date ➤ Specifies in arguments: ① Date (the DateTime type) and culture (the string type) ② Culture (the string type), ③ The true or false value (the bool type), to display the result to display the result with a capital letter or with a small letter. ④ The true or false value (the bool type), depending on which the system culture or designer localization will be used ➤ Returns the string value	{MonthName(DateSerial(2016,1,1))} - the result is January, because the 1 of January 2016 is set. {MonthName(DataSource.Column)} - the result is the name of the month for each Column value. {MonthName(DateSerial(2016,2,1),"de")} - the result will correspond to the de culture, i.e. Februar. {MonthName(DataSource.Column,"en")} - all the names of months will correspond to the en culture. {MonthName(DateSerial(2016,1,1), false)} - the name of the month will be in lower case. {MonthName(DataSource.Column, true)} - the name of the months will start with a capital letter. {MonthName(DateSerial(2016,1,1), false)} - the name of the month will correspond to the culture used by the system. {MonthName(DataSource.Column, true)} - the names of months will correspond

		to the culture that corresponds to the culture of the report designer.
{Second()}	Displays seconds: ❯ Specifies time in arguments (the DateTime type) ❯ Returns the long value	{Second(DataSource.Column)} - seconds will be displayed from each value. For example, if time is 16:22:36, then the result is 36.
{TimeSerial(,,)}	Displays time: ❯ Specifies hours, minutes, seconds in arguments (the long type) ❯ Returns the TimeSpan value	{TimeSerial(1,14,20)} - the result is 01:14: 20, 1 hour, 14 minutes, 20 seconds.
{Year()}	Displays year: ❯ Specifies date in arguments (the DateTime type) ❯ Returns the long value	{Year(DateSerial(2016,1,2))} - the result will be 2016, since the date is January 2, 2016. {Year(DataSource.Column)} - the years from each Column value will be displayed.
Math:		
{Abs()}	Displays the absolute number. ❯ Specifies the number is arguments (the double,decimal,long type) ❯ Returns respectively double, decimal, long	{Abs(-42)} - the result is 42 {Abs(DataSource.Column1)} - the result will be absolute numbers from the values of Column1, i.e. without considering the number.
{Acos()}	Displays the angle value in radians. ❯ The cos values in arguments (the double type)	{Acos(-1)} - the angle in radians will be calculated for the value cos = -1, i.e. the angle will be ~ 3.14. {Acos(DataSource.Column

	Returns the angle value in radians of the double type	1)} - for all cos values, the angle in radians will be calculated.
{Asin()}	Displays the angle value in radians. ❯ The sin value in arguments (the double type) ❯ Returns the value of the angle in radians of the double type	{Asin(0)} - the angle in radians will be calculated for the value sin = 0, i.e. the angle is 0 {Asin(DataSource.Column1)} - for all sin values, the angle will calculated in radians.
{Atan()}	Displays the angle value in radians. ❯ The tan value in arguments (the double type) ❯ Returns the value of the angle in radians of the double type	{Atan(-1)} - the angle in radians will be calculated for the value tan = -1, i.e. the angle will be ~ -0.79 {Atan(DataSource.Column1)} - for all tan values the angle in radians will be calculated
{Ceiling()}	Displays the maximum integer value for a specified number ❯ The value is specified in arguments (the double, decimal type) ❯ Returns the value of the angle in radians of the double and decimal type	{Ceiling(25.124)} - It is worth noting that when this function is used, the number is not rounded. {Ceiling(25.9)} - the result is 26 {Ceiling(DataSource.Column)} - for all Column values, the nearest maximal integers will be found and displayed.
{Cos()}	Calculates and displays the cos value: ❯ The value of the angle in radians is specified in arguments (the double type) ❯ Returns double, decimal values	{Cos(0)} - the result is 1. {Cos(DataSource.Column1)} - for all values, the cos of the angle will be calculated.

{Div()}	Displays the result of the division of one argument to another: ❯ In arguments the following is specified: **1** The dividend and divisor (the double, decimal, long type). **2** The dividend and divisor and value, that is the result, if the divisor is equal to 0. ❯ Returns the value of the double, decimal, and long types	{Div(2,1)} - the result is 2, because 2 / 1 = 2 {Div(2,0,4)} - the result is 4, because the divisor is 0 and the third argument will be displayed {Div(DataSource.Column1, DataSource.Column2,Data Source.Column3)} - the results of dividing the Column1 values by the values of Column2 will be displayed. In this case, if Column2 contains zero values, then, instead of the result of division, in this line, the values from Column3 will be displayed.
{Exp()}	Displays the result of raising to the specified degree the number e: ❯ The arguments indicate the degree to which the number e must be risen (the long type) ❯ Returns the value of the double type	{Exp(4)} - the number e will be raised to the 4th degree. {Exp(DataSource.Column1)} - each value from Column1 will be the degree to which the number e will be raised.
{Floor()}	Displays the minimum integer value to the specified number: ❯ The value is specified in arguments (the double, decimal type) ❯ Returns the value of the double, decimal types	{Floor(123.59)} - the result will be 123, because this is the nearest minimum integer. It should be noted that this function does not round numbers. Floor(101.99)} - the result is 101 {Floor(DataSource.Column1)} - for all Column1

		values, the nearest minimum integers will be found and displayed.
{Log()}	Calculates the natural logarithm: ❯ The value is specified in arguments (the double type) ❯ Returns the value of the double type	{Log(x)}, where x is a number or an expression, the result is a calculation of the natural logarithm.
{Maximum(,)}	Compares the two values and displays the maximum: ❯ Two values are specified in arguments (the long, decimal, double type) ❯ Returns the value of the long, decimal, double types	{Maximum(5,9)} - the result is 9. {Maximum(DataSource.Column1,DataSource.Column2)} - all the Column1 values are equal to the Column2 values. The report will display the maximum numbers.
{Minimum(,)}	Compares the two values and displays the minimum: ❯ Two values are specified in arguments (the long, decimal, double type) ❯ Returns the value of the long, decimal, double types	{Minimum(5,9)} - the result is 5. {Minimum(DataSource.Column1,DataSource.Column2)} - all the Column1 values are equal to the Column2 values. The report will display the minimum numbers.
{Round()}	Rounds up the value to an integer or up to a certain number of decimal: ❯ In arguments, the following is specified: ① The value (the decimal, double types), ② Number of characters to	{Round(7.56)} - the result is 8 {Round(DataSource.Column1)} - all Column1 values will be rounded according to the mathematical rounding rules.

	which the fractional part should be rounded (the int type) ❯ Returns the value of the decimal, double types	{Round(5.7896541897,3)} - the result is 5.789 {Round(DataSource.Column1,2)} - all values from the data column will be rounded up to two decimal places in the fractional part, according to the mathematical rounding rules.
{Sign()}	Displays an indicator. For positive numbers 1, 0 - for all zero values, -1 - for negative values: ❯ The value is specified in arguments (the long, decimal, double types). ❯ Returns the value of the long type	{Sign(256)} - the result is 1. {Sign(0)} - the result is 0. {Sign(-157)} - the result is -1. {Sign(DataSource.Column1)} - to each value from Column1, depending on the sign of the number, an indicator will be assigned.
{Sin(0)}	Calculates sin of an angle: ❯ The value of an angle in radians is specified in arguments (the double type). ❯ Returns the value of the long type	{Sin(0)} - the result is 0. {Sin(DataSource.Column1)} - for all values, the sin angle is calculated.
{Sqrt()}	Calculates the square root of the number: ❯ The number is specified in arguments (the double type). ❯ Returns the value of the double type	{Sqrt(4)} - the result will be 2, because the square root of 4 is 2. {Sqrt(DataSource.Column1)} - for all Column1 values, the square root will be calculated.
{Tan()}	Calculates tg of an angle: ❯ The value of an angle in radians is specified in arguments (the double type).	{Tan(90)} - the result is ~ -1.995 {Tan(DataSource.Column1)} - for all values, the tan of the angle will be

	❯ Returns the value of the long type	calculated.
{Truncate()}	Displays only the integer part without rounding: ❯ The value is specified in arguments (the double, decimal types). ❯ Returns the value of the double, decimal types	{Truncate(Sqrt(5))} - the result will be number 2, because the square root of 5 is ~ 2.236, i.e. The whole part in this number is 2. {Truncate(DataSource.Column1)} - only the integer part of all Column1 values will be displayed.
Print State:		
{IsNull(,)}	Identifies null values in the specified data column. If there is a null value the result is true, otherwise - false. ❯ In arguments, the following is specified: ❶ The data source (the object type) ❷ The column name (the string type). ❯ Returns the value of the bool type	{Next(DataSource.Column)} - in the rendered report, instead of null values, the true values will be output, and instead of other values false values will be shown.
{Next(,)}	Displays the value from the next line. If the value of the next line is null, the result is 0. ❯ The data source is specified in arguments (the object type) and a column name (the string type). ❯ Returns the value of the object type	For example, the Column column contains values 2, 5, 9. Then, using the function {Next (DataSource, "Column")}, the first value will be 5, the second 9, and the third will be empty.

{NextIsNull(,)}	Compares the value of the string with the value of the next line. If the value of the next line is 0 or null, the result is true, otherwise - false. ❯ In arguments, the following is specified: ① The data source (the object type) ② The column name (the string type). ❯ Returns the value of the bool type	For example, the Column data column contains the values 2, 0, 9. Then, using the function {NextIsNull (DataSource, "Column")}, the first value is true, the second is false, the third is true.
{Previous(,)}	Displays the value from the previous line. If the value of the next line is null, the result is 0. ❯ In arguments, the following is specified: ① The data source (the object type) ② The column name (the string type). ❯ Returns the value of the bool type	For example, the Column column contains values 2, 5, 9. Then, using the function {Previous (DataSource, "Column")}, the first value will be empty, the second value will be 2, the third value will be 5.
{PreviousIsNull(,)}	Compares the value of the string with the value of the previous row. If the value of the previous line is 0 or null, the result is true, otherwise - false. ❯ In arguments, the following is specified: ① The data source (the object type). ② The column name (the	For example, the Column data column contains the values 2, 9, 0. Then, using the function {PreviousIsNull (DataSource, "Column")}, the first value is true, the second is false, the third is false.

	string type). ❯ Returns the value of the bool type.	
Programming Shortcut:		
{Choose()}	Displays the value by index. ❯ The arguments, specify the index and values. ❯ Returns values by index.	All product groups are grouped by category: expensive goods, medium price goods, cheap goods. An index is assigned to each group: expensive - index 1, average - index 2, cheap - index 3. The report should be displayed instead of their index - category. In this case, you can use the Choose function. {Choose(DataSource.Column1, "expensive", "average", "cheap")} - instead of index 1, the value expensive will be displayed, instead of index 2 - average, instead of index 3 - cheap.
{IIF(,,)}	Used to display a particular value, depending on the condition: ❯ In arguments, the condition is specified, the value if the condition is true (true) and the value if the condition is false (false) ❯ Returns the value of the object type	In the inventory report, you need to track the number of items. The logistician's task is that, when the quantity of goods is coming to 0 (less than 6), it is necessary to order these goods. In order to highlight critical positions in the report

		visually you can use the function {IIF („)}
		{IIF(DataSource.Column1 > 6,"Minimum","Normal")}, where DataSource.Column1 - the column with the values of the quantity of goods, 6 - the extreme quantity of goods, Minimum - the value that will be displayed if the stock of goods is less than 6, Normal - the value to be displayed if the stock of goods is 6 or more.
{Switch()}	Assigns the specified value when the specified condition is complete: ❯ In arguments, specify the condition and the value that will be assigned, if the condition is complete. Such condition-value pairs can be specified ❯ Returns the value of the object type	For example, a list of employees is displayed in the report, and you need to display their position: Nancy is the lead project manager, Andrew is the chief developer, the remaining employees (6 people) are Juniors. In this case, the Switch function will have three pairs of "condition-value" arguments: {Switch(Employees.FirstName == "Nancy", "Manager", Employees.FirstName == "Andrew", "Developer", Employees.FirstName! = "", "Junior")}

Strings:

{Arabic()}	Converts these numbers to Arabic numerals: ➤ The value is specified in arguments (the string or int types) ➤ Returns the value of the string type	{Arabic(2)} - the number 2 will have an Arabic spelling. {Arabic(DataSource.Column1)} - all the numbers from Column1 will have an Arabic spelling.
{DateToStr()}	Converts date to text value: ➤ In arguments, the following is specified: ① Date (the DateTime type) ② Boolean values (true or false) for displaying the header that starts with a capital letter or with a lowercase letter. ➤ Returns the value of the string type	{DateToStr(DataSource.Column1)} - all dates from Column1 will be displayed in text form. {DateToStrPl(DataSource.Column1,true)} - dates will be displayed in text form, in Polish and the first character is a capital letter. {DateToStrPl(DataSource.Column1,false)} - dates will be displayed in text form, in Polish and the first character is a lowercase letter. {DateToStrPtBr(DataSource.Column1)} - the dates will be displayed in text form in the Brazilian language.
{Insert(,,)}	Inserts a value after a certain character into another value: ➤ In arguments, the following is specified: ① The value in which to insert text (the string type), ② The number of a character, after which the	{Insert("25",2," dollars")} - in the value 25, after the second symbol, the value dollars will be inserted, i.e. the result will be 25 dollars. {Insert(DataSource.Column1,2,DataSource.Column2)}

	value is inserted (the int type), ③ The value for insertion (the string type) ❯ Returns the value of the string type	- in the Column1 value, after the second character, Column2 values will be inserted. For example, Column1 - contains the value of Category, Column2 - Products, then the result will be CaProductstegory.
{Left()}	Displays the specified number of characters from the left side of the value: ❯ The value is specified in arguments of the string type string and the number of characters to be displayed (the int type) ❯ Returns the value of the string type	{Left("Beverages", 4)} - only four characters from the Beverages value will be displayed, i.e. the result will be Beve. {Left(DataSource.Column1, 2)} - only the first two characters for each Column1 value will be displayed.
{Length()}	Displays the number of characters for the specified value: ❯ The value is specified in arguments (the string type) ❯ Returns the value of the int type	{Length("Beverages")} - the result will be the number 9, because the value Beverages consists of nine characters. {Length(DataSource.Column1)} - for each Column1 value, the number of characters will be calculated and this result will be displayed.
{Mid()}	Displays characters from a value. In this case, you can set the reference position: ❯ In arguments, the following is specified:	{Mid("Beverages",2,3)} - three symbols will be displayed, after the first two, i.e. the result will be ver.

	① The value (the string type) ② Index of the reference position (the int type) ③ Number of characters to display (the int type) ❯ Returns the value of the int type	Mid(DataSource.Column1, 3,2)} - 2 characters will be displayed after the first three for all values.
{Persian()}	Converts specified numbers to numbers in Persian: ❯ The value is specified in arguments of the string or int types ❯ Returns the value of the string type	{Persian(5)} - number 2 will have Persian spelling. {Persian(DataSource.Column1)} - all the numbers from Column1 will have Persian spelling.
{Remove()}	Deletes the specified number of characters from the index of a specific position: ❯ In arguments, the following is specified: ① The value (the string type) ② Index of the reference position (the int type) ③ Number of characters to delete (the int type) ❯ Returns the value of the int type	{Remove("Beverages",2,3)} - after the second character, three characters will be deleted, i.e. the result is Beages. {Remove(DataSource.Column1,3,2)} - for all values from Column1, two characters will be deleted after the first three.
{Replace(,,)}	Replaces certain characters or their combination with other characters: ❯ In arguments, the following is specified: ① The value (the string type) in which the replacement will be made	{Replace("Beverages","ver", "NEW")} - in the value Beverages, the ver characters will be replaced by the characters NEW, i.e. the result is BeNEWages. {Replace(DataSource.Colu

	② Characters to be replaced (the string type) ③ Characters to be inserted (the string type) ❯ Returns the value of the string type	mn1, "rex","sum")} - for Column1 values, in which the combination of rex characters occurs, rex will be replaced by sum. In values where there is no combination of rex, a replacement will not be done.
{Right()}	Displays the specified number of characters from the right side of the value: ❯ The value is specified in arguments of the string type and the number of characters which should be displayed (the int type) ❯ Returns the value of the string type	{Right("Beverages",3)} - three characters from the right side of the value will be displayed, i.e. ges. {Right(DataSource.Column 1,4)} - for each Column1 value, four characters will be displayed from the right side.
{Roman()}	Converts Arabic numerals to Roman numerals: ❯ In the arguments, specify the number (the int type) ❯ Returns the value of the string type	{Roman(4)} - the number 4 will have a Roman spelling. {Roman(DataSource.Colu mn1)} - all the numbers from Column1 will have a Roman spelling.
{Substring()}	Displays a certain number of characters from the specified position: ❯ In arguments, the following is specified: ① The value (the string type) from which the characters will be displayed ② The index of position (the int type), i.e. how many characters are skipped ③ Number of characters to display (the int type)	{Substring("Beverages",6,3)} - the first six characters are skipped and three characters will be displayed, i.e. the result is ges. {Substring("Beverages",0,2)} - two characters will be displayed starting with zero, i.e. the result will be Be.

	❯ Returns the value of the string type	{Substring(DataSource.Column1,1,4)} - the first character is skipped and four are counted starting from the second one. This is the result for each Column1 value, which is displayed in the report.
{ToCurrencyWords()}	Displays the currency value as the text: ❯ In arguments, the following is specified: ① Numeric value (the double, decimal, long types) ② You can also specify values (true or false) to display them with a capital letter and display cents (type bool) ③ Single and plural formats for currency and cents (the string type) ④ It is possible to specify the ISO code (the string type) ⑤ You can also specify a base unit for the integer part and a fractional In addition, various combinations of arguments are possible. There are also some types of this function that support different cultures. ❯ Returns the value of the string type	{ToCurrencyWords(100)} - the used currency is dollars of USA, so the result will be One hundred dollars and zero cents. {ToCurrencyWords(Products.UnitPrice)} - all product prices will be displayed as text. {ToCurrencyWords(100,true,false)} - the result will be displayed with the first capital letter (since it is set to true) and without displaying cents (since it is set to false), i.e. the result will be One hundred dollars. {ToCurrencyWordsEnGb(1.25,"EUR",2)} - the ISO code EUR will be applied and the result will be one euro and twenty-five cents. {ToCurrencyWordsEnIn("dollars","cents",1.25M,0,true)} - the base unit for the

		integer part as dollars will be specified, the fractional part - cents, the number for conversion 1.25, then the number of decimal signs to convert and the value true means that the entry will start with the capital letter.
{ToLowerCase()}	Displays the value in lowercase: ❯ The value is specified in arguments (the string type) ❯ Returns the value of the string type	{ToLowerCase("EURO")} - the result is euro. {ToLowerCase(DataSource.Column1)} - all values of this column will be displayed in lowercase.
{ToOrdinal()}	Converts numerals to ordinal: ❯ The value is specified in arguments (the long type) ❯ Returns the value of the string type	{ToOrdinal(25)} - - the result is 25th. {ToOrdinal(DataSource.Column1)} - all the values of this column will be converted to ordinal numerals.
{ToProperCase()}	Converts the text to the format - the first character is capital, the rest characters are in lowercase: ❯ The value is specified in arguments (the string type) ❯ Returns the value of the string type	{ToProperCase("dOllars")} - - the result is Dollars. {ToProperCase("dollars")} - - the result is Dollars. {ToProperCase("dOLLARS")} - - the result is Dollars. {ToProperCase(DataSource.Column1)} - all values from this column will be with the first capital letter and the rest ones in lowercase.
{ToUpperCase()}	Displays the value in uppercase: ❯ The value is specified in arguments (the string type)	{ToUpperCase("dollars")} - the result is DOLLARS. {ToUpperCase("dOllars")} - the result is DOLLARS.

	➤ Returns the value of the string type	{ToUpperCase("dOLLARS")} - the result is DOLLARS. {ToProperCase(DataSource.Column1)} - all values will be written in uppercase.
{ToWords()}	Displays the numerals as text: ➤ In arguments, the following is specified: ① A numeric value that will be converted to text (decimal, double, long) ② True or false values whether to display the first character with a capital letter ③ True or false values to return null and empty values ④ It is also possible to specify true or false values to provide the feminine form for the result ➤ Returns the value of the string type	{ToWords(100)} - the result is one hundred. {ToWords(100, true)} - the result is One hundred. {ToWordsEnIn(0,false)} - the result is Zero. {ToWordsEnIn(0,true)} - there will be no results. {ToWordsEs(100,true,true)} - the result starts with a capital letter and in feminine form, i.e. Cien {ToProperCase(DataSource.Column1)} - all values will be displayed in text.
{Trim()}	Trims the spaces at the beginning or end of the line: ➤ The value is specified in arguments (the string type) ➤ Returns the value of the string type	{Trim(" <1 dollars> ")} - the result in this case is <1 dollars>". {Trim(DataSource.Column1)} - the spaces before each value and after each value will be truncated.
{TryParseDecimal()} {TryParseDouble()} {TryParseLong()}	Checks the value for conversion to decimal, double, long: ➤ The value is specified in	{TryParseLong("100")} - The value can be converted to long. {TryParseLong(" { 100")} -

	arguments (the string type)	the result is false. The
	❯ Returns a value of the bool type. If true, then the conversion will be successful, otherwise it will be false.	value cannot be converted to long. {TryParseLong(DataSource.Column1)} - each value will be checked on possibility to be converted to long.

Totals in-depth

3.35.1 Totals

In the report, besides the list of data and its title, totals are present. This can be the amount, quantity, minimum, average, maximum value for a particular source, band or page. Depending on the desired result, you should select the type of the total function. All results of the functions can be divided conditionally into two types:

❯ Associated with bands. In this case, the results are calculated at the time of the report creation process. Every time, when one operation is performed with Data band, a single value is calculated. Appropriately, the text component with the total should be placed on any bands which associates directly with any band that is associates with the Data band.

❯ Not associated with bands. In this case, calculation of totals is not associated with the operation of rendering the Data band. Consequently, the text component with the total functions can be located anywhere in the report. It is worth noting that all functions have the Totals prefix. It is the format - {Totals.Functions ()}.

3.35.1.1 Totals Associated with Bands

In order to calculate and display the total, you should place a text component in the report, call the editor and go to the Summary tab.

❶ The **Expression** field. This field specifies an expression of calculating totals. The expression can be specified manually or it will be generated automatically, depending of type of other parameters.

❷ The **Summary Function** field. In this field, the function of calculating totals is selected.

❸ In this field you can specify the **Data band** by which the total will be calculated.

4 In this field you can specify by the data column which values will be used to calculate the total.

5 Using the radio buttons, you can set the object for calculating totals:

> ❯ **Report**. The total will be calculated for the entire report.
> ❯ **Column**. Totals will be calculated by every column in the report.
> ❯ **Page**. Totals will be calculated by every page of the rendered report.

6 The Running Total parameter. If the flag is checked, then the total will be calculated as running. If unchecked, then the total will be calculated only by the project (report, column, page).

7 The Conditions parameter. If the flag is checked, then the condition will be considered when calculating totals. If unchecked, then the total will be calculated without considering conditions.

8 The field, which specifies an expression of a condition.

The type of the total function result

By default, the function for calculating totals return the Decimal type (except for the functions - Count and CountDistinct). However, you can also make calculations using two other data types - Double, and Int64. For the function returns the result of the calculation using the Double data type, add the Latin letter D in the upper register to the name of the function. For calculations using the Int64 type, you should add the Latin letter I in the upper register. This separation will allow to avoid losses in the calculation of totals.

Function	Type of returned value
Sum()	Decimal
SumD()	Double
SumI()	Int64

⚠ Notice: The letters I, D can be added to any function except Count and CountDistinct. These functions always return the Int64 type.

Some words about the function syntax

When using the C# programming language, all the functions should be written strictly in compliance with the register.

❯ Sum (expression) - the sum is calculated by automatically identified object.

❯ Sum (band, expression) - the sum is calculated by the particular object.

❯ SumIf (band, expression, condition) - the sum is calculated by the object with the condition.

❯ expression - the expression for calculation;

❯ band - the name of the band to execute calculation;

❯ condition - the condition of including the calculation into the expression.

In the case with calculations by a page or a container, the syntax is the same except for the addition of the Latin letter c as the prefix to the function name:

❯ cSum (expression) - calculation of the sum by the page or container;

❯ cSum(band, expression) - calculation of the sum by the page or container and object on it;

❯ cSumIf (band, expression, condition) - calculation of the sum by the page or container and object on it, under certain conditions.

To calculate totals by the column, add the col prefix to the function name:

❯ colSum (expression) - the sum is calculated by the column;

❯ colSum (band, expression) - the sum is calculated by the column and an object in it;

❯ colSumIf (band, expression, condition) - the sum is calculated by the column and objects in it, under certain condition.

The Count function differs from the rest of the functions that it has no expressions for calculation. The syntax for this function is shown below.

❯ Count() - calculates the number of rows;

❯ CountIf(condition) - calculates the number of rows by the condition;

❯ Count(band) - calculates the number of rows by the object;

❯ CountIf(band, condition) - calculates the number of rows by the object and condition;

❯ cCount() - calculates the number of rows by the page and container;

❯ cCount (band) - calculates the number of rows by the page (container) and an object on it;

❯ cCountIf(band, condition) - calculates the number of rows by the page (container) and an object on it under certain condition;

❯ colCount() - calculates the number of rows by the column;

❯ colCount (band) - calculates the number of rows by the column and the object in this column;

❯ colCountIf(band, condition) - calculates the number of rows by the column and the object in the column under certain condition.

Showing totals in any place

Typically, the components, in the text expression of which the function call is specified, are placed on the footer band on the Data band. There are several types of footer bands:

❯ ReportSummaryBand - the band is used to display totals for the entire report;
❯ PageFooterBand - the band is used to display totals by the page;
❯ FooterBand - the band is used to display totals by the list;
❯ GroupFooterBand - the band is used to display totals by the group.
❯ ColumnFooterBand - the band is used to display totals by the column.

Position of components with functions in any of the bands mentioned above allows the report generator to determine exactly to what data band this function is applicable. Also a component with the functions can be placed on the Data band. In this case, on each data row, the result of the function calculation by all rows will be displayed.

If you want to display the total, for example, on the Header band, then this is done using a script. However, in Stimulsoft Reports, the component with the function may be in any band of the report.

⚠ Notice: The components with functions can be placed anywhere in the report.

It is also allowed to place a component with the function on a page and on other pages of the report template. For example, it is possible to calculate the sum of values by the list and output it in the header list. Another example, calculate the number of rows in the list and output the value at the beginning of the page. At the same time, there is a limitation: you must specify the Data band, in which the result will be calculated:

❯ {Sum (DataBand1, Products. UnitsInStock)}. In this case, the total will be calculated for the Products.UnitsInStock column values for each row of DataBand1.
❯ {Count (DataBand1)}. In this case, the number of rows of DataBand1 will be calculated.

Expressions with functions

To calculate the totals, it is possible not to specify additional arguments in the expression. For example, for the Count function, it is optional or only one argument can

be set for the Sum function - an expression that should be calculated. All this is possible if the report generator can determine to which Data band those functions are related.

> ⚠ Notice: The report generator can determine the relationship between functions and specific Data band, if the component with this function is related to the band with the Data band. In other words, the component with the function is located on the Header and Footer bands that relate to this Data band.

Otherwise, in the arguments, you should specify the data source or a Data band on which it is necessary to calculate the total. The following can be specified in expressions:
> The object which values will be calculated - {Sum (DataSource.Column)}
> The object and various mathematical operations with them - {100 + Sum (DataSource.Column) * 2}

Calculation of totals by the page

In order to calculate the total by the page or panel, you should add the Latin letter "c" in lower case as a prefix to the name of the function:

> {AcCount (DataBand1)} - the report engine calculates the number of lines on one page or a panel.

> ℹ Notice: The calculation of totals by the page has the same principle as for the panel.

When calculating totals by the panel or page, it is desirable to specify the Data band by which goes the calculation of the aggregate function. It is necessary because there may be more than one Data band on one page.

On one page or panel you can use any number of aggregate functions. Stimulsoft Reports does not have any restrictions with this. It is allowed to combine the totals by the page with the condition. For example:

> {CountIf(DataBand1, Products.UnitsInStock = 0)} - the report engine calculates the number of items on this page which are equal to zero.

Calculation of totals by column

In order to calculate the total by a column, you must add the prefix col (from the word column) in the lowercase to the name of the function. For example:

> {ColCount ()} - the report engine calculates the number of rows in each column.

⚠ Notice: The calculation of totals by a column in Stimulsoft Reports has one limitation. Totals can be calculated only by the columns on the page. Calculation of totals by the columns on the Data band are not allowed.

When calculating totals by the column, it is desirable to place text components with functions on ColumnHeader, ColumnFooter, Header or Footer bands. You can calculate an unlimited number of totals by the column. There are no restrictions in this. It is also allowed to combine the footers by the column with the condition:

> {ColCountIf(DataBand1, Products.UnitsInStock = 0)} - the report engine calculates the number of rows in each column, where the condition will be executed.

Calculating totals in the event code

Using Stimulsoft Reports you can calculate functions in code of the report event. It provides the ability to calculate the more complex functions. Also in this case, you can refer to the calculated value from the code in the process of calculation and to influence this process. In order to make this calculation, you should create a variable in the data dictionary, which will store the value of the function.

ℹ Notice: Do not use variables declared in the code to store the result of calculation of functions. You must use the variables from the data dictionary.

When creating a variable, the data type of the variable is indicated. For example, Decimal, and the initial value, for example, 0. Then, in the Data band, indicate an expression to increment variable in the Rendering event. For example, if you want to calculate the sum of the values by the field Products.UnitPrice field, the expression will be the following:
> Variable + = Products.ItemsInStock;

In order to display the result of calculation, you should place the text component with the expression:
> {Variable}

Also, you must have a text component with the expression {Variable}, set the Process At property to the End of Report value. It is necessary that the report generator calculates

the value of the variable after processing the remaining components.

Calculation of totals with condition

Sometimes, when calculating totals, it is necessary to consider certain values. In this case, the condition is set in the function of calculating totals. For example, it is necessary to sum the values that are greater than zero. To add a condition to the function of calculating the totals, you should to add a suffix If (the Latin alphabet) to the function name, and add an additional argument with the condition:

➤ {SumIf (Products.UnitsInStock, Products.UnitsInStock> 1)}. In this case, the amount of Products.UnitsInStock values will be calculated which is greater than 1.
➤ {CountIf (Products.UnitsInStock == 0)}. In this case, the number of rows with a zero value in the column is calculated UnitsInStock

> 🛈 Notice: If you want to make a calculation using a Double or Int64, you must first add the Latin letter D or I, and then the word If. For example: {SumDIf(Products.UnitsInStock,Products.UnitsInStock> 0)}.

Totals and automatic changing the size of the component

> ⚠ Notice: When rendering a report, at the moment, when the size of the component is determined, the result of the calculation of the total function is still unknown. This should be considered when installing the automatic resizing for the components in which the calculation of totals is done. Otherwise, an issue may arise when the size of the component will not be correct in relation to the result of calculation of the total function.

Totals with the disabled Data band

The Data band can be disabled in a variety of ways. For example, it can be disabled by a certain condition or it may have a zero height. By default, when rendering a report, the report engine does not take into account disabled data bands and will not process them. However, if it is necessary to calculate totals by the disabled Data band, then you should set the CalcInvisible property for this band to true. In this case, the report will only be displayed the Data included bands, and calculation of totals will be executed considering the Data band.

Calculating totals in Master-Detail reports

When calculating totals in hierarchical reports, there are some issues in calculating the result. Consider an example based on the Master-Detail Report. Suppose the report shows a list of product categories. Categories, in this case, are master entries, and products are detail entries:

Suppose you want to count the number of products to be displayed in the report. If you add the Footer band with the function Count() to the band with a list of products (detail records), then, by each category (master record), totals will be calculated:

	Master Detail			Stimulsoft
	The sample demonstrates how to create a Master-Detail report.			Date: July 2016
	Beverages Soft drinks, coffees, teas, beers, and ales			
	Name	Quantity per unit	Price	Units in stock
1	Chai	10 boxes x 20 bags	$18.00	39.00
2	Chang	24 - 12 oz bottles	$19.00	17.00
3	Chartreuse verte	750 cc per bottle	$18.00	69.00
4	Côte de Blaye	12 - 75 cl bottles	$263.50	17.00
5	Guaraná Fantástica	12 - 355 ml cans	$4.50	20.00 ✓
6	Ipoh Coffee	16 - 500 g tins	$46.00	17.00
7	Lakkalikööri	500 ml	$18.00	57.00
8	Laughing Lumberjack Lager	24 - 12 oz bottles	$14.00	52.00
9	Outback Lager	24 - 355 ml bottles	$15.00	15.00
10	Rhönbräu Klosterbier	24 - 0.5 l bottles	$7.75	125.00
11	Sasquatch Ale	24 - 12 oz bottles	$14.00	111.00
12	Steeleye Stout	24 - 12 oz bottles	$18.00	20.00
				Count: 12

If you add the Footer band with the function Count() to the band with the categories, the result is the number of master entries in the report, i.e., the number of categories. However, in the Master-Detail report, you can calculate the totals immediately for all detail records. In this case, you must specify the names of both (master and detail) bands as a function with a colon: Count(MasterBand:DetailBand).

The result of the Count(MasterBand:DetailBand) function is the number of products in all categories.

3.35.1.2 Totals not Related to Bands

The calculation of totals in reports can be made by specifying an expression, for example, {Sum(DataBand1)}. At the same time, the totals are calculated when rendering the report: each time when an operation with DataBand in carried out, a single value is calculated. Then, all calculated values are added together and the total value will be displayed. In this case, totals are associated with bands. The calculation of totals in Stimulsoft Reports can be made in another way - instantly. In other words, calculate the total not associated with bands. To do this, use the special prefix Totals before the function with the separator ".", For example, {Totals.Sum(DataBand1)}. Calculation of functions with the prefix Totals occurs where the function was called, as opposed to the totals associated with bands, the calculation of which is performed during rendering the report.

Totals Functions:

Function	Description	Sample

{Avg()}when	Calculates the arithmetic mean: ➤ In arguments, specify an object or two objects ➤ Returns values of different types (double, decimal, long, DateTime, TimeSpan), depending on the function selected.	{Avg(DataSource.Column1)} - the arithmetic mean of the Column1 column will be calculated. {AvgDate(DataSource.ColumnDate)} - the average of the date on the ColumnDate column will be calculated. {AvgTime(DataSource.ColumnTime)} - the average time by the ColumnTime column will be calculated. All functions can have two arguments. For example, the report uses several Data bands. It is necessary to calculate the arithmetic mean for the first Data band. In this case, the first argument is the band, the second is the object for calculation - {Avg(DataBand1,DataSource.Column2)}. As a result, the arithmetic mean of Column2 will be calculated, but only for the DataBand1 values.
{Count()}	Calculates the number of values or the number of unique values: ➤ In arguments, specify a value, object or two objects ➤ Returns the values of various long types	{Count()} - the result is the number of entries in the data source. {Count(DataBand1, DataSource.Column1)} - the result is the number of entries in Column1 column for DataBand1. {CountDistinct(DataSource.Column1)} - the result is the number of unique entries in DataSource.Column1. {CountDistinct(DataBand2, DataSource.Column2)} - the result is the number of unique entries in Column2 column for DataBand2.
{First()}	Displays the first value from the specified object: ➤ In arguments, specify an object or two objects	{First(DataSource1.Column1)} - the result is the first value of Column1 from the DataSource1. {First(DataBand2, DataSource.Column2)} - the result is the first value of Column2 of the

	❯ Returns the values of various object types	DataBand2 band.
{Last()}	Displays the last value from the specified object: ❯ In arguments, specify an object or two objects ❯ Returns the values of various object types	{Last(DataSource1.Column1)} - the result is the last value of Column1 from the DataSource1. {Last(DataBand2, DataSource.Column2)} - the result is the last value of Column2 of the DataBand2.
{Max()}	Displays the maximum value from the specified object: ❯ In arguments, specify an object or two objects ❯ Returns the values of various double, decimal, long, DateTime, TimeSpan, string types depending on the function selected.	{Max(DataSource1.Column1)} - the result is the maximum value from Column1 of DataSource1. {MaxDate(DataSource1.ColumnDate)} - the result is the maximum date from the ColumnDate of the DataSource1. {MaxTime(DataSource1.ColumnTime)} - the result is the maximum time from the ColumnTime of the DataSource1. {MaxStr(DataSource1.Column1)} - all values will be sorted alphabetically. The result is the last value. {Max(DataBand2, DataSource.Column2)} - the result will be the maximum value of Column2 of the DataBand2 band.
{Median()}	Displays the mean (non-arithmetic) value from the list: ❯ In arguments, specify an object or two objects ❯ Returns the values of various double,	Suppose, Column1 contains 5 values: 2, 5, 6,1,7. The {Median(DataSource1.Column1)} function displays the average value from this list, i.e. the result is 6. {Median(DataBand2, DataSource.Column2)} - the result is the average value of Column2 of the DataBand2.

	decimal, long types, depending on the function selected.	
{Min()}	Displays the maximum value from the specified object: ❯ In arguments, specify an object or two objects ❯ Returns the values of various double, decimal, long, DateTime, TimeSpan, string types depending on the function selected.	{Min(DataSource1.Column1)} - the result is the minimum value from Column1 of DataSource1. {MinDate(DataSource1.ColumnDate)} - the result is the minimum date from the ColumnDate of the DataSource1. {MinTime(DataSource1.ColumnTime)} - the result is the minimum time from the ColumnTime of the DataSource1. {MinStr(DataSource1.Column1)} - all values will be sorted alphabetically. The result is the first value. {Min(DataBand2, DataSource.Column2)} - the result is the minimum value of Column2 of the DataBand2 band.
{Mode()}	Displays the value that is most common in the list of values: ❯ In arguments, specify an object or two objects ❯ Returns the values of various double, decimal, long types depending on the function selected.	{Mode(DataSource1.Column1)}. Suppose, Column1 contains a list of values: 2, 2, 6, 7, 7, 8, 7, 6, 5, 9, 4. In this case, the result is 7, because it is most often repeated in the list of values. {Mode(DataBand2, DataSource.Column2)} - the result will be the value from Column2 of the DataBand2, which is the most common.
{Rank(,)}	Display the rank of the value. The prefix Totals is mandatory: ❯ Specify in arguments: ① Objects for	{Totals.Rank(DataBand1,DataSource.Column1)}. Suppose, Column1 contains a list of values: 44, 9, 36, 55, 71. In this case, the values will be sorted in ascending order, i.e. 9, 36, 44, 55, 71 and to each of them a rank will be assigned. The number 9 will receive rank 1; 36 - rank 2;

| | processing and assigning rank (the object type) ❷ The value (true or false) for assigning a tight or not tight rank ❸ Sorting direction of values. ❯ Returns the values of various long types | 44 - rank 3; 55 - rank 4; 71 - rank 5. By default, calculates a tight rank and sorts the values for assigning rank by ascending order

{Totals.Rank(DataBand1,DataSource.Column1, true, StiRankOrder.Dess)} - in this case, there will be a tight rank since it is set to true. When the rank is assigned, the values will be sorted in descending order since StiRankOrder is set to Desc. For sorting in ascending order (used by default), you should set to Asc (StiRankOrder.Asc).

An example of a not tight rank is {Totals.Rank(DataBand1, DataSource.Column1, false, StiRankOrder.Asc)}. Assume Column1 contains a list of values: 44, 9, 44, 9, 31, 64,68, 71. The values are assigned in ascending order, i.e. 9, 9, 31, 44, 44, 44, 68, 71. In this case, the ranks will be as follows: 9 - rank 1, 9 - rank 1, 31 - rank 3, 44 - rank 4, 44 - rank 4, 44 - rank 4, 68 - rank 7, 71 - rank 8. In other words, when assigning a rank to a number, the rank of the previous value and the number of values with this rank are taken into account. |
|---|---|---|
| {Sum()} | Displays the result of the sum of the values: ❯ Specify in arguments: ❶ Objects for processing and assigning rank (type object) ❷ Condition ❸ Summation expression ❯ Returns the values of various long, decimal, double, | {Sum(DataSource1.Column1)} - the result is the sum of all Column1 values in the DataSource1.

{SumDistinct(DataSource1.Column1)} - the result is the sum of all the unique Column1 values in the DataSource1.

SumTime(DataSource1.Column1) - the result is the sum of the time from Column1 in the DataSource1.

{Sum(DataBand2,DataSource2.Column2)} - the result is the sum of the values from Column2 of the DataBand2. |

	TimeSpan types	{SumDistinct(DataBand1,DataSource.Column1, DataSource.Column2)} - the result is the sum of the Column2 values that correspond to the unique values from Column1 of the DataBand2.

Samples to calculate totals not associated with bands.

For example, there is a Master-Detail Report, which is a list of products by categories:

In this report, the result can be calculated for each category for the entire report. It is also possible to calculate the proportion of each category of the total. To begin, let's calculate the amount of product in a category. To do this, add the Footer band in the report template, put a text component with an expression of calculating totals {Sum(DataBand2,Products.UnitPrice)}. For the summation of values, the Sum function is used, its arguments specify the object by which the totals and data column will be calculated, the values of which will be summarized. Since it is necessary to calculate the

amount of product by each category, the object for calculating totals will be the detailed Data band, i.e. DataBand2. Values in the UnitPrice column indicate the price of each product. Therefore, the sum of these values will be the total for the category:

In this case, the result is associated with the Data band. To calculate the total by the report, use the functions which are not associated with bands. For this, add a prefix Totals to the function, through the "." separator. As an object, you should specify the data source. The expression for calculating totals in the report, will be {Totals.Sum(Products, Products.UnitPrice)}. The result is displayed on the master band.

Each time, when the master band is printed in the report, the total by the report will be shown. Using the results of calculations, it is possible to calculate the share of each category of the grand total. The result is displayed as a percentage. To calculate the proportion, you should divide the total by the category on the total by the report - {(Sum(DataBand2,Products.UnitsInStock) / Totals.Sum(Products, Products.UnitsInStock))}. For the text component, in which the share is displayed, set the percentage format. The result is displayed on the master band.

Thus, you can calculate any total in the report. To calculate the total not associated with bands you should use the prefix Totals to the name of the function, and use the "." separator.

3.36   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.

3.36.1   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.

3.36.1.1 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.
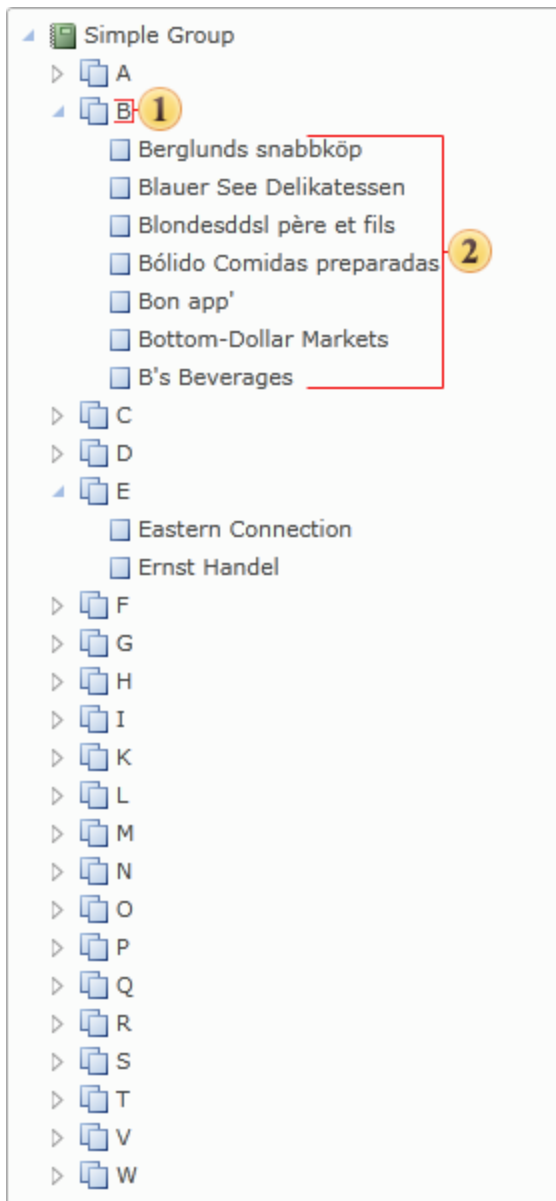
3.36.1.2 Bookmarking Using Code

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:
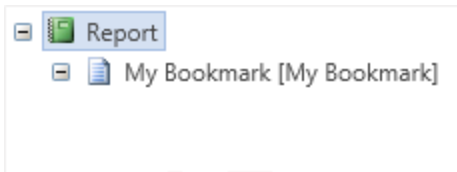
To add a bookmark of the second level to the tree it is necessary write the following code:

```
Bookmark["My Bookmark"].Add("Bookmark Level2");
```
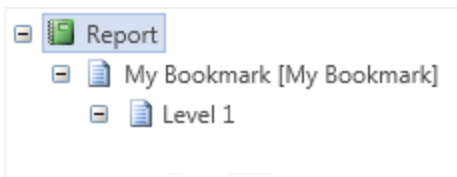

...and for the third level:

```
Bookmark["My Bookmark"]["Level2"].Add("Bookmark Level3");
```
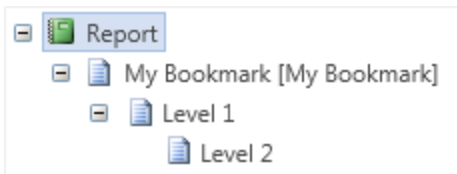

To create all three bookmarks the code sample shown above can be used. Stimulsoft Reports automatically checks presence of each bookmark in a tree and will add ones which should be added. Sometimes it is required to organize navigation using bookmarks. If it is necessary to find components the **Interaction.Bookmark** 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.

3.36.1.3 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:
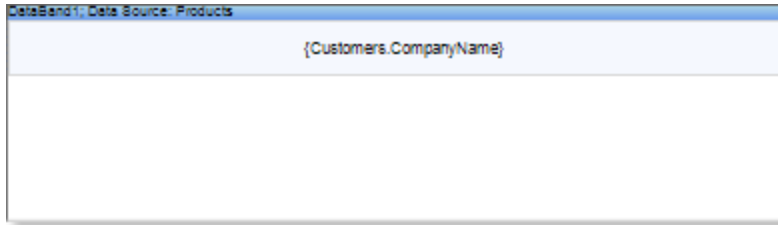
When rendering the report a list of companies will be built, but the tree of bookmarks will not be shown. To show the hierarchy of bookmarks it is necessary to specify an expression (see below an axample):

%\{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:
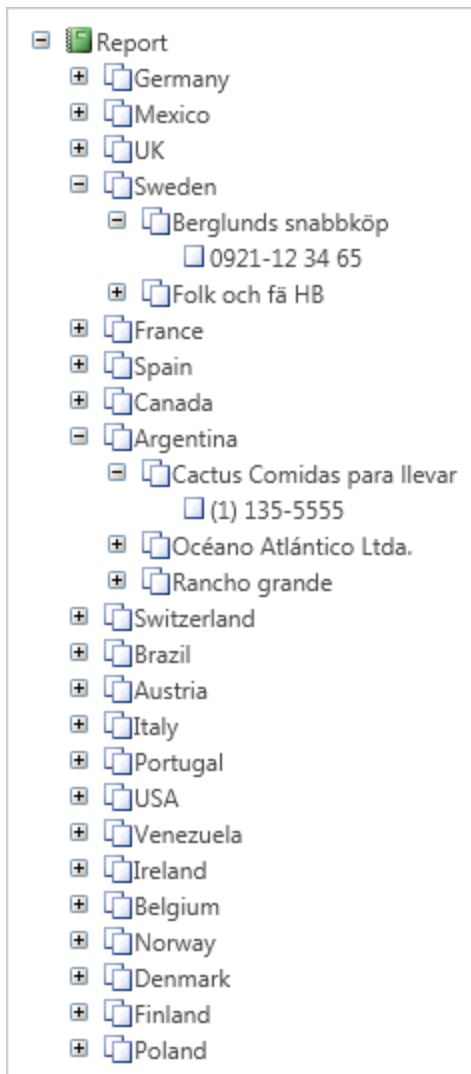
3.36.1.4 Bookmark Nesting

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 **Group Header** band, is one level higher then the bookmark, created by the **Data** band, 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.

3.36.2 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.

3.36.2.1 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.

3.36.2.2 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.

3.36.2.3 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

3.36.3 Drill-Down Reports

In Stimulsoft Reports it is possible to create an interactive report with detailing. The report detailing refers to additional interpretation of data in the report. Usually interpretation is done when you click on any item. After that, there occurs a detailed report rendering in a new tab in the viewer. The picture below shows the viewer window with detailed tabs:
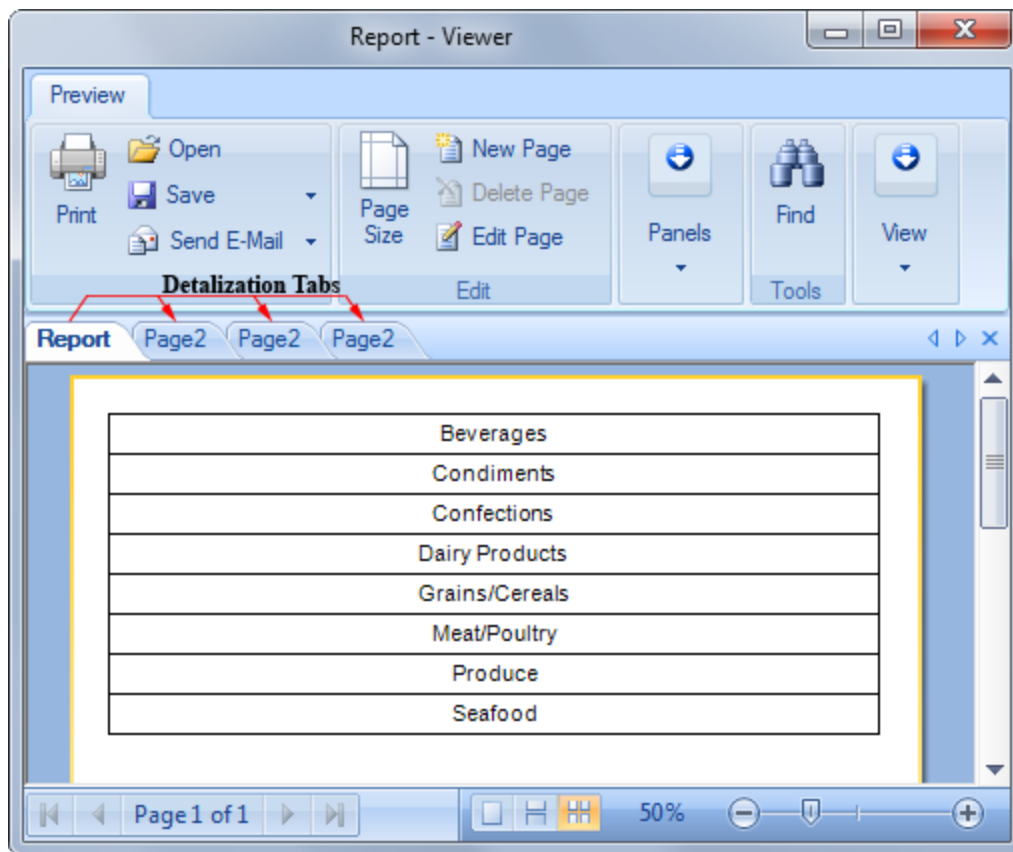
It should also be noted that the specification can be multi-level. In other words, detailing can also be interpreted, i.e. an hierarchy of detailing can be built. For example, a report with the names of categories will have details of products within a specific category. A report with products will have detailing by producers, for a particular product, etc. The picture below schematically shows the levels of detailing:
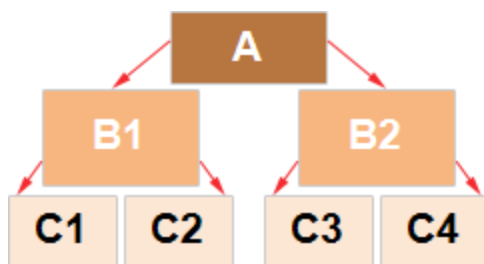
As can be seen from the picture above, a report can be interpreted as reports **B1** and **B2**. This is the first level of detailing. Reports **B1** and **B2**, in turn, have detailing as reports **C1**, **C2**, **C3** and **C4**. This is a detailing of the second level. Consider the creation of frill-down reports in more detail.

3.36.3.1 Drill-Down Report Using Report Pages

The drill-down report using a report page means an interactive report in which detailing goes using a different page of this report template. To create this report, you should set the value of the **Interaction.Drill-Down Page** property for a component, which should be detailed. The value specifies a page with detailed information. Consider the example of a **Drill-Down Report** using the page. The **Data Band** and a text component in it should be placed in the first page of the report template. Specify the data source **Shippers** for the band. In the text component indicate the expression **{Shippers.ShipperID}** and **{Shippers.CompanyName}**. On the second page of the report put a **Data Band** and a text components in it, select the data source **Orders** for this band. Insert the expressions in the text components: **{Orders.ShipVia}**, **{Orders.ShipName}** and **{Orders.ShipCountry}**, respectively. The picture below shows two pages of the report template:
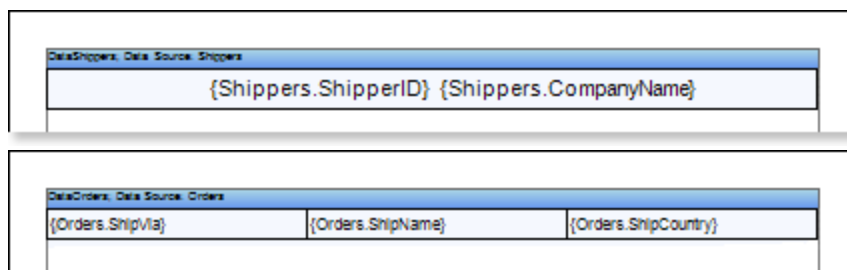
Also, add the **Header Band** on a page with detailed data. Then, select the text component with expressions **{Shippers.ShipperID}** and **{Shippers.CompanyName}** and change the values of some properties. The **Interaction.Drill-Down Enabled** property must be set to **true**. Then, set the value of the **Interaction.Drill-Down Page** property to the page on which the detailed data are placed. In this case, it is the **Page2**. The picture below shows a window for selecting detailing pages:
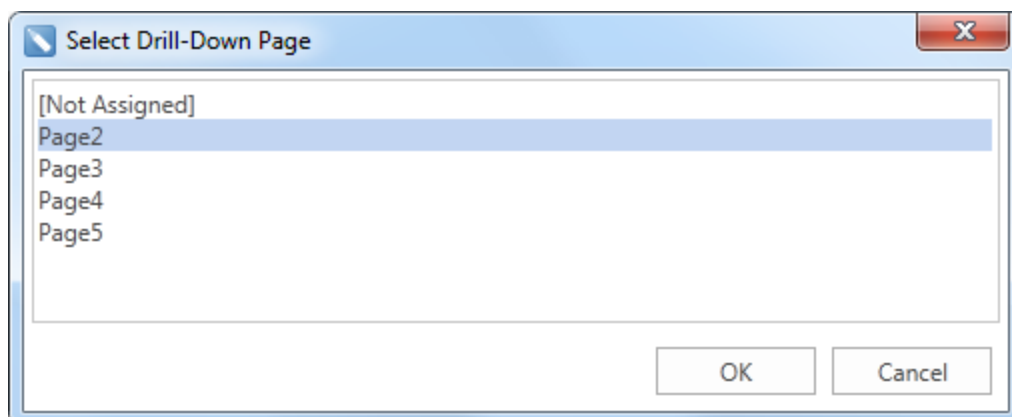
Also, specify the **Drill-Down Parameters**, if necessary. In each setting you should change the following properties: **Name** and **Expression**. In this case, define a detailed parameter with the name **ShipperID** and the expression **Shippers.ShipperID**. Set data filtering in the **Data Band**, which will contain detailed data, . To do this, add a filter and specify a filtering expression: **(int)this["ShipperID"] == Orders.ShipVia**. After that, you should render a report. The picture below shows a rendered page of the report:
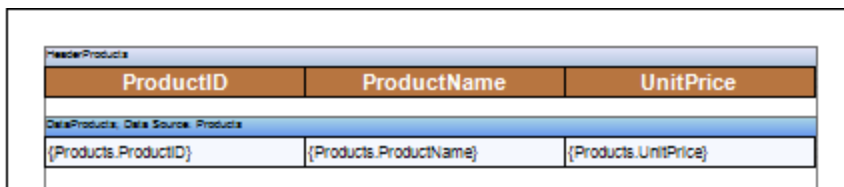
As can be seen from the picture above the page with the main data is rendered. To display detailed information, you should click the rendered text component. Then, the report generator, considering the **Drill-Down Options** and filtering data on the **Data Band**, renders the second page of the report template. The picture below shows a schematic detailing of the report:
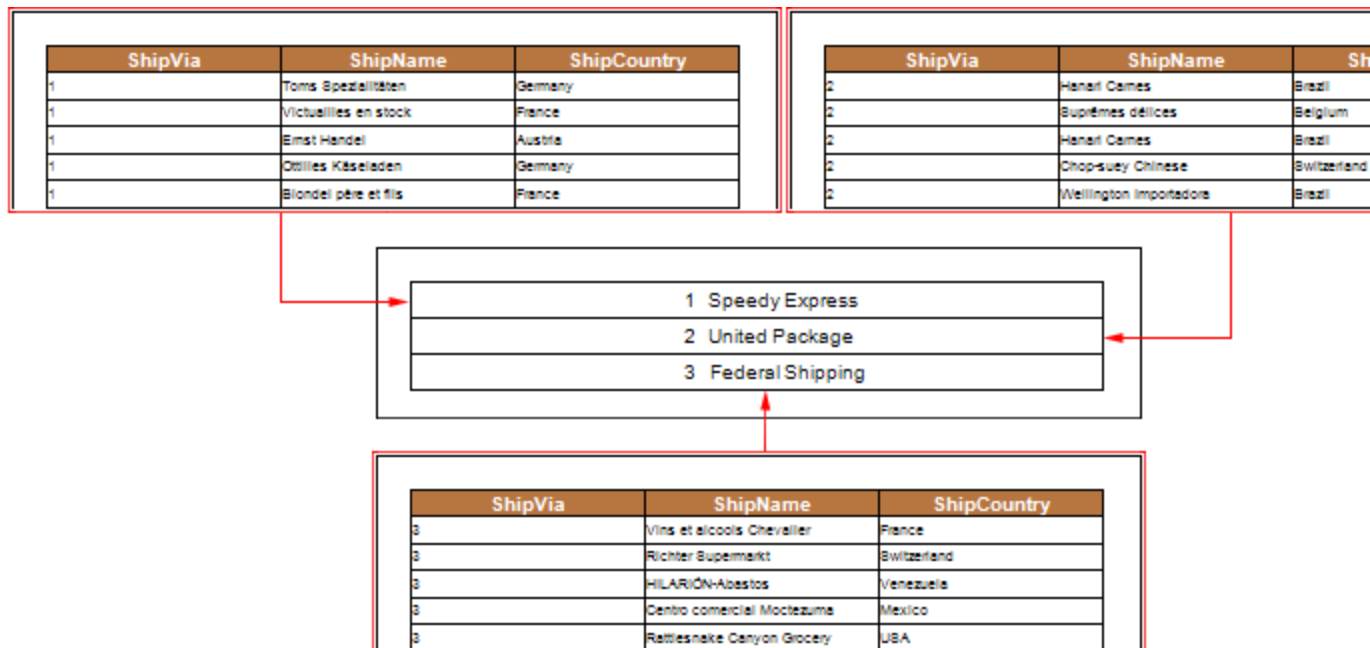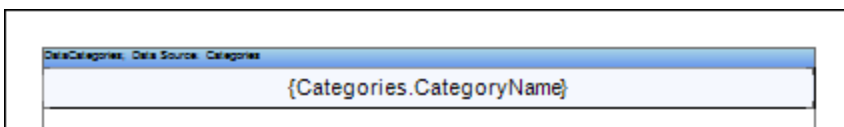
3.36.3.2 Drill-Down Reports Using External Report

The drill-down report with another (external) report means an interactive report in which the main and detailed data are located in different reports. It is possible to create such a report using the **Interaction.Drill-Down Report** property. Consider the example of a Drill-Down Report using an external report. First, create a report with detailed data. This report will contain a list of products and their prices. Put the **Data Band** in the page of the report template with text components which contain expressions: **Products.ProductID**, **Products.ProductName** and **Products.UnitPrice**. For this band, you should select the data source **Products**. Also add the **Header Band**. The picture below shows a page template with detailed information:

Add a filter with the expression **(int)this["CategoryID"] == Products.CategoryID** in the **Data Band**. After that, you must save the report template. For example save the report to: **D:\\Products.mrt**. Now create a report that will contain the main data in this example, the category names. Put the **Data Band** with a text component in the page template. The text component will contain the expression **Categories.CategoryName**. For this band, you should select the data source **Categories**. The picture below shows a page of the report template with the main data.

Then, select the text component and change the values of some properties. The **Interaction.Drill-Down Enabled** property must be set to **true**. Then, set the value of the **Interaction.Drill-Down Report** property to the full path to the report with detailed data.

Also, specify the **Drill-Down Parameters**. In each parameter you must change the following properties: **Name** and **Expression**. In this case, define a detailed parameter with the name **CategoryID** and the expression **Categories.CategoryID**. Then render a

report. The picture below shows a page of the rendered report:

As can be seen from the picture above template page will be rendered with the main data. To display the detailed data, click the rendered text component. The report generator will run the report and render it, considering the parameters of the detailing and filtering. The picture below shows schematically the report:
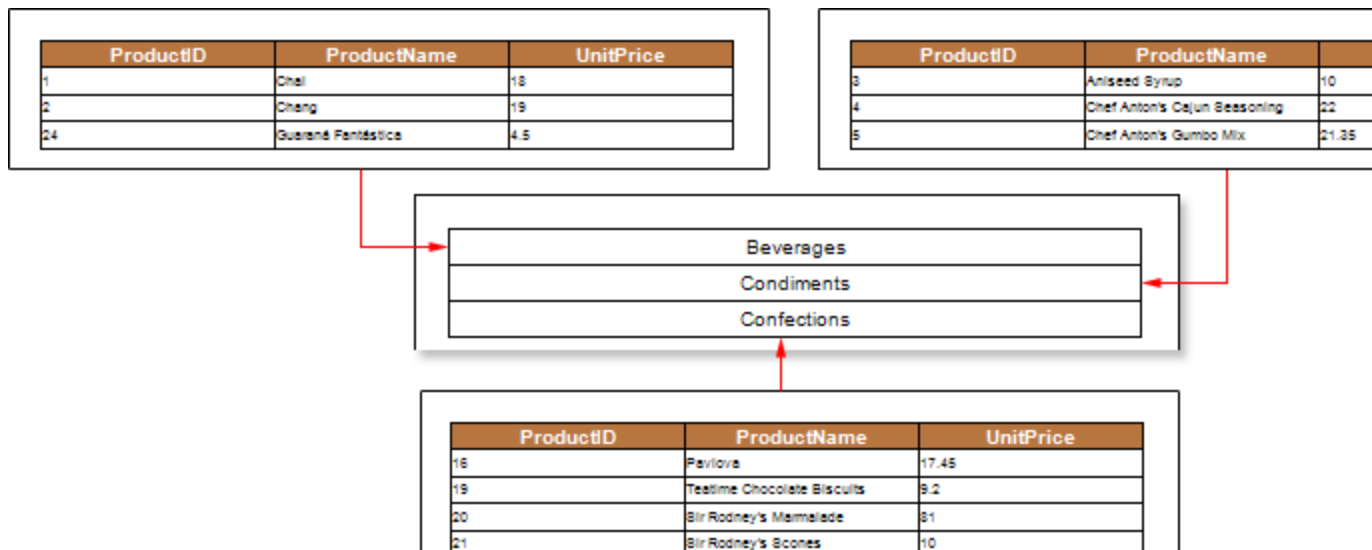
3.36.3.3 Interactive Selection

One of the drill-down types is the interactive selection. The Interactive Selection can be used to produce data detailing on the same page, on which the main data are placed. Creating a report with the interactive selection is possible using the **Interaction.Selection Enabled** property. Only a **Data Band** has this property. Consider the example of a report using the interactive selection. Open a report with the list of categories and products related to these categories. The picture shows a report template:

Select the **Data Band** in order to enable interactive selection. In this case, the band that contains the names of categories (the band which has a text component with the expression **Categories.CategoryName**) will be selected. Set the **Interaction.Selection Enabled** property of this selected band to **true**. After that, add a filter to the detailed band, if necessary. In this example, the filter will be added to the Data Band that contains information about products. Set a filtering expression, in this case it is **DataCategories.SelectedLine == Products.CategoryID**. Then, render a report. The picture below shows a page of the rendered report with interactive selection:

As can be seen from the picture above, the category **Beverages** was selected. This category has been detailed and displayed showing products in this category. Also, in this picture you can the category **Dairy Products** highlighted when the cursor is hovered. In addition, it should be noted that in the interactive selection the multi-level nesting may also be present.

3.36.3.4 Drill-Down Parameters

When you create an interactive report using **Drill-Down** relations, there is a possibility in the report generator to specify the parameters to be passed from the main report to the detailed one. For example, you can pass a parameter to be used for filtering data in a detailed report. Also, you can initialize properties (**Report Alias**, **Report Title**, **Report Description**) of the detailing a report by specifying them in the parameters of the detailed report. Suppose there is an interactive report that contains the category names and details of products related to these categories. Let's make each detailed tab has the category name by which it is open. To do this, change the values of properties for the group **Drill-Down Parameter**:

1 Specify the name of the parameter in the field of the **Name** property. In order to initialize a report property, you must specify its name in the name of the detailed parameter. In this case, you must specify the **ReportAlias**.
2 In the field of the **Expression** property specify an expression that is evaluated each time you pass a parameter to the report. In this case, you must specify the expression **Categories.CategoryName**.

Now, in the rendered report, a tab with the detailed data will have the category name, which has been interpreted. The picture below shows a report that was built with the tabs of detail:

As can be seen from the picture above, the categories **Beverages**, **Confections**, **Grains/Cereals**, **Produce** were detailed. And the tab, which is located on the detail of these categories have names of categories, respectively.
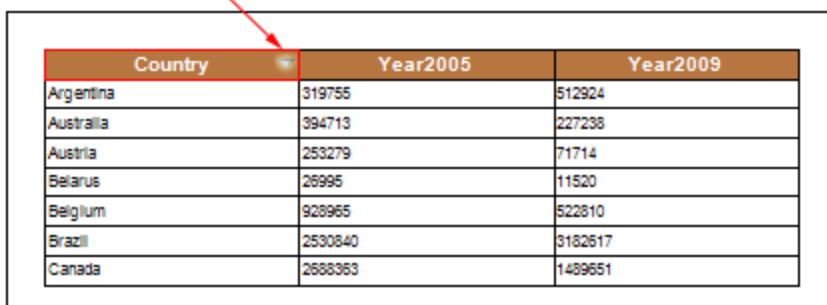
Detailed description of using parameters can be found at Drill-Down Report Using Page in Report and Drill-Down Report Using External Report.

3.36.4 Dynamic Sorting

In **Stimulsoft Reports** it is possible to use dynamic sorting. Dynamic sorting provides the ability to change the sorting direction in the report. Sorting the data can be
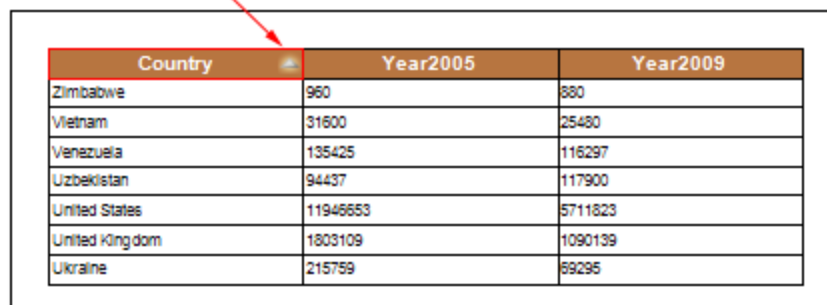
performed both on a single data column as well as in several ones. Set the **Interaction.Sorting Enabled** property of the component, by clicking on which the dynamic sorting by one column will be enabled, to **true** and change the value of the **Interaction.Sorting Column** property. The value of this property is the data column, by which dynamic sorting will be done. It should be noted you can specify only one data column for one component. Then, select the component to which dynamic sorting was set. Dynamic sorting is carried out in the following directions: **Ascending** and **Descending**. Each time you click the component, the direction is reversed. The picture below shows a report page with dynamic sorting:

Click this component
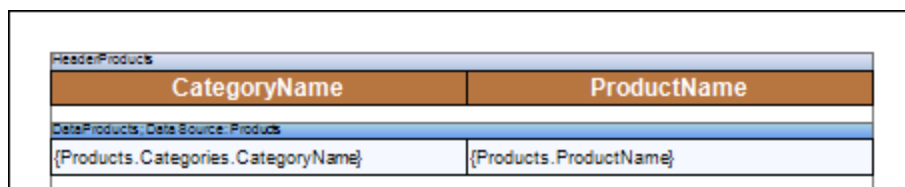
Country	Year2005	Year2009
Argentina	319755	512924
Australia	394713	227238
Austria	253279	71714
Belarus	26995	11520
Belgium	928965	522810
Brazil	2530840	3182617
Canada	2688363	1489651

Click this component

Country	Year2005	Year2009
Zimbabwe	960	880
Vietnam	31600	25480
Venezuela	135425	116297
Uzbekistan	94437	117900
United States	11946653	5711823
United Kingdom	1803109	1090139
Ukraine	215759	69295

If you need to sort by multiple columns simultaneously, it can be done by pressing the Control button. Consider the following example. Suppose there is a report that contains the names of categories and a list of products. The picture below shows the report template:

HeaderProducts

CategoryName	ProductName

DataProducts: Data Source: Products

{Products.Categories.CategoryName}	{Products.ProductName}

When rendering the report without sorting, data are taken from the data source sequentially. In order to enable dynamic sorting you need to select the component

when clicking it the sort direction will be changed. In this example, select text components in the **Header Band**. Then set the **Interaction.Sorting Enabled** properties for both components to **true**. In the fields of the **Interaction.Sorting Column** properties specify the data column to be used for sorting data. In this case, specify the column **{Products.Categories.CategoryName}** for the text component with the expression **CategoryName**, and for the text component with the expression **ProductName** specify the column **{Products.ProductName}**. Render a report. In order to sort data by multiple columns, you must click the components holding the **Control** button and change the sorting direction. The picture below shows a report page rendered with dynamic sorting by multiple columns:

CategoryName ▼1	ProductName ▲2
Beverages	Steeleye Stout
Beverages	Sasquatch Ale
Beverages	Rhönbräu Klosterbier
Beverages	Outback Lager
Beverages	Laughing Lumberjack Lager
Beverages	Lakkalikööri
Beverages	Ipoh Coffee
Beverages	Guaraná Fantástica
Beverages	Côte de Blaye
Beverages	Chartreuse verte
Beverages	Chang
Beverages	Chai
Condiments	Vegie-spread
Condiments	Sirop d'érable
Condiments	Original Frankfurter grüne Soße
Condiments	Northwoods Cranberry Sauce

As can be seen from the picture above, when sorting by multiple columns, the data are sorted first by the first column. In this case, the categories are sorted in the **Ascending** direction. Then, data are sorted by the second column. In this case, the products are sorted in the **Descending** direction, but within each category. In other words, in the products category **Beverages** is ordered in the direction from **Z** to **A**, in the category **Condiments**, too, from **Z** to **A**, etc. To disable sorting by multiple columns, you must release the **Control** key and click the component with dynamic sorting.

3.36.5 Dynamic Collapsing

Sometimes you need to show a report in a compact form. In Stimulsoft Reports you can find the ability to dynamically collapse information in the preview window. A report with dynamic collapsing is an interactive report, in which collapsing blocks can expand/ collapse its contents clicking the block title. Dynamic collapsing is usually used in reports with grouping, Master-Detail, hierarchical reports. Dynamic collapsing can be multilevel. Consider an example of using dynamic collapsing in the report. Let's have a report that contains a list of products that are grouped by category. The picture below schematically showed the report with a multilevel collapsing:
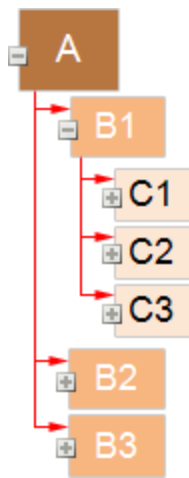
As can be seen from the picture, the collapsing unit **A** contains a collapsible blocks **B1**, **B2**, **B3**. This is dynamic collapsing of the first level. In turn, the block **B1** contains a collapsible blocks **C1**, **C2**, **C3**. This is dynamic collapsing of the second level, etc. Consider the example of a dynamic collapsing of the report with the group. Let's have a report that contains a list of products that are grouped by category. Below is a picture with a report with grouping:

Enable dynamic collapsing, where the title of the collapsing unit will be group titles, i.e. in this case, the category names. To do this, return to the report template (see the picture).

Select the component that will be a title of the collapsing block, i.e. in this example, the **Group Header** band. Then, set the **Interaction.Collapsed Enabled** property to **true**. In the field of the **Interaction.Collapsed** property specify an expression **{GroupLine! = 1}**. Render a report. The picture below shows a report page rendered with dynamic collapsing:

Now, when rendering a report, the group will have a look as expanding/collapsing blocks. To expand/collapse the block, you should click the title block. In this case, the group header. On the component for which the dynamic collapsing is enabled, is displayed if the block is collapsed the icon ⊞ is displayed and the icon ⊟ is displayed if the block is expanded. Note that you can collapse blocks with the the group footer. To do this, set the **Interaction.Collapse Group Footer** property to **true**.

3.36.6 Reports with Contents

Sometimes 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.
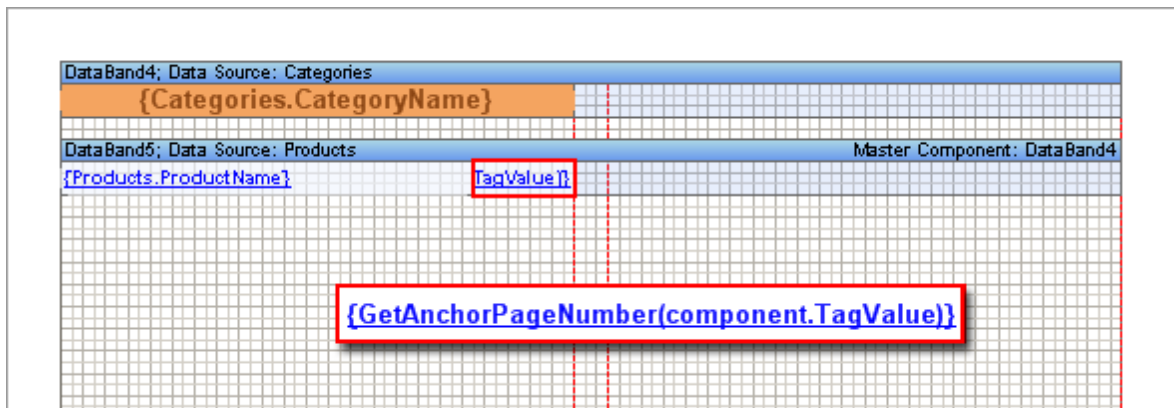
As an anchor name the value of the **Tag** property is used. For filling the **Tag** property the following expression is used:

{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.

3.36.7 Editing

In our reporting we have the ability to edit some of the components of a rendered report in the viewer, or in the preview tab. As a rule, it must be made before printing or exporting. The components that can be changed are:
❯ Text;
❯ Text in Cells;
❯ Rich text;
❯ Checkbox.

In order to make it possible to edit the report components, you should set the Editable property of these components to Yes. Then, you can modify these components in the viewer using the tool Editor. In text components editing means changing the text, and in the checkbox editing means changing the value (true or false).

For PDF and Word documents:
By default, when you export a PDF document you can edit it. But it is possible to include the mode in which after exporting editing will be available only for the report

components with the Editable property enabled. If No is set, then you can edit all components, unless it is not limited with safety parameters. If you select Yes then you can only edit components with the Editable property enabled. The Word document can also be editable. However, with the parameter Restrict Editing it is possible to allow editing only the components that have the Editable property set to Yes. For this set Restrict Editing to Except Editable Fields.

4 Dashboards

The **Dashboard** component is a scalable area on which you can place elements of data analysis. All elements placed on the dashboard can be related to each other or split into groups of related elements. The dashboard panel is created in the report designer, and viewed in the preview panel in the report designer or in the report viewer.
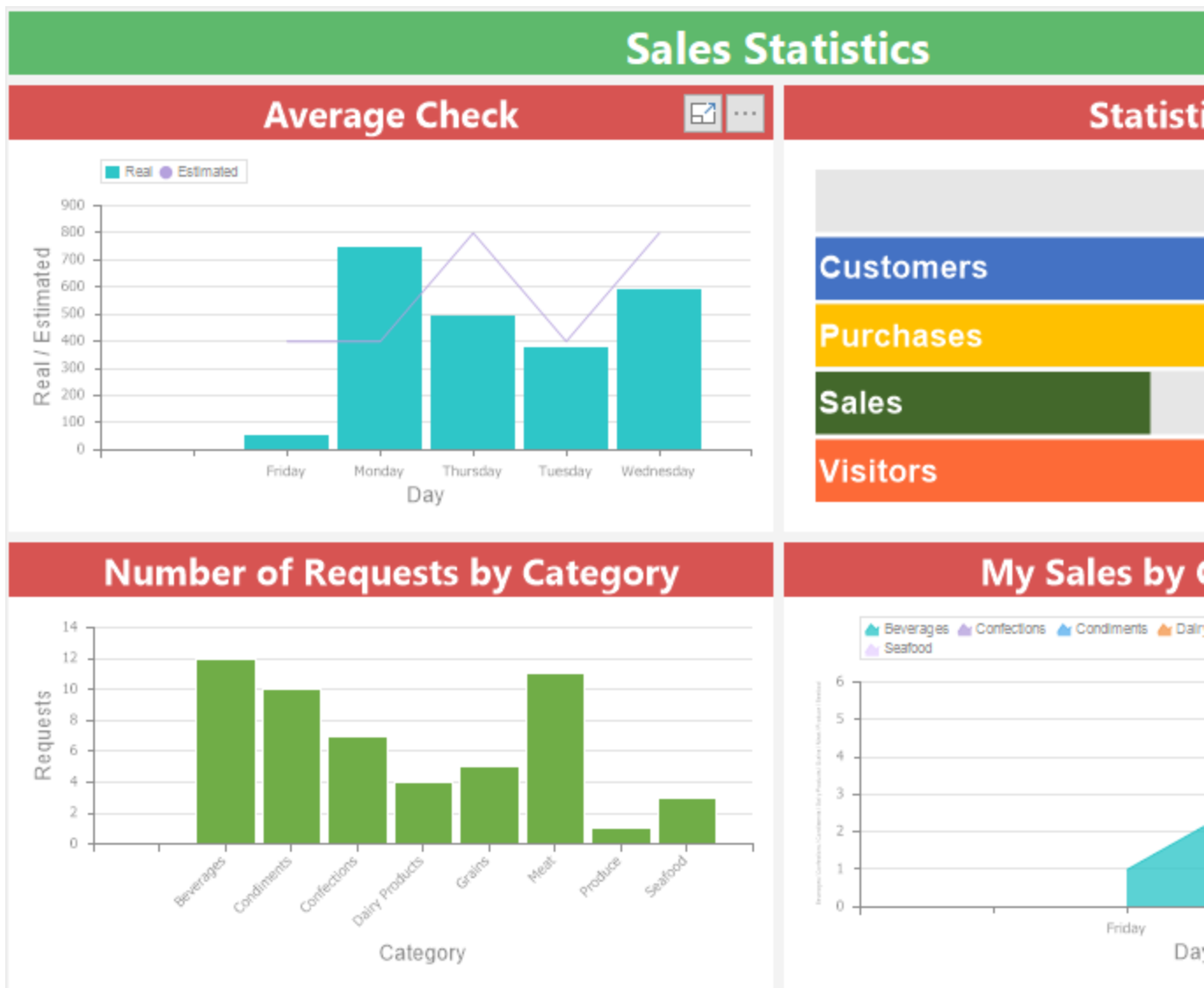
Creating dashboards

Please do the following in the report designer to add a new dashboard in the report:

❯ Click the **Dashboard** button on the **Insert** tab;
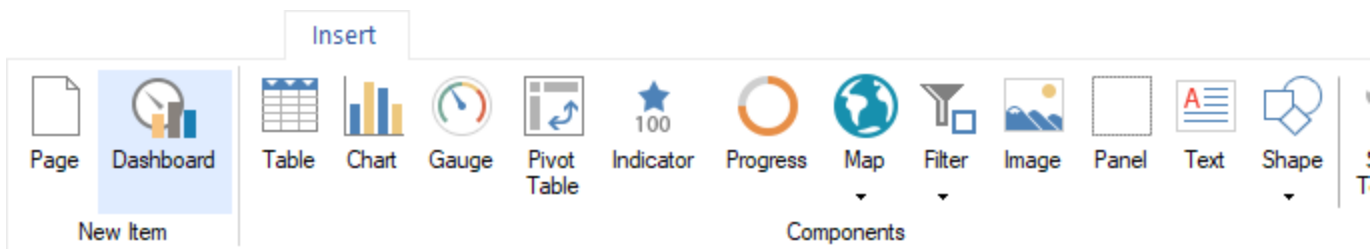
❯ Select **New Dashboard** in the context menu of the page title or dashboard panel.
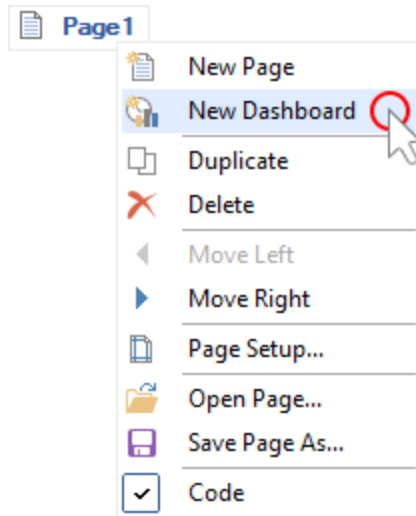
The size of a dashboard

When rendering a report, the dashboard is stretched (or squeezed) to the entire area of the report viewer. The elements of the dashboard will also be stretched in width and height, in proportion to the original dimensions in the report designer.

Information

The data filtering elements of the drop-down type (drop-down list, date picker, drop-down hierarchical list) do not stretch in height.

When creating the dashboard in the designer, its size has a working area that looks like a white sheet with a grid. In order to change the size of the dashboard in the report designer, you should:

❯ Double-click on the working area of the dashboard;
❯ Specify the width and height of the dashboard in pixels.
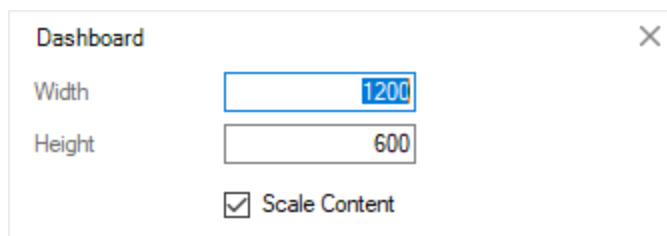
Information

When you change the size of the working area of the dashboard in the report designer, the elements can stretch (shrink) or keep their size unchanged. It depends on the **Scale Content** parameter. If this option is enabled, then, when resizing the dashboard, all elements will also be stretched or shrunk. If this parameter is disabled, only the size of the working area of the dashboard will be increased, while the size of the elements will remain unchanged.

All elements of the dashboard are grouped into the following ones according to their functionality:

Elements of data analysis
> Table;
> Chart;
> Pivot;
> Indicator;
> Progress;
> Maps;

Elements of data filtering
> List;
> Combo box;
> Tree View;
> Tree View Box;
> Date Picker.

Other dashboard elements
> Panel;
> Text;
> Image;
> Shapes.

Adding items to the dashboard
Do the following to add an element to the dashboard:
> Drag items from the **Toolbox** or the **Insert** tab to the working area of the dashboard;
> Select items on the **Toolbox** or the **Insert** tab, and left-click in the dashboard panel.

Changing the item type

You can change the type of the element without redesigning it. To do this:

❯ Select an element that needs to be changed on the dashboard;

❯ Left-click on the **Change Type** button;

❯ In the menu that opens, select the element which you want to convert the current one to.
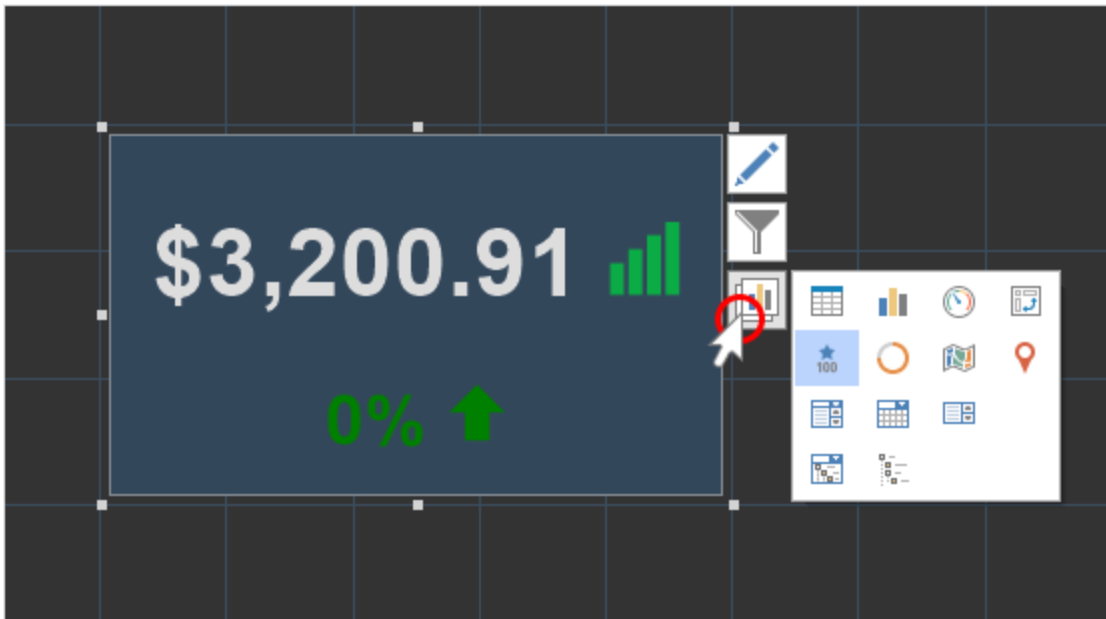
4.1   Appearance

In addition to creating dashboards, an important role plays the setting of this panel and its elements. In this chapter, we will describe the parameters of the dashboard and its elements.
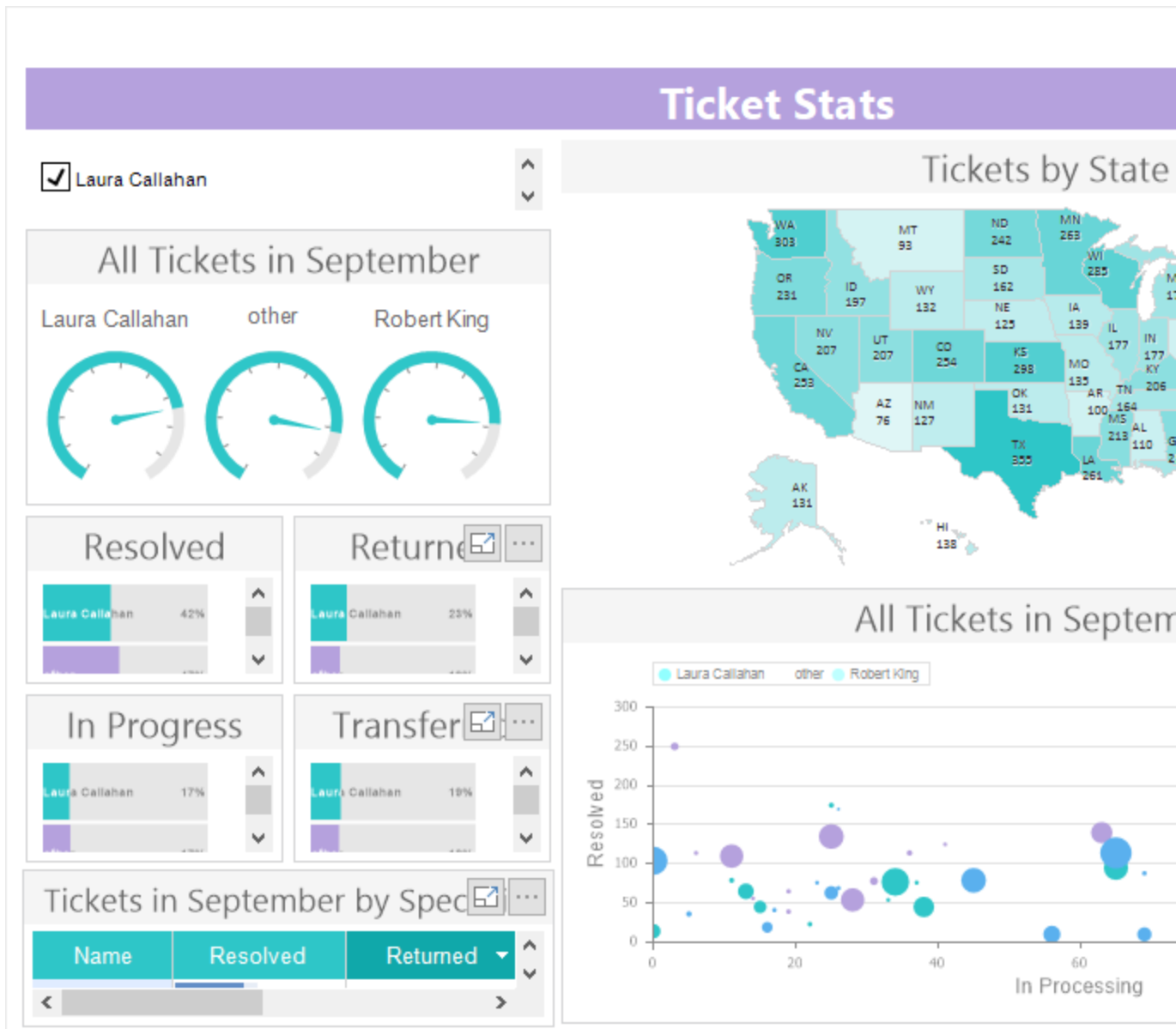
Information

Some elements, besides those listed below, may also have individual design options.

Styles of a dashboard and elements

When creating a dashboard, the report designer contains predefined styles. The first style from the list is applied to the dashboard. For all newly added elements on this dashboard, the current color scheme of the dashboard is used. By default, when you

change the style of the dashboard, the newly selected color scheme will be applied to all elements on this panel. However, for each component of the dashboard you can assign your style.

In order to change the style of the dashboard, you should:
> Left-click on the empty area of the dashboard;
> Select the dashboard style on the **Home** tab, in the styles menu.
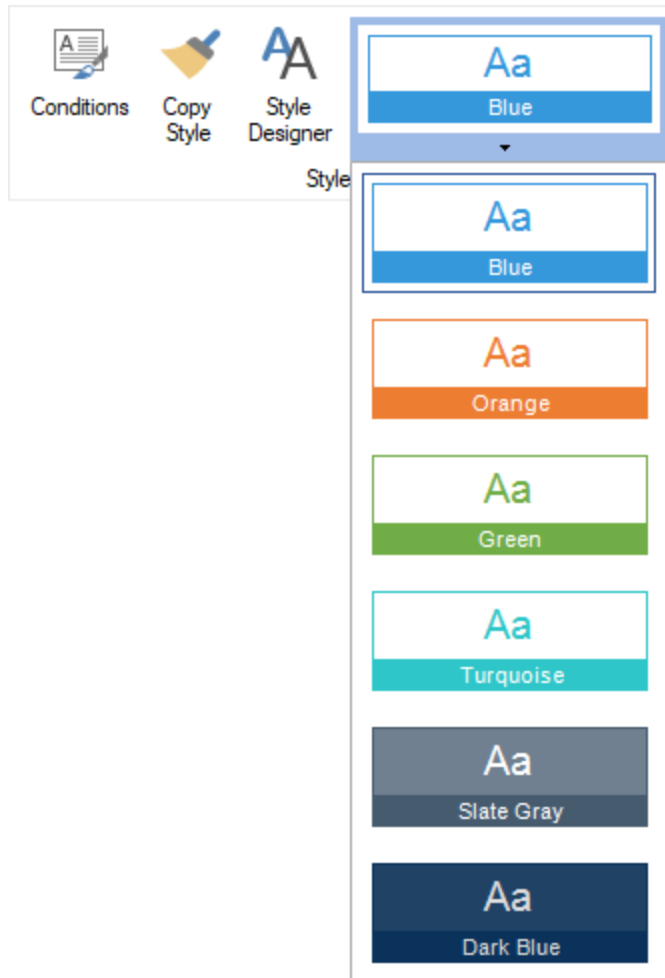
In order to change the style of an element in the dashboard, you should:
> Select an dashboard element;
> Select the desired element style on the **Home** tab, in the style menu.

Information

In this case, if you change the style of the dashboard, the color scheme of the element will not change.

In addition, you can create custom styles for the elements of the dashboard. To do this, call the Style Designer and create styles for the elements. You can also assign the created style using the style menu on the Home tab or using the Style property of the element.

Background

One of the settings for the design is to set the background color of the element. By default, the background color is used from the assigned style. In order to change the background color of the dashboard or its elements you should:

❯ Select the dashboard or element;

❯ Change the value of the **Back Color** property in the property panel.

❯ After that, select the background color from the drop-down list.

Also, you can change the background color of the element on the Home tab in the report designer:

❯ Select the dashboard or element;

❯ Use the **Background** tool to select a background color from the palette or specify a custom color.
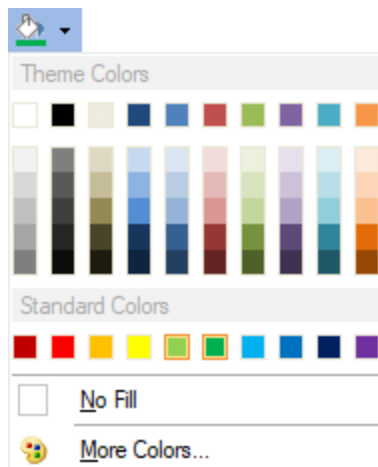
Text color

When customizing the design, you can change the text color of a specific item. To do this:

❯ Select an element;

❯ Select the required color from the drop-down list in the **Fore Color** property.

Information

The [Table](#) element also has its own color for each column. To do this, select the data field in the **Table** element editor and change the text color.

Margin and padding

Each element in the dashboard can define the margin and padding of the element. To do this:

❯ Select an item on the dashboard;

❯ Change the values of the **Margins** and **Padding** property groups on the property panel.

You can also customize the type, borders, size and color of the borders of the element. To do this:

❯ Select an item on the dashboard;

❯ Change the type, size, sides, borders color using the **Border** property group on the property panel, or tools on the **Home** tab in the report designer.
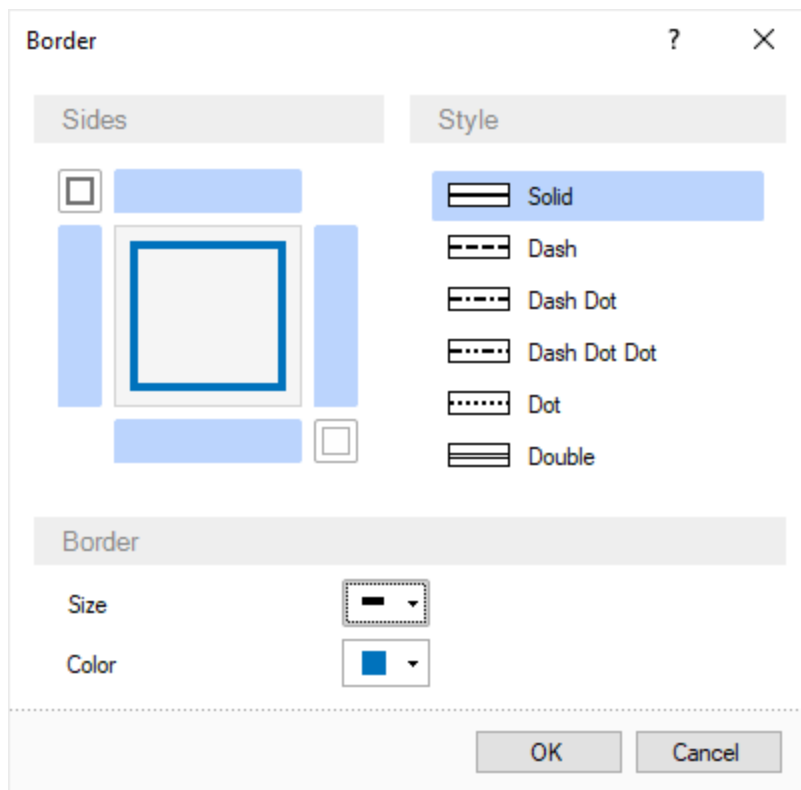
Element titles

The titles of elements on the dashboard can be created in various ways. For example, using the **Text** element. However, elements also have the ability to enable and configure an element title. In order to include the title:

❯ Move the cursor to the top of the element;

❯ In the upper right corner, check the box to enable the title display or uncheck the box to disable the title display. By default, the title of the elements is enabled.
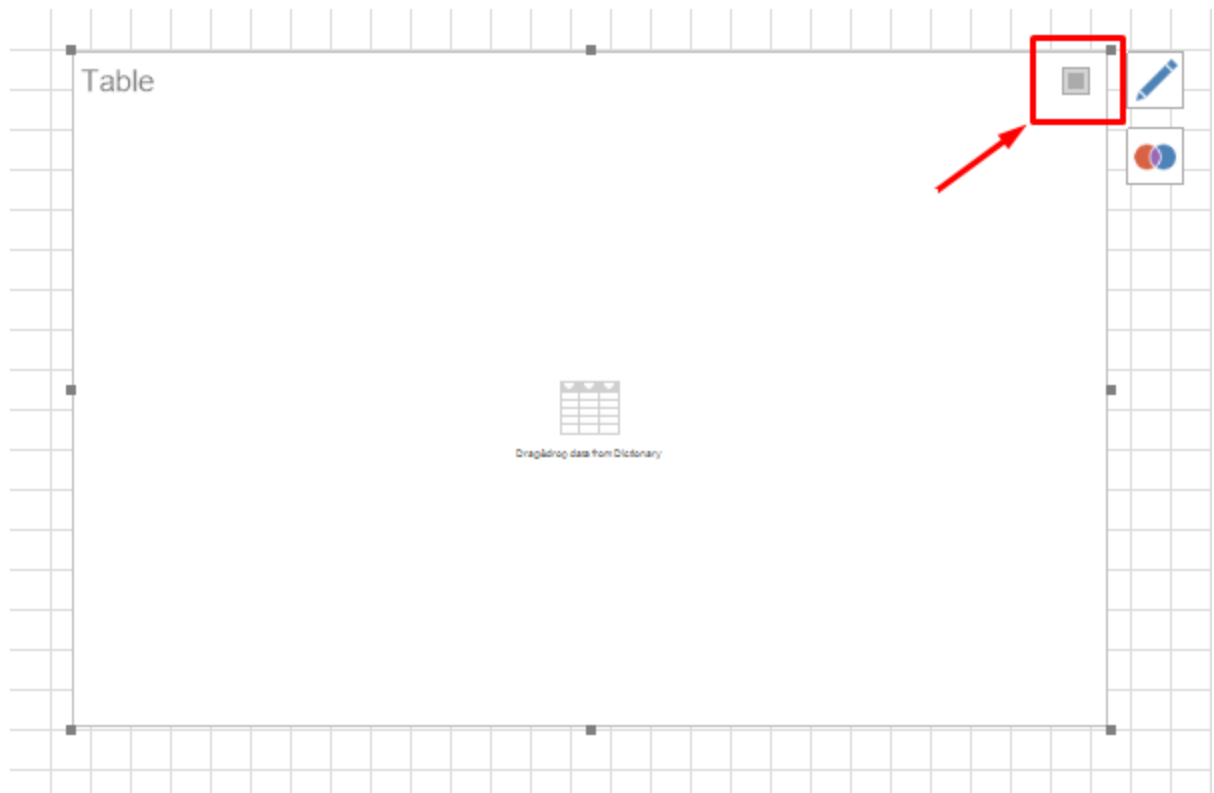
❯ You can also enable or disable displaying of the title by setting the **Visible** property from the **Title** group on the property panel to **true** or **false**.

In order to change the title text you should do the following:
❯ Double-click the input pointer on the header area on the item.
❯ Enter the title text.
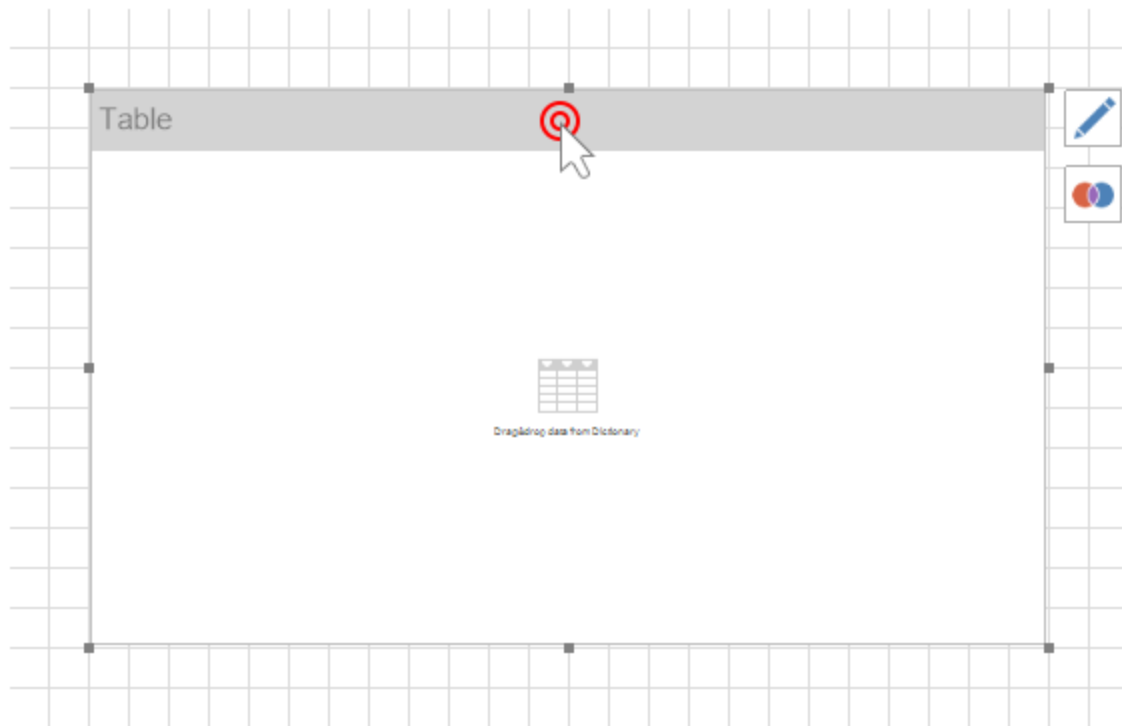
You can also change the title on the property panel:
> Select an item;
> In the **Title** property group, change the **Text** property value.

In addition, text of the title can also be changed:
> Align the title horizontally;
> Header background color;
> The color of the text and its font.

Text formatting
You can apply formatting to the elements of the dashboard. You should do the following steps:
> Select an item on the dashboard;
> Using the **Text Format** tool on the **Home** tab of the ribbon panel, apply the format to the element values.
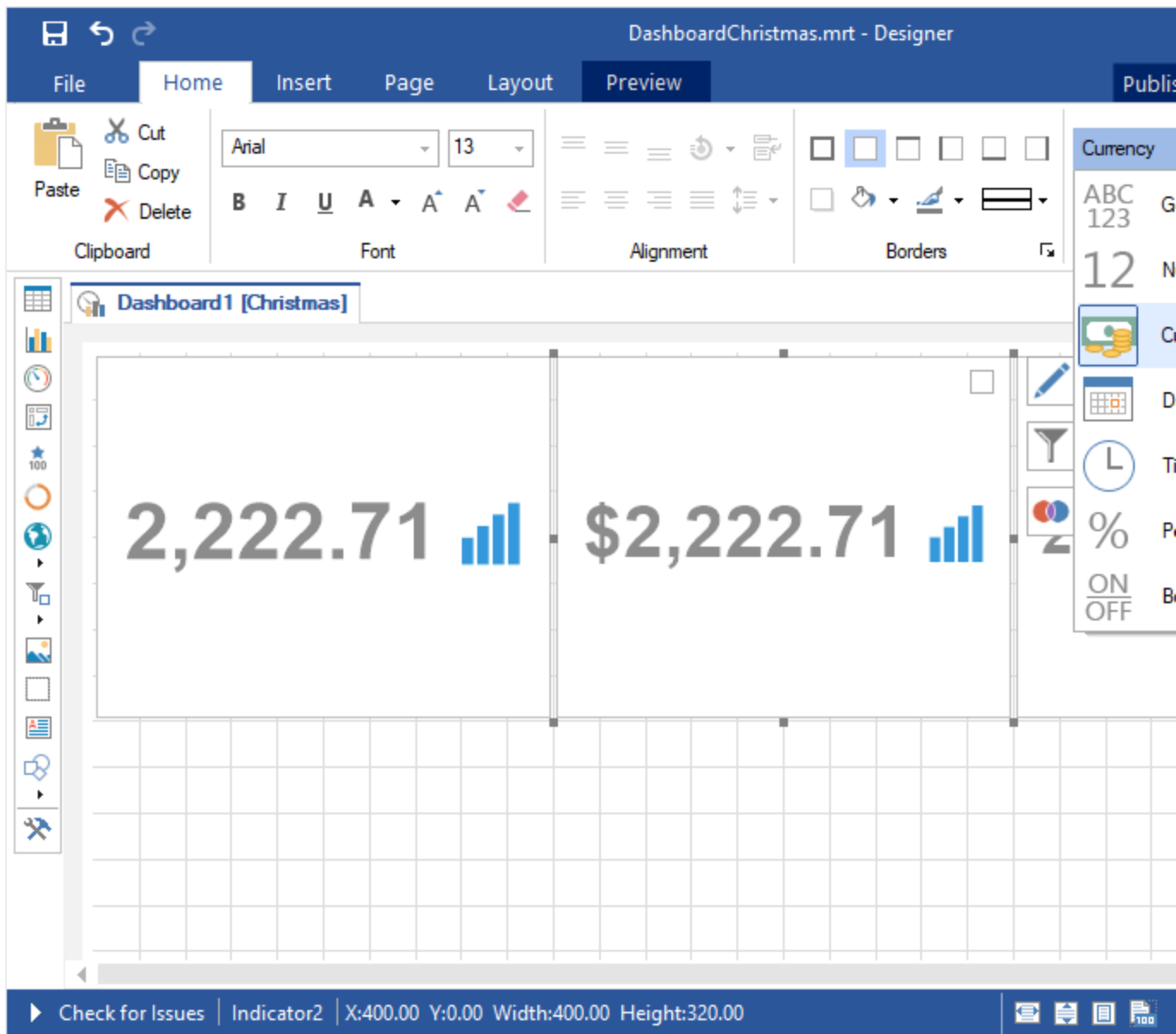
Also, you should remember that for the Table and Pivot Table elements you can set the formatting for the values of each data field. You should do the following steps:
➤ In the Table or Pivot Table element editor, select the data field;
➤ Select a format using the **Text Format** tool on the **Home** tab of the ribbon panel.

4.2 Data

Elements of data analysis can work with different data sources. Before starting the design of the dashboard, you should read the following chapters:
➤ Data Sources;

❯ Relations between data sources;
❯ Data transformation.

All data sources of dashboard elements form a virtual data table for the current dashboard. This is necessary for the interaction of the dashboard elements with each other.

> **Information**
>
>
> When designing a dashboard, data from various sources can be used in the elements. In this case, for correct analysis and comparison of data between these sources, relationships should be established. Otherwise, interactive actions with the elements of the dashboard may lead to incorrect data processing and incorrect displaying of the result.

Data fields

There are fields in which data fields are indicated in the editor of the dashboard elements. Each data field has an expression which results of processing are the data values for the current dashboard item. The data field expression can be a reference to a data column or a variable.
❯ If a reference to a data column is specified, the values of the data column will be the values of the data field on the basis of which the current element of the dashboard will be rendered.
❯ If a reference to a variable is specified, the value of the variable will be the value of the current data field. You should know that at this moment we support the variable of the Value is specified.
❯ Also, you can manually specify the values of the data field. To do this, enter a value or a list of values in the **Expression** field of the current data field. To enter values manually, you should use the **List()** or **Array()** functions using the "," separator between values.

A function can be applied to the expression of data fields. In that case, the values of the data field will be the values processed using this function.

You can add a new data field using on of the ways below:
❯ Drag and drop a data column into an item field. In this case, a new data field will be created with reference to the data column you dragged.

❯ Select the **New Field** command from the context menu of the element editor.

In the Table element, the data fields can be of the following types:
❯ **Measure**. By default, this field type applies to all numeric data types. Also, this type of data field is used if it is necessary to group the values of the current data field by the values of another data field.
❯ **Dimension**. This type of field is used by default for non-numeric data types. When grouping data, the values of this data field will be a condition of grouping for the values of other data fields.

Changing the type of data field is carried out in the **Table** element editor, using the **Measure** and **Dimension** buttons:
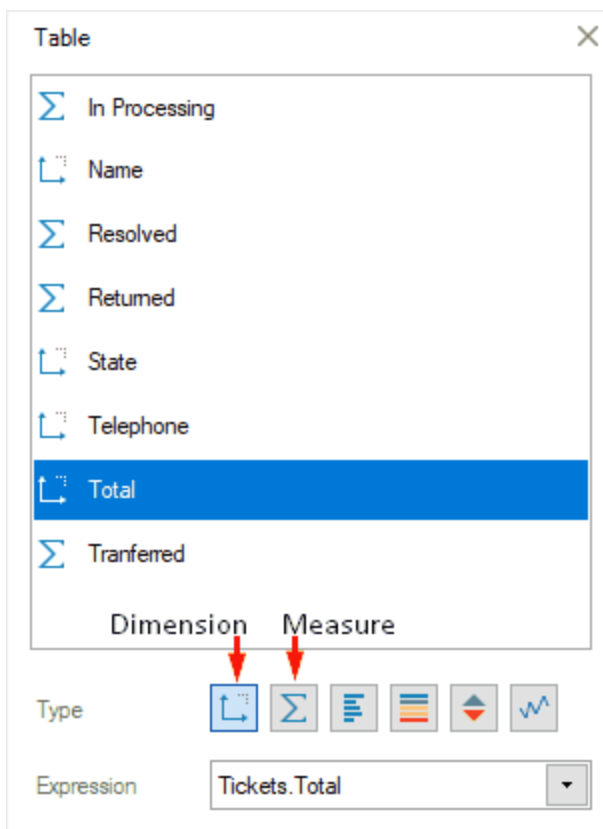
Data field expression
The data fields of the element have the **Expression** field. In this field, you can see the expression of the current data field, and there is also a drop-down menu with a list of commands:
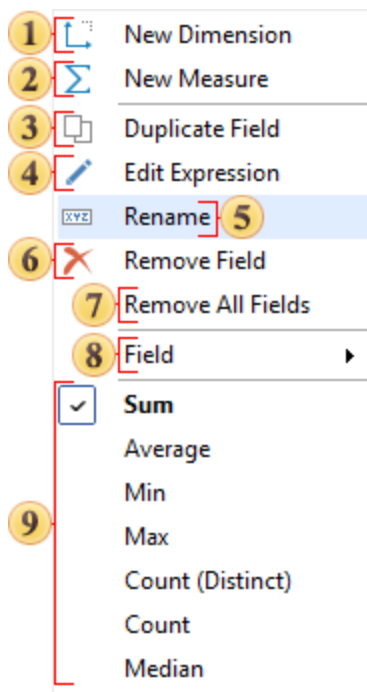
1 The command is used to create a field of the **Dimension** type.

2 The command is used to create a field of the **Measure** type.

3 The command is used to create a duplicate of the current data field.

4 The command calls the expression editor for the current data field.

5 The command is used to change the name of the current field. Also, you can select a field in the list and press the F2 key.

6 The command is used to delete the current field.

7 The command is used to remove all data fields from the current item field.

8 The command contains menus and submenus with a list of data sources from the report dictionary and data columns in these sources. With this command, you can select the data column for the current field.

9 A list of the most frequently used functions that can be applied to the expression of the current field. Depending on the type of data, this list of functions may vary.

Adding data to an element

Drag and drop the data source or columns from the dictionary to the element or its editor. In this case, data fields will be created with references to data columns.

Information

When you drag a data source into the dashboard, a **Table** element with all columns of this data source will be added.

❯ Select the data field in the element editor, using the **Field** command, select a data column. In this case, the expression of the data field will be a reference to the selected data column.
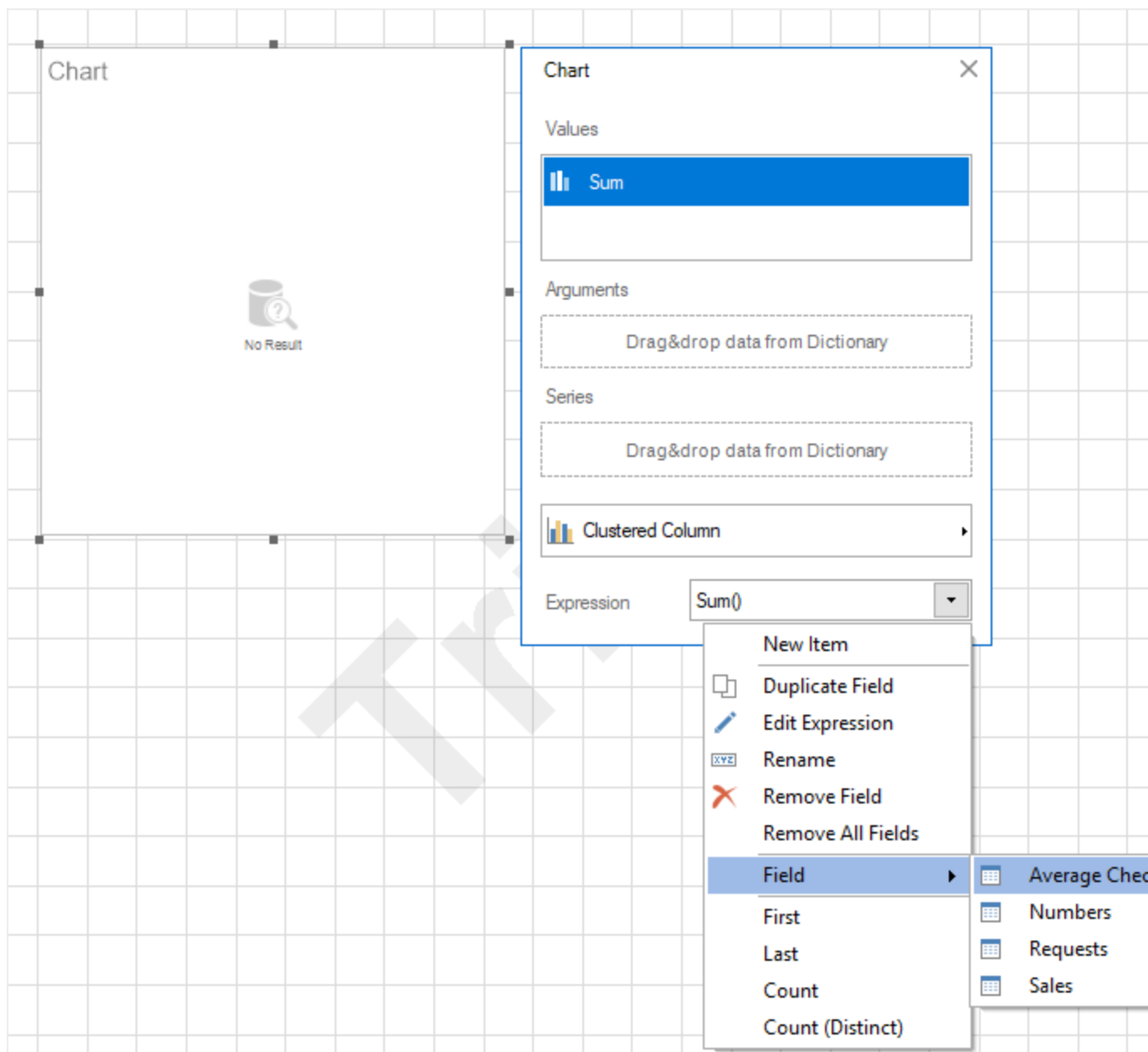
❯ Select the data field, call the Expression editor to create an expression for this field;
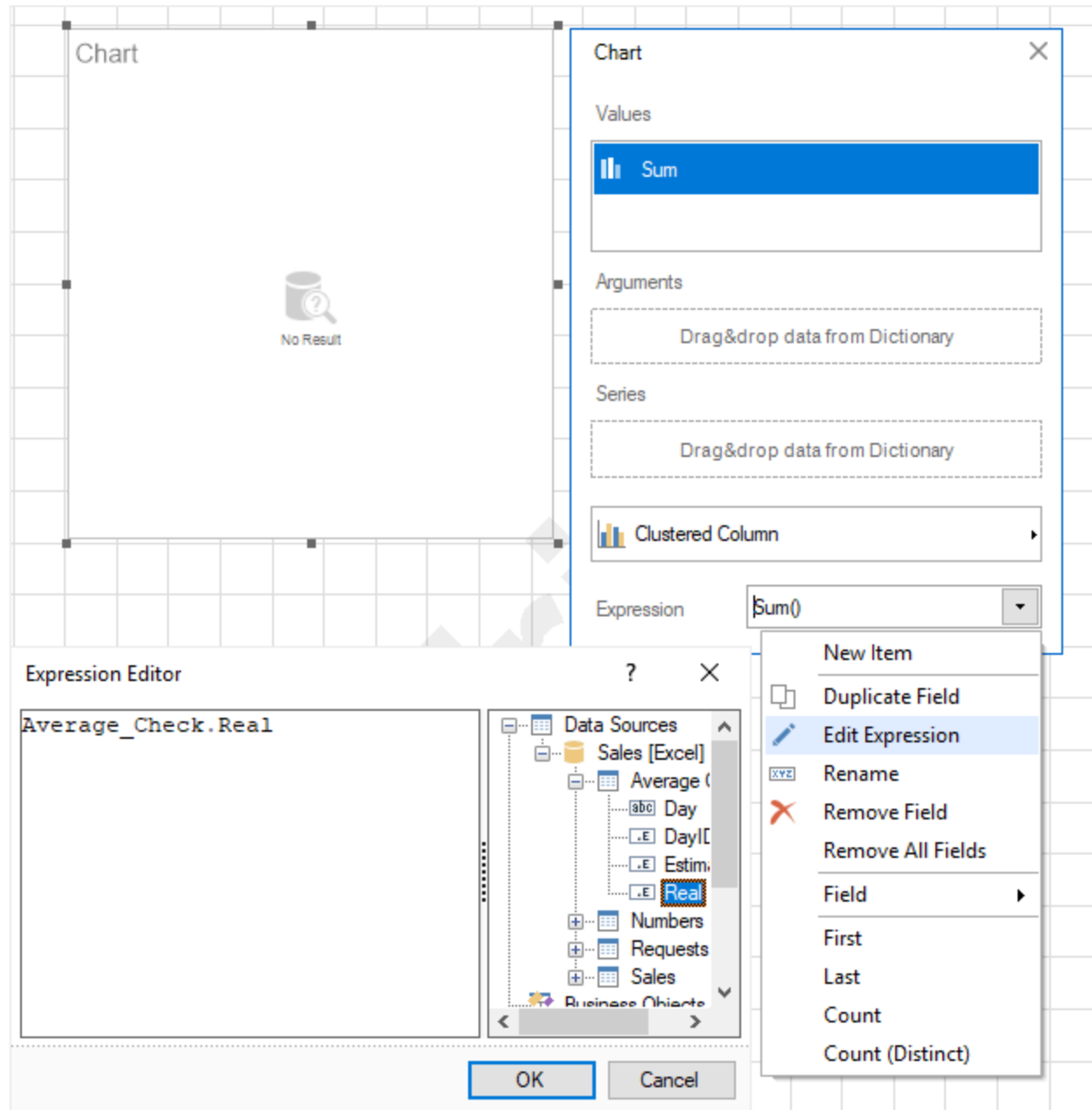
❯ Select the data field and change the expression manually.
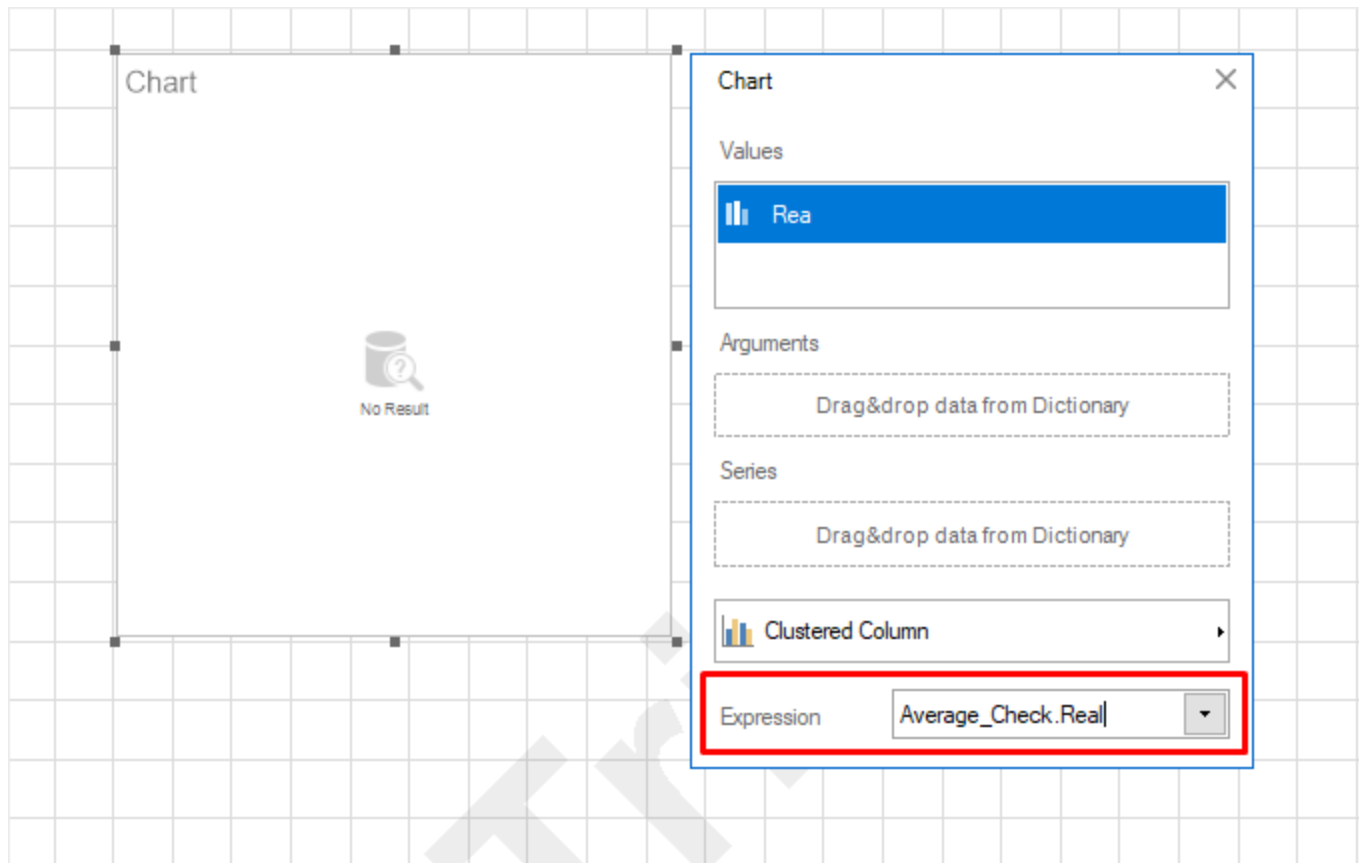
Putting values manually

In the elements of the dashboard, you can enter one value for the current data field or you can specify a list of values. In order to enter a single value, you should:

❯ Call the item editor;

❯ Create a new data field;

❯ In the **Expression** field you should enter the value for the current data field.

To enter a list of values, you should to the following:

➤ Call the item editor;

➤ Create a new data field;

➤ In the **Expression** field you should enter the **List()** or **Array()** function with the list of values with the "," separator.

Removing data from elements

❯ Select the field in a specific field of the element editor, and click the **Remove Field** button (see the picture below).
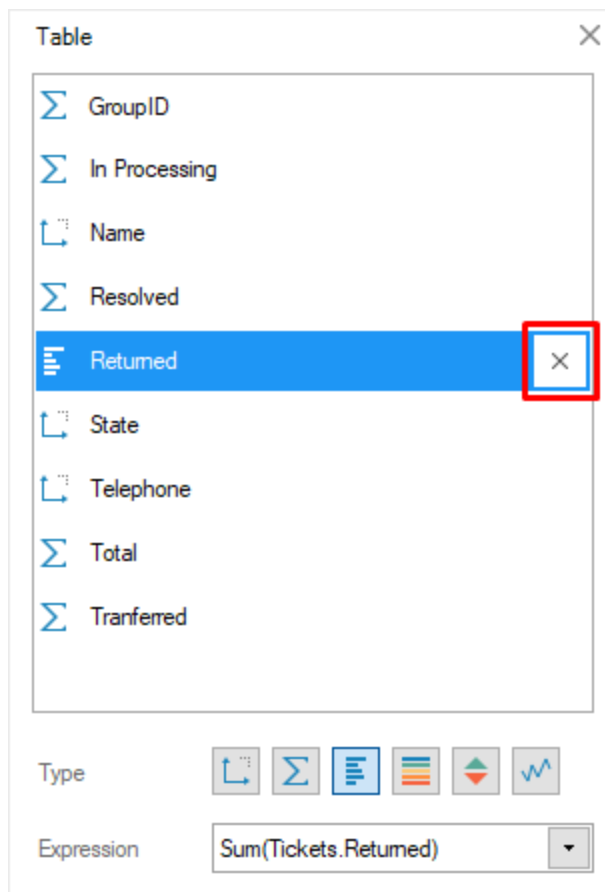
> Select the **Remove Field** command in the context menu of the current data field.
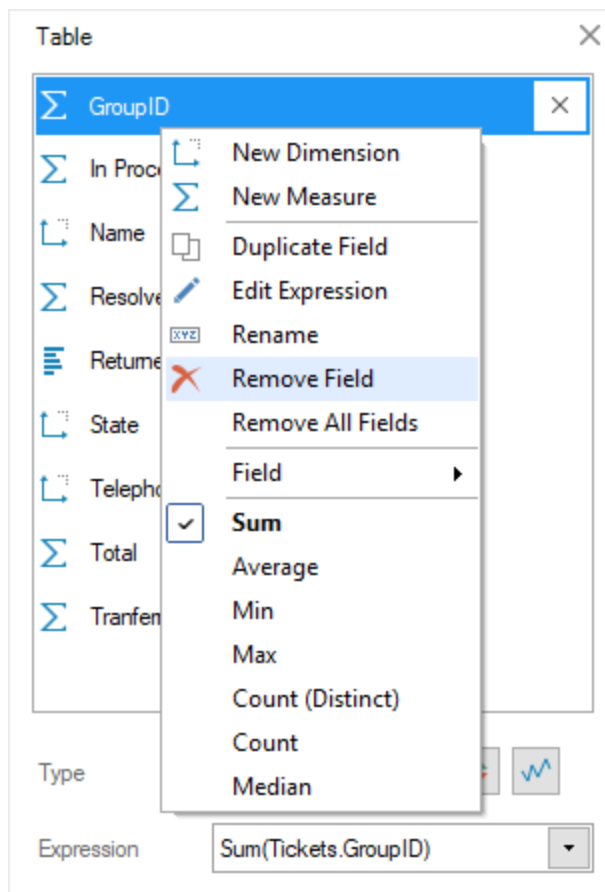
> Select the **Remove All Fields** command in the context menu of the field of the dashboard element, .
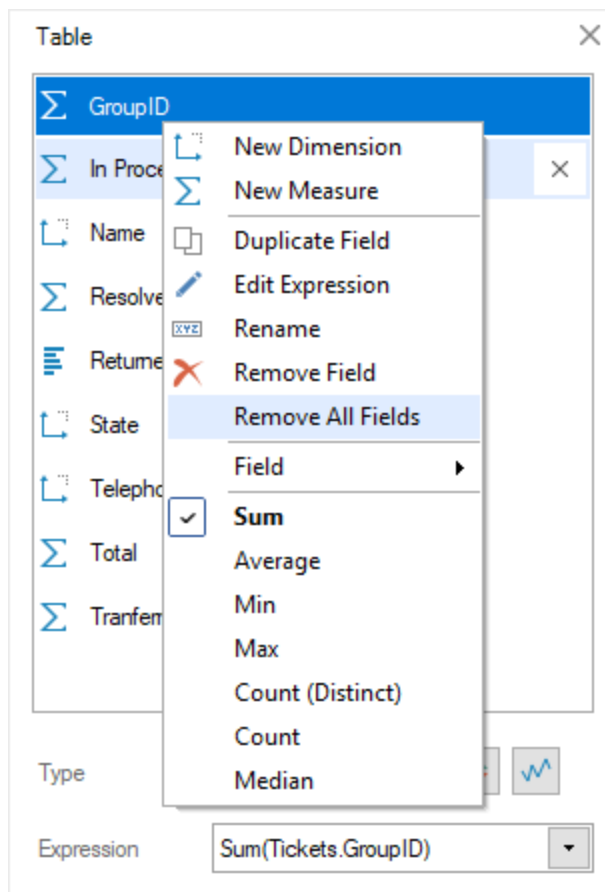
List of functions

Depending on the type of values, the list of functions used may vary. The table below contains a complete list of functions that can be applied to data fields.

Function	Description
Functions that are available from the menu of the Expression field	
Count()	Calculates the number of values in the current data field.
DistinctCount()	Calculates the number of unique values in the current data field.
First()	Shows the first value of the current data field.

Last()	Shows the last value of the current data field.
Sum()	Shows the result of the sum of values in the current data field.
Avg()	Calculates the arithmetic average for the values of the current data field.
Min()	Shows the minimum value from the current data field.
Max()	Shows the maximum value from the current data field.
Median()	Shows the average (non arithmetic) value from the current data field.
Year()	Shows the year from date encoding.
Quarter()	Shows the quarter from the date encoding.
Month()	Shows the month from date encoding.
Day()	Shows the day from date encoding.

Functions that can be added from the data dictionary or entered manually

Mode()	Shows the most frequently repeating values in the current data field.
List()	Enters a list of values for the current data field of an item.

Array()	Enters an array of values for the current data field of an item.
ToUpperCase()	Converts all data field values to uppercase.
ToLowerCase()	Converts all data field values to lowercase.
ToProperCase()	Sets the first character value to uppercase, and the remaining characters to lowercase.
Insert(„)	Inserts text into data field values, after a specific character. Three arguments are specified through the "," delimiter: ❶ Data field; ❷ The ordinal number of the character after which another value will be inserted. ❸ The value to be inserted.
Replace(„)	Replaces certain characters in values. Three arguments are specified through the "," delimiter: ❶ Data field; ❷ A character or combination of characters that needs to be replaced. ❸ The value to be replaced.
Remove(„)	Removes the specified number of characters in the values. Three arguments are specified

	through the "," delimiter: **1** Data field; **2** The ordinal number of the character from what the removal starts. **3** Number of characters to remove.
DayOfWeek()	Shows day of week from date encoding.
DaysInMonth()	Shows the number of days in a month.
DaysInYear()	Shows the number of days per year.
Month()	Shows the number of the month.
ISO2()	Shows the two-letter code of the geographical object.
ISO3()	Shows the three-letter code of the geographical object.
NormalizeName()	Shows the names of the geographical objects by default.
Left(,)	Shows the specified number of characters from the left side of the value. Two arguments are specified through the "," delimiter: **1** Data field; **2** The number of characters to show.
Mid(,,)	Shows characters from a value. Three arguments

	are specified through the "," delimiter: ❶ Data field; ❷ The ordinal number of the character with which to start the display. ❸ The number of characters to show.
Right(,)	Shows the specified number of characters from the right side of the value.. Two arguments are specified through the "," delimiter: ❶ Data field; ❷ The number of characters to show.
Substring(,,)	Shows characters from a value. Three arguments are specified through the "," delimiter: ❶ Data field; ❷ The ordinal number of the character with which to start the display. ❸ The number of characters to show.

4.3 Groups

By default, all elements on the dashboard are related to each other, which means that data filtering of one element affects the data filtering of other elements. However, when designing a dashboard, it is possible to split the elements of the dashboard into groups. For example, you want to display statistics for two unrelated companies in one dashboard. In this case, the elements of the dashboard should be split into groups, where the first group is one company and the second group is another company.
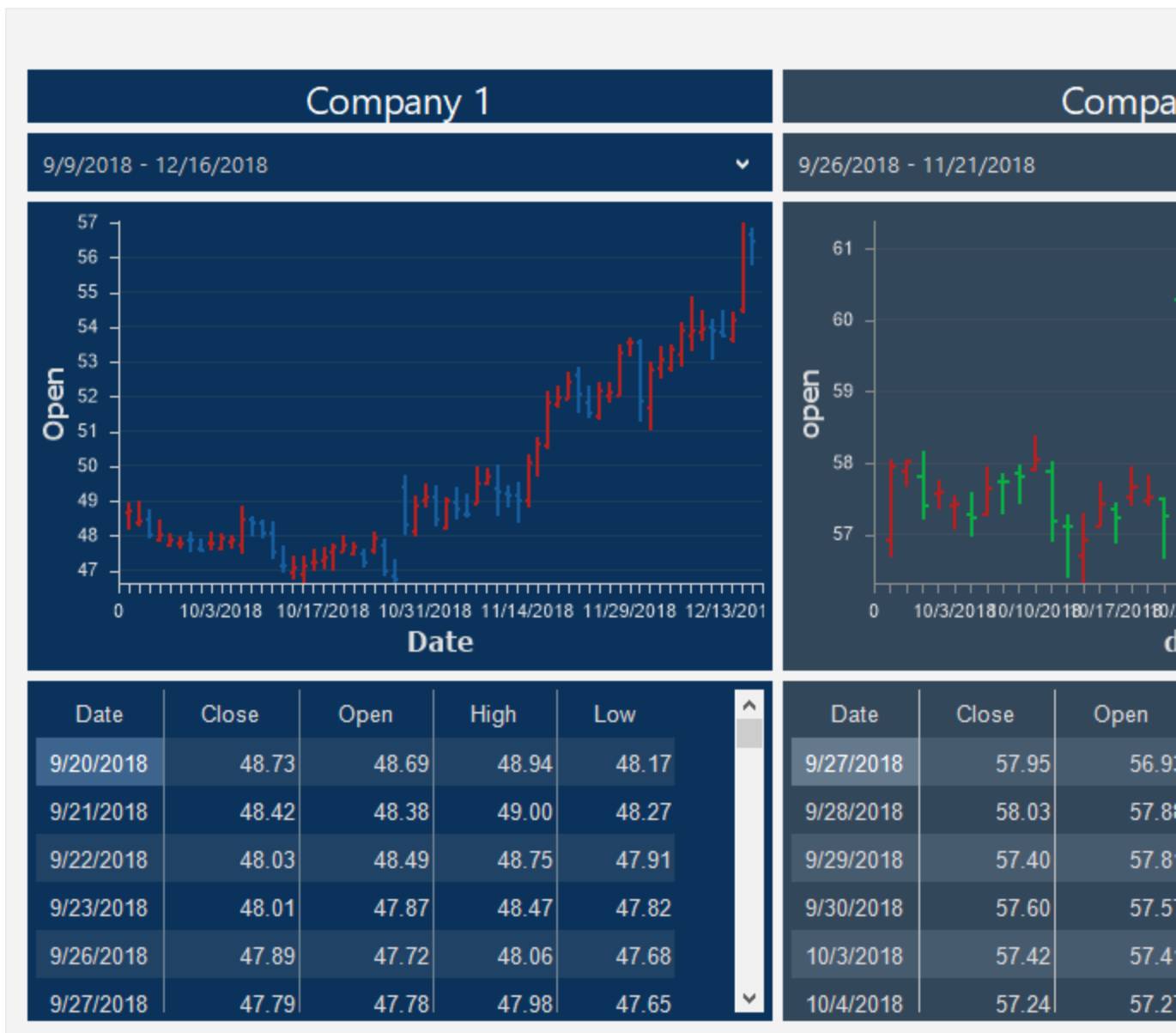
The belonging of an element to a group can be determined using the **Group** property. By default, this property is empty for an element, and it does not belong to any group. The group of elements in the dashboard is a set of elements for which the value of the **Group** property matches.

Information

You should know that using the **Group** property, you can also create relationships between elements that are located on different dashboards within the same report.

For this, those elements must belong to the same group, the values of the **Group** property of those elements must be identical.

Creating groups

Do the following to create a group of items on the dashboard:

❯ Select the elements;

❯ Specify any value in the **Group** property .

To add an element to a group, you should do the following:

❯ Select an element in the dashboard;

❯ In the **Group** property, specify the group value that is the same as for other elements in this group.

Removing elements from the group

❯ In order to remove an element from the group you should select it in the dashboard;

❯ Delete a specified value from the **Group** property.

You can also select several elements in the dashboard and delete the value from the **Group** property.

4.4 Table

Table is an element of data analysis, which provides the ability to display data field values in **Measure** and **Dimension** modes, as well as apply **Data Bars**, **Color Scale**, **Indicator**, **Sparklines** to data field values. In addition, the table element has settings for data aggregation — filtering, sorting, replacing values, calculating a cumulative total, etc.

Information

When dragging a data source to the dashboard, a table element will be created with all the data columns of this source.

The options for displaying the values of the **Table** element are made in its editor. In order to call the editor, you should:
❯ Double-click on the **Table** element;
❯ Select the **Table** element, and select the **Design** command in the context menu;
❯ Select the **Table** element, and click the **Browse** button of to the **Columns** property on the property panel.

Table element editor
In the editor of the **Table** element, you may add data fields, the order in which they are displayed in the table, the deletion, and the insertion of graphical indicators of data analysis are determined.
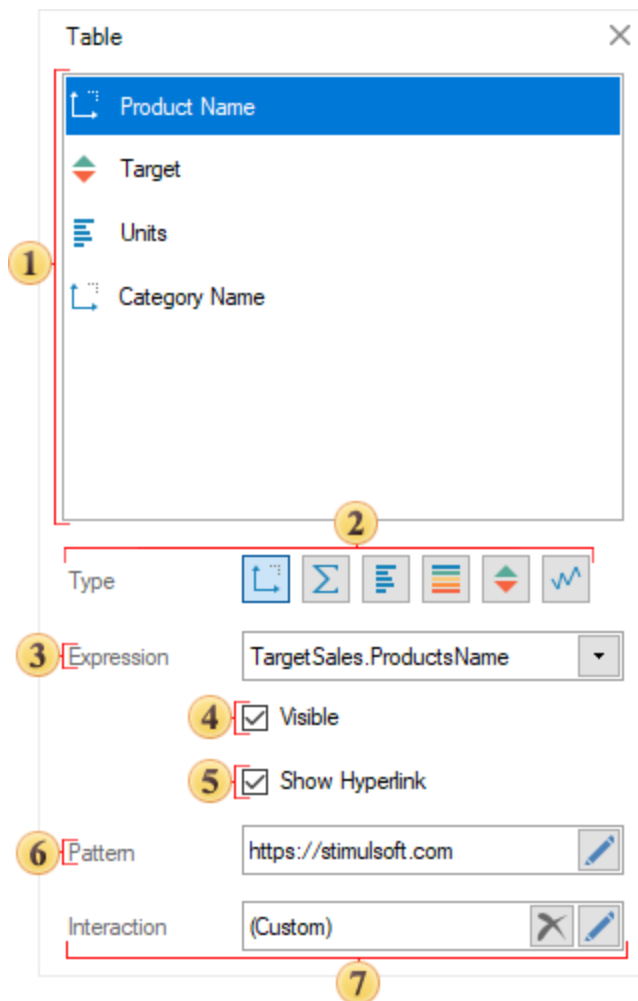
1 The list of data fields of the Table element.

2 The type of values of the selected data field:

❯ **Dimension**, the type in which the value of the data field will be displayed in the initial state.

❯ **Measure**, a type in which various functions can be applied to the values of a data field.

❯ **Data Bars**, the type in which different functions can be applied to the values of the data field, and a data bar will be added for each value of this field.

❯ **Color Scale**, the type in which different functions can be applied to the data field values, and a color scale will be added for each value of this field.

❯ **Indicator**, the type in which different functions can be applied to the values of the data field, and an indicator will be added for each value of this field.

❯ **Sparklines**, a type in which different functions can be applied to the values of a data field, and a sparkline will be added to each value of this field. By the way, in this case, sparkline also has several types - a graph, area, data bar, a win/loss. Also for a

sparkline graph or area, you can define a starting point mode.

3 The **Expression** field of the selected data field.

4 The **Visible** parameter provides the ability to enable or disable the display of the selected column in the dashboard table.

5 The **Show Hyperlink** parameter allows you to set a hyperlink for the current field values. This option is available only if the data field type is defined as Dimension.

6 In the **Pattern** field, a hyperlink is specified for the values of the current data field. This field is available only if the **Show** hyperlink option is enabled.

7 The **Interaction** parameter provides the ability to configure interactive actions for the current data field of an item.

The order of elements output

The order of the fields in the editor from top to bottom, displays the sequence of their output in the **Table** element, from left to right. In order to change the order of the output fields in the table you should change their order in the editor. To do this:

❯ Move the cursor to the required field;

❯ Press the left button of the mouse and, without releasing it, drag the field to a specific place.

Size mode

By default, the table has a fixed width of columns both in the report designer and in the report viewer. However, you can enable stretching the table. You may do this the following way:

❯ Select the **Table** element in the dashboard.

❯ Set the **Fit** value to the **Size Mode** property on the property panel. In this case, the table will stretch across the width of the element. However, in the viewer the width of the columns cannot be less than the preset width. To prevent the table from stretching by the width of the element, set the **Resizing Method** property to **AutoSize**.

Grouping data in a table

In order to group the data in the **Table** element, it is necessary for the data fields which values are to be grouped, to switch the mode from **Dimension** to **Measure**. For example, if there are three data fields in the table - a list of categories, products, the number of orders for each product from different states, then to group by product, follow the fields with the number of orders for different states to switch the item type from **Dimension** to **Measure**.

In the case of grouping data into categories, it is also necessary for the data field with the list of products to change the element type from **Dimension** to **Measure**.

Menu of a header of value columns

Each data field added to the editor is a column of values in the **Table** element. In this case, for each column a column header values will be created. The text of this header is the name of the data field in the **Table** element editor. Each header of the value column contains a drop-down menu, in which the commands for sorting and filtering by the values of the current column can be found. In order to call the drop-down menu of the header, you should click the left button of the mouse.
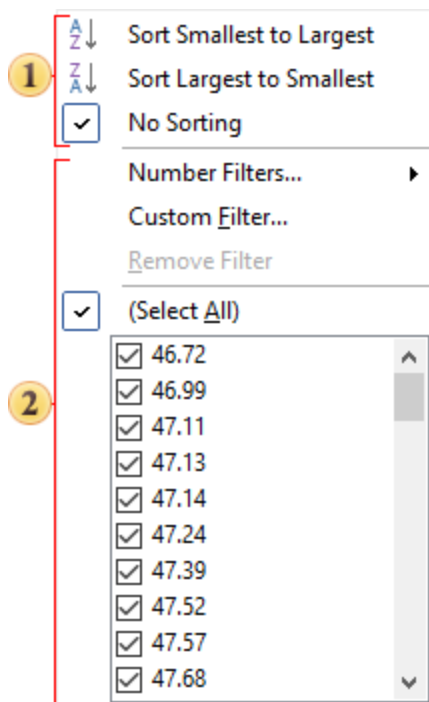
1 Commands to sort the table data by the values of the current column. In this case, the data is sorted according to the same principle as in data conversion.

2 Commands to filter table data and related items by the values of the current column. In this case, the data filtering is carried out on the same principle as when converting data - a typical filter, a custom filter, the selection of values.

Information

You can disable the sorting and filtering commands in the value column header menu using the interaction parameters of the Table element.

List of Table properties

The list shows the name and description of the properties of the **Table** element, which you may find in the properties panel of the report designer.

Name	Description
Columns	Customizes the list of columns in the **Table** element. When you click the **Browse** button, the

	element editor will be opened in which you can add, delete, enable or disable data fields. Each data field is a column in the **Table** element.
Data Transformation	Customizes the data transformation of the current item.
Group	Adds the current item to a specific group of items.
Interaction	Sets interaction of the Table element.
Size Mode	Sets the size mode of the columns of the element: **AutoSize** - optimal column widths will be calculated; **Fit** - the columns will be proportionally stretched across the entire width of the element.
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of the element - color, sides, size, and style.
Font	A group of properties defines the font family, its style, and size for the values of the element.
Fore Color	Specifies the color of the values of the element. By default, this property is set to **From Style**, i.e. the color of the values will be

	obtained from the settings of the current element style.
Header Font	A group of properties that allows you to define a font family, its style and size for the headers of the values of the Table element.
Header Fore Color	Determines the color of the headers of the values of the element. By default, this property is set to **From Style**, i.e. the color of the value headers will be obtained from the settings of the current element style.
Style	Selects a style for the current element. The default it is set to **Auto**, i.e. the style of this element is inherited from the style of the dashboard.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Layout	Customizes the controls of the item in the viewer.
Margin	A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define padding (left, top,

	right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ **Fore Color** allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style ❯ The group property **Font** that allows you to define the font family, its style and size for the title of the current element. ❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The **Visible** property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then the element header will be

	disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an item. ❯ The **Allow Move** option allows or prohibits moving an item. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property is set to **False**, then this element

is not tied to the current location.

4.5 Chart

Chart is a graphical element of data analysis, using which the data can be processed and the result is displayed as graphs.
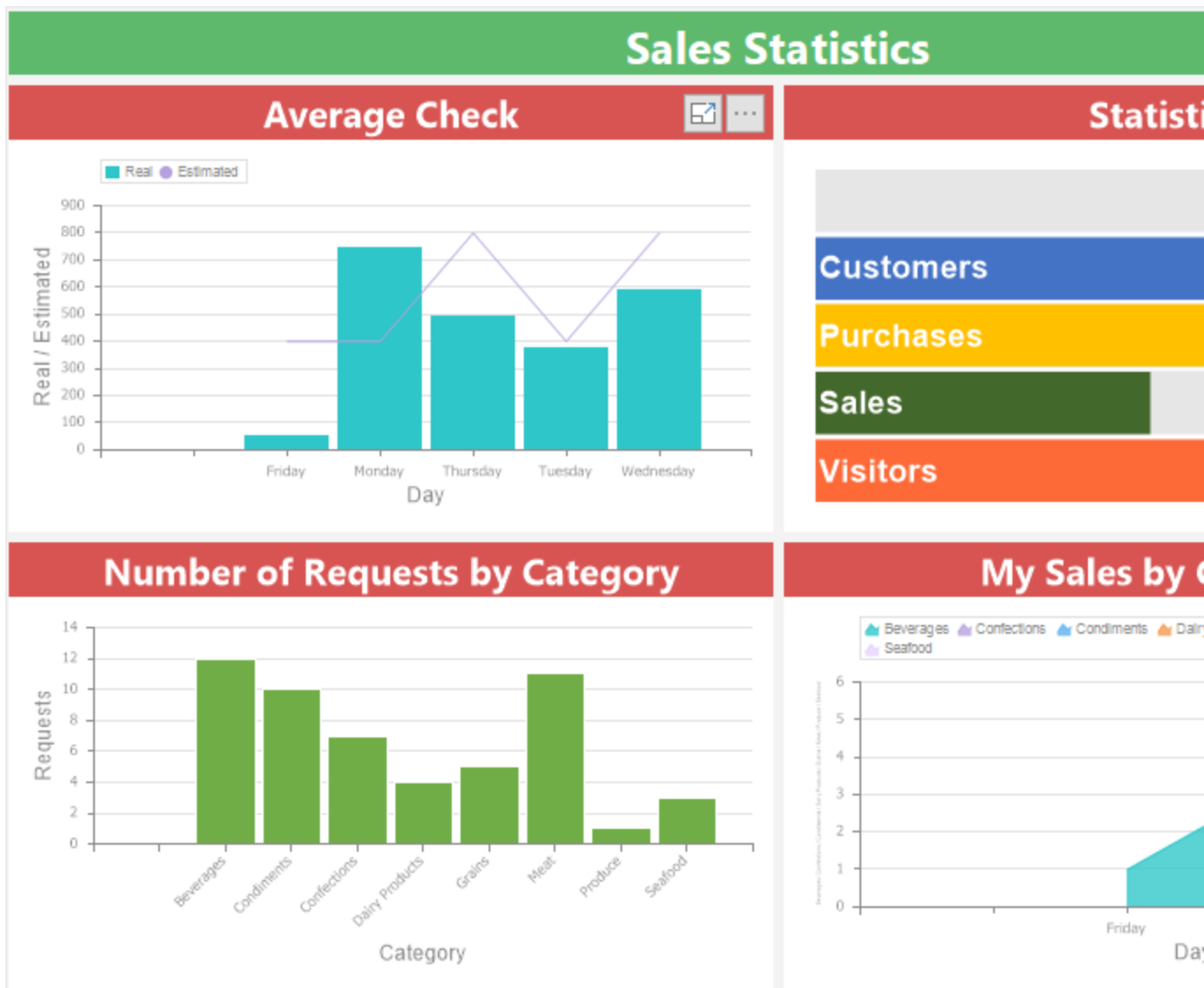
You can configure the **Chart** element in the special editor. In order to call the chart editor, you should:

➤ Double-click the left mouse button;

➤ Select the **Chart** item and select the **Design** command in the context menu;

➤ Select the **Chart** item, and, on the property panel, click the **Browse** button for the

Values, **Arguments** or **Series** properties.

In the chart editor, you can do the following:
> Specify data fields with values for the chart;
> Specify chart arguments;
> Specify the rows of the chart;
> Choose a chart type;
> Modify the expression of the selected item.

Information

The chart area is configured using the **Area** group on the property bar. You can adjust the horizontal, vertical lines and etc.

Chart values
In order to create a chart in the dashboard, at least one data field specified in the **Value** field is required:
> Drag and drop the data column from the dictionary into the **Value** field, and for

newly added items - into the editor or chart area.

❯ Create **New Field**. Set the expression for this element, the processing result of which will be the values for the chart.

Also, the chart can specify the arguments and series. If the arguments and series are not specified, then all element values will be processed and displayed using one graphic element. For example, if three data fields are created in the **Value** field, then three graphical elements will be displayed in the **Chart**.
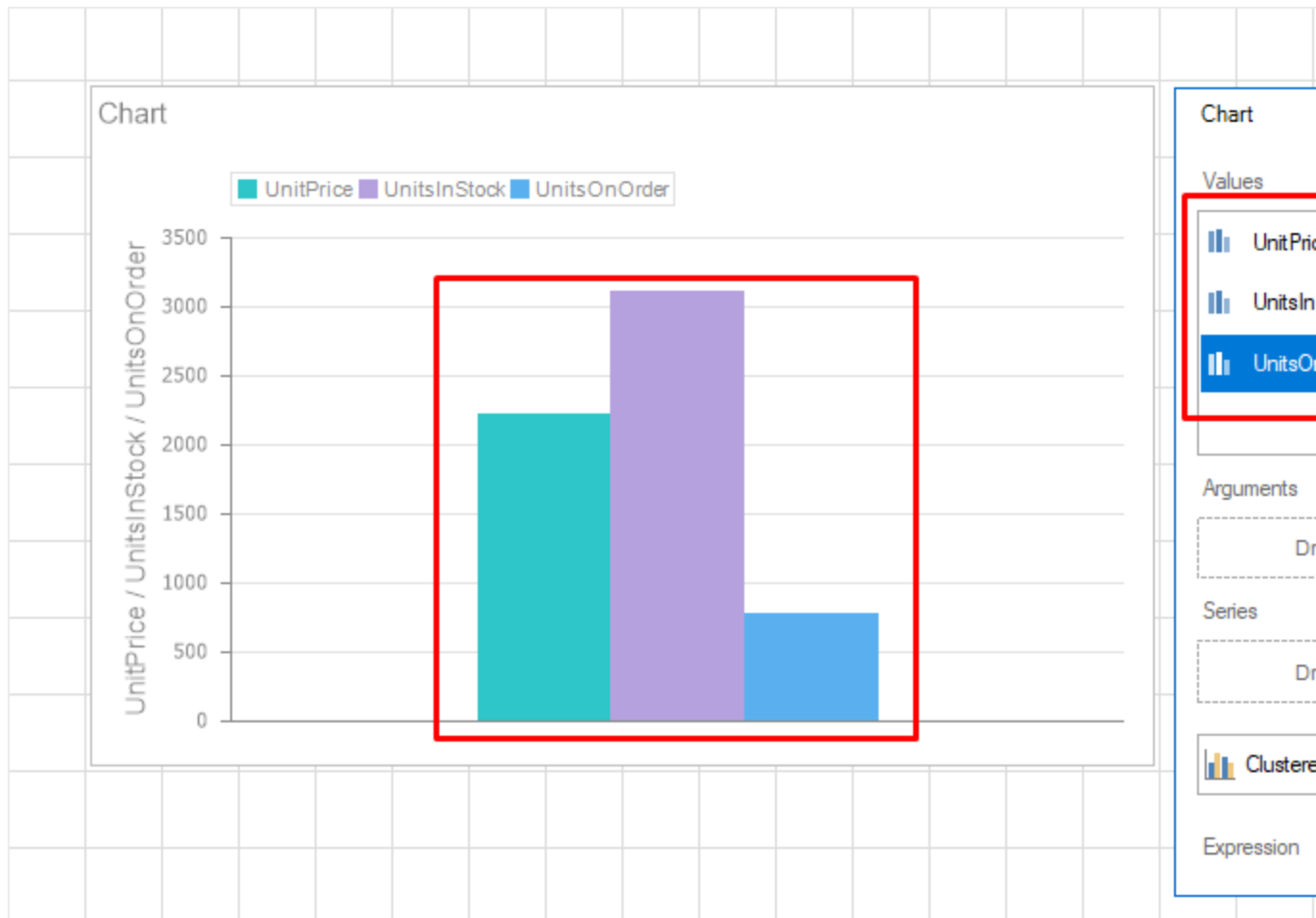
Information

For some types of charts require setting values in several fields. For example, for financial charts you need to specify the value in the fields - **Open**, **Close**, **Max**, and **Min**. In this case, you should create at least one data field for every Value field.

Chart types

Depending on the type of a chart chosen, the data will be displayed using one or another graphic element. You can display several types of charts within the same chart element.

Information

Within the same **Chart** element, not all types are compatible. It is impossible to display the **Funnel** and **Gantt** in one element.

You should know that only one type of a chart can belong to one data field. If it is necessary to display the same data field with different types of charts within the same **Chart** element, you should create several duplicates of this data field in the **Value** field and specify one of the chart types for every copy.

In order to change the type of a chart, you should do the following:
> Double-click the left mouse button on the **Chart** item;
> Click the chart type button in the editor;
> Select the chart type you need.
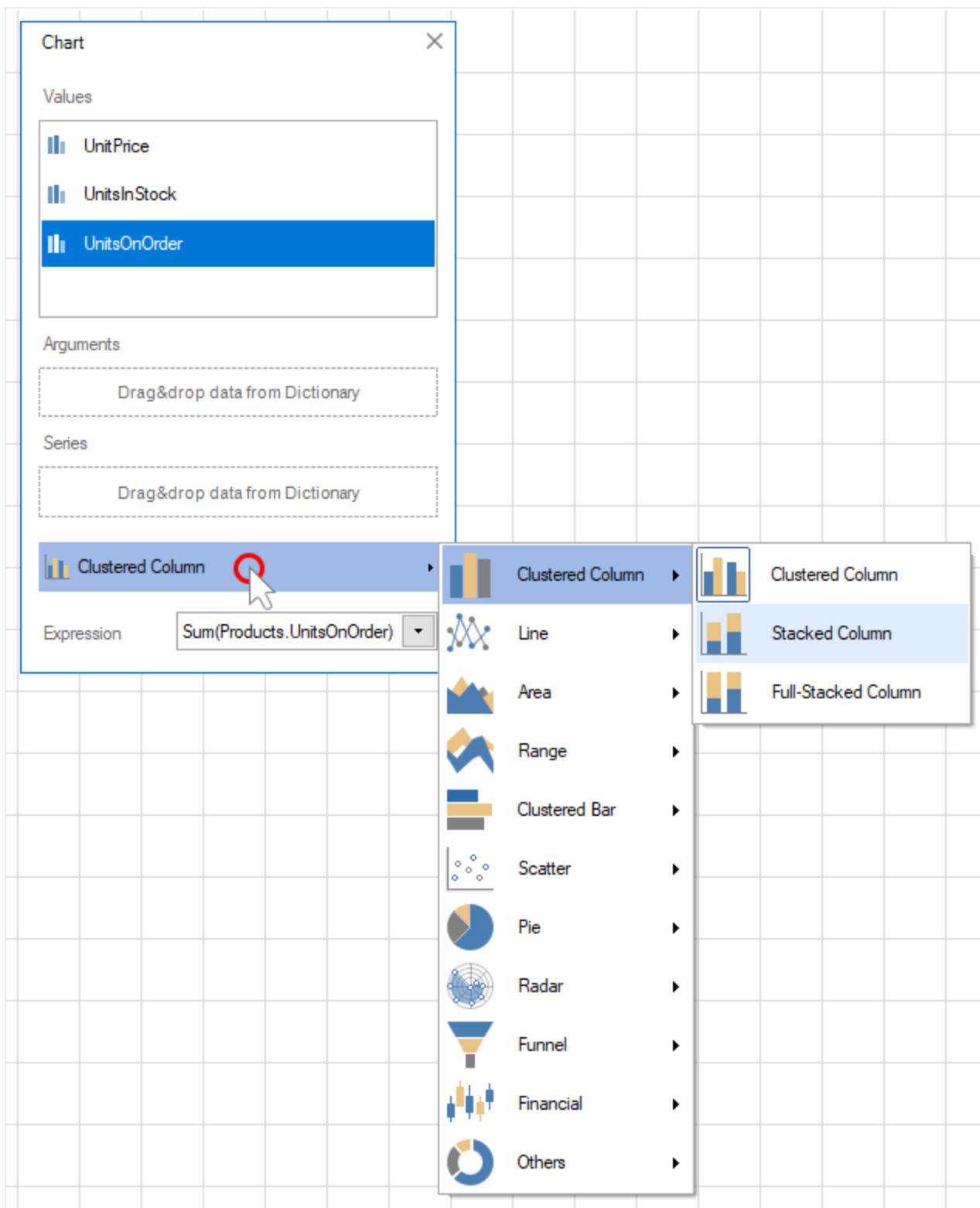
Arguments

The argument refers to data that is associated with the values of the chart. In other words, every value of the chart will correspond to some value. For example, product prices are related to the list of products, i.e. every product has its own price. In this case, in the chart, each product will be represented in a separate graphic element.

Also, for product prices, an argument may be a category of products. In this case, for each category of products, a graphic element will be presented. The value of this graphic element will be the sum of the prices of products which are included into this category.
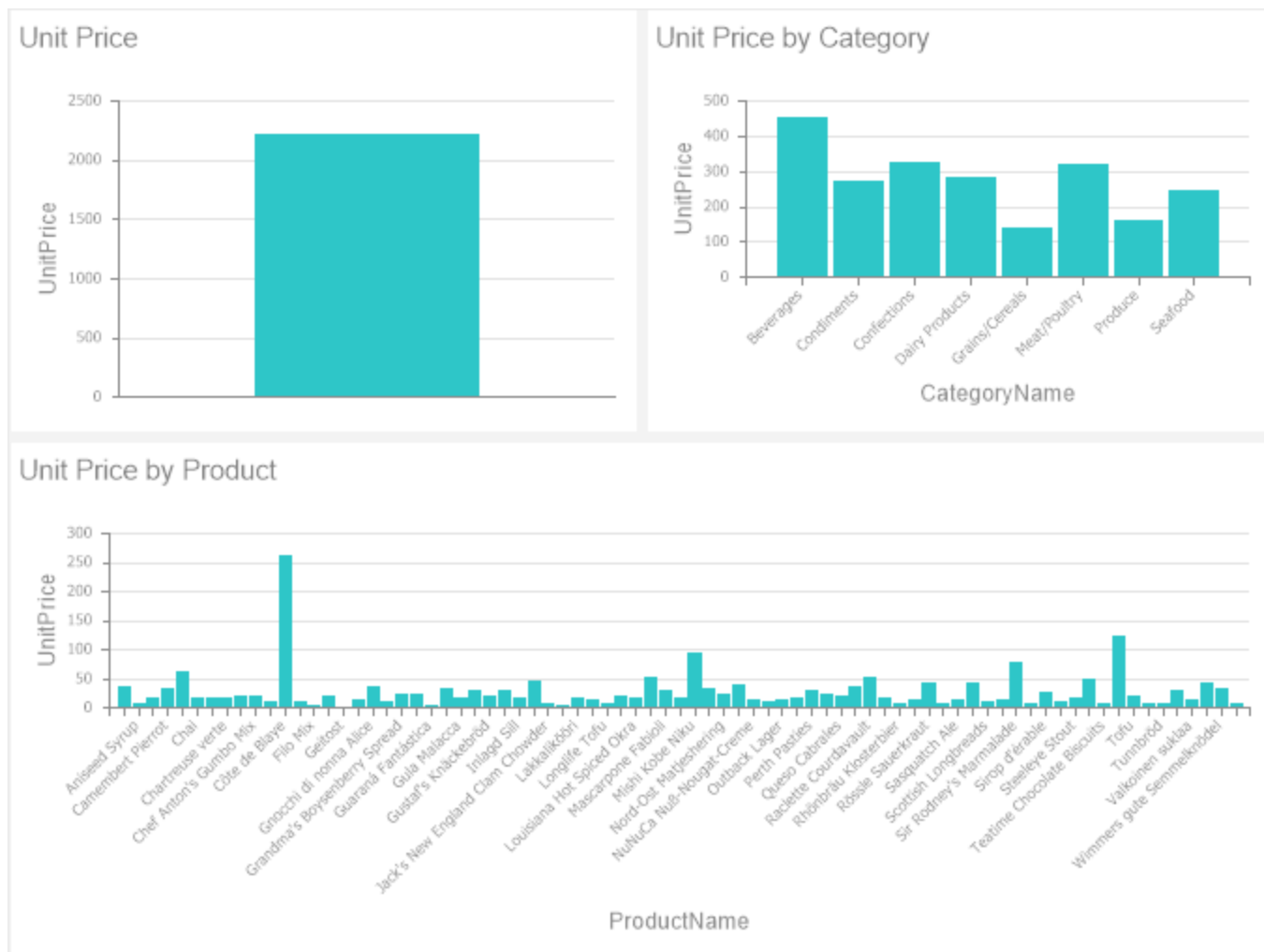
For charts with an area of X - Y, the arguments are the values along the X axis (except for bar charts). In the case of other chart types, the arguments are separate chart segments.

In order to set chart arguments, you should to the following:

❯ Double-click the mouse left button on the **Chart** element;

❯ In the element editor, drag ad drop the data column from the dictionary to the **Arguments** field.

Create **New Field** in the **Arguments** field. Set the expression for this element, the processing result of which will be the arguments for the chart.

Information

In the chart editor, you can specify multiple data fields for the **Arguments** field.

Series

Series of charts are graphical elements with or without arguments and grouped by a specific value.

For example, you have a chart with product prices (chart values) and a list of products (chart arguments). If you add an element to the series of the chart with the category data for these products, then a list of products will be displayed for every category. Below is a chart with prices for every product category and one argument.
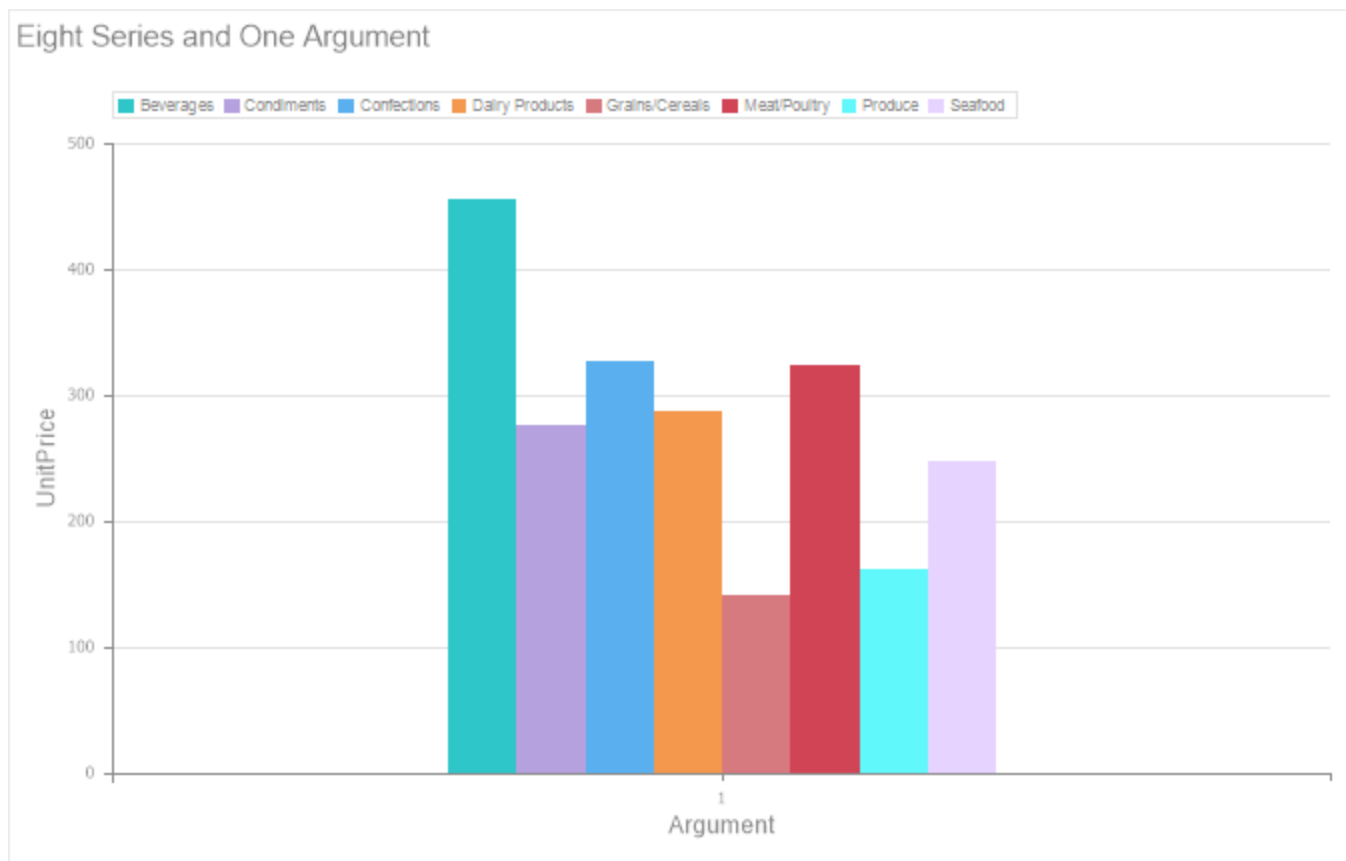
In order to set the series of a chart, you should do the following:

❯ Double-click the mouse left button on the Chart item;

❯ In the element editor, drag and drop the data column from the dictionary to the **Series** field.

❯ Create **New Item** in the **Series** field. Set an expression for this element, the processing result of which will be series for the chart.

Information

The chart axes are configured using the **X Axis** and **Y Axis** property groups (you can find them on the property panel). You can customize axis labels and titles.

The Color Each property

By default, the graphical elements of a diagram within one series have one color. However, if you need to display each graphic element in a separate color, you should:

❯ Select the Chart element;

❯ set the **Color Each** property to **true** on the property panel.

Chart Legend

The chart legend is a description for the graphic elements. If the chart has series, the legend will automatically be enabled. The legend shows:

❯ **Marker** is a special graphic icon with the color of the graphic element to which it belongs.

❯ The value of the series for a specific graphic element of the chart;

❯ If the arguments are set for the chart, it shows the value of the argument for a specific graphic element of the chart.

List of properties

The list shows the name and description of the properties of the element which you may find in the properties panel of the report designer.

Name	Description
Values	Sets a data field for chart values. When you click the **Browse** button, the element editor will be opened in which you can add or remove data value fields.
Arguments	Sets a data field for chart arguments. When you click the **Browse** button, the element editor will be opened in which you can add or remove argument data fields.

Series	Sets a data field for chart series. When you click the **Browse** button, the element editor will be opened in which you can add or remove series data fields.
Data Transformation	Customizes the data transformation of the current item.
Group	Adds the current item to a specific group of items.
Interaction	Customizes the interaction element of the chart.
Top N	Customizes the top values of the element.
Area	A property group that is used to customize the chart area: ❯ The **Color Each** property is used to set a unique shade for every graphic element of the chart. If this property is set to **true**, then colors from the style collection will be applied to graphic elements. Every graphic element will have its own color. After all the colors from the collection are used, the same colors with a lightening coefficient will be applied to the other graphic elements. Thus, every graphic element will have a certain shade. If this property is set to **false**, then the graphic elements of one series will use one color from the collection of style colors. ❯ The **Grid Lines Horizontal** group of properties is used to

	change the color and visibility of horizontal grid lines. If the **Visible** property is set to **true**, the horizontal grid lines will be displayed. ❯ The **Grid Lines Vertical** group of properties is used to change the color and visibility of horizontal grid lines. If the **Visible** property is set to **true**, the vertical grid lines will be displayed. ❯ The **Interlacing Horizontal** group of properties is used to change the color and visibility of horizontal interlacing. If the **Visible** property is set to **true**, the horizontal interlacing will be displayed. ❯ The **Interlacing Vertical** group of properties is used to change the color and visibility of vertical interlacing. If the **Visible** property is set to **true**, the vertical interlacing will be displayed. ❯ The **Reverse Horizontal** property is used to mirror the chart area horizontally. If the property is set to **true**, the area will be displayed horizontally. ❯ The **Reverse Vertical** property is used to mirror the chart area horizontally. If the property is set to **true**, the area will be displayed vertically.
Format of arguments	Customizes the formatting of the arguments the Chart element.
Legend	A property group that is used to

	customize the chart legend: ❯ The **Horizontal Alignment** property is used to determine the horizontal position of the legend on the Chart element. The legend can be located in the chart area or outside of it. ❯ The **Labels** group of properties is used to change the color and font of the legend label. ❯ The **Title** group of properties is used to customize the title of the legend - specify the text of a title, change its color and font. The **Vertical Alignment** property is used to set the vertical position of the legend on the Chart element. The legend can be located in the chart area or outside of it.
X axis	A group of properties that is used to customize the value axis: ❯ The **Labels** property group is used to customize labels of the X axis - change color and font, specify the rotation angle, text before or after labels, their position and alignment. ❯ The **Title** property group is used to customize the title of the X axis - specify the text of the axis title, change its color and font, direction, alignment, and position.
Y axis	A group of properties that is used to customize the value axis: ❯ The **Labels** property group is used to customize labels of the Y axis - change color and font,

	specify the rotation angle, text before or after labels, their position and alignment. ❯ The **Title** property group is used to customize the title of the Y axis - specify the text of the axis title, change its color and font, direction, alignment, and position.
Format of values	Customizes the <u>formatting of the values</u> of the Chart element.
Back Color	Changes the background color of the Chart element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of a table - color, sides, size, and style.
Negative Series Color	Customizes the list of colors for negative values of the rows of the Chart element.
Series Color	Customizes the list of colors for the values of the rows of the Chart element.
Style	Selects a style for the current element. The default it is set to **Auto**, i.e. the style of this element is inherited from the style of the dashboard.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing

	the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Layout	Customizes the controls of the element in the viewer.
Margin	A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the Table element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ **Fore Color** allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style ❯ The group property **Font** allows you to define the font family, its style and size for the title of the current element. ❯ The **Horizontal Alignment**

	property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The **Visible** property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and

	movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property is set to **False**, then this element is not tied to the current location.

4.6   Gauge

Gauge is an element of the dashboard panel using which you can display the processed value from the data field.
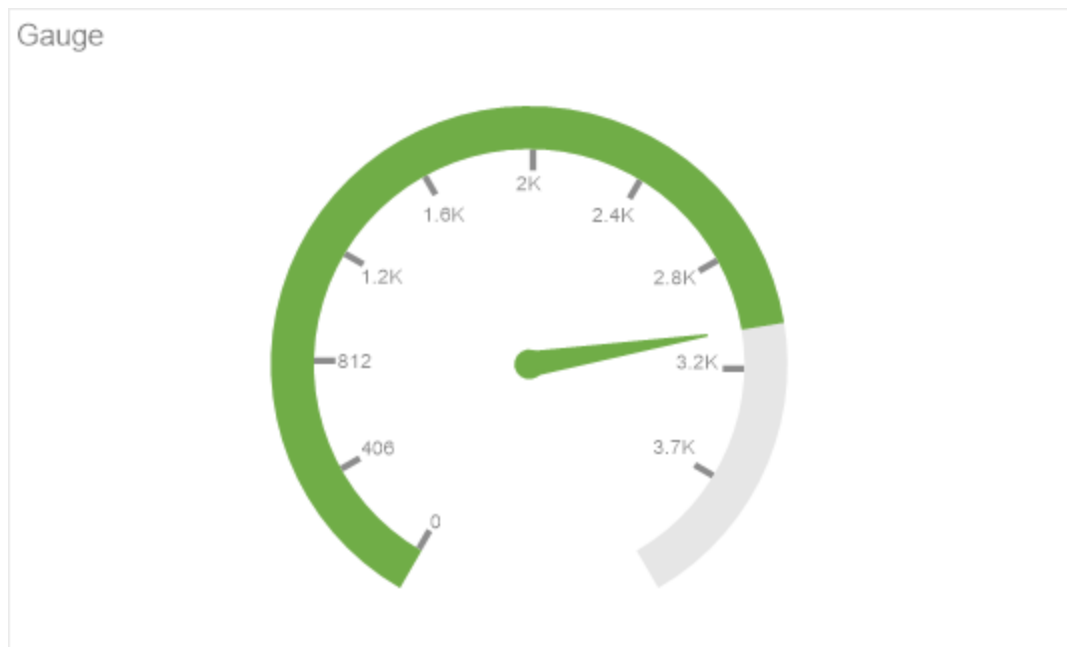
You may adjust the **Gauge** element in its editor. In order to call the editor, you should:
❯ Double-click the mouse left button on the item;
❯ Select the **Gauge** element, and select the **Design** command in the context menu;

❯ Select the **Gauge** element, and, on the property panel, click the **Browse** button of the **Value** or **Series** properties.

In the Gauge editor you can:
❯ Specify a data field with values for the gauge;
❯ Specify the series of the gauge;
❯ Select **Type** of the gauge;
❯ Modify **Expression** of the selected element;
❯ Select and adjust the range of the gauge;
❯ Set the color palette of the gauge scale.

Gauge values

In order to create a gauge on the dashboard panel, you need a data field in the **Value** field. To do this:

❯ Drag and drop the data column from the dictionary to the **Value** field, and for newly added items, drag it into the editor or the gauge area.

❯ Create **New Field**. Set the expression for this element, the processing result of which will be the value for the gauge.
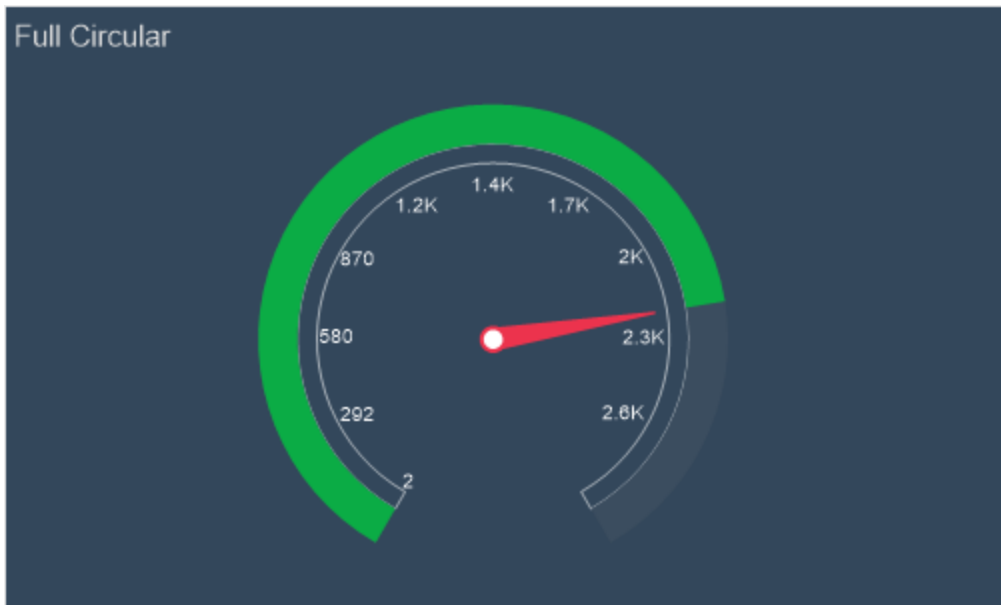
Series of gauge

A gauge series is a separate graphic element for a specific segment of values selected by a specific condition. The condition in this case will be the values of the data field which is indicated in the **Series** field.

For example, a data field with product prices is specified in the **Gauge** field. Without specifying the series one graphic element will be displayed, the value of which will be the sum of the prices of all products.

Gauge
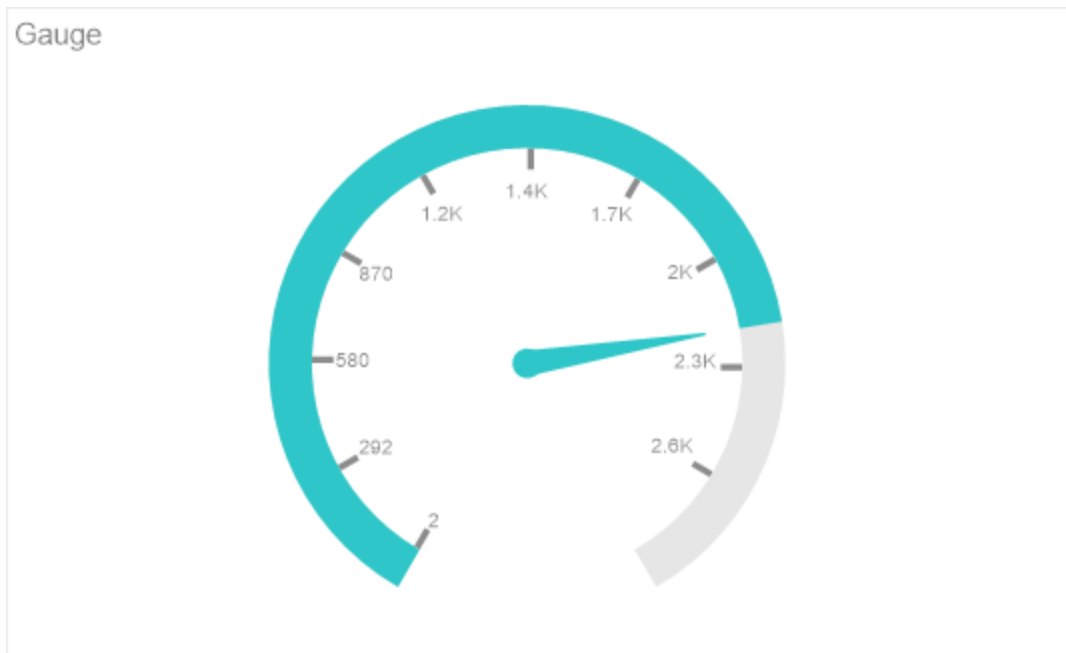
If you specify a data field with a list of products in series then, for every product, a graphic element will be displayed, the value of which will be the price of this product.

Gauge by Product
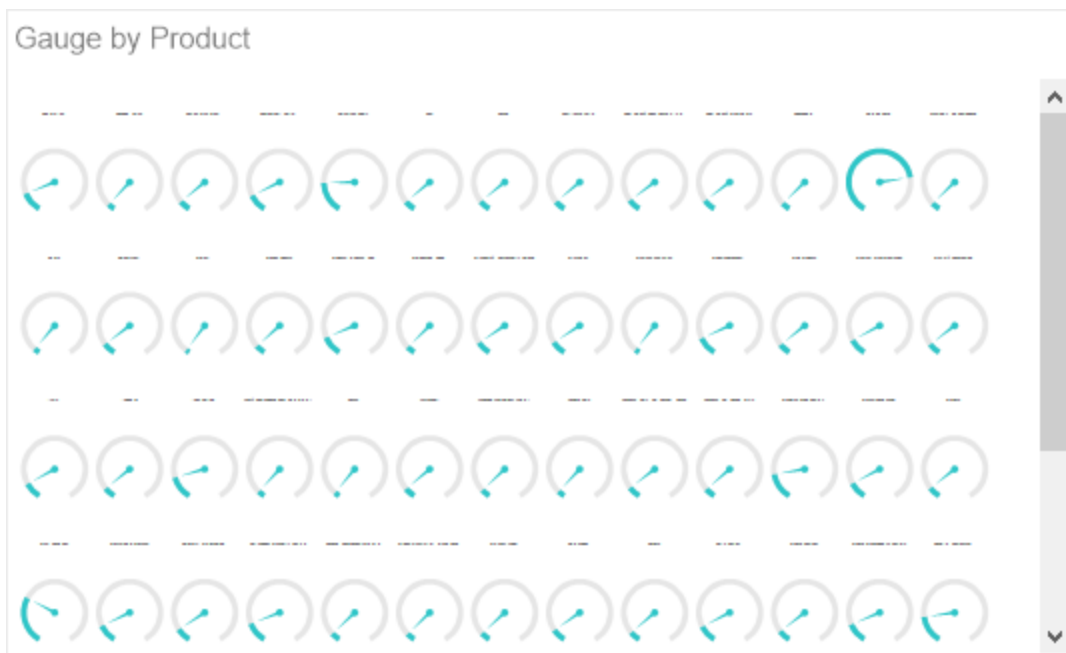
If you specify a data field with a list of product categories in series, then a graphic element will be displayed for every category. The value of this graphic element will be

the sum of the prices of products included into this category.
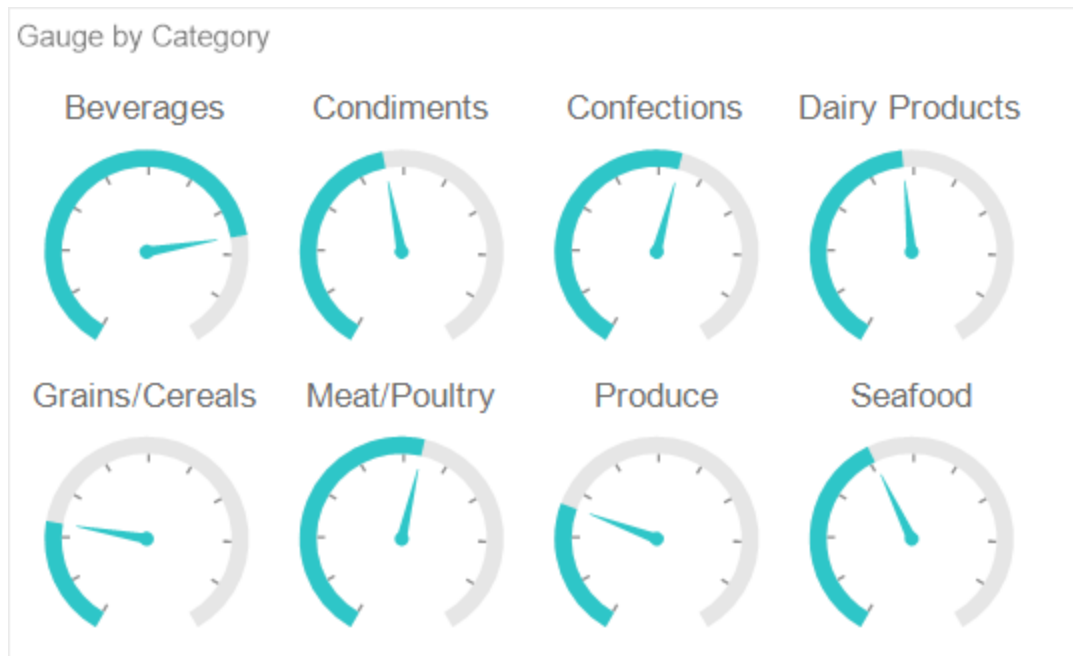
In order to specify the series of the gauge, you should do the following:

❯ Double-click the mouse left button on the gauge;

❯ In the element editor, drag and drop the data column from the dictionary to the **Series** field.

❯ Create **New Field** in the **Series** field. Set the expression for this element, the processing of which will be series for the gauge.

Gauge types

The gauge can be of the following types:

❯ Full Circular;

❯ Half-Circular;

❯ Linear.
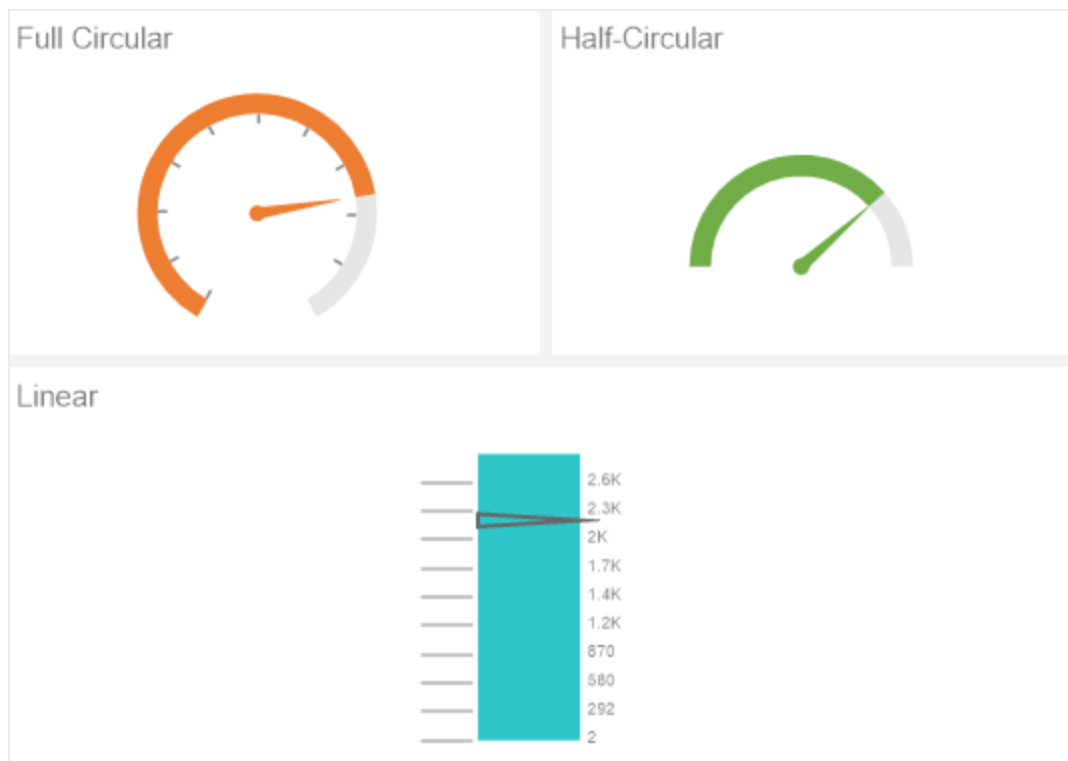
In order to change the type of a gauge, you should:
> Double-click the mouse left button on the gauge;
> Using the control buttons, select one of the types of a gauge.
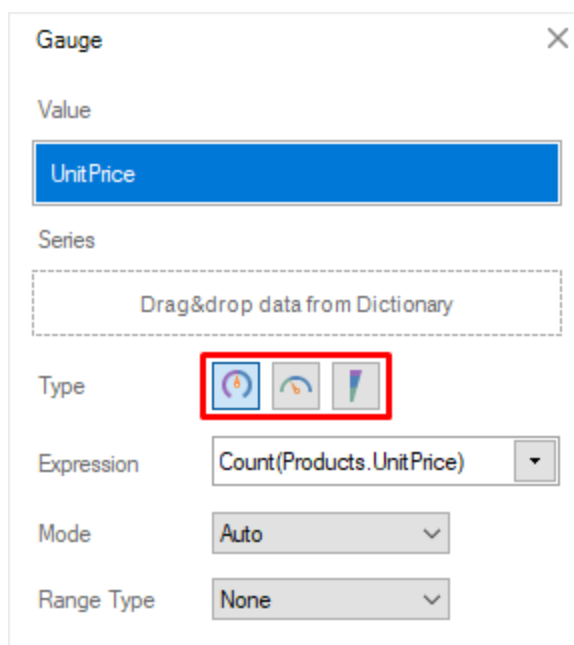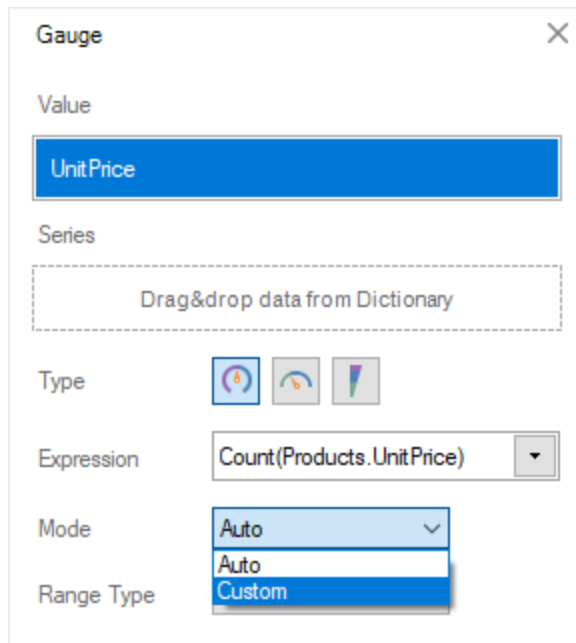
Information

Within the limits of one **gauge** element it is possible to use only one type of a gauge. Even in the case of multiple series, the type of the gauge will be the same for all values.

Gauge range of values

Regardless of the gauge type, its values and series, you can define a range of values. By default, the **Autorange** mode is used. In this case, the initial and final value of the gauge scale is calculated automatically. However, if you need to specify a specific range of values, you should do the following:

❯ Double-click the mouse left button on the gauge;
❯ In the **Mode** field, left-click on the menu.
❯ Select **Custom** in the drop-down list.

❯ In the **Minimum** field you should set the initial value of the gauge scale;
❯ In the **Maximum** field you should set the maximum value of the gauge scale;

Select the **Auto** value in the **Mode** field in order to enable the automatic mode for calculating the range of values of the gauge scale.

Multi-colored scale

By default, the gauge scale is monochromatic. However, you can adjust the color for a specific range of the scale. To do this:

❯ Double-click the mouse left button on the gauge;

❯ Left-click in the **Range Type** field.

❯ Select **Color** in the drop-down list.

Then you should:
> Determine the type of values for the scale range - **Percentage** or **Value**;
> Customize the list of ranges;
> Customize every range by specifying the start - end values of the range and its color.

List of properties
The list shows the name and description of the properties of the element which you

may find in the properties panel of the report designer.

Name	Description
Value	Sets the data field of the current item values. If the data field is set, this value will include two properties: ❯ **Expression** is used to change the expression of the **Value** data field of this element; ❯ **Label** is used to change the name of the data field in the **Value** field of the current item.
Series	Sets the data field of the rows of the current item. If the data field is set, this value will include two properties: ❯ **Expression** is used to change the expression of the **Series** data field of this element; ❯ **Label** is used to change the name of the data field in the **Series** field of the current item.
Calculation Mode	Sets the mode of calculating the value scale of the gauge. If the value is set to **Auto**, the scale will be calculated automatically. If the value is set to **Custom**, then the **Minimum Value** and **Maximum Value** properties will be displayed, which will specify the scale of values of the gauge.
Data Transformation	Customizes the data transformation of the current item.
Group	Adds the current item to a specific group of items.

Minimum	Sets the initial value of the scale of the gauge. This property is displayed only if the **Calculation Mode** property is set to **Custom**.
Maximum	Sets the final value of the scale of the gauge. This property is displayed only if the **Calculation Mode** property is set to **Custom**.
Range Mode	Changes the type of values to calculate the range of the scale of values. If the property is set to **Percentage**, the size of the range of the value scale is calculated in relative values. The scale of the gauge is 100 percent. If the value of this property is set to **Value**, the size of the range of the scale of values is calculated in absolute values.
Range Type	Sets the range mode. If the property is set to **None**, the scale of values will have the same color. If the property is set to **Color**, then the scale of values can be divided into ranges of a specific color.
Short Value	Applies an abbreviation for the values of the gauge. Has two values - true and false.
Type	Changes the type of a gauge - Full Circular, Half-Circular, Linear.
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.

Border	A group of properties that allows you to customize the borders of the element - color, sides, size, and style.
Font	A group of properties defines the font family, its style, and size for the values of the element.
Fore Color	Specifies the color of the values of the element. By default, this property is set to **From Style**, i.e. the color of the values will be obtained from the settings of the current element style.
Style	Selects a style for the current element. The default it is set to **Auto**, i.e. the style of this element is inherited from the style of the dashboard.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Layout	Customizes the controls of the item in the viewer.
Margin	A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define padding (left, top,

	right, bottom) of the graphic element area from the margin of the value area.
Title	A group of properties that allows you to customize the title of the element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ **Fore Color** allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style ❯ The group property **Font** that allows you to define the font family, its style and size for the title of the current element. ❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The **Visible** property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then

	the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property

	is set to **False**, then this element is not tied to the current location.

4.7 Pivot

Pivot is an element of the dashboard, which is used to process, group and summarize data values by rows and columns of this table.

Pivot Table

Pivot Table			CategoryName				
ProductName		OrderID	Beverages	Condiments	Grains/Cereals	Meat/Poultry	Seafo
▶ Alice Mutton						1443	
		10289		10			
		10405		10			
		10485		10			
		10540		10			
		10591		10			
		10702		10			
◢	Aniseed Syrup	10742		10			
		10764		10			
		10849		10			
		10857		10			
		11017		10			
		11077		10			
		Total		120			
▶ Boston Crab Meat							75
▶ Carnarvon Tigers							168
▶ Chai			684				
▶ Chang			836				
▶ Chartreuse verte			540				
▶ Chef Anton's Cajun Seasoning				440			

To display the pivot element you should to add at least one data field in the **Totals** field. Element settings of the **Pivot** table is implemented in the element editor. In order to call the editor, you should:

❯ Double-click the **Pivot** item;

❯ Select the **Pivot** item, and select the **Design** command in the context menu;

❯ Select the **Pivot** item, and, on the property bar, click the **Browse** button of the **Columns** property.

Editor of the Pivot table

In the editor of the **Pivot** table, you can add elements with data and edit expressions for these elements, as well as adjust the top values of the element.
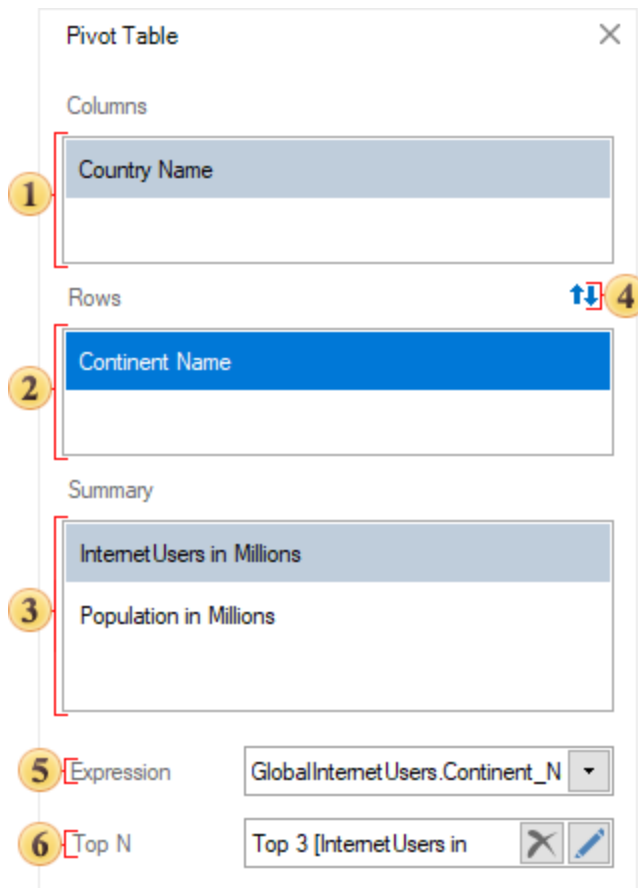
1. The **Columns** field indicates the data fields for the rows of this table;
2. The **Row** field indicates data fields for the columns of this table;
3. The **Summary** field indicates the data fields for the resulting cells of this table;
4. The **button** is used to change data fields between **Columns** and **Rows** fields.
5. The **Expression** field in which the expression of the selected data field is displayed.
6. The **Top N** parameter is used to customize the list of maximum or minimum values of the pivot table. The top values are set in the Top N values editor. To call the editor, click the **Edit** button in the current field. In order to reset the top values, click the **Remove** button in the current field.

Totals

At the intersection of the columns and rows of the pivot table you can see cells. A value from the corresponding cell of the data source will be added to this cell, i.e. the value from the data source cell formed at the intersection of the corresponding column and rows in the data source. Then, all values of each row and each column will be summed up and displayed in the resulting cells of the pivot table. Also, in the **Totals** field you can specify several data fields. In this case, cells will be added to the pivot table both for the first data field and for the second one.

Pivot Table

Pivot Table		Beverages	Condiments	Grains/Cereals	Meat/Poultry	Seafood	Total
ProductName		Beverages	Condiments	Grains/Cereals	Meat/Poultry	Seafood	Total
Chai	1	39					39
	2						
Chang	1	17					17
	2	40					40
Chartreuse verte	1	69					69
	2						
Chef Anton's Gumbo Mix	1						
	2						
Côte de Blaye	1	17					17
	2						
Guaraná Fantástica	1	20					20
	2						
Gula Malacca	1		27				27
	2						
Röd Kaviar	1					101	101
	2						
Røgede sild	1					5	5
	2					70	70
Sasquatch Ale	1	111					111
	2						

Rows

This field of the pivot table indicates the data fields which values will form the rows of the pivot table. Also in this field you can specify multiple items. In this case, the data fields must be related with each other, because the values of the top data field in this field are "predecessor" for the values of the lower data field. For example, if the top data field contains a list of categories, and the bottom contains a list of products.

In this case, in the pivot table, each category will be a separate line in the pivot table. However, each category will contain its own list of products that will form the rows of the pivot table within that category.

Pivot Table

Pivot Table		CategoryID					
CategoryName	ProductName	1	2	5	6	8	Total
▶	Beverages	273					273
◢ Condiments	Chef Anton's Gumbo Mix						
	Gula Malacca		27				27
	Sirop d'érable		113				113
	Total		140				140
▶	Grains/Cereals			61			61
▶	Meat/Poultry				21		21
▶	Seafood					201	201
	Total	273	140	61	21	201	696

Columns

This field of the pivot table indicates the data fields which values will form the rows of the pivot table. Also in this field you can specify multiple data fields. In this case, the data fields must be related to each other, because the values of the top data field in this field are "predecessor" for the values of the lower data field. For example, if the top data field contains a list of categories, and the bottom contains a list of products.
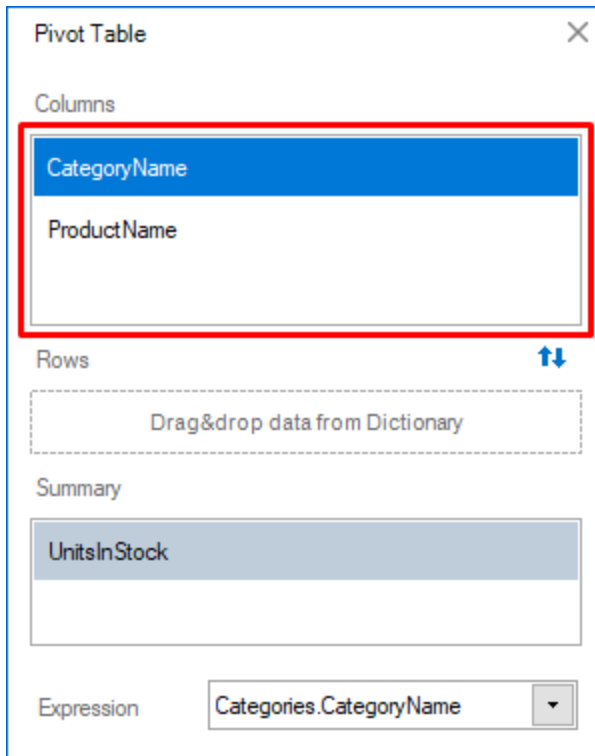
In this case, in the pivot table, each category will be a separate column in the pivot table. However, each category will contain its own list of products that will form the columns of the pivot table within that category.

Pivot Table

Pivot Table		CategoryName, ProductName						
CategoryID	▸Beverages	◢ Condiments				▸ Grains/Cereals	▸Me
		Chef Anton's Gumbo Mix	Gula Malacca	Sirop d'érable	Total		
1	273						
2			27	113	140		
5						61	
6							
8							
Total	273		27	113	140	61	

List of properties

The list shows the name and description of the properties of the element which you may find in the properties panel of the report designer.

Name	Description
Columns	Customizes the list of columns in the element. When you click the **Browse** button, the element editor will be opened in which you can add or delete data fields in the **Columns** field.
Rows	Customizes the list of rows in the element. When you click the **Browse** button, the element editor will be opened in which you can add or delete data fields in the **Rows** field.
Summary	Customizes the list of values in the element. When you click the **Browse** button, the item editor will be opened in which you can add or remove data fields in the **Summary** field.
Data Transformation	Customizes the data

	transformation of the current element.
Group	Adds the current item to a specific [group of items](#).
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of the element - color, sides, size, and style.
Style	Selects a style for the current element. The default it is set to **Auto**, i.e. the style of this element is inherited from the style of the dashboard.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Layout	Customizes the controls of the item in the viewer.
Margin	A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows

	you to define padding (left, top, right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ Fore Color allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style ❯ The group property **Font** that allows you to define the font family, its style and size for the title of the current element. ❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The **Visible** property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then

	the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property

	is set to **False**, then this element is not tied to the current location.

4.8   Indicator

Indicator is an element of the dashboard that represents the ability to display the aggregated value of the data field, as well as the rate of increase of this value to the target. In addition, the growth rate and the aggregated value of the indicator can be grouped by a condition.
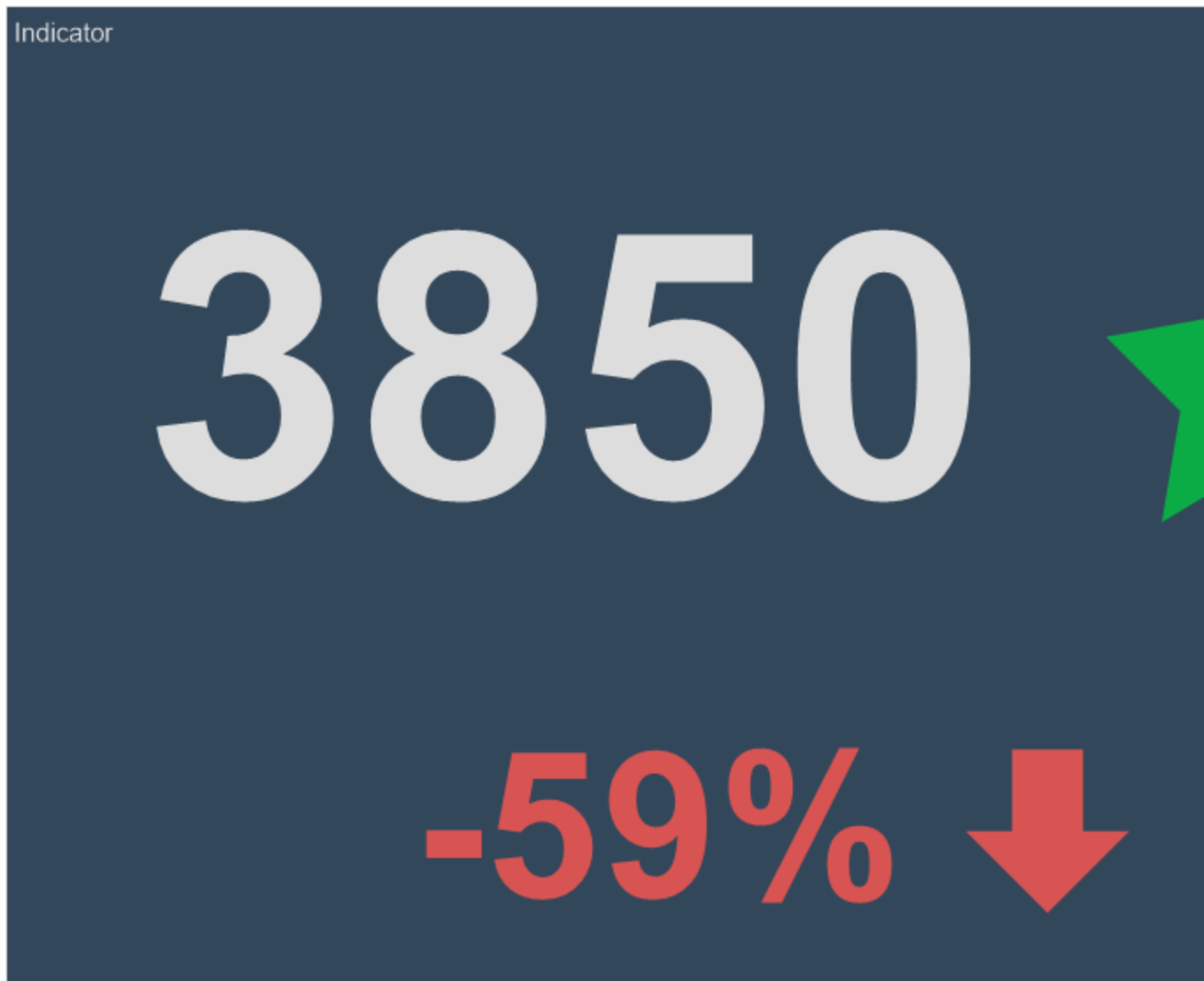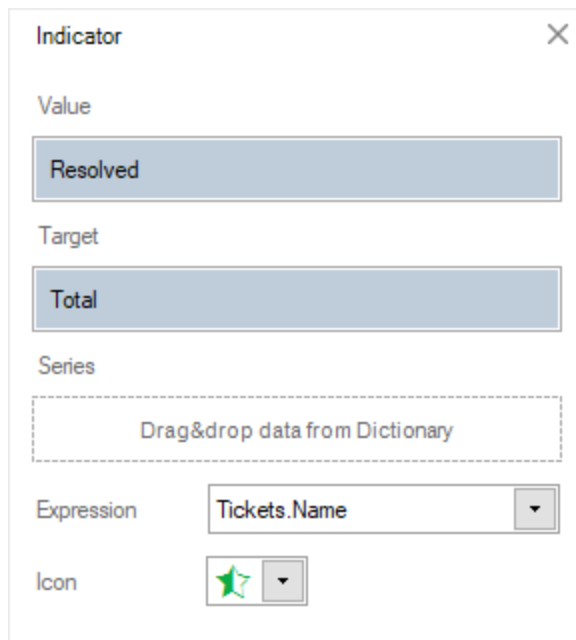
To display the **Indicator**, you should add a data item in the **Value** field. In this case, the value will be displayed with a specific graphic element. Also in order to display the

growth rate, it is necessary to set the data element in the **Target** field. The settings of the **Indicator** element can be done in the element editor. To call the editor, you should:

> Double-click on the **Indicator** element;

> Select the **Indicator** element and select the **Design** command in the context menu;

> Select the **Indicator** element, and, on the property panel, click the **Browse** button of the Value, Target, and Series properties.

Indicator editor

In the **Indicator** editor, you can add elements with data, edit the expressions of these elements, select a graphic element to indicate the value.

In the **Indicator** editor you can:

> Specify the data field for the Indicator value;

> Specify the data field for the target value of the Indicator;

> Specify the data field for the series of the Indicator;

> Select a graphic element to display the value.

Indicator value

In the value field, you can specify only one data field. All values of this field will be aggregated, i.e. a function will be applied to them. By default, this is a summation function for numeric values. If a field with non-numeric values is added, then by default the function of the number of rows in this data field is applied to them.

Information

In the **Indicator** element, you can specify only the value. In this case, the aggregated value of the data field will be displayed with a specific graphic element, without a growth rate.

Target value

To use the indicator to display the growth rate, besides the value in the indicator, it is necessary to indicate the target value. The target value is the aggregated value of the

data field specified in the **Target** field of the indicator. In the **Target** field, you can specify only one data field. By default, the summation function for numeric values is applied to the data field in the **Target** field. If a data field with non-numeric values is added, then, by default, the function of counting the number of rows in this data field is applied to it.
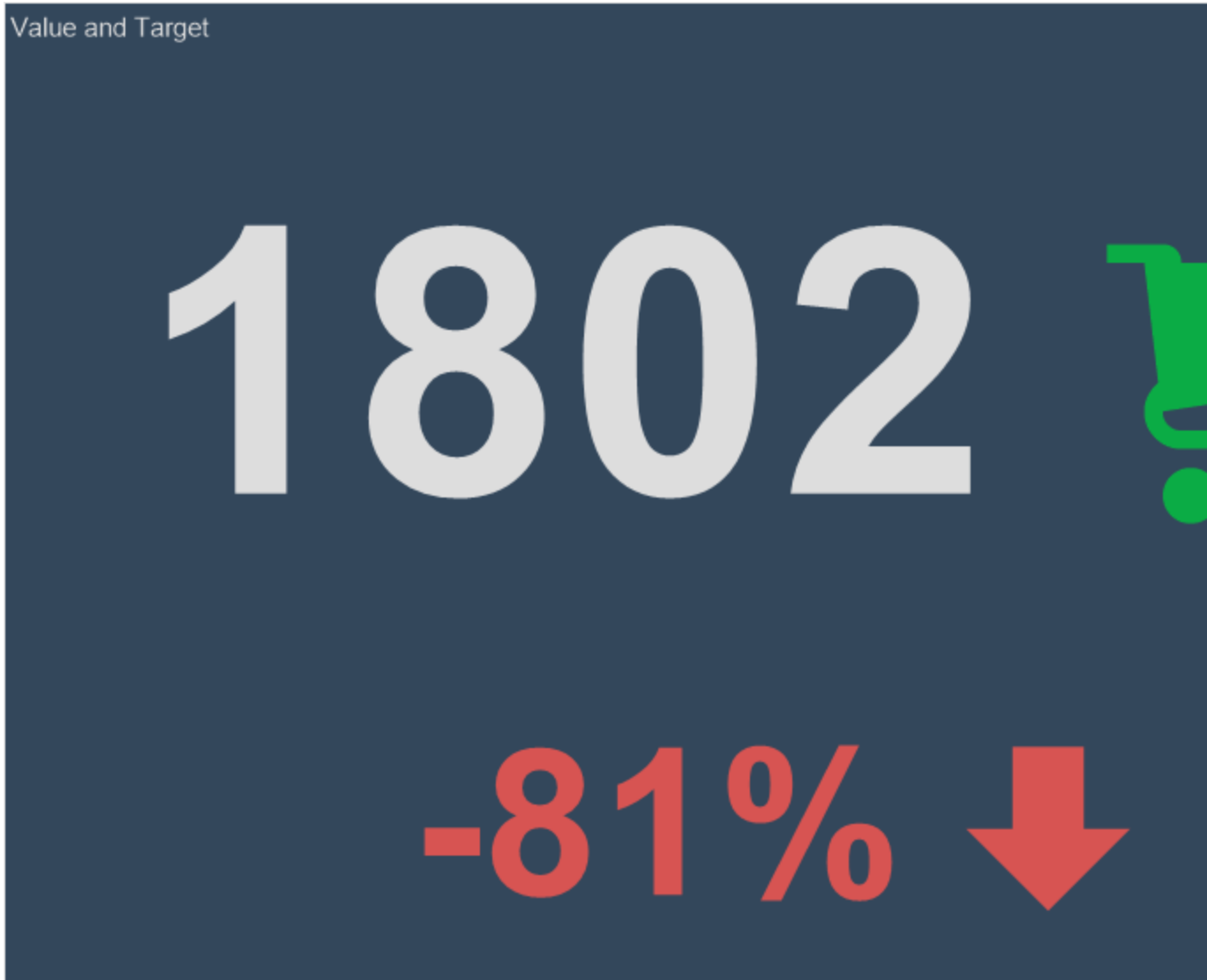
Information

If only the target value is specified in the **Indicator** element but the value of this indicator is not indicated, then the growth rate in the indicator will be -100 percent.

Indicator series

The indicator series is a separate indicator for a specific segment of values selected by a specific condition. The condition in this case will be the values of the data element that is specified in the **Series** field.

For example, in the **Value** field on the indicator, a data field with the number of orders issued is set, and in the Target field you set the planned number of orders. Without a series, only one indicator will be displayed. The indicator value will be the aggregated value of the data field specified in the Value field. All data field values form the **Target** field will also be aggregated. Based on the value and the target value, the indicator will be displayed with the growth rate.

If you specify a data field with a list of products in series, then the indicator will be displayed for every product, i.e. for each product, the number of orders issued and the rate of growth of orders for each product will be displayed.

Series by Product

Alice Mutton
Aniseed Syrup
Boston Crab Meat
Camembert Pierrot
Carnarvon Tigers
Chai
Chang
Chartreuse verte
Chef Anton's Cajun Seasoning
Chef Anton's Gumbo Mix
Chocolade
Côte de Blaye
Escargots de Bourgogne
Filo Mix
Fløtemysost
Geitost
Genen Shouyu
Gnocchi di nonna Alice
Gorgonzola Telino
Grandma's Boysenberry Spread
Gravad lax
Guaraná Fantástica
Gudbrandsdalsost
Gula Malacca
Gumbär Gummibärchen
Gustaf's Knäckebröd

If you specify a data field with a list of product categories in series, then an indicator will be displayed for each category, i.e. the indicator value and growth rate will be calculated by processing the values and growth rate of all products included into this category. In other words, the values and growth rate of each product will be grouped into categories to which they relate.

Series by Category

Beverages	167.7
Condiments	152.1
Confections	115.8
Dairy Products	117.9
Grains/Cereals	92.4
Meat/Poultry	49.5
Produce	30.0
Seafood	210.3

In order to set the series of the indicator, you should:
> Double-click the left mouse button on **Indicator**;
> Drag and drop the data column from the dictionary to the **Series** field.
> Create **New Field** in the **Series** field. Set the expression for this element, the processing of which will be the values of the series of the indicator.

Graphic element of value
When creating an indicator for a value, you can select a graphic element. To do this:
> Call the editor of the element;
> Click the **Browse** button in the **Icon** field, and select the icon in the drop-down list.
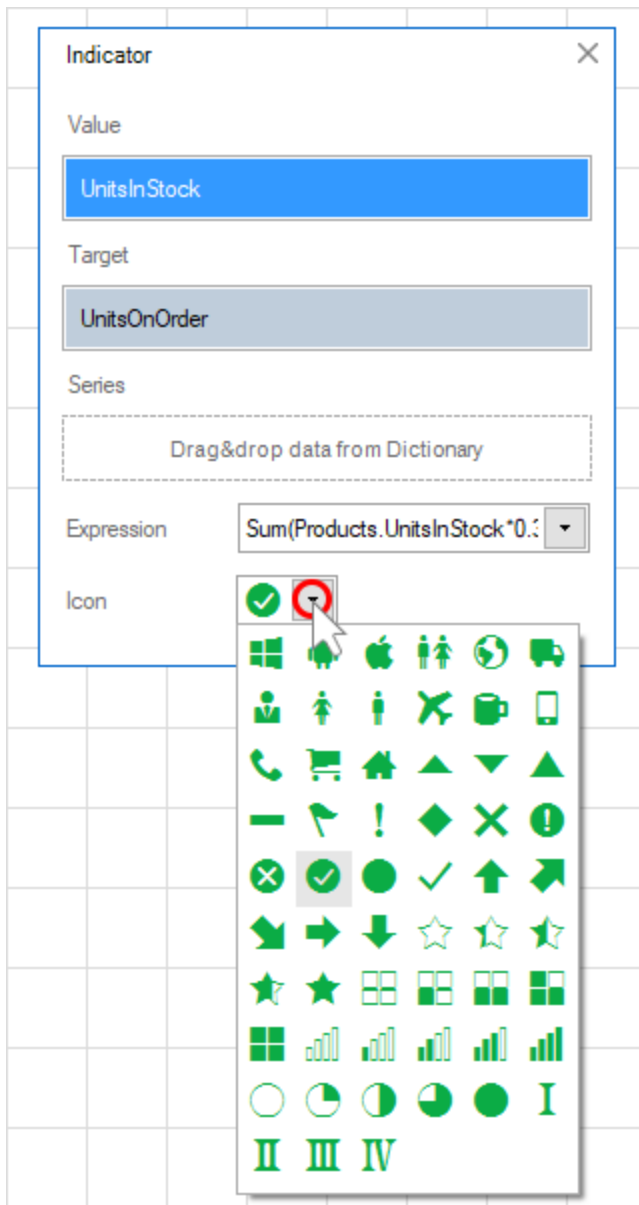
Information

If series are specified in the indicator, then icons for values are not used. In this case, there is no **Icon** field in the **Indicator** editor.

List of properties

The list shows the name and description of the properties of the element which you may find in the properties panel of the report designer.

Name	Description
Value	Sets a data field to set data value in the element. If the data field is set, this value will include two properties: ❯ **Expression** is used to change the expression of the **Value** data field of this element; ❯ **Label** is used to change the name of the data field in the **Value** field of the current item.
Target	Sets the target data field of the current item. If the data field is set, this value will include two properties: ❯ **Expression** is used to change the expression of the **Target** data field of this element; ❯ **Label** is used to change the name of the data field in the **Target** field of the current item.
Series	Sets the data field of the series of the current item. If the data field is set, this value will include two properties: ❯ **Expression** is used to change the expression of the **Series** data field of this element; ❯ **Label** is used to change the name of the data field in the **Series** field of the current item.
Data Transformation	Customizes the data tranformation of the current element.
Group	Adds the current item to a specific group of items.

Icon	Changes the value icon of the element.
Icon Set	Selects a set of icons for the values of the element.
Format	Sets the formatting values of the element.
Top N	Customizes the top values of the element.
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of the element - color, sides, size, and style.
Font	A group of properties defines the font family, its style, and size for the values of the element.
Fore Color	Specifies the color of the values of the element. By default, this property is set to **From Style**, i.e. the color of the values will be obtained from the settings of the current element style.
Glyph Color	Changes the color of the glyph.
Style	Selects a style for the current element. The default it is set to **Auto**, i.e. the style of this element is inherited from the style of the dashboard.
Enabled	Enables or disables the current item on the dashboard. If the

	property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Layout	Customizes the controls of the item in the viewer.
Margin	A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define padding (left, top, right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ Fore Color allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style ❯ The group property **Font** that allows you to define the font

	family, its style and size for the title of the current element.
	❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right.
	❯ The **Text** property is used to set the title text of the current element.
	❯ The **Visible** property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option

	enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property is set to **False**, then this element is not tied to the current location.

4.9 Progress

Progress is an element of the dashboard panel that represents the ability to display the growth rate (relative share) of a value relative to the target value.
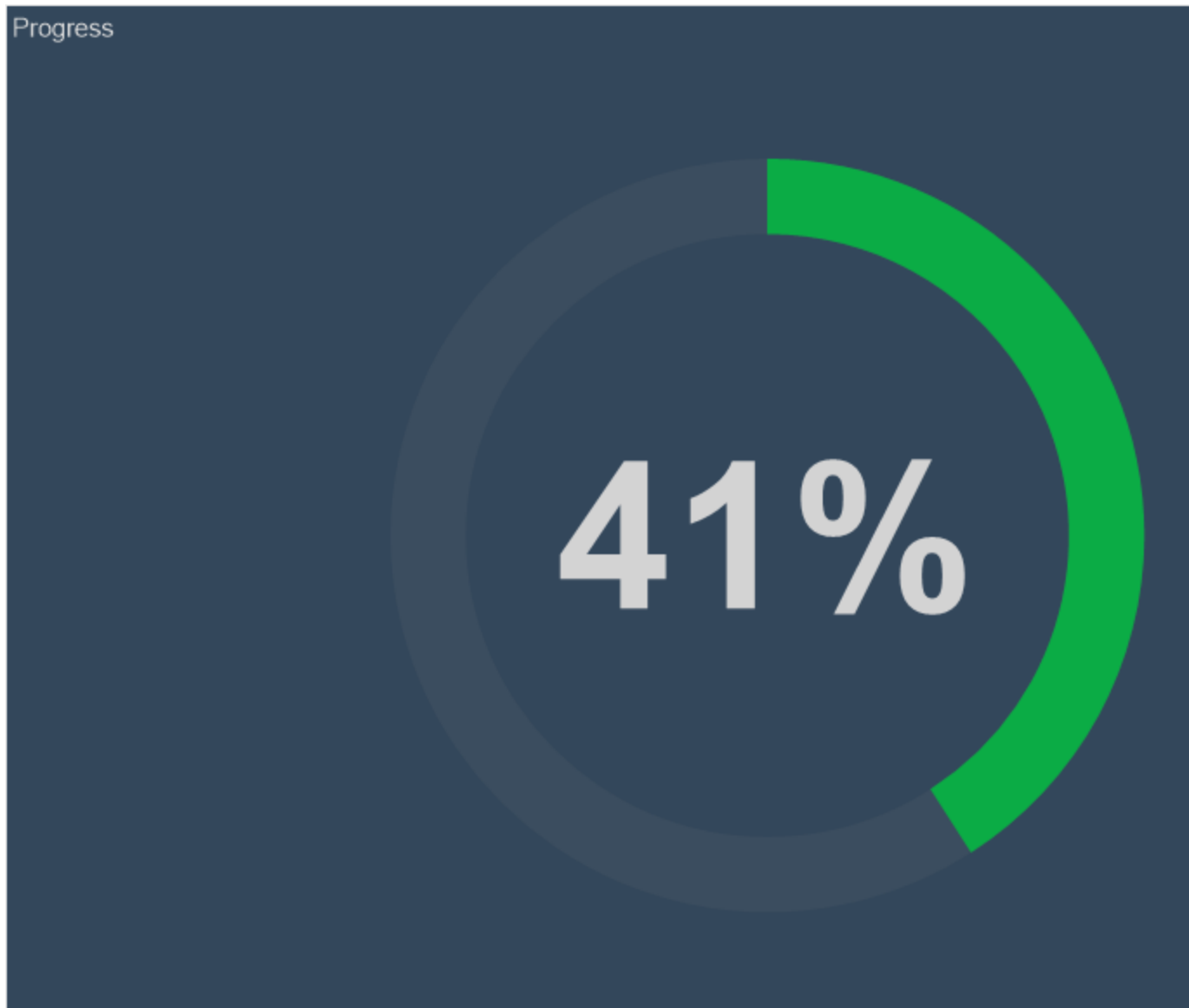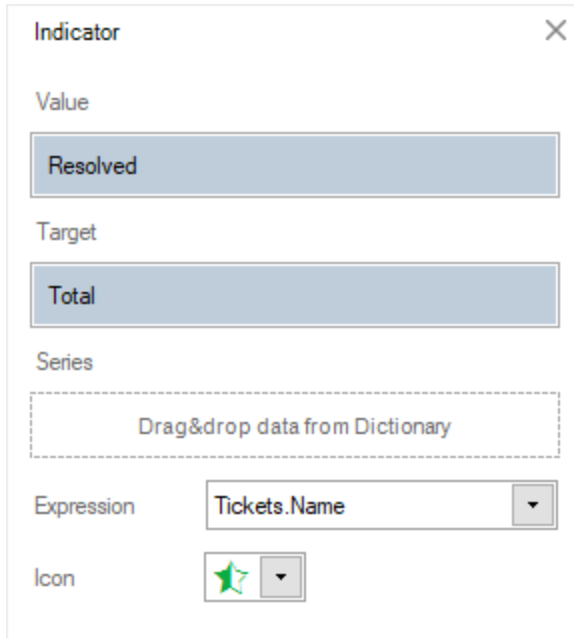
In order to display **Progress**, you need to add a data field to the **Value** and **Target** fields. In this case, using the graphical element, the growth rate of the value in relation to the target will be displayed. You can setup the **Progress** element in the editor. In order to call the editor, you should:

❯ Double-click on the **Progress** element in the dashboard panel;

❯ Select the **Progress** element and select the Design command in the context menu;

❯ Select the **Progress** element, and, on the property panel, click the **Browse** button of the **Value**, **Target**, and **Rows** properties.

Progress editor

In the **Progress** editor, you can add elements with data, edit the expressions of these elements, select the type of a graphic element to display the calculated value.

In the **Progress** editor you can:
> Specify the data field for the Progress value;
> Specify the data field for the target Progress value;
> Specify the data field for the Progress series;
> Select the type of the graphic element.

Progress values

In the **Value** field, you can specify only one data field. All values of this field will be aggregated, i.e. a function will be applied to them. By default, this is a summation function for numeric values. If a data field with non-numeric values is added, then, by default, the function of the number of rows in this data field is applied to them.

Information

Without a target value, the growth rate will always be 100 percent.

Target value of progress

In order to display the growth rate with the help of progress, besides the value in the progress it is necessary to indicate the **target** value. The target value is the aggregated value of the data field specified in the **Target** field of progress. Only one data field can

be specified in this field. By default, the summation function for numeric values is applied to the data field in the **Target** field. If a field with non-numeric values is added, then by default the function of counting the number of rows in this data field is applied to it.

Progress

97%
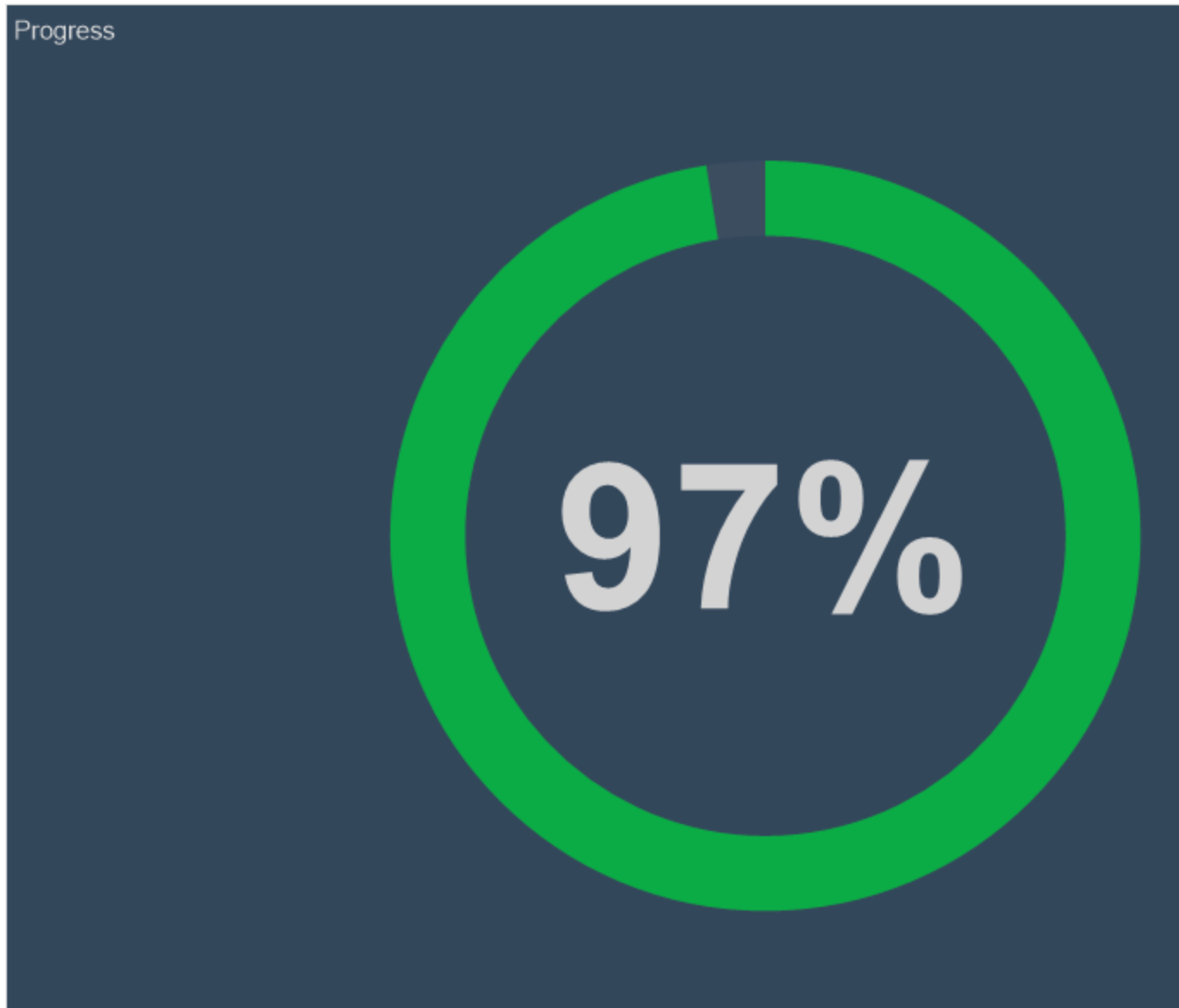
Information

If only a target value is specified in the **Progress** element, but no value is specified, then the growth rate in progress will be 0 percent.

Progress series

A series of progress is a separate progress for a specific segment of values selected by a certain condition. The condition in this case will be the values of the data field, which is specified in the **Series** field.

For example, in the **Progress** value field, a field with the number of orders issued is set, and the planned number of orders is set in the **Target** field. One progress will be displayed without specifying a series. The value of progress will be the growth rate (the value relative to the target value).
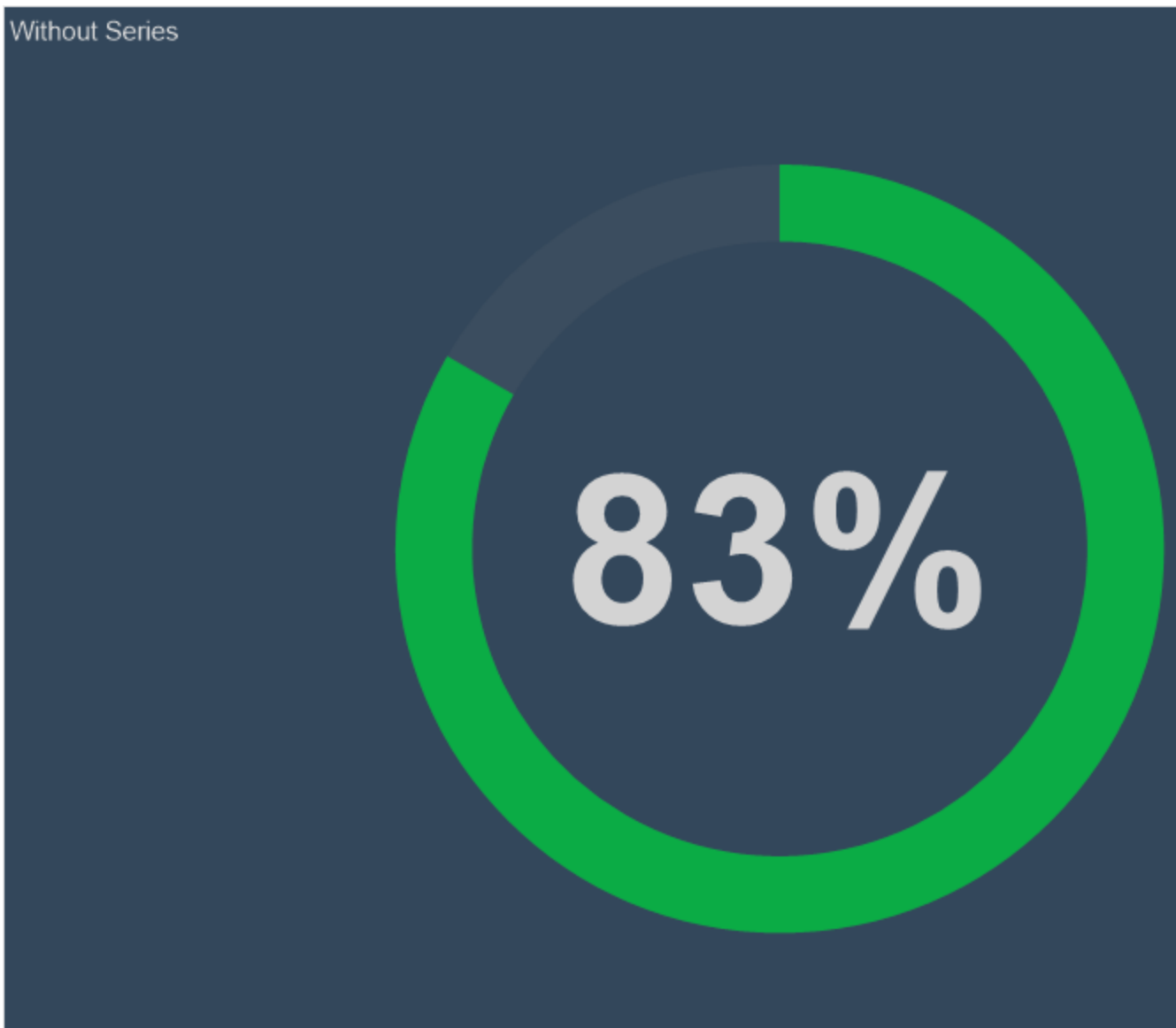
If you specify a data field with a list of products in series, then the progress will be displayed for every product, i.e. the growth rate will be displayed for every product.

If you specify a data field in the rows with a list of product categories, then progress will be displayed for every category, i.e. growth rate will be calculated by aggregating the growth rate of all products included into this category. In other words, the growth rate of every product will be grouped into the categories to which they relate.
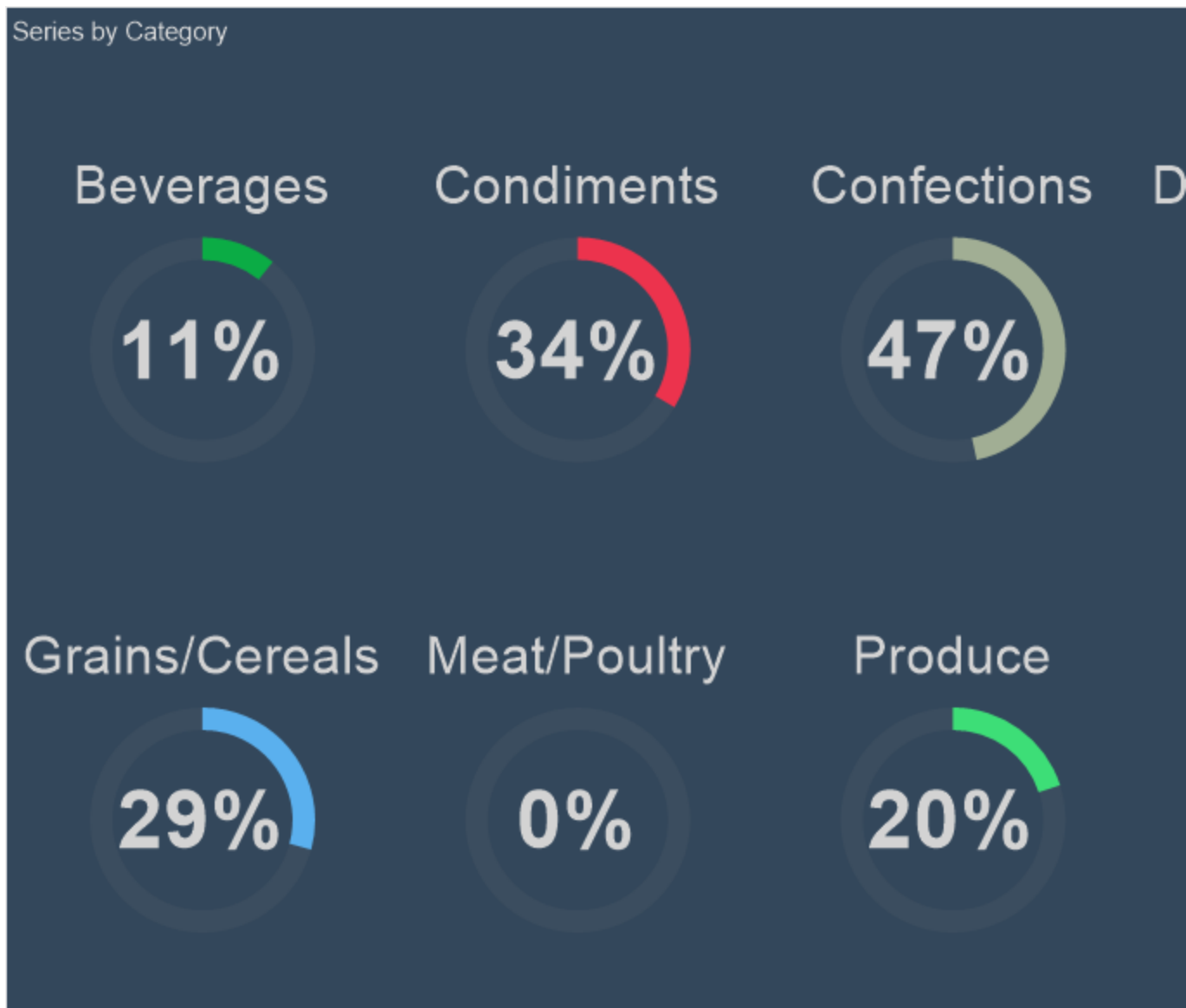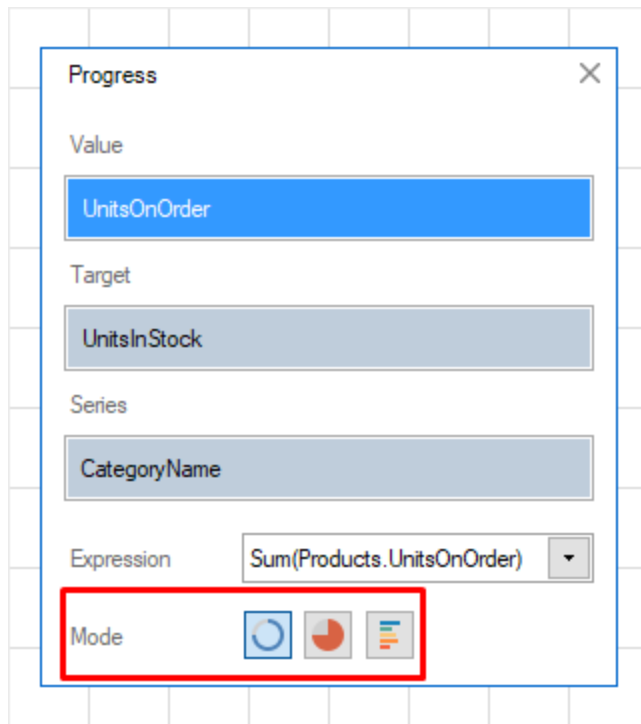
In order to set the series of progress, you should:

❯ Double-click the left mouse button on the **Progress** element;

❯ Drag and drop the data column from the dictionary to the **Series** field in the editor.

❯ Create **New Field** in the **Series** field. Set the expression for this data field, the processing result of which will be the values of the progress series.

Progress types
When creating the progress, you can select the type of graphic element with which the growth rate value will be displayed. To do this:

> Call the editor of the **Progress** element;
> Use the buttons to select the mode of the graphic element - Circle, Pie, Data Bars.

Below are the three elements of progress with different modes.
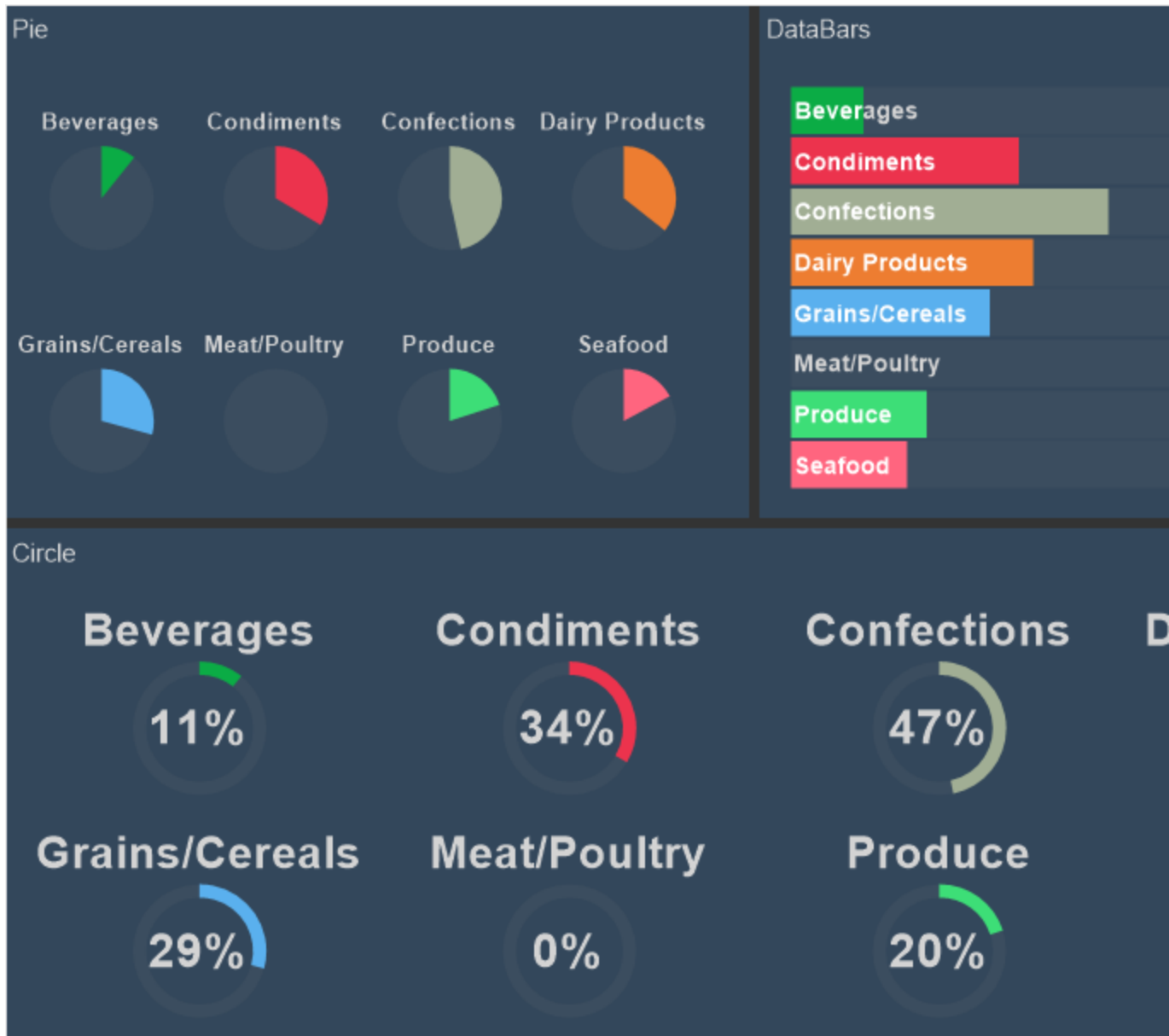
Information

Within one **Progress** element, you can select only one type of graphic element.

List of properties

The list shows the name and description of the properties of the element which you may find in the properties panel of the report designer.

Name	Description
Value	Sets a data field to set data value in the element. If the data field is set, this value will include two properties: ❯ **Expression** is used to change the expression of the **Value** data field of this element; ❯ **Label** is used to change the name of the data field in the **Value** field of the current item.
Target	Sets the target data field of the current item. If the data field is set, this value will include two properties: ❯ **Expression** is used to change the expression of the **Target** data field of this element; ❯ **Label** is used to change the name of the data field in the **Target** field of the current item.
Series	Sets the data field of the series of the current item. If the data field is set, the **Series** group will include two properties: ❯ **Expression** is used to change the expression of the **Series** field of this element; ❯ **Label** is used to change the name of the data field in the **Series** field of the current element.
Data Transformation	Customizes the data transformation of the current element.
Group	Adds the current item to a

	specific group of items.
Mode	Changes the type of the graphic element- DataBars, Circle, Pie.
Format	Sets the formatting values of the element.
Top N	Customizes the top values of the element.
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of the element - color, sides, size, and style.
Font	A group of properties defines the font family, its style, and size for the values of the element.
Fore Color	Specifies the color of the values of the element. By default, this property is set to **From Style**, i.e. the color of the values will be obtained from the settings of the current element style.
Series Color	Customizes the list of colors for the series of the element.
Style	Selects a style for the current element. The default it is set to **Auto**, i.e. the style of this element is inherited from the style of the dashboard.
Enabled	Enables or disables the current item on the dashboard. If the

	property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Layout	Customizes the controls of the item in the viewer.
Margin	A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define padding (left, top, right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ Fore Color allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style ❯ The group property **Font** that allows you to define the font

	family, its style and size for the title of the current element. ❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The **Visible** property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option

	enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property is set to **False**, then this element is not tied to the current location.

4.10 Maps

Map is an element of the dashboard, which provides the ability to display data with reference to geographic location.

Region Map

Online Map

When designing a dashboard for displaying maps, you can use the following elements:

❯ Region map, provides the ability to display any value with reference to a geographic object.

❯ Online map, provides the ability to display any object by geographic coordinates on an online Bing map.

4.10.1 Region Map

Region Map provides the ability to display any value with reference to a geographic object.
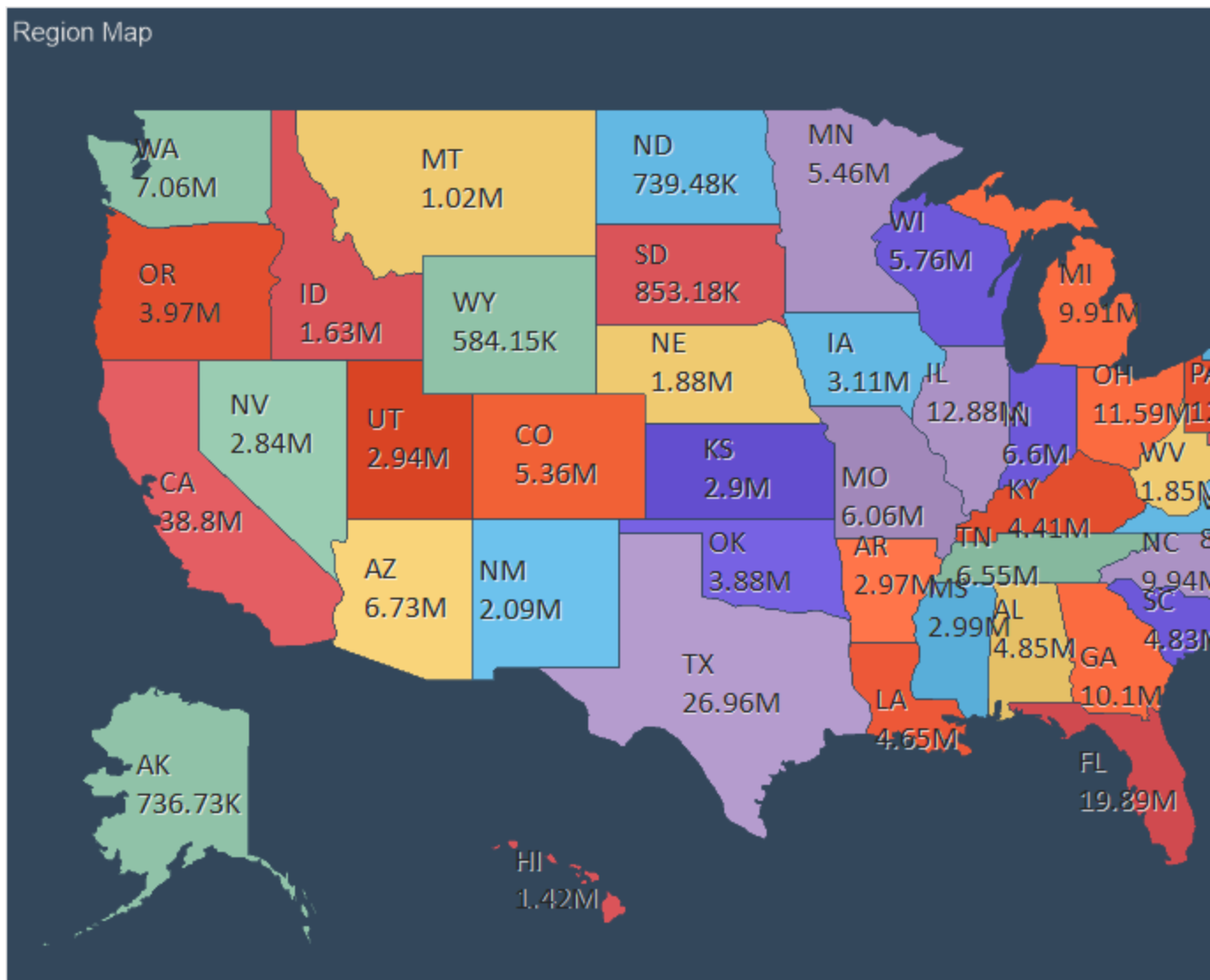
Region Map

The **Region Map** element can be placed anywhere on the dashboard. This item is configured in its editor. In order to call the editor, you should:

❯ Double-click on an item;

❯ Select the **Region Map** element, and select the **Design** command in the context menu;

To resize the **Region Map** element you should:

❯ Select an item on the dashboard panel;

❯ Increase or decrease the size of the element vertically, horizontally or diagonally.

Region Map editor

In the region map, you can display any value, with reference to a specific geographical object. The list of geographical objects depends on the selected map view.

Below is the editor of the Region Map element when manually filling in the data.

1 **Data from** parameter provides the ability to define a data source:
> **Manual** by setting a value for each map element;
> **Data Columns** by filling in the appropriate fields.

2 The table contains **Key**, **Name**, **Value** and **Color**. Also, if a map with grouping or heat map with grouping is selected, a column for the grouping keys of map elements will be present. By default, keys and map elements are filled. All that is needed is to enter a value for a specific map element, and specify the key of grouping, if necessary.

3 Parameter that provides the ability to select the type of a map. To change the type of

the map you should:
> Left-click on the field;
> Select the type of a map in the drop-down list.

4 The **Map Type** parameter is used to change the type of the **Region Map** element. The map may be of the following type:

> **Individual** - every **Map Key** is a separate geographical object. Each geographical object will have its own value.

> **Group** - by any condition, Map Keys will be combined into a group of geographical objects.

> **Heatmap** - every **Map Key** is a separate geographical object, and the values of all geographical objects of the map will also be analyzed. For a geographic object with a maximum value, the specific color will be defined, for a geographic object with a minimum value, another specific color will be defined. The color of other geographical objects will be obtained by mixing these colors.

> **Heatmap with Group** - by any condition, map keys will be combined into a group of geographical objects. After grouping of geographical objects, their values will be analyzed. In every group, the geographic object with the maximum value will have one color, and the geographic object with the minimum value will have another color. The color of the remaining geographical objects in the group will be obtained by mixing these colors.

5 The **Display Name Type** parameter allows you to select the display mode for the names of map elements:

> **None** - map names for every map element will not be displayed;
> **Full** - names for every map element will be displayed in full;

❯ **Short** - names for every map element will be abbreviated.

6 The **Show Values** parameter is used to display the values of map elements. If the box is checked, then its value will be displayed for every map element. If the box is not checked, the values of the map elements will not be displayed.

7 The **Color Each** parameter allows every element of the map to define its own color. This option is available only for an individual card. If the check box for the option is checked, then each map element will have a specific color; if the check box is not checked, all map elements will have one color.

Consider the setting of the Region Map editor, if the data will be obtained from the data fields. To do this, you should select the **Data Columns** value in the **Data from** parameter. Below is a map editor with data fields:

1 The **Data from** option is used to specify a data source:
> ❯ **Manually** - setup a value for every element of the map;
> ❯ From the **Data columns** by filling in the appropriate fields.

2 The **Key** field indicates a data field with a list of keys of map elements of a certain type.

3 The **Name** field indicates a data field with names for map elements of a certain type.

4 The **Value** field indicates a data field with values for every map element of a certain type.

5 The **Color** field indicates a data field that contains the color as **#FFFFF** for every map key.

Information

If the **Color** field is empty and an individual map type is selected, the **Color Each** option will be available in the editor. This option is used to automatically apply an individual color to every map element.

---

6 This parameter is used to select the type of a map. To change the type of the map you should:
> ❯ Left-click on the field;
> ❯ Select the type of a map from the drop-down list.

7 The **Expression** field displays the expression of the selected data item.

8 The **Map Type** parameter is used to change the type of the **Region Map**. There are several types of the map:
> ❯ **Individual** - every **Map Key** is a separate geographical object. Each geographical object will have its own value.
> ❯ **Group** - by any condition, **Map Keys** will be combined into a group of geographical objects.
> ❯ **Heatmap** - every **Map Key** is a separate geographical object, and the values of all geographical objects of the map will also be analyzed. The specific color will be defined for a geographic object with a maximum value; for a geographic object with a minimum value, another specific color will be defined. The color of other geographical objects will be obtained by mixing these colors.
> ❯ **Heatmap with Group** - by any condition, map keys will be combined into a group of geographical objects. After grouping of geographical objects, their values

will be analyzed. In every group, the geographic object with the maximum value will have one color, and the geographic object with the minimum value will have another color. The color of the remaining geographical objects in the group will be obtained by mixing these colors.

9 The **Display Name Type** option allows you to select the display mode for the name of the map elements:

> **No** - map names for every map element will not be displayed;
> **Complete** - names for every map element will be displayed in full;
> **Short** - names for every map element will be abbreviated.

10 The **Show Value** parameter is used to display the values of map elements. If the check box is checked, then its value will be displayed for every map element. If the box is not checked, the values of the map elements will not be displayed.

List of properties

The list shows the name and description of the properties of the element which you may find in the properties panel of the report designer.

Name	Description
Key	Sets a data field of keys in the element. If the data field is set, this value will include two properties: > **Expression** is used to change the expression of the **Key** data field of this element; > **Label** is used to change the name of the data field in the **Key** field of the current item.
Name	Sets a data field to set names in the element. If the data field is set, this value will include two properties: > **Expression** is used to change the expression of the **Name** data field of this element; > **Label** is used to change the

	name of the data field in the **Name** field of the current item.
Value	Sets a data field to set data value in the element. If the data field is set, this value will include two properties: ❯ **Expression** is used to change the expression of the **Value** data field of this element; ❯ **Label** is used to change the name of the data field in the **Value** field of the current item.
Group	Sets a data field to set grouping in the element. If the data field is set, this group will include two properties: ❯ **Expression** is used to change the expression of the **Group** data field of this element; ❯ **Label** is used to change the name of the data field in the **Group** field of the current item.
Color	Sets a data field with colors for the keys of the current item. If the data field is set, the **Color** group will include two properties: ❯ **Expression** is used to change the expression of the **Color** data field of this element; ❯ **Label** is used to change the name of the data field in the **Color** field of the current element.
Color Each	Enables or disables the use of individual colors for each key of a map.
Data Transformation	Customizes the data

	transformation of the current item.
Group	Adds the current item to a specific group of items.
Interaction	Sets interaction of the Region Map element.
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of the element - color, sides, size, and style.
Style	Selects a style for the current element. The default it is set to **Auto**, i.e. the style of this element is inherited from the style of the dashboard.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Layout	Customizes the controls of the item in the viewer.
Margin	A group of properties that is used to define margins (left, top, right, bottom) of the map area from

	the border of this element.
Padding	A group of properties that is used to define padding (left, top, right, bottom) of the map area from the border of this element.
Title	A group of properties that allows you to customize the title of the element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ Fore Color allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style ❯ The group property **Font** that allows you to define the font family, its style and size for the title of the current element. ❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The Visible property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the

	element title will be included. If this property is set to **False**, then the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then

	the current item is bound to the current location. If this property is set to **False**, then this element is not tied to the current location.

4.10.2 Online Map

Online Map is used to display any object by geographic coordinates on the online map from Bing.
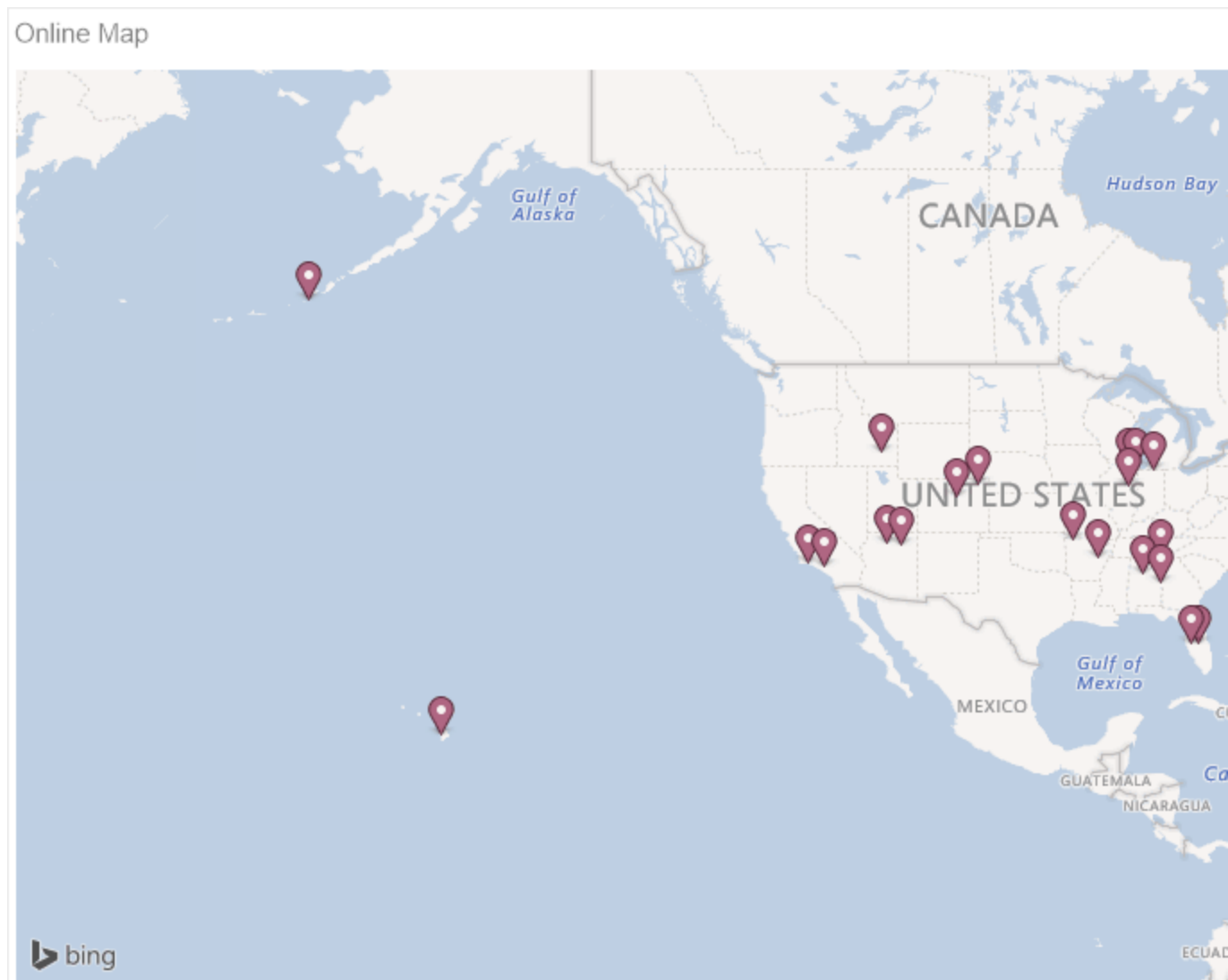
The **Online Map** element can be placed anywhere on the dashboard. This item is configured in the element editor. In order to call the editor, you should:
> Double-click on an item;

❯ Select **Online Map**, and select the **Design** command in the context menu;

❯ Select **Online Map**, and, on the property panel, click the **Browse** button of the **Latitude** or **Longitude** properties;

In order to resize an item on the **Online Map**, you should:

❯ Select an item on the dashboard;

❯ Increase or decrease the size of the element vertically, horizontally or diagonally.

Online map editor

Online map is used to display any object by geographic coordinates and works only with data elements.
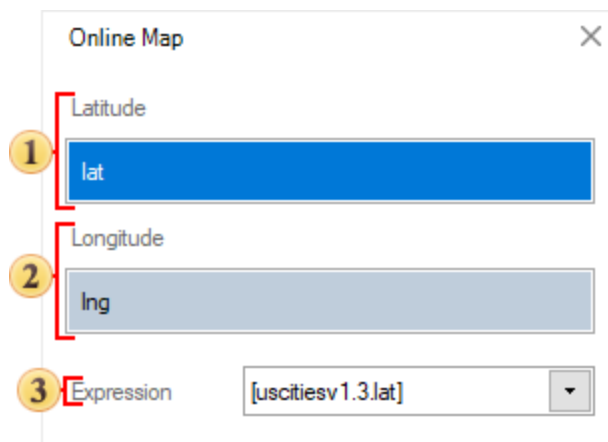
❶ The **Latitude** field indicates the data field with the latitude value of the geographical object.

❷ The **Longitude** field indicates the data field with the longitude value of the geographical object.

❸ The **Expression** field displays the expression of the selected data field.

List of properties

The list shows the name and description of the properties of the element which you may find in the properties panel of the report designer.

Name	Description
Latitude	Specifies a data field with the latitude of the geographical objects of the Online Map

	element. If the data field is set, the Latitude group will include two properties: ❯ **Expression** is used to change the expression of the **Latitude** field of this element; ❯ **Label** is used to change the name of the data field in the **Latitude** field of the current element.
Longitude	Specifies a data field with the longitude of the geographical objects of the Online Map element. If the data field is set, the Longitude group will include two properties: ❯ **Expression** is used to change the expression of the **Longitude** field of this element; ❯ **Label** is used to change the name of the data field in the **Longitude** field of the current element.
Data Transformation	Customizes the data transformation of the current item.
Group	Adds the current item to a specific group of items.
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of the element - color, sides, size,

	and style.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Layout	Customizes the controls of the item in the viewer.
Margin	A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ Fore Color allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title

	will be obtained from the settings of the current element style ❯ The group property **Font** that allows you to define the font family, its style and size for the title of the current element. ❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The Visible property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element.

	❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property is set to **False**, then this element is not tied to the current location.

4.11 Image

Image is an element with which you can display various graphical objects (photo, logo, picture, etc.) on the dashboard. The Image element supports the following types of graphics - BMP, PNG, JPEG, TIFF, GIF, PNG, ICO, EMF, WMF, SVG.
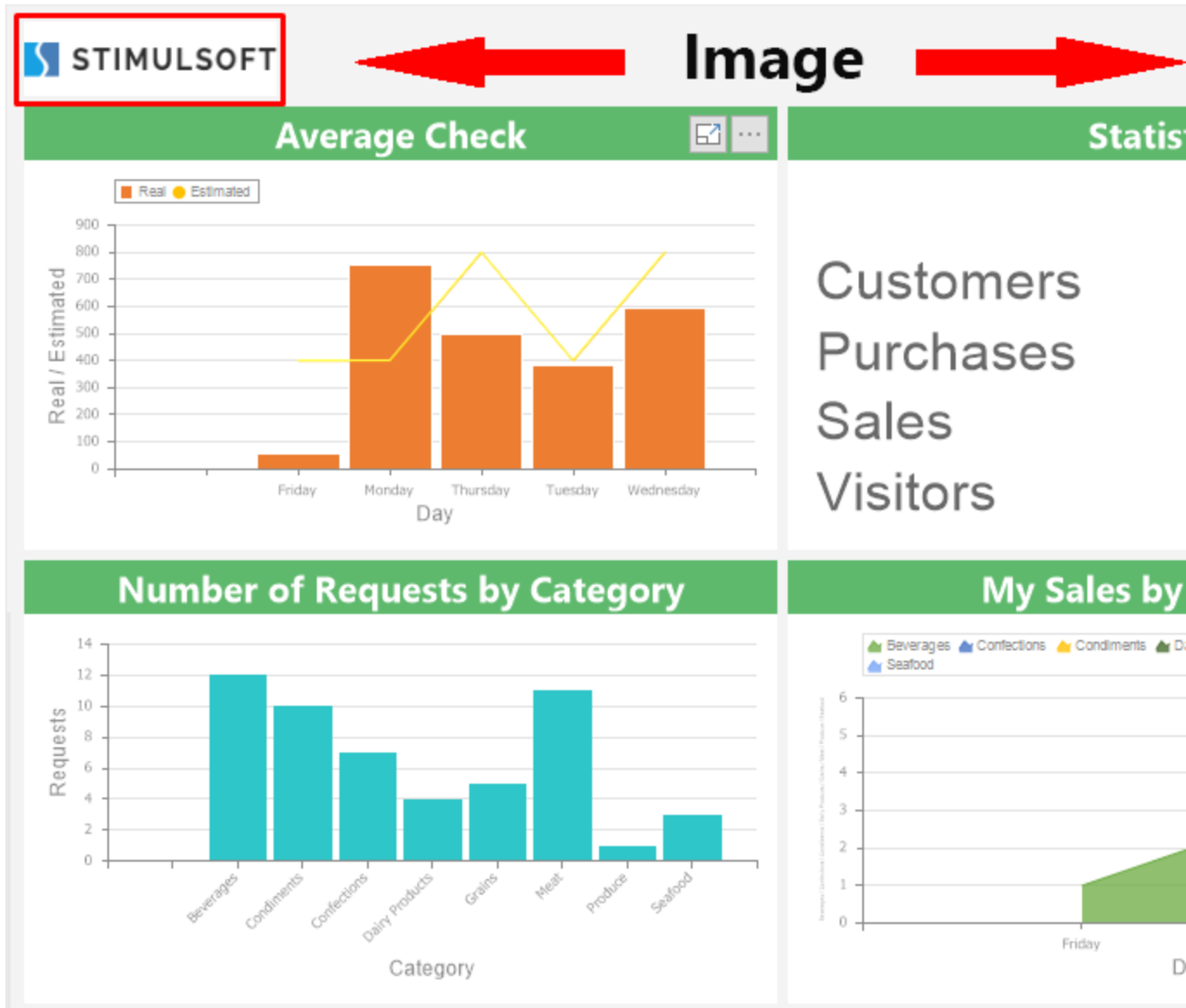
The image can be placed anywhere on the dashboard. Setting up the source for the image element is carried out in its editor. In order to call the editor, you should:

❯ Double-click on the Image element;

❯ Select the **Image** item, and select the **Design** command in the context menu;

❯ Select the **Image** item, and, on the property panel, click the **Browse** button of the Image, **Hyperlink** properties of the image.

In order to resize an image element you should:

❯ Select it in the dashboard;

❯ Increase or decrease the size of the element vertically, horizontally or diagonally.

Image editor

In the editor you can indicate the source of the image for the current element. Within one element, you can display only one graphic object (picture, logo, photo, etc.).
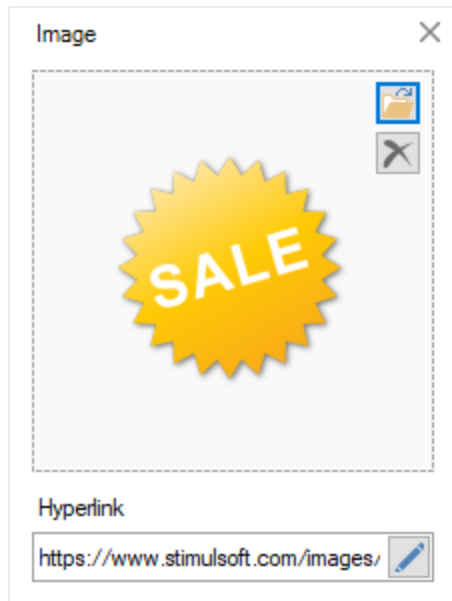
> In the **Image** field you can upload an image from the local storage.
> In the **Hyperlink** field, the link to the graphic object is indicated. This can be either a URL or a link to a report resource (resource://logo).

Information

Since only one graphic object can be displayed in one element, an image can have only one source. It follows that if the graphic object is directly loaded in the **Image** element, the receipt hyperlink will not work. To do this, delete the graphic object from the **Image** field.

Element settings

Any graphic object added to the element is stretched to the entire area of the element, with the exception of the specified margins and padding. The setting of the graphic object in the element is carried out using buttons on the ribbon panel or using properties on the property panel. All these properties are located in the Image

additional group:

❯ The **Aspect Ratio** property. When stretching an image, its proportions may be broken. In order to stretch the Image element while maintaining the proportions of the graphic object, you should set the Aspect Ratio property to **true**.

❯ The **Horizontal Alignment** property is relevant if the **Aspect Ratio** property is set to **true**. Horizontal alignment of the graphic object within the Image element. You can also specify the horizontal alignment using the buttons on the Ribbon panel.

❯ The **Vertical Alignment** property is relevant if the **Aspect Ratio** property is set to **true**. Vertical alignment of the graphic object within the Image element. You can also specify the vertical alignment using the buttons on the Ribbon panel.

List of properties

The list shows the name and description of the properties of the element which you may find in the properties panel of the report designer.

Name	Description
Image	Uploads a picture in the element. When you click the Browse button, the element editor will be opened in which you can add or delete an image or a hyperlink to the image.
Hyperlink	Sets a hyperlink in the element. When you click the Browse button, the element editor will be opened in which you can add or delete an image or a hyperlink to the image.
Ratio	Provides the option of the aspect ratio of the image in the current element. If the property is set to **True**, then the aspect ratio of the image in the current element will be saved. If this property is set to **False**, then the aspect ratio will not be taken into account and the image will not stretch

	proportionally.
Horizontal alignment	Changes the horizontal alignment of the image in the current element.
Vertical alignment	Changes the vertical alignment of the image in the current element.
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of the element - color, sides, size, and style.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Margin	A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the

	Table element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ Fore Color allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style ❯ The group property **Font** that allows you to define the font family, its style and size for the title of the current element. ❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The Visible property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current

	item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property is set to **False**, then this element is not tied to the current location.

4.12 Panel

The **Panel** is an element of the dashboard on which other elements can be placed, including other panels and dashboards.
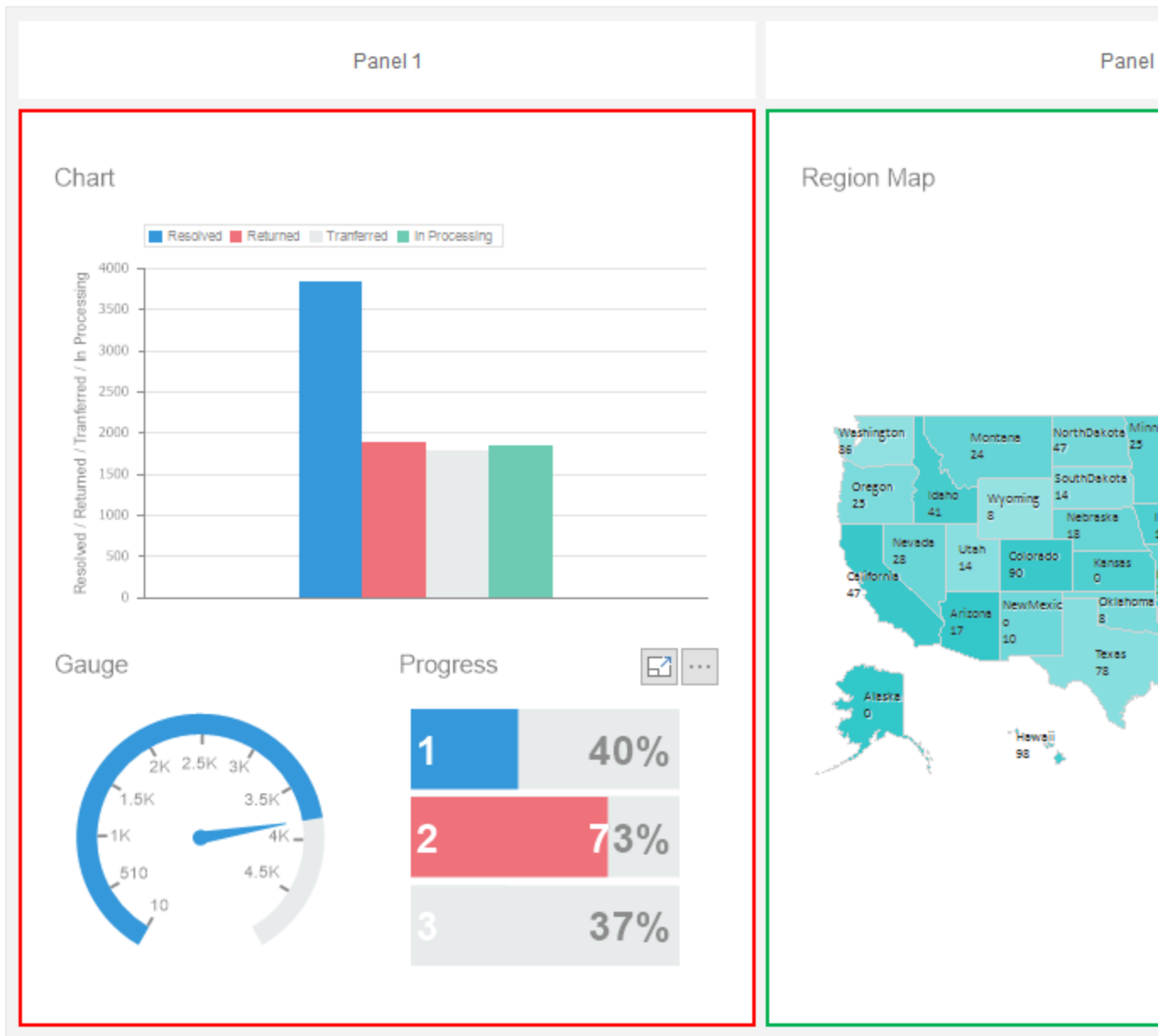
The **Panel** element is stretched along with the dashboard by height and width. To resize the **Panel** element you should:

➤ Select a dashboard element;

➤ Increase or decrease the size of the element vertically, horizontally or diagonally.

List of properties

The list shows the name and description of the properties of the element which you

may find in the properties panel of the report designer.

Name	Description
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of the element - color, sides, size, and style.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Margin	A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to

	use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property is set to **False**, then this element is not tied to the current location.

4.13 Text

Text is an element that allows you to display any text or the result of a specified expression on the dashboard. An expression can be a reference to a system variable or a data column.
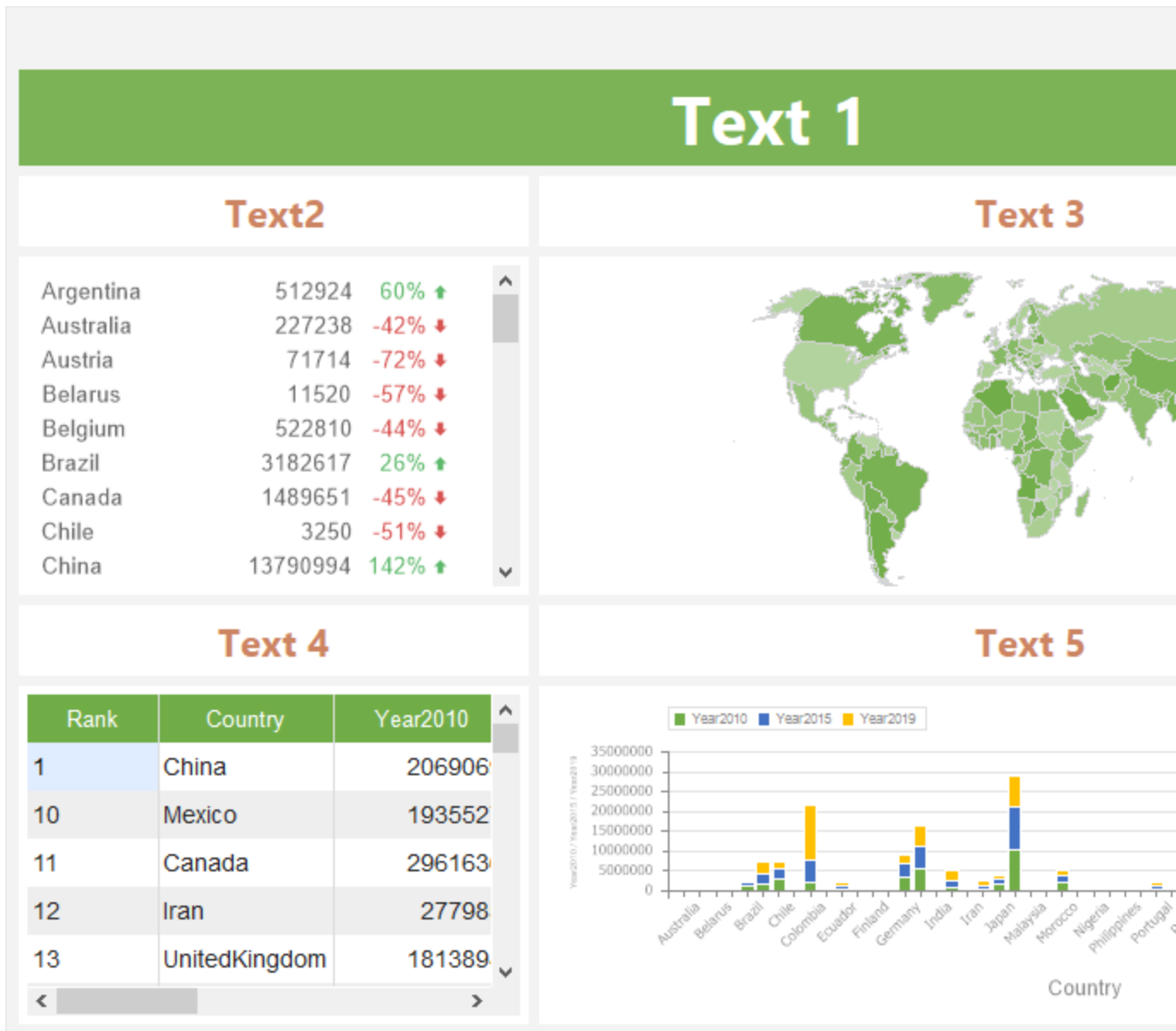
Information

If the text element contains a reference to a system variable, then the result that appears in this element will be the value of the system variable. For example, if a reference to the **Today** system variable is specified, the result will be the current date and time of the operating system. If the text element expression is a reference to a data column, then the result that is displayed using this element will be the current value of this data column.

The **Text** element can be placed anywhere on the dashboard. The text element is configured in its editor. In order to invoke the **Text** element editor, you should:

❯ Double-click on this item;

❯ Select the **Text** element and select the **Edit** command in the context menu;

❯ Select the **Text** element, and, on the property panel, click the **Browse** button of the **Text** property.

In order to resize the text element you should:

❯ Select it on the dashboard;

❯ Increase or decrease the size of the element vertically, horizontally or diagonally.

Text editor

In the **Text** editor you can change the content of this element. You can customize the design of the text in the editor or using the control buttons on the Ribbon panel.
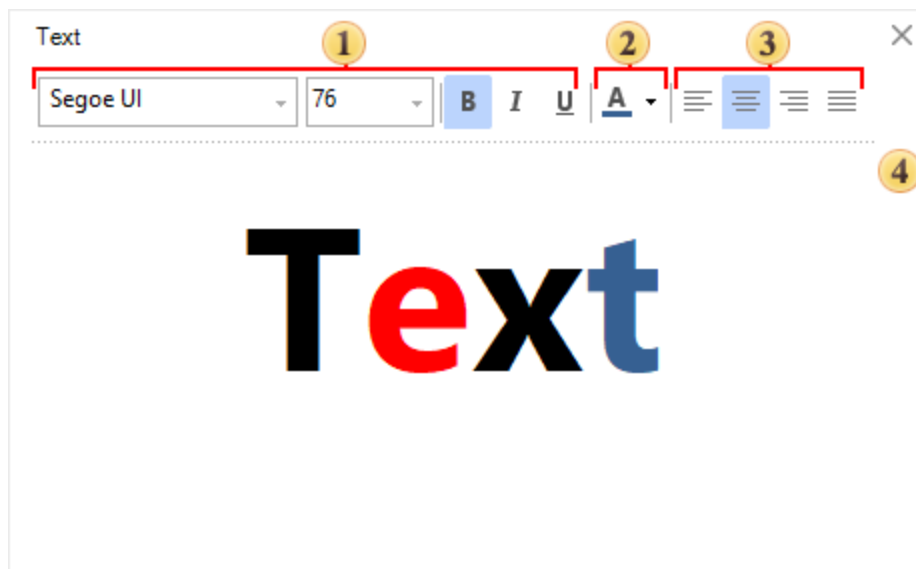
1 The group of commands for managing text fonts - font family, font size, font type.

2 Sets the color of the text or its characters. Each character of the text can select its own color. To do this, select the character in the field and select a color from the palette or enter a color value in the RGBA format.

3 Commands for horizontal text alignment in the Text element area - left, center, right, width.

4 Text input field.

Information

Similar commands to work with text ① - ③, are located on the **Home** tab of the Ribbon panel in the report designer. Select the **Text** element and change its font settings, text color, text alignment. In addition, on the Ribbon panel on the Home tab, you can align text vertically - top, bottom, center.

Also, the text element can change the **Back Color** and the borders of the element. In more detail can be found in the chapter Appearance.

List of properties

The list shows the name and description of the properties of the **Text** element which you may find in the properties panel of the report designer.

Name	Description
Text	Specifies text in a Table element. When you click the **Browse** button, the editor of the element will be opened, in which you can add or delete text, as well as customize its appearance.
Back Color	Changes the background color of the element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of a table - color, sides, size, and style.
Fore Color	Specifies the color of the values of the element. By default, this property is set to **From Style**, i.e.

	the color of the values will be obtained from the settings of the current element style.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Margin	A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the element: ❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element. ❯ **Fore Color** allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings

	of the current element style ❯ The group property **Font** allows you to define the font family, its style and size for the title of the current element. ❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right. ❯ The **Text** property is used to set the title text of the current element. ❯ The **Visible** property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option

	enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property is set to **False**, then this element is not tied to the current location.

4.14 Shape

Shape is an element with the help of which various shapes can be displayed on the dashboard.
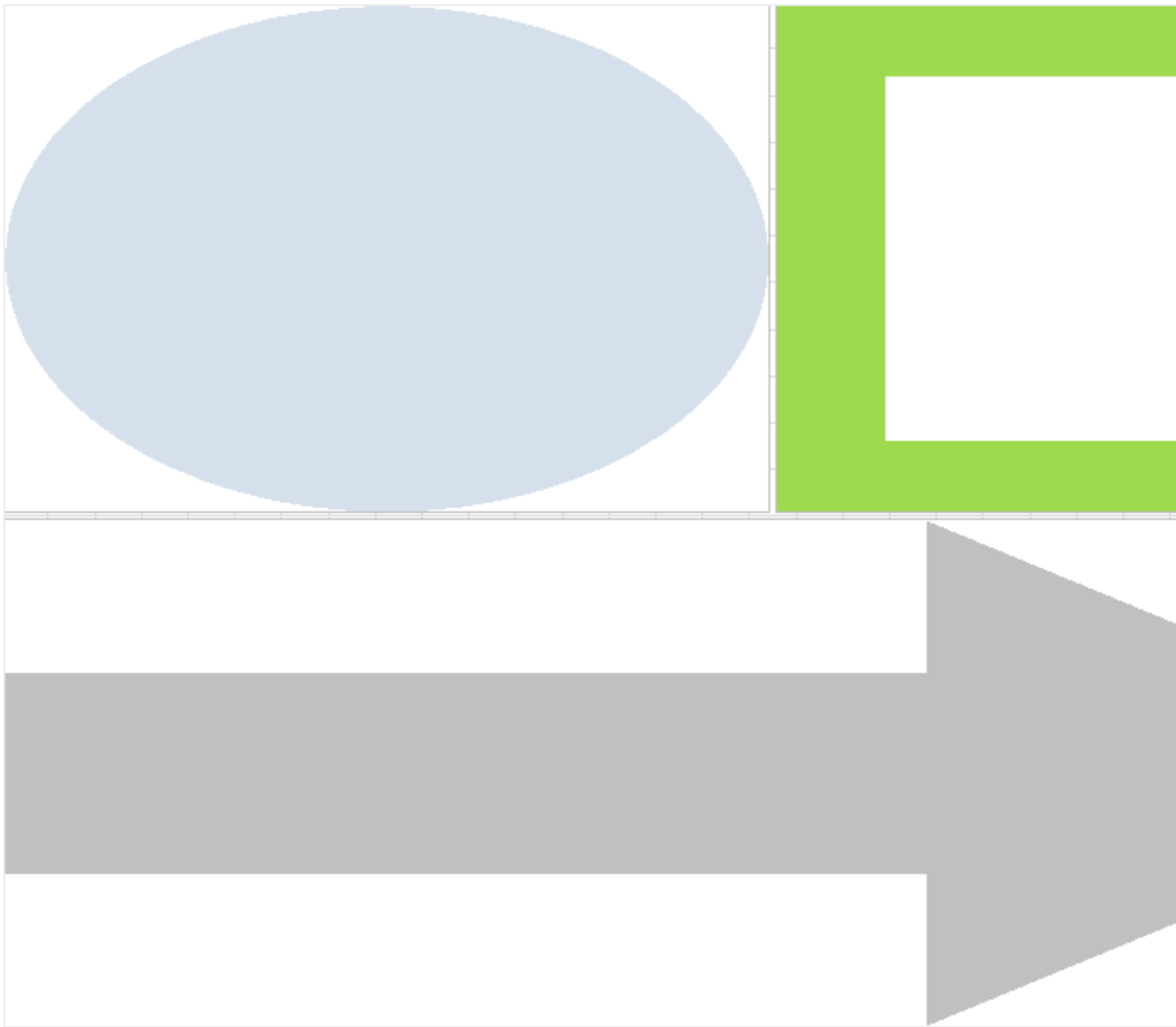
The **Shape** element can be placed anywhere on the dashboard. The setting of the **Shape** element is carried out in the shape editor. In order to invoke the editor of this element, you should:

> Double-click on a **Shape**;

> Select the Shape element, and select the **Design** command in the context menu;

In order to resize the Shape element you should:

> Select it in the dashboard;

> Increase or decrease the size of the element vertically, horizontally or diagonally.

Shape editor
Shape settings can be found in the Shape editor.
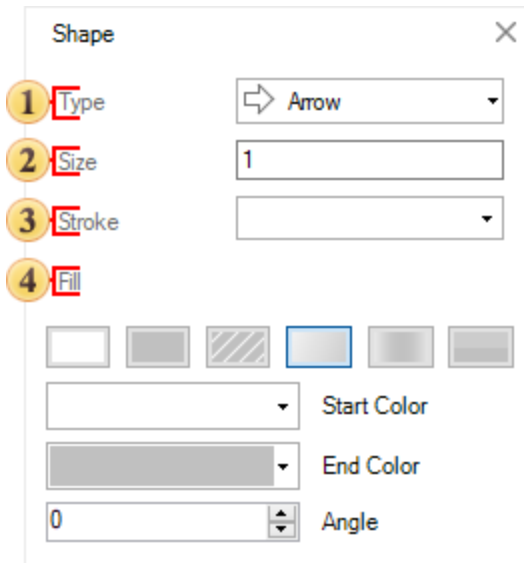
① **Type** - determines the type of shapes. Click on the value field and select a shape from the drop-down list.

② **Size** - changes the size of the stroke of shapes.

③ **Stroke** - changes the stroke color of shapes. Click on the value field and select a shape from the drop-down list.

④ **Fill** - changes the fill type of shapes and, depending on the type selected, override the colors, angle, scale, focus, blending, and hatching. For example, for the Solid type, only one color can be defined. If the **Gradient fill** type is selected, then the starting and ending colors must be selected. You can also change the gradient angle.

Information

All these parameters are represented as identical properties on the property panel. You can configure settings by selecting the **Shape** element and changing the values of these properties in the properties panel.

List of Shape properties
The list shows the name and description of the properties of the **Shape** element which you may find in the properties panel of the report designer.

Name	Description
Fill	A property group that is used to change the brush and fill color of a shape in the current element.
Shape Type	Changes the type of a shape in the current element.
Size	Changes the stroke width of a shape for the current element.
Stroke	Changes the stroke color of a shape.
Back Color	Changes the background color of the Shape element. By default, this property is set to **From Style**, i.e. the color of the element will be obtained from the settings of the current element style.
Border	A group of properties that allows you to customize the borders of a table - color, sides, size, and style.
Enabled	Enables or disables the current item on the dashboard. If the property is set to **True**, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to **False**, this element is disabled and will not be displayed when previewing the dashboard in the viewer.
Margin	A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.
Padding	A group of properties that allows

	you to define indents (left, top, right, bottom) of the columns from the range of values.
Title	A group of properties that allows you to customize the title of the element:
	❯ The **Back Color** property provides the ability to change the background color of the title of the current item. By default, this property is set to **From Style**, i.e. the background color will be obtained from the style settings of the current element.
	❯ **Fore Color** allows you to change the text color of the title of the current item. By default, this property is set to **From Style**, i.e. the text color of the title will be obtained from the settings of the current element style
	❯ The group property **Font** allows you to define the font family, its style and size for the title of the current element.
	❯ The **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right.
	❯ The **Text** property is used to set the title text of the current element.
	❯ The **Visible** property is used to enable or disable displaying of the title of the current item. If the property is set to **True**, then the element title will be included. If this property is set to **False**, then

	the element header will be disabled.
Name	Changes the name of the current element.
Alias	Changes the alias of the current item.
Restrictions	Configures the permissions to use the current item in the dashboard: ❯ The **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed. ❯ The **Allow Delete** option enables or disables the deletion of an element. ❯ The **Allow Move** option allows or prohibits moving an element. ❯ The **Allow Resize** option enables or disables resizing of an element. ❯ The **Allow Select** option enables or disables the element selection.
Locked	Locks or unlocks resizing and movement of the current element. If the property is set to **True**, the current element cannot be moved or resized. If this property is set to **False**, then this element can be moved and resized.
Linked	Binds the current location to the dashboard or another element. If the property is set to **True**, then the current item is bound to the current location. If this property

	is set to **False**, then this element is not tied to the current location.

4.15   Interaction

When viewing the dashboard, its elements may have certain interactive features. In other words, depending on the impact of users on the elements of the dashboard, various actions can be carried out.
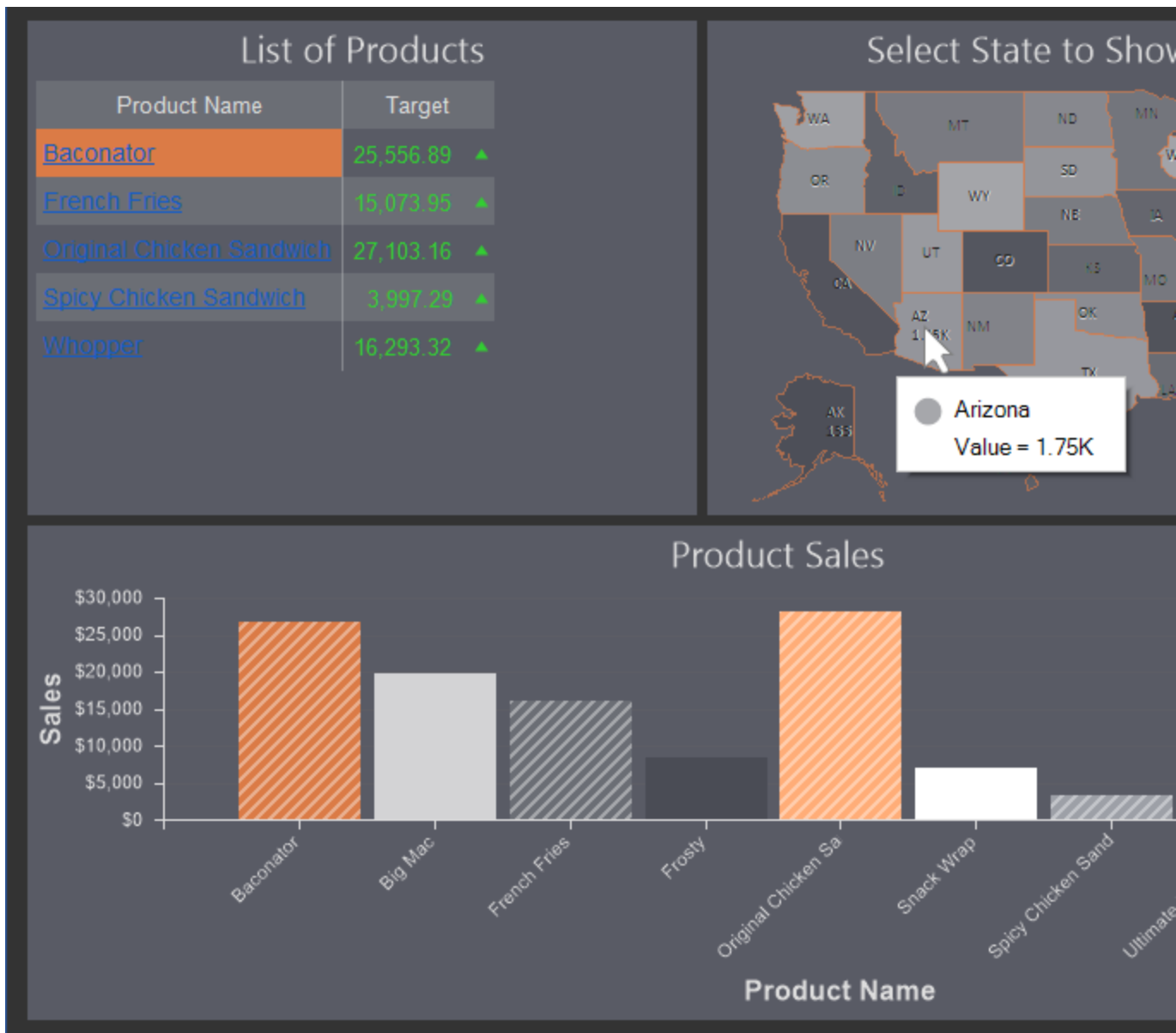
The user actions for viewing the dashboard include:

❯ Cursor hover over the value of the dashboard element;

❯ Single click with the left button of the mouse or touch the element value of the dashboard.

The interactive actions of the dashboard panel include:

❯ Displaying additional information on the values of the dashboard element as a **Tool Tip**. It can occur only when the user hovers the cursor.

❯ Filtering data. It can occur only when a user clicks on an element.

❯ Following the hyperlink. It can occur only when a users clicks on an element.

You can specify interactive element actions in the **Interaction** editor. In order to call the editor you should:

❯ Select an element on the dashboard in the report designer;

❯ Click the **Browse** button for the **Interaction** property on the property panel.

Currently, interactive actions are available for the following items:

❯ Chart;

❯ Regional map;

Information

Configuring the interaction of the Table will be described below.

Interaction editor

In this editor, every action is represented as a separate group of parameters with which you can customize an interactive action.
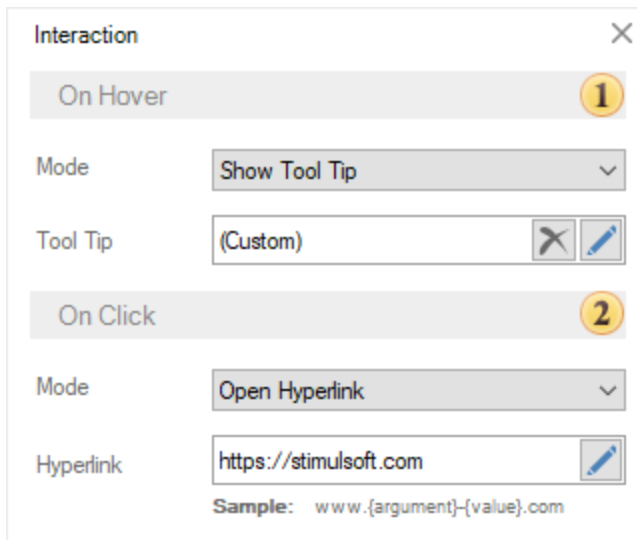
❶ The **On Hover** group of parameters is used to define settings for an interactive action when hovering the cursor over the value of a dashboard element. The **Mode** parameter allows you to select the type of the interactive action:

▶ **None** - when you hover the cursor over the value of the dashboard element, no action will occur

▶ **Show Tool Top** - when the user hovers the cursor over the value of the element of the dashboard, certain information will be displayed. Also, if you select this type of action, the **Tool Tip** option will be available. You can create and design a custom tool tip. By default, a standard tool tip is used for every element.

▶ **Show Hyperlink** - when the user hovers the cursor over the value of the element, the hyperlink specified in the **On Click** group will be displayed.

❷ The **On Click** group of parameters allows you to define an interactive action when you click the left button of the mouse or touch the value of a dashboard element. The **Mode** parameter allows you to select the type of interactive action:

▶ **None** - when clicking on the value of the dashboard element, no action will occur;

▶ **Apply Filter** - when you click on the value of the dashboard item, the dashboard data will be filtered through the interrelation of its elements.

▶ **Open Hyperlink** - when you click on the value of the dashboard item, the follow by the hyperlink will occur. Also, when this type of action is selected, the **Hyperlink** parameter will be displayed, in the value field of which you should specify a hyperlink.

Table element interaction

In the **Table** element, you can configure interaction both for each column of this table, and for the entire element. You can setup interactive actions in the **Interaction** editor.

For a column of the **Table** element you can:
> Specify a hyperlink for all values of this column;
> Display a tool tip or a hyperlink when you hover over a column value;
> Filter by value by a single click.

For the entire **Table** element, you can:
> Enable or disable sorting in value column titles;
> Enable or disable data filtering commands in value column titles.

To invoke the **Interaction** editor for a column of values, you should:
> Select the data field in the **Table** element editor;
> Click the **Edit** button of the **Interaction** parameter.
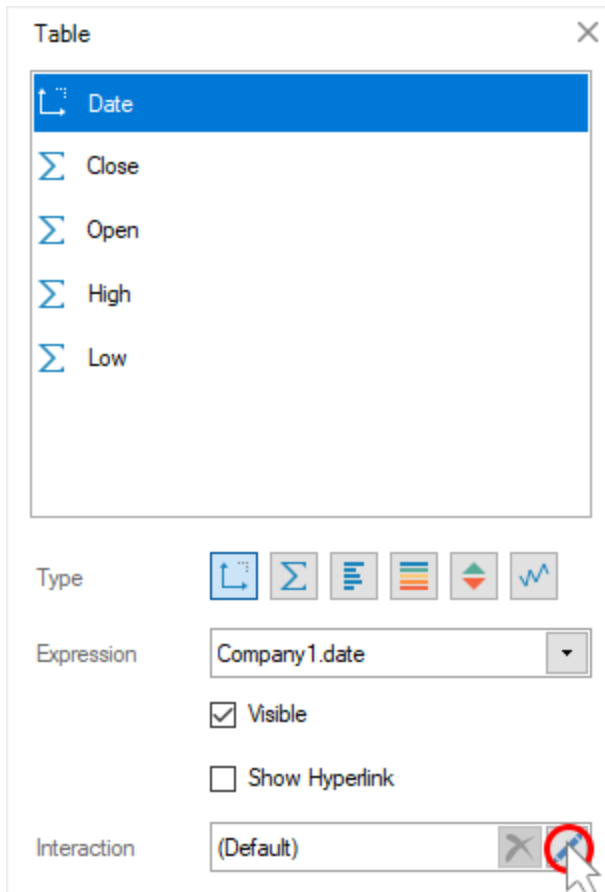
To invoke the **Interaction** editor of the **Table** element you should:
> Select an item on the dashboard panel;
> Click the **Browse** button on the **Interaction** property in the property panel.

The Interaction editor of the Table element
It contains options to enable or disable commands to control sorting and filtering data in the titles of value columns.
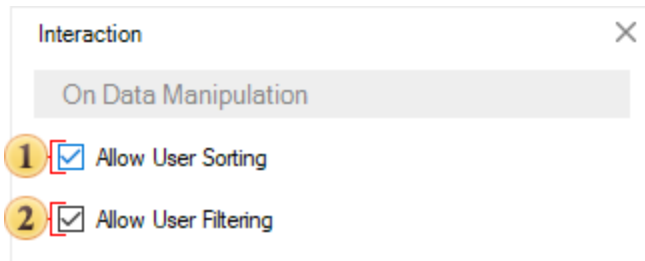
1 The **Allow User Sorting** parameter allows you to enable or disable displaying sorting commands in the value column title menu. If the check box is checked for this parameter, then sorting commands will be displayed in the title menu.

2 The **Allow User Filtering** parameter allows you to enable or disable displaying filtering commands in the value column title menu. If the check box is checked for this parameter, then sorting commands will be displayed in the title menu.

Information

You should know that, if the check boxes are cleared for the **Allow User Sorting** and **Allow User Filtering** parameters, the title menu will be disabled for value columns.

The Interaction editor for Table data fields
You can configure Interactive actions for every data field. To do this, select the data field and click the **Edit** button in the **Table** editor.
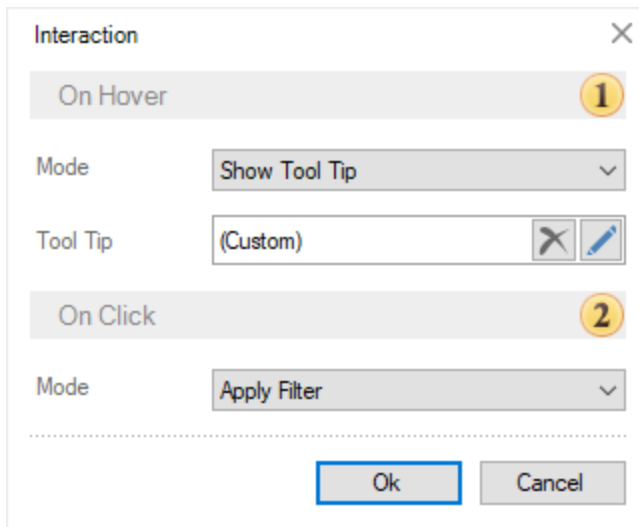
❶ The **On Hover** group of parameters is used to define settings for the **Interactive** action when you hover over a data field value. The **Mode** parameter allows you to select the type of the interactive action:

❯ **None** - when you hover the cursor over the value of data fields, no action will occur;

❯ **Show Tool Top** - when the user hovers the cursor over the value of the data fields of the dashboard, certain information will be displayed. Also, if you select this type of action, the **Tool Tip** option will be available. You can create and design a custom tool tip.

❯ **Show Hyperlink** - when the user hovers the cursor over the data fields, the hyperlink for this data filed will be displayed. If the hyperlink is not set, then the value itself will be displayed when hovering over the value.

❷ The **On Click** group of parameters allows you to set an interactive action when you click with the left button of a mouse or touch the value of a data field. The **Mode** parameter allows you to select the type of an interactive action:

❯ **None** - when you click on the value of the data field, no action will occur. You should know that, if a hyperlink is specified for the values of the data field, then you can follow by this hyperlink.

❯ **Apply Filter** - when you click on the value of the data field, the data of the dashboard will be filtered through the interrelation of its elements.

5   Data

This section provides information on using the data dictionary. The examples in this section show the processes of creating a database connection from the Report Designer, creating a data source to work with variables and their use in reports.

> ⚠ Note: When creating report templates you should consider the value of the Retrieve Only Used Data property.

5.1 Data Dictionary

Each report contains the data dictionary. The data dictionary contains information about the data used to create reports. This information includes: connections to databases, data sources and their relations, variables, and business objects. Also the report data dictionary may not have any information about the data, but the report will be rendered. The report data dictionary is displayed in the **Dictionary** panel. The picture below shows the **Dictionary** panel:
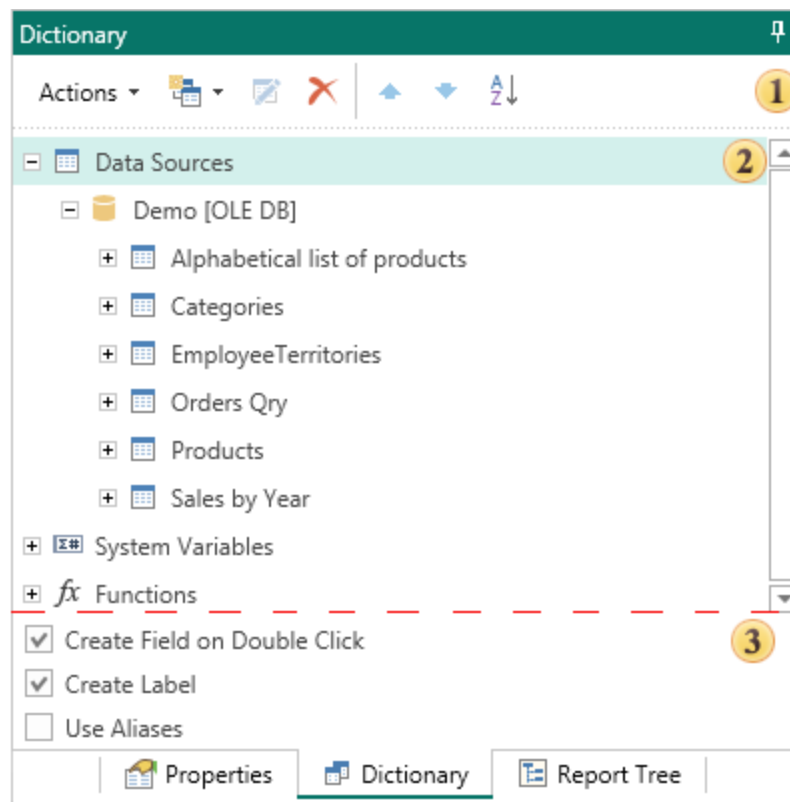
① The **Data Dictionary** panel. Contains the basic controls in the dictionary.
② The **Information** panel. Displays information about the data as a tree.
③ The **Settings** panel. Used to enable/disable some options to work with the data dictionary.

5.1.1 Control Panel

The basic elements to control data dictionary can be found on the control panel. The picture below shows the control panel:

❶ The **Actions** menu. This menu contains the main control commands for the data dictionary;

❷ The **New Item** menu. In this menu the basic commands to create new elements in the data dictionary are placed;

❸ The **Edit** button provides an opportunity to change any element, which can be edited;

❹ Using the **Delete** button one can delete any item in the data dictionary available for deleting;

❺ Pressing the **Up/Down** buttons, the selected item in the data dictionary is moved one position up/down;

❻ The **Sorting Items** menu. In this menu one can select the sorting direction: **Ascending**, **Descending**. Also in this menu, one can enable **Auto Sort**. The picture below shows the Sorting Items menu:
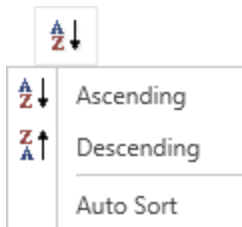
The **Ascending** option sorts the information in order from **A** to **Z**; The **Descending** option sorts the information in order from **Z** to **A**. The **Auto Sort** sorts in order from **A** to **Z**. One should note that the items are sorted within functional groups. For example, data sources within the data sources group are not mixed with the variables and the variables within the variables group are not mixed with the data sources, etc. Also note the nesting of elements of the data dictionary.

5.1.1.1 Actions Menu

In the **Actions** menu the main commands to control the data dictionary are located. The picture below shows this menu item:
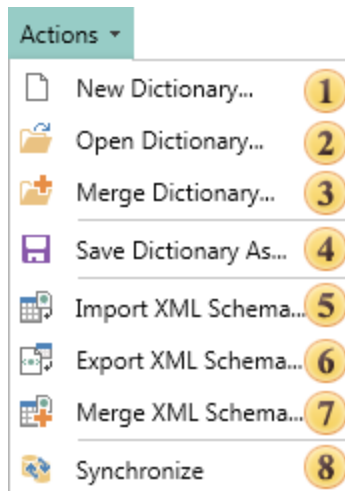
1 The **New Dictionary...** command is used to create a new data dictionary in an editing report;

2 The **Open Dictionary...** command invokes a dialog box in which one should specify the path to the previously saved data dictionary, select it and click Open. In this case, the current data dictionary is replaced with the specified data dictionary.

3 If it is necessary to add a data dictionary to the data dictionary in the report, you can use the **Merge Dictionary...** command. Using this option, the user will see a dialog box in which it is possible to specify the path to the previously saved data dictionary, select it and click Merge. Then, the selected data dictionary will be added to the data dictionary in the report. If the current data dictionary and the data dictionary, which will be added, have the same items, the existing items will be replaced on data items from the added data dictionary;

4 The **Save Dictionary As** command invokes a dialog box in which it is possible to specify the path by what data dictionary, the name of the saving *.dct file will be saved, click the Save button. After that, the data dictionary of a report will be saved;

5 Using the **Import XML Schema...** command it is possible to import information about the data from the selected XML schema to the data dictionary. After clicking this item, a dialog box will be invoked where a user must specify the path to a previously saved XML schema, select it and click Open;

6 Using the **Export XML Schema...** command it is possible to save the data dictionary as an XML schema. After clicking this item, a dialog box will be invoked where one must specify the path to save the XML schema and the *.xsd file name. Then click the Save button;

7 If it is necessary to add more information about the data from the selected XML schema to the information about the data in the data dictionary, click the Merge XML Schema... command. A dialog box will be invoked where one must specify the path to the XML schema, information from which will be added, select it and click Open;

8 The **Synchronize** command provides the ability to synchronize the contents of a data dictionary with the data that are registered for the report. This command synchronizes the registered data in a data store and data dictionary of a report. Moreover, the data can be passed to the report from both the program and be connected in the report. If data were registered using the **RegData** or **RegBusinessObjects** methods then, when running the report designer, they will be synchronized. It is necessary to note that if the data are registered in a report as connections to databases, then synchronization will not be performed automatically. This remark is not related to a connection in the report, generated for the **XML** data. For data that are registered in the report and receive the information from databases using queries, one must use the wizard to create a new data source. A wizard to create a new data source provides the ability to add tables from the database automatically.

5.1.1.2 **New Item Menu**

Commands using which it is possible to add new items to the data dictionary of a report can be found in the **New Item** menu. The picture below shows the **New Item** drop down list:
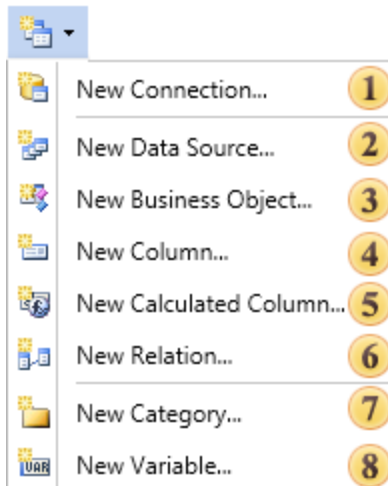
1 If you want to create a new data source in the data dictionary of a report, you must select the **New Data Source...** command. The type of the data source depends on the type of connection. When using this command, a wizard to create a new data source that provides the ability to add more than one data table in a data dictionary of a report. It is necessary to know that this is just a method of describing the data source.

2 To add a description of a new business object to the data dictionary of a report you should select the **New Business Object...** command. It should be remembered that for each created business object, you must pass real business objects from the program.

Since, as already mentioned before, only a method of describing data is created in the data dictionary. So, without real business objects, it will not work.;

③ Add a new column in the selected data source or a business object using the **New Column...** command. Also, if the data column is added to the report data dictionary, but it does not really exist in the database, it can lead to incorrect report rendering.

④ In the report data dictionary, it is possible add a new calculated column in the selected data source. Use the **New Calculated Column...** command for this. In contrast to the simple data column, for proper report rendering, it is not necessary for a new calculated data column be placed in the database.

⑤ The command to add the variable to the data dictionary.

⑥ In order to organize a new relation between the data sources, you should use the New Relation... command. It is worth to note that relations can be created only between data sources and cannot be created between business objects. Therefore, if needed to create the relation between business objects, the **RegData** method should be used instead of the **RegBusinessObjects** method. The **RegData** method converts the business object into the ADO.NET DataSet. As a result, you can work with this business object by means of ADO.NET. Accordingly, it will provide an opportunity to add new relations between business objects and use them.

⑦ If you want to add a new category of variables in the report data dictionary, you should use the **New Category...** command. All variables are organized in a two-level structure, where the variable can be located both in the main list and in the category, which is located in the main list. Such a category can be created with this command.

⑧ The **New Variable...** command provides an opportunity to add a new variable into the data dictionary. If, when calling this command, any category of variables has been selected in the data dictionary, then the variable will be created in this category. If no category in the data dictionary has been selected or the Variable element has been selected in the data dictionary, then the new variable will be created at the top level of the variables list.

5.1.2 Data Sources

The **Data Source** is a structural description of the data used for the report. The Data Source is like a program "layer" which provides data from the database and its conversion and to the report generator. In other words, the data source is a description of the methods, parameters, and data access methods.

> ℍ **Notice**: The description of data does not contain actual data. Filling the data is carried out at the time of the report rendering process.

To create a data source you should select the **New Data Source** command in the New Item menu of

the data dictionary or from the context menu:
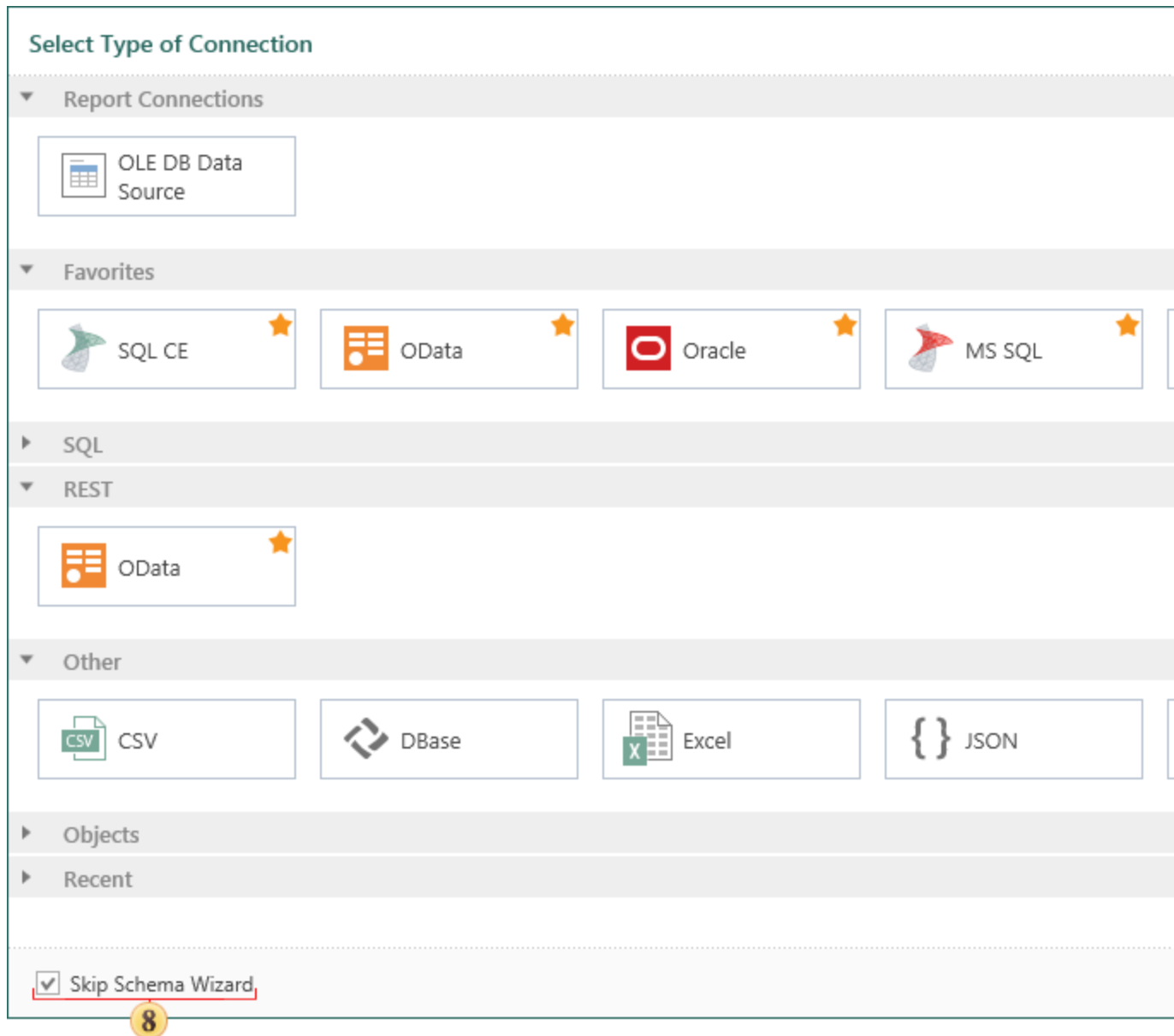
Before the new data source is created, you need to connect to the data storage. In the dialog of creation the data source, connection types are grouped:

1 The **Connection** group contains already created links to the data storage. If no connection is established, this group will not be displayed.

2 The **Favorites** group lists the types of connections that have been marked by the user. In other words, the user can create a list of connections, checking them with stars. To do this, move the cursor to the upper right corner of the connection and press the left mouse button (in the case of the touch UI, simply press the input pointer). If the star is orange, the connection is added to the list of favorites. In order to remove the connection from the list of favorites, you must click on the "burning" star:

In the left picture, the star is not checked, the connection is not selected. In the right picture, the connection is selected. If no one is checked with the star then this group will not be displayed.

3 This group contains a list of all connections that support SQL connection strings.

4 This group contains data sources to connect to the data store using REST protocol.

5 The **Other** group contains commands to create connections to data stores such as the XML, Excel, JSON, CSV, Dbase.

6 To create a connection to the database containing objects, you should use this group. For example, for passing business objects from the repository in the report.

7 This group contains previously created connections. In other words, ever created connections to the data stores but not available in the current report are located in this group.

8 The **Skip Schema Wizard** parameter. When you create a data source, the following methods exist to obtain them from data storage:

> Get the data scheme. In this case, you will see a hierarchical list of data in the form of tables, views, stored procedures, etc. The user should select the required sources with flags;

> Generate a query to obtain data. For more details read about queries here.

To determine the method of obtaining the data is possible by means of the **Skip Schema Wizard** parameter. If you want to retrieve the database schema, you should uncheck this option. If you need to go to the creation of a query, check the flag for this parameter. It should be borne in mind that you can go to the creating of a query from the form of retrieving data by clicking the **New Query** button.

Once the connection is established, depending on the type of the data source and the **Skip Schema Wizard** value, the create data source form is created.

5.1.2.1 Queries

Queries are text script forms, which are used to extract data from tables and making them available in the report generator. Queries is that they get data from database tables and create them on the basis of a temporary table. The data in the temporary table will be filtered, grouped, sorted and ordered, according to the query parameters. Then, the temporary table is passed to the report generator. Applying queries provides the ability to avoid duplication of data in tables and provides maximum flexibility for searching and displaying data in a database. Most of queries are used to fetch data from the database and transfer them to the report generator. Not all data source types support **SQL** queries. If the type of a data source supports **SQL** queries, the **New Data Source** dialog will display the **Text Query** with the query. The picture below shows a

New Data Source dialog, where in the **Query Text** field a query for fetching is created.

As can be seen from the picture above the **CategoryName** column of the **Categories** data source, and the columns **ProductName**, **UnitPrice**, **UnitsInStock** from the **Products** data source will be selected. The relation between data sources is organized by the key data column **CategoryID**. In order to get the data column, click the **Retrieve Columns** button after building the query. In addition, before a query you can change the type of the data source. The following types are available: **Stored Procedure** and **Table**.

❶ The **Name in Source** field. In this field, you can enter the name or you can click the [...] to call a list

of names. The list of names will be displayed in the **Name in Source** dialog;

2 In the **Name** field specifies the data source name that appears in the report generator;

3 The alias of the data source should be indicated in the **Alias** field;

4 Command to control queries. This panel has the main items to control text queries. Click the **Run** button to run the query for execution.

5 The **Query Text** field. This field specifies the text of the query.

6 The menu to select the data source type. The following types of data source are **Table** and **Stored Procedure**. The picture below shows the selection menu of the data source type:
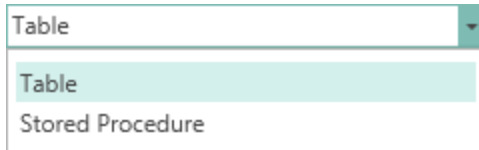
7 Commands to manage data. This panel lists commands such as creating a new column, the new calculated columns, the new parameter. Among other things, this panel has the **Retrieve Columns** command.

8 The **Columns** panel. This panel displays the data source columns, and parameters. Properties of the selected column or parameter are located on the property bar.

5.1.2.1.1 Parameters

When creating a query it is possible to use the **Parameter** object. This object is designed to send additional conditions for selecting data into a query. For example, if you need a query to use a value entered by the user each time the query is executed, you can create a query using parameters. The **Parameter** object can only be used with **SQL** data sources. These data sources are typically have the **Text Query** field. To insert a parameter in the query, you must click the **New Parameter** button. The picture below shows the toolbar, on which the **New Parameter** button can be found:
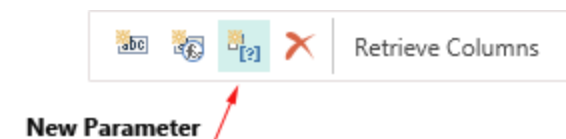
After clicking this button a new parameter will be created. This parameter will be displayed in the **Parameters** tab in the **Columns** panel. The picture below shows an example of the **Columns** panel with the **Parameters** tab:
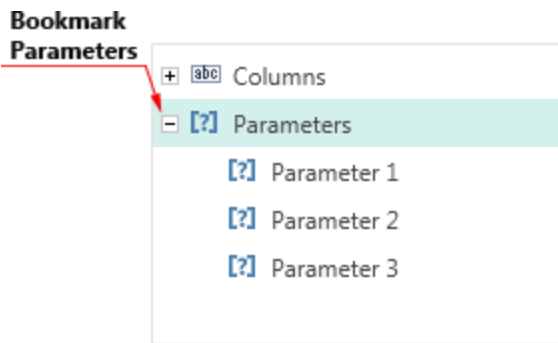
Each parameter has a property with which you can change its settings. The picture below shows the panel of parameters properties:

1 For each parameter you can specify a value that is used to populate the parameter. The value can be an expression, **const**, variable, etc. For example, **{x + y}** or **{variable}**.

2 The **Name** property. Used to change the parameter name. This feature works only for named parameters.

3 The **Size** property provides an opportunity to change the size of the type used in the parameter. Keep in mind that each type in the database has its own size. Therefore, when using a query, you must specify the correct type size. For some adapters, database size may be omitted, but generally if the size is not specified or is incorrect, then the queries using these parameters will be performed incorrectly.

4 Use the **Type** property in order to change the parameter type. The values of the properties are in the drop-down list, and are a list of types used in the parameters for a particular database. It should be noted that a list of types differs depending on the database.

Also, you must specify the parameter in the query. Here is an example of schematic position of parameters in the query:
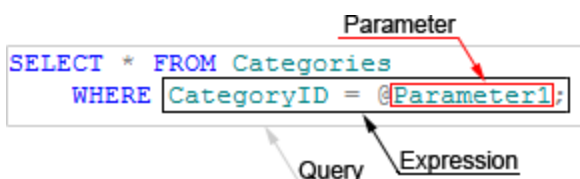
As a rule, the **@** symbol is used to specify a parameter in the query. The **@** symbol is used with named parameters, i.e. after the **@** symbol goes the name of the parameter. But in some databases (for example in **OleDB**), the **@** symbol cannot be perceived by the adapter and database queries with parameters will not work. In this case, you can use unnamed parameters. For specifying unnamed parameters in the query the **?** character is used. After the **?** character, the parameter name is not specified. In this case, the order of parameters in the **Parameters** tab is important. As indications of the **?** characters in the query, parameters will be taken sequentially from the **Parameters** tab in the top-down direction. Consider the following example. Suppose there are three parameters that are specified in the query:

```
[?] Parameters          SELECT * FROM Products
    [?] 1                        WHERE CategoryID = ?
    [?] 2                        AND ProductID = ?
    [?] 3                        AND OrderID = ?;
```
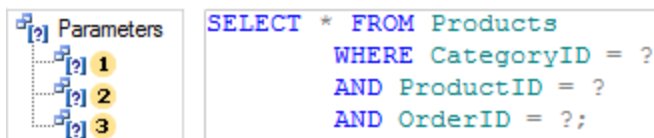
Since, in this case, unnamed parameters (marked with **?**) are used, then, when running, the query parameters will be taken from the **Parameters** tab in the top-down order. The picture below schematically presents a comparison of parameters of the **Parameters** tab to the parameters in the query:

```
SELECT * FROM Products
       WHERE CategoryID = 1
       AND ProductID = 2
       AND OrderID = 3;
```
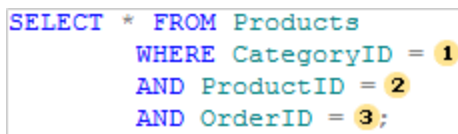
In this case, the parameters used in this example, can have names, but when using the **?** character they play no role. Once a query to parameters is created and executed, the parameters will also be displayed in the **Dictionary**, in the created data source in the **Parameters** tab. The picture below shows an example of the **Dictionary** panel and placing parameters in it:
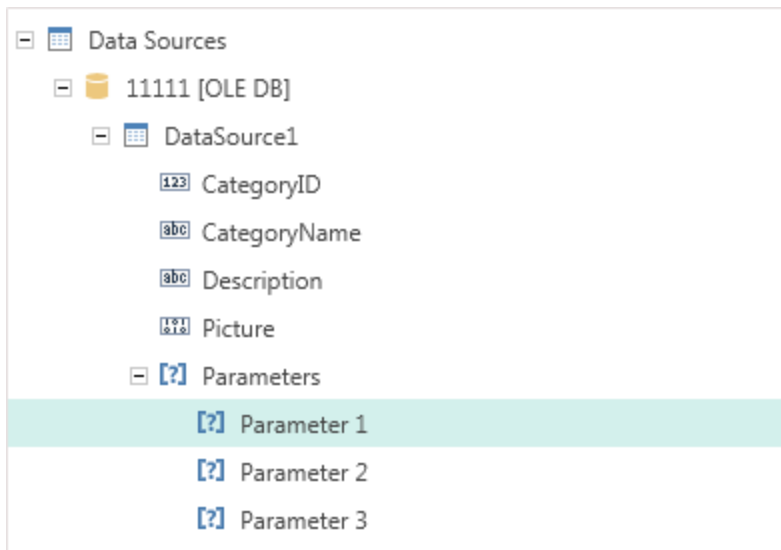
To edit a parameter separately from the data source, select the **Parameter** in the data dictionary and click **Edit** on the toolbar in the dictionary or select **Edit** item in the context menu of the selected parameter. After pressing the button or selecting **Edit**, the user will be shown the **Edit Parameter** dialog, in which you can edit the selected parameter. The picture below shows an example of the **Edit Parameter** dialog:
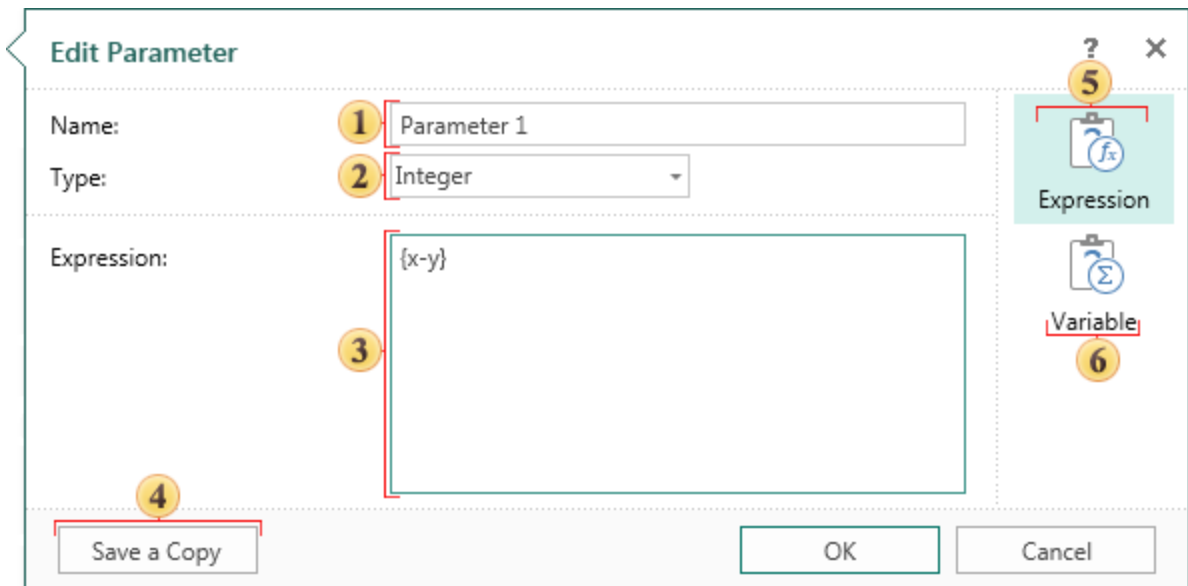
❶ This field displays the parameter **Name**, which can be edited;

❷ This field displays the **Type** of the parameter, which can be edited;

❸ The **Expression** field displays used expressions in a query parameter, which, if necessary, can be edited;

❹ The **Save a Copy** button saves a copy of the edited parameter by assigning the

Copy postfix in the parameter name.

5 The **Expression** tab. An expression, link to the data column, etc is specified as a value of the parameter.

6 The **Variable** tab. A variable is specified as a value of the parameter.

Using variable as SQL parameter

A variable can be specified as a value in the parameter. In this case, values of the variable will be the values of the parameter when requesting data. There are two ways to use a variable in a query as a parameter:

➤ Create a variable in the data dictionary. Open the data source for editing. Create a parameter in the data source. Specify a variable as the value of this parameter. Insert the parameter in the text of the query.

➤ When creating or editing a variable, set the **Allow using as SQL** parameter check box:

Register this variable in the text of the query, using the special "**@**" symbol before the variable name:

```
select * from Products
        where Discontinued = @Variable1
```

Click **OK**. Now the variable is present in the data source and is used as a parameter in the query.

5.1.2.1.2 Query Builder

The **Query Builder** is a visual component that allows creating queries visually. Creating a query using a designer allows complete controlling the query parameters and building of complex conditions of data selection using simple visual user interaction. The picture below shows the **Query Builder** dialog:
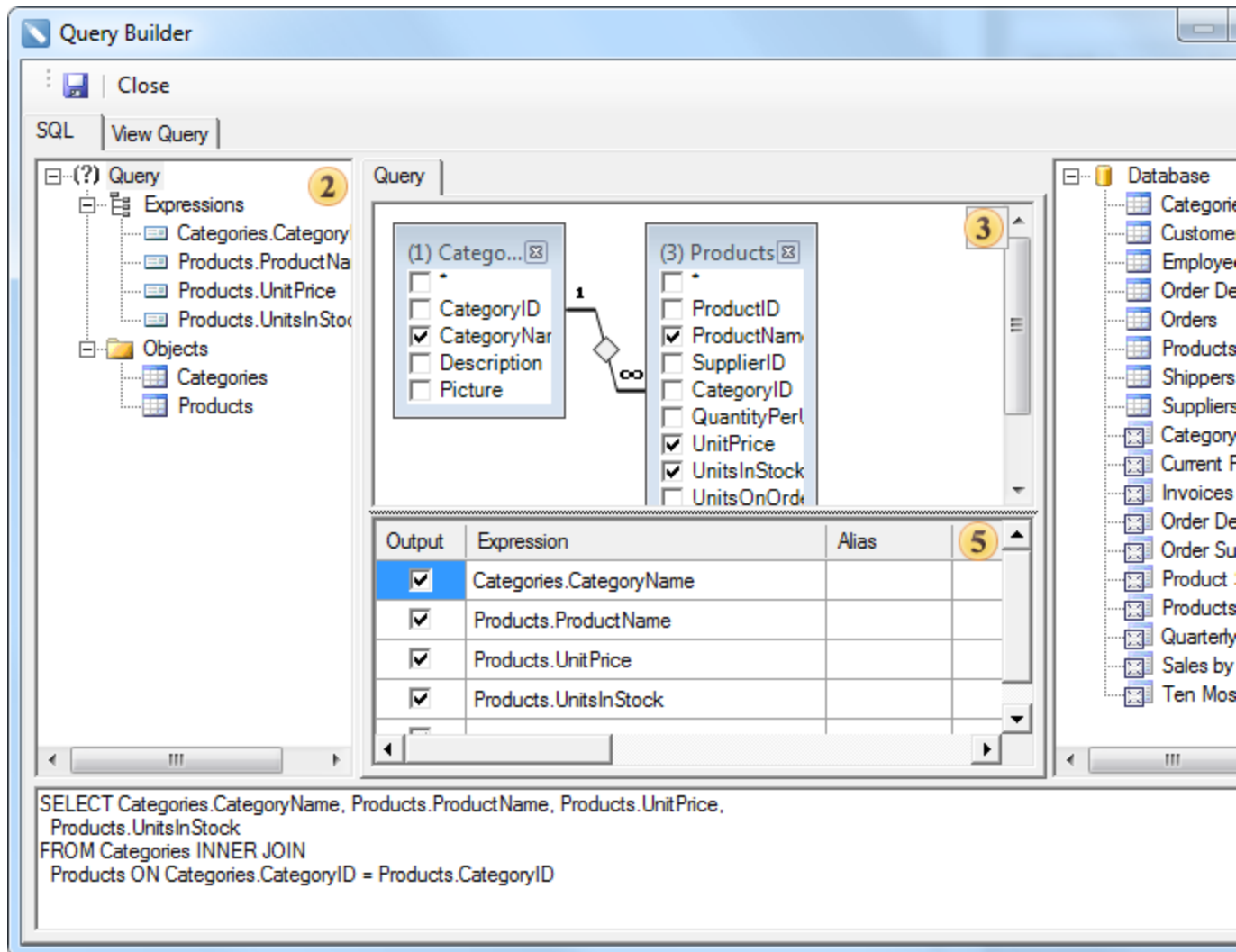
① **Control Panel**. Contains the Save button (saves the query) and the Close button (closes the query builder);

② **Query tree panel**. This panel shows the query tree.

③ **Query design panel**. This panel is an area in which the query is visually represented. In this area, you can determine the initial database objects and derived data sources, as well as define relations between data sources, configure the data source properties, and references.

④ bar databases. This panel displays the database and included in her data sources;

⑤ Table panel. This panel shows a table in which rows are data columns used in the query and columns are operations. In this table, you can define data columns, aliases, sorting type, sorting order, grouping, criteria.

⑥ This panel displays a query built on the panel **③** as a code.

The Query Builder contains the **View** tab, which provides an opportunity to display data

columns selected by the query. There operations in the query should also be taken into account. The picture below shows the **View** tab in the Query Builder:

Click the **Save** button to add the created query text into the **Query Text** field.

5.1.2.2 Creating Data Source

Consider an example of creating a new data source. It is worth noting that before you can create a data source, you must setup a connection. If there is no connection, then go to **Dictionary**, select New the **New Data Source** command in the **New Item** menu:
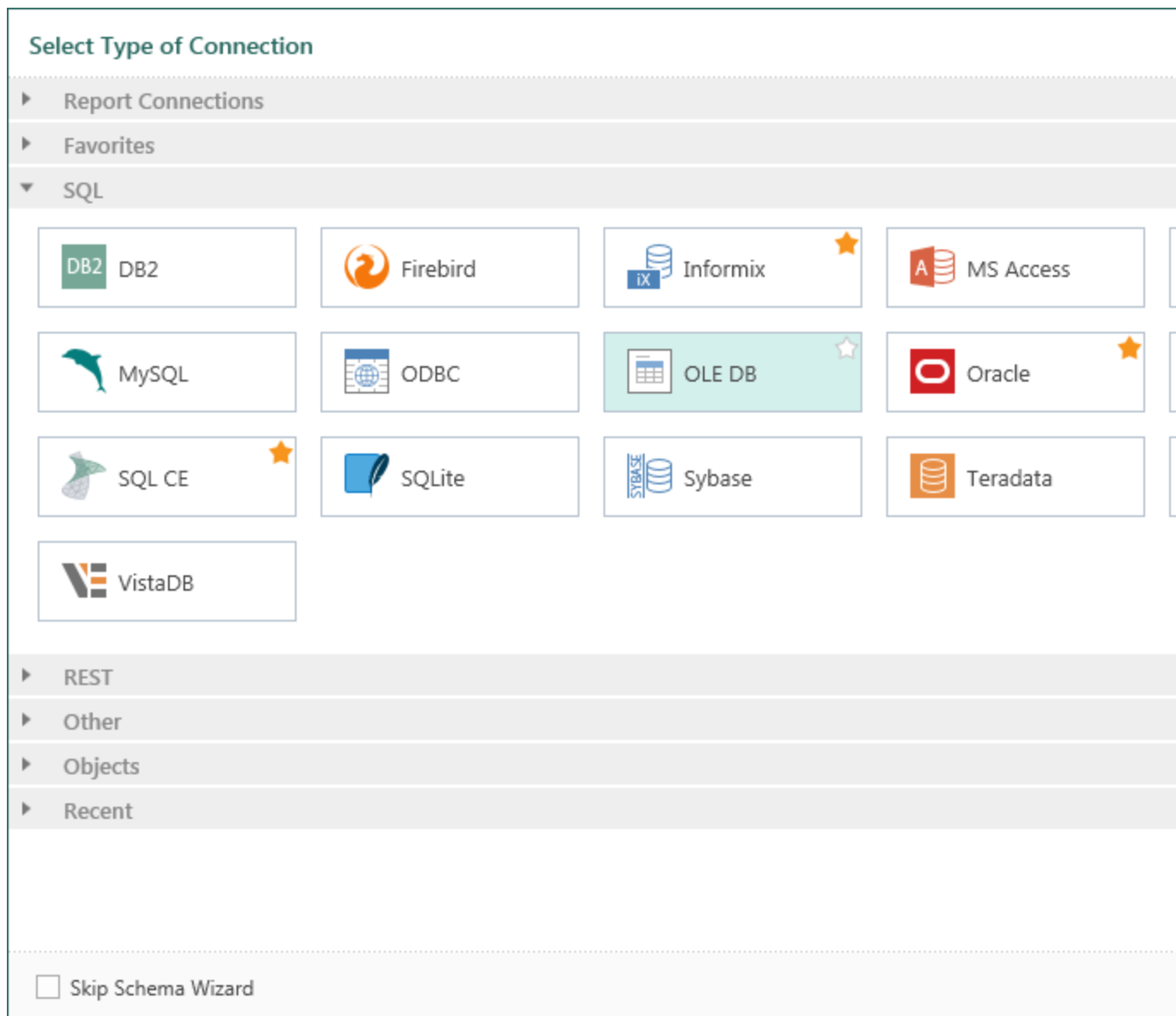
In the opened menu, select the type of a connection, for example, OleDB. The form to create the connection will be opened:

Specify the connection name, alias, and the connection string. Also you can find buttons to call the query builder, to clean the connection string, the button to check the connection and the button of the connection string template (for OleDB pattern is as follows: *Provider=SQLOLEDB.1; Integrated Security=SSPI; Persist Security Info=False; Initial Catalog=myDataBase; Data Source=myServerAddress*). To verify the connection string, press the **Test** button. In this case, if the connection string does not contain errors, the user will see the **Connection was successful** window. If the connection string contains an error, the user will be shown a window with the text of the error which was returned by the database server in response to the attempt to create the connection. After clicking the **Ok** button, a new connection will be created.

Next, the following ways to create a new data source are possible:
❯ Obtaining data by schema;
❯ Creating a request for retrieving data.
It specifies how to create a data source, such parameter as **Skip Schema Wizard**. If it is checked, then after creating a connection, the user will see a the query form. If the checkbox is not checked, the data schema will be retrieved.

Consider a ways of obtaining data in detail.

Retrieving the data schema (the checkbox of the Skip Schema is not set)
After you create the connection you will go to the **Select Data** dialog. To get a list of tables from the database, you must click the **Refresh** button in this window. You can also enable/disable the **Auto Refresh** mode check/uncheck the checkbox. If checked, the wizard will automatically update the list of data tables. The list in this window is represented as tabs which are arranged in a hierarchical form. The

Home tab is a category (for example, Queries, Tables, Views, Procedures).

For creating a new data source you should select the data table. It is also possible to exclude the table data columns from the future data source. For this purpose, it is necessary to open the selected table and uncheck the flag next to the name of the column that you want to exclude. By default, if you select the data table, all the columns in this table are checked. They will be added to the new data source. Each selected data table will be a single data source, ie, one table is one source. The picture below shows the **Select Data** window with selected data tables and data columns selected:
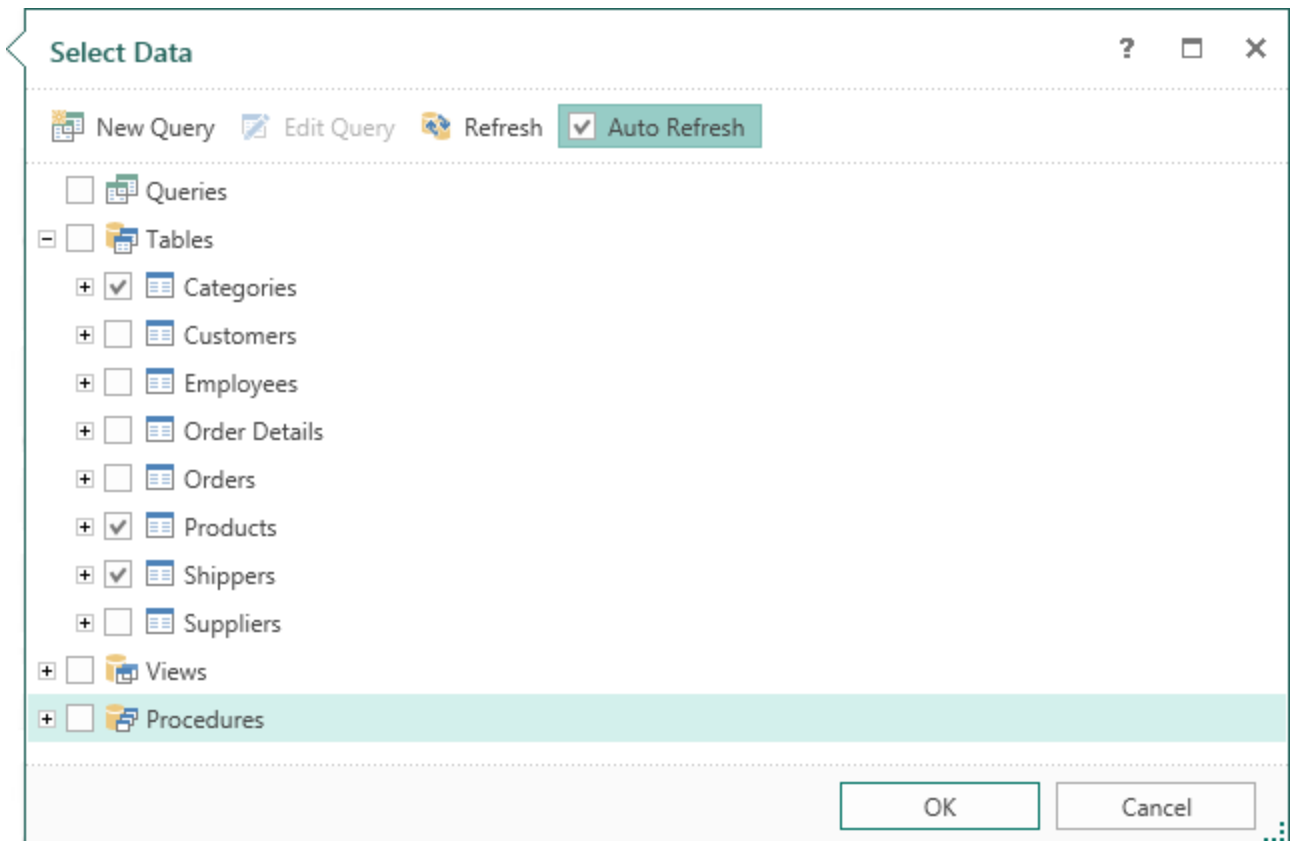
After clicking Ok, the **Categories**, **Products** and **Shippers** data sources will be created. The picture below shows the data sources created in the **Dictionary**:
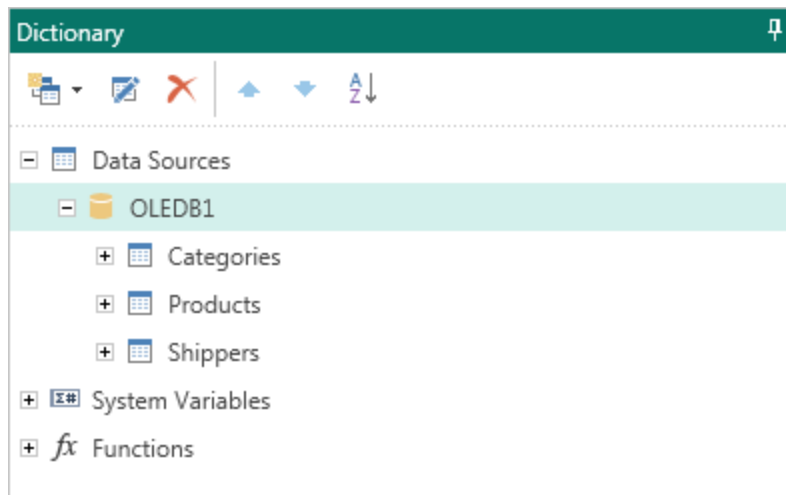
Now, report templates will be created on the basis of these data descriptions.

Retrieving data without schema (the checkbox of the Skip Schema is set)
After the connection has been created and Skip Schema is checked and you will go to the Edit Data Source dialog to create the query:

In this window you must define parameters such as the **Name in Source**, **Name**, **Alias**. Also, in the **Query Text** field, it is necessary to form a database query and execute it. If the request is successful, press the Retrieve Columns button.

Each column contains properties such as the **Name in Source**, **Name**, **Alias**, and **Type**. To change the values of these properties, high you should select the data column, and,

in the Properties panel, to change and edit them. It is also possible to add or remove a data column. To add a column, click the **New Column** button or the **New Calculated Column** button. To delete a column, it is necessary to select it and click **Delete**.

After clicking Ok, a new source will be created.

Editing the data source
Any created data source can be edited. To do this, select the data source, click the **Edit** button on the toolbar in the **Dictionary**, or select the **Edit** button in the context menu of the data source. After clicking the button or selecting the **Edit** item of the context menu, the user will see the **Edit Data Source** dialog. It has the same tools and fields as a second dialog - **New Data Source**.
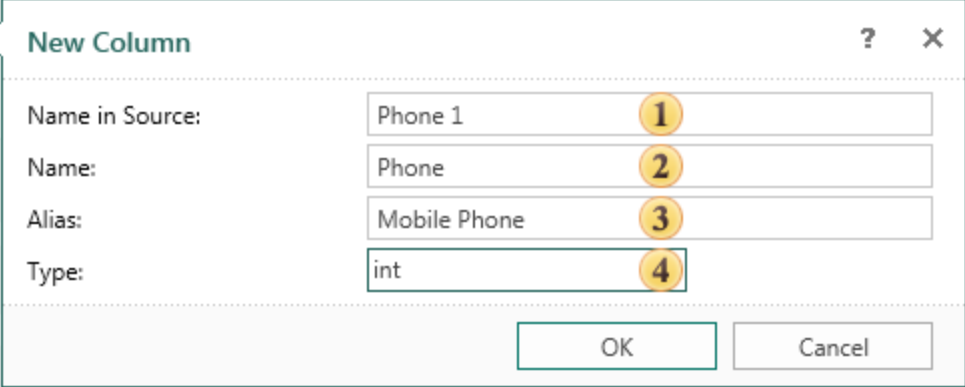
Depending on the type of source, this box may not have the **Query Text** field, because not all connections support SQL queries. All changes will be applied after pressing the **Ok** button. The **Save a Copy** button saves a copy of the edited data source, with the Copy postfix in the name of the data source.

5.1.2.3 Creating and Editing Data Columns

Creating data columns

To create a new column select the data source, which will be added to the data column, and select **New Column...** in the **New Item** menu or the context menu of the selected data source. After selecting this option the **New Column** dialog will be invoked. In this dialog you should specify new columns. The picture below shows a **New Column** dialog:
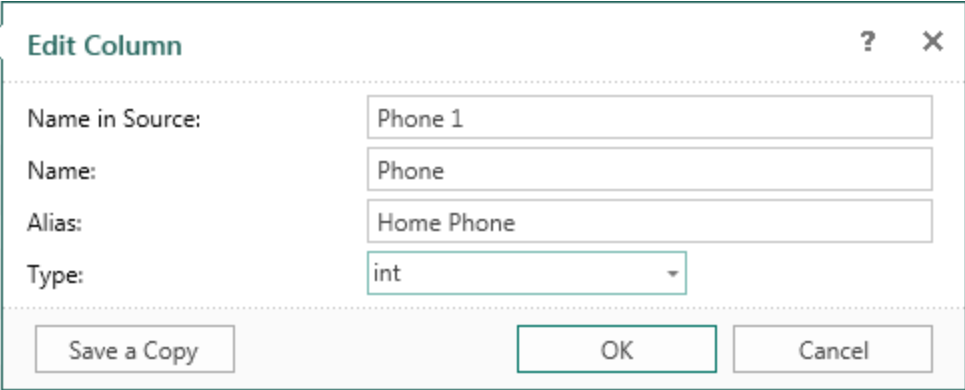
① The **Name in Source** field. Specifies the name in the data source (not in the report).
② The column **Name**. Used to call the new column in the report.
③ The column **Alias**. Specified in the Alias.
④ The **Type** field. Used to select the type of data that will be contained in the new column.

After clicking **Ok**, a new data column in the selected data source will be created. It should be noted that the data column generated this way is only a description of the (virtual) data columns and it does not contain real data. If the database does not have this column, then when calling the database, the report generator will produce an error.

Editing data columns

The data column can be edited. To do this, you must select **Edit** in the context menu of the selected column, or click the **Edit** button on the toolbar in the data dictionary. After that, the user will be shown the **Edit Column** dialog, where you can change settings such as **Name in Source**, **Name**, **Alias** and **Type** of the edited column. Press **Ok** to apply changes. The picture below shows the **Edit Column** dialog:

The **Save a Copy** button saves a copy of the edited data column, with the assignment of the Copy postfix in the name of the data column.

5.1.2.4 Calculated Data Column

The calculated data column is calculated on the base of an expression that can be used by other data columns into an existing data source. The expression can be a name of the non-calculated column, constant, function, or any combination, connected to one or more operators. The expression cannot be a nested query. The calculated data column is a virtual column that is not stored physically in the data source. The values of the calculated data column are updated each time you access to them in the query. Also, the values of calculated column are updated every time you change the columns included into the calculated expression. Before you add a calculated column, you must connect at least one data source. Consider the creation of calculated data column in the data source Auto. The following columns are in this data source: Rank, Country, Year2000, Year2005, Year2009. Columns Year2000, Year2005, Year2009 contain data about cars produced in 2000, 2005, and 2009. Create a calculated data column, which will contain data on the growth of production cars in 2009 relative to 2000, the results are displayed in percentages. The picture below shows the data column of Year2000 and Year2009:

Year2000	Year2009
2069069	13790994
10140796	7934516
12799857	5711823
5526615	5209853
3114998	3512916
1681517	3182617
801360	2632694
3032874	2170078
3348361	2049762
1935527	1557290

To create a new calculated column you should call the **New Calculated Column** dialog and fill in the dialogue form. The dialog can be called from the context menu of data source or from the **Actions** menu. The picture below shows the **New Calculated Column** dialog:

❶ The **Name** column is used to call this calculated column in the report. Enter in the **Name**.

❷ The **Alias** column is used as a prompt. Enter in the **Alias**.

❸ The **Type** field provides the ability to choose the data type that will contain the new calculated column.

❹ The **Dictionary** button contains a drop-down menu that displays the structure of the data dictionary. In this menu you can select data columns, business objects, or system variables that will be added to the calculation of expression of the calculated data column.

❺ The **Value** field is used to define an expression for calculating the values of the new calculated data column.

In this example, the calculation expression will contain data columns Year2000 and Year2009 from the data source Auto, and the type of data in a new calculated column will be double. After the column is created, you should place a text component with a reference to this data column. In this example, the text component will contain a link **{Auto.NewCalculatedColumn1}**. As the result of calculations is necessary to be displayed in the percentage, then this text component should change the format, i.e. set the **Percentage** format. Below is a report with the calculated data column:

Year2000	Year2009	NewCalculatedColumn1
2069069	13790994	15,00%
10140796	7934516	127,81%
12799857	5711823	224,09%
5526615	5209853	106,08%
3114998	3512916	88,67%
1681517	3182617	52,83%
801360	2632694	30,44%
3032874	2170078	139,76%
3348361	2049762	163,35%
1935527	1557290	124,29%

5.1.2.5   Data From Other Data Source

In the report generator you can create a data source based on existing data sources. The **Data from other Data Source** provides analogical features like the query to the database. When creating a data source using the visual interface, in the process of creating a data source, to perform sorting, grouping, filtering, and calculating of totals using aggregate functions. Consider the example of creating data from other data sources. Suppose there is a **Master-Detail** report, to which each category corresponds a number of products. The picture below shows a page of the **Master-Detail** report (shown partially):

As can be seen from the picture above, the name of the category, product name (related to this category) and the price of the product are displayed in the report. If you want to create a report that displays the name of the category and the total value of all products included in this category, it can be done in various ways. But the easiest way is to create a data source based on another data. To do this, select **Data from other Data Source** item in the **New Data Source** dialog and setup the data source you create. The picture below shows the second form of the **New Data Source** dialog:
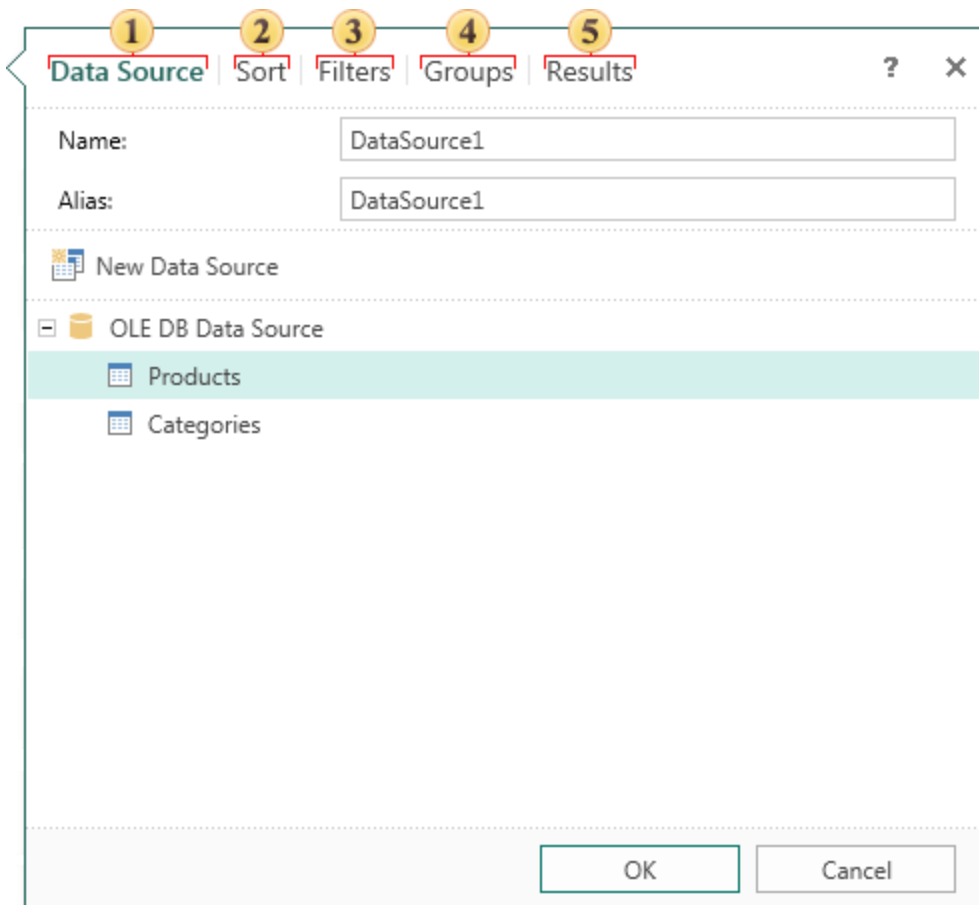
As can be seen from the picture above, the process of creating data from other sources includes the following steps:

① **Data Source**. On this stage, you must specify the Name of a new data source and its Alias. In our example, the alias name and the data source name is DataSource1. You should also select a data source on which to setup a new one. In this case, the selected data source Products. This step is optional.

② Sorting criteria are specified in the Sort step. On this stage you should specify the data column to be used for sorting, and to select the sorting direction. This step is optional.

③ Set conditions of filtering data in a new data source on stage Filters. To filter the data you need to add a filter to specify an expression or a condition that will be filtered. This step is optional.

④ To specify the conditions of grouping data in a new data source, you can do the step **Groups**. To group the data you should indicate the data column by which the data will be grouped, and select your destination of groups location. Data column, by which grouping will be performed will present in the new data source. In this example, using the relation, between data sources **Categories** and **Products**, indicate grouping by the

data column **CategoriesName**, which contains the names of categories. This step is optional.

5️⃣ The last step is **Results**. In this step, you can make the calculation on a data column with aggregate functions. The picture below shows the Results tab:

Column	Aggregate Function	Name
UnitPrice	Sum	UnitPrice.Sum

As can be seen from the picture, this tab should indicate the following parameters:

➤ Select the data column in the **Column** field that will be present in the new data source or from which data will be collected to calculate the aggregate. This field is mandatory. For example, the data column **UnitPrice** is selected. It contains data on the products prices.

➤ The **Aggregate Function** menu is a list of aggregate functions that can be used to calculate the selected data columns. Aggregate functions can be omitted in this case, the data column will contain data, which are in the data column, which is the basic one. In this example, select the aggregate function **Sum**, which summarizes the data.

➤ In the **Name** field specify the column name, which is used to refer to this calculated column in the report.

Now for the report rendering the data source **DataSource1** can be used, which contains two data columns: **CategoryName** and **UnitPrice.Sum**. The picture below shows a report, based on data from a data source **DataSource1**:

1	Beverages	455.75
2	Condiments	276.75
3	Confections	327.08
4	Dairy Products	287.3
5	Grains/Cereals	141.75
6	Meat/Poultry	324.04
7	Produce	161.85
8	Seafood	248.19

As can be seen in the picture above, each category corresponds to the total value of all products included in this category.

5.1.2.6   Data From Cross-Tab

In Stimulsoft Reports you can create a data source based on cross-table, i.e. you can create a new source, which columns will be columns of the rendered cross-table, and strings are the strings of the rendered cross-table. Consider an example of creating a data source based on the cross-table. The picture below shows a report page with the rendered cross-table:

Products					CategoryName				
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	43.9	17.45		7	71.8	53	62.5	270.65
Brazil	4.5								4.5
Canada		28.5	49.3			31.45			109.25
Denmark								21.5	21.5
Finland	18		36.25						54.25
France	281.5			89				13.25	383.75
Germany	7.75	13	89.13		33.25	123.79	45.6	25.89	338.41
Italy				79.3	57.5				136.8
Japan		15.5				97	33.25	37	182.75
Netherlands			22.25						22.25
Norway				60.0					60.0
Singapore	46	19.45			14				79.45
Spain				59					59
Sweden								60	60
Sweden					30				30
UK	37	10	112.7						159.7
USA	46	146.40					30	28.05	250.45
Total	455.75	276.75	327.08	287.3	141.75	324.04	161.85	248.19	2222.71

To create a data source based on cross-table, you should call the **New Data Source** dialog and select the **Data from CrossTab** item. The picture below shows the **New Data Source** dialog:
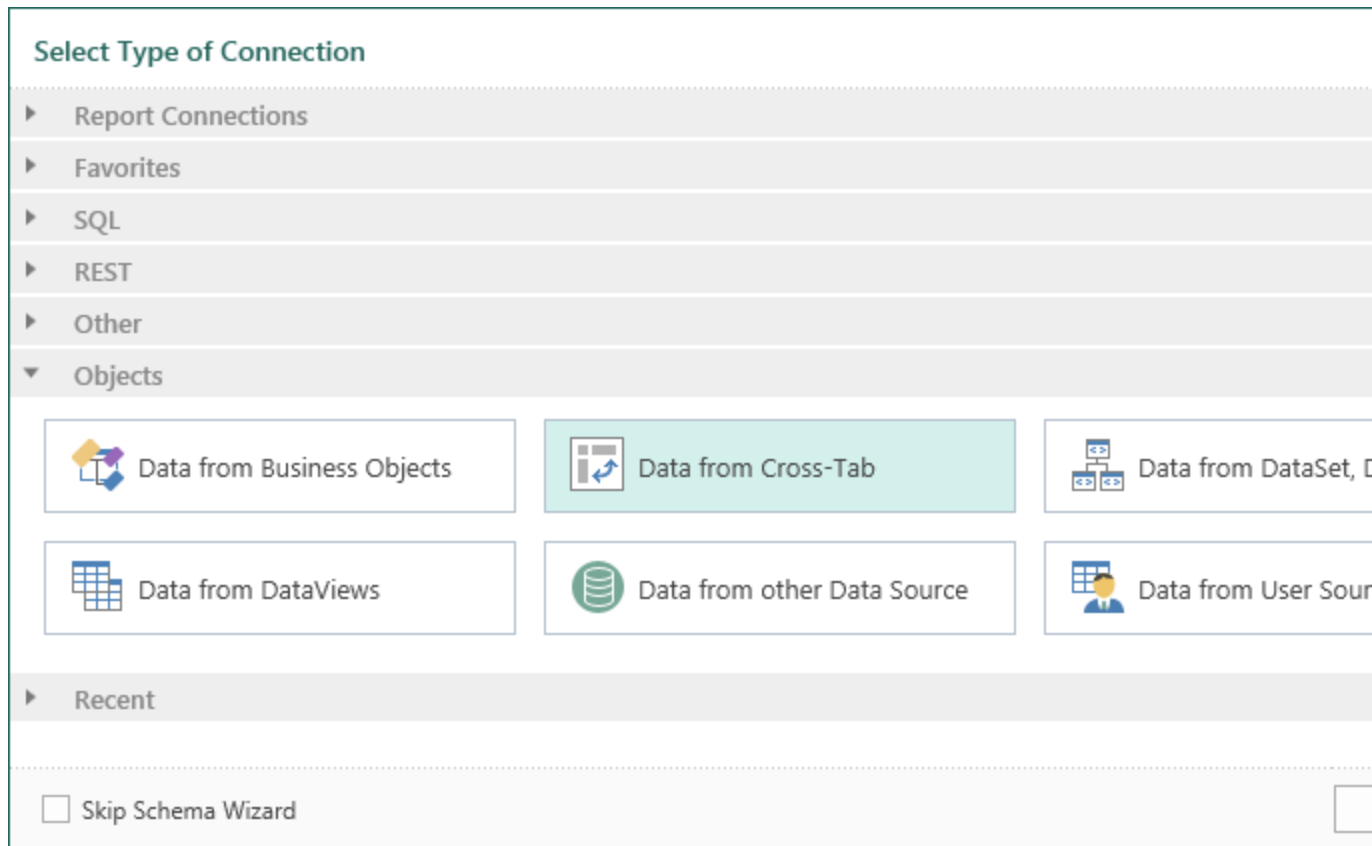
After clicking **Ok**, in the next dialog form **New Data Source**, you should indicate the **Name** of the new data source and cross-table, which will be used as a basis. You can also specify the **Alias** of the new data source. The picture below shows the second form of the **New Data Source** dialog:
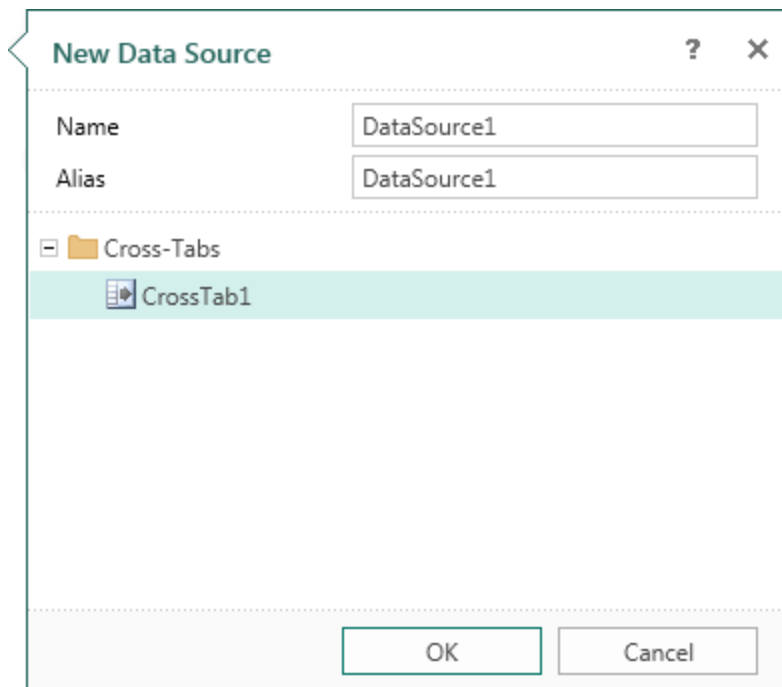
After clicking **Ok**, you will create a data source **DataSource1**, which will contain the columns **ShipCountry**, **CategoryName**, **UnitsPrice**. The data source on the base of the cross-table is a virtual data source that does not contain real data. Filling this source occurs when rendering the cross-table. Therefore, a report that will use this data source, for example, to render a report with the list, must contain the cross-table on the base of which the data source was created. For example, create a report with the list. Put the cross-table in the first report page, and in the second page, put the **DataBand** with text components, which will contain the expressions **{DataSource1.ShipCountry}**, **{DataSource1.CategoryName}**, **{DataSource1.UnitsPrice}**. The picture below shows a part of the report page with the rendered list:

UK	Beverages	37
UK	Condiments	10
USA	Condiments	146.40
USA	Produce	30
Japan	Meat/Poultry	97
Japan	Seafood	37
Spain	Dairy Products	59
Japan	Produce	33.25
Japan	Condiments	15.5
Australia	Confections	17.45
Australia	Meat/Poultry	71.8

When rendering a report, the report generator fills created data source **DataSource1** with data from the cross-table and display the data as a list.

5.1.2.7 Custom Data Sources

If you want to build a report based on the custom data then, in Stimulsoft Reports, you can create custom data based on custom data sources. To do this, you should select Data from User Sources in the New Data Source window, and in the next New Data Source dialog box, configure a custom data source. The picture below shows the form **New Data Source**:
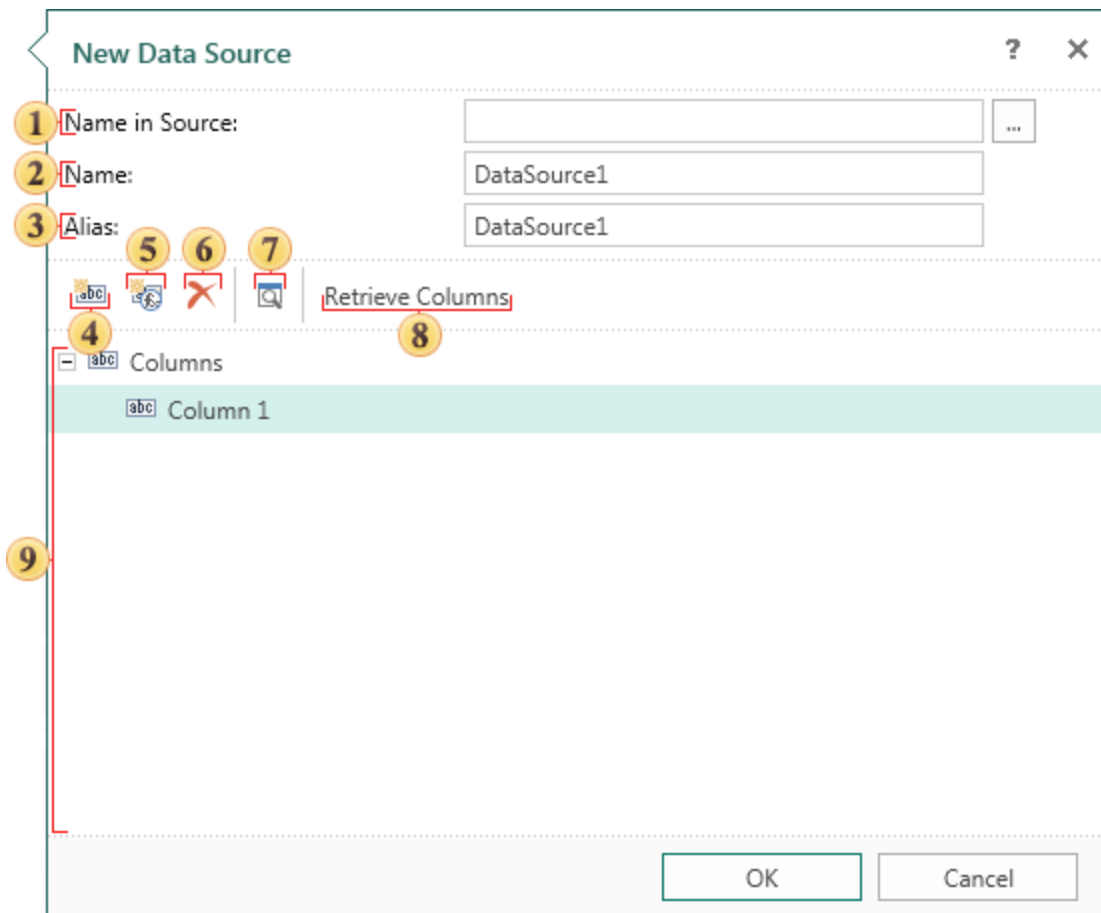
Setting the data source is done using the following controls:

1 The **Name in Source** field. Specifies the name of a connection or database. When creating data on the base user data sources this is not mandatory to fill this field.

2 The source name that is used to access the report is indicated in the Name field. This field is mandatory.

3 The alias of the source is indicated in the **Alias** field. This field is not mandatory.

4 Using the **New Column** button you can add the new column to the data source.

5 The new calculated data column can be added to the data source using the **New Calculated Column** button.

6 The **Delete** button deletes the selected **9** data column or deletes all data columns when the **Columns** tabs is selected.

7 Preview the query.

8 Using the **Retrieve Columns** button you can get all the columns from the database. In this case, there is no connection to the database and the query is not built, so the button is no longer relevant.

9 This panel displays the data source structure.

5.1.2.8 **Retrieve Only Used Data**

Sometimes it is enough to change the value of one property to significantly increase the speed the report rendering. When working with the report template, the data dictionary does not contain any real data. Data in the dictionary are located only as a description of the data structure. Execution of all queries and data transfer from the storage is carried out at the moment of the report rendering process. At this time, the entire structure of the dictionary is filled with real data. In other words, if 200 data sources are created in the dictionary then the actual data are transferred from the storage to all those sources. The more data to be transmitted from the storage to the dictionary, the longer is the time of the report rendering process. However, not always all data sources are used in the report. In order to significantly reduce the time of the report rendering getting only real data for data sources used in the report, you should set the **Retrieve Only Used Data** report property to **true**.

Consider an example. For example, a MS SQL database that contains data tables, stored procedures, and views is used. The picture below shows the data structure of the dictionary:
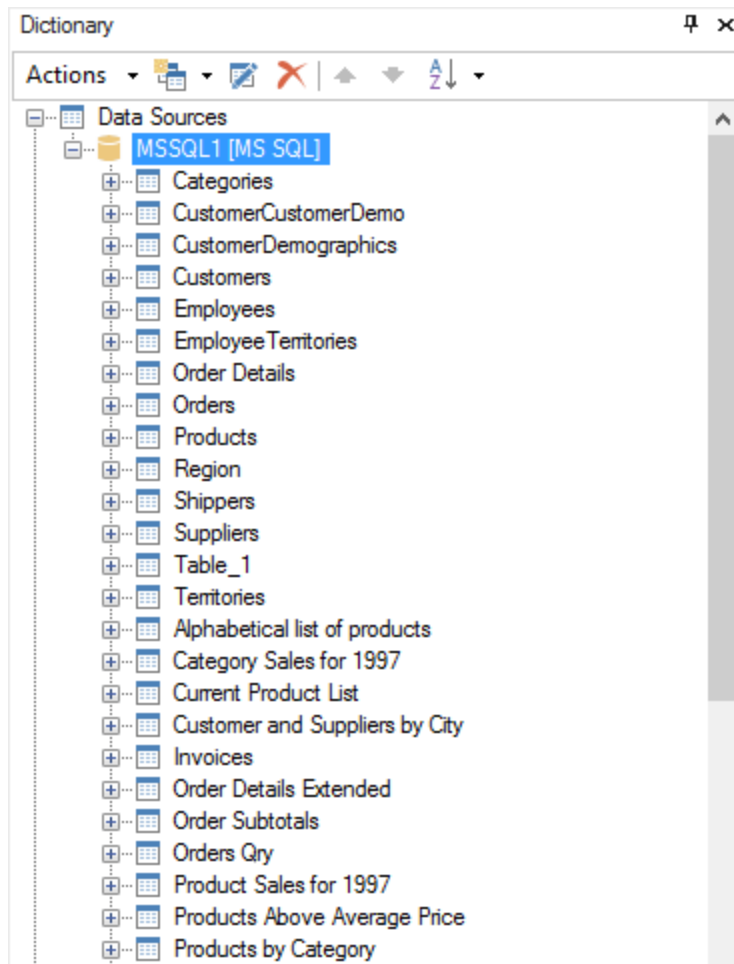
Each table contains data from one to the plurality of data columns, with at least one data row. For example, only the **CustomerCustomerDemo** data source will be used in the report.

❯ The **Retrieve Only Used Data** property is set to **false**

In this case, when rendering the report, data will be transferred from the database for each table in the data dictionary, and then the dictionary in the report itself. In other words, every table will be filled with actual data. Then, the report generator, selects the data used in the report and displays them in a structured way. Time of the report rendering depends on how fast data is transferred and the data size. The faster the data will be transmitted, the faster the report will be rendered. The picture below schematically shows the data transferring, if the **Retrieve Only Used Data** property is set to **false**:

Retrieve Only Used Data = false
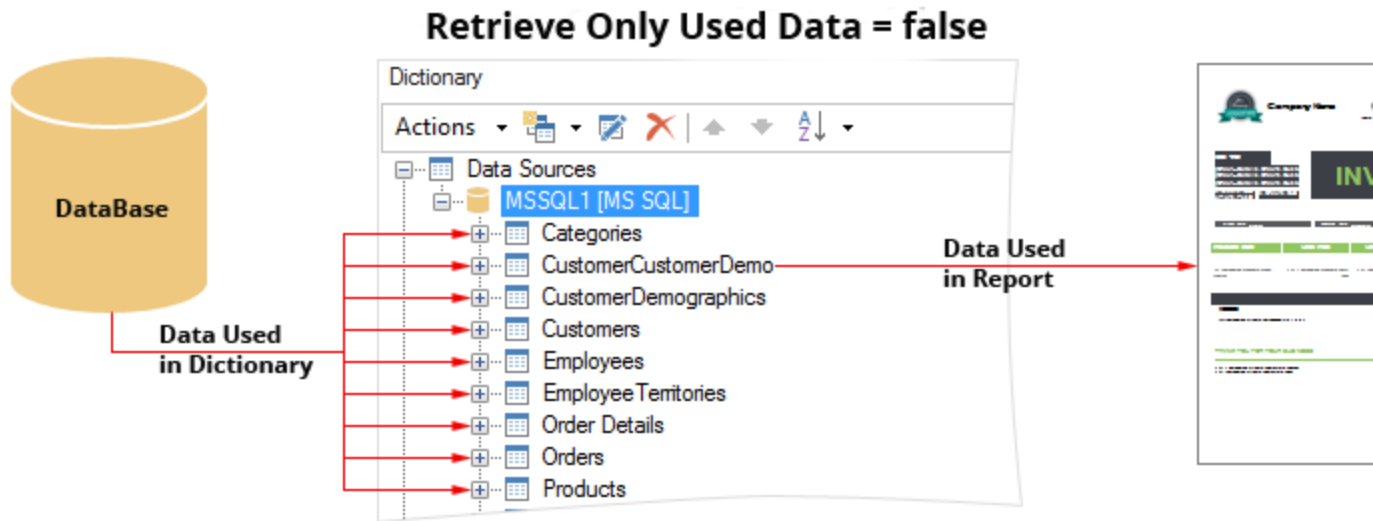
> The **Retrieve Only Used Data** property is set to **true**

In this case, when rendering a report, the report generator will analyze the report structure and transfer data only for tables used in this report. In the current example, the data will be transferred only for the **CustomerCustomerDemo** table. The rendering time of the report, in this case, will be much less. If the report will be used by more than one table, the data will be transferred to the several tables only. The picture below schematically shows the data transfer, if the **Retrieve Only Used Data** property is set to **true**:

Retrieve Only Used Data = true
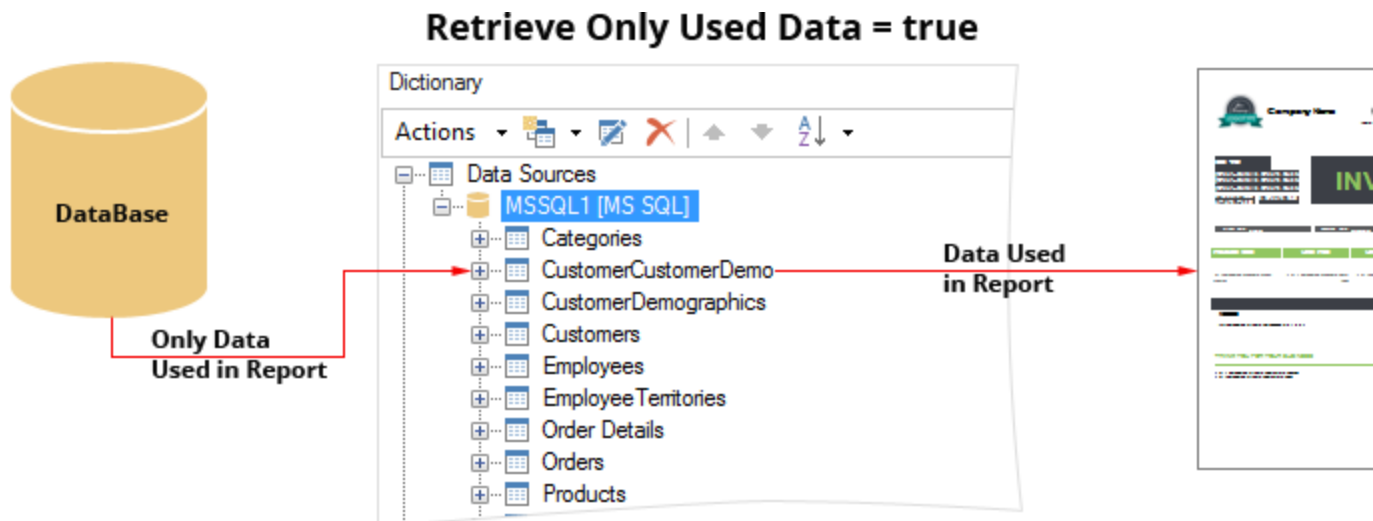
Information: An alternative method is to remove unused data sources from the data dictionary. However, sometimes it is necessary that the whole structure is present. For example, for the further development of the report or, say, when one and the same dictionary is used for a variety of reports.

5.1.3   Relation

Relation is created between data sources and defines how should data from these sources be bind. When creating a relation, keys which play a role of data columns, are indicated. As a result, a relation is a connection between data sources on the basis of one or more key data columns. The Relation provides the ability to filter, sort, display data when accessing the same data source via a relation from another data source. Let's review the following example. The picture below shows two data sources - **Categories** and **Products** (partially):

CategoryID	CategoryName
1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood

CategoryID	ProductName	UnitPrice	UnitsInStock
1	Chai	18	39
1	Chang	19	17
2	Aniseed Syrup	10	13
2	Chef Anton's Cajun Seasoning	22	53
2	Chef Anton's Gumbo Mix	21,35	0
2	Grandma's Boysenberry Spread	25	120
7	Uncle Bob's Organic Dried Pears	30	15
2	Northwoods Cranberry Sauce	40	6
6	Mishi Kobe Niku	97	29
8	Ikura	31	31
4	Queso Cabrales	21	22
4	Queso Manchego La Pastora	38	86

The relation is organized by the key data columns. Key data columns are present in the data sources, among which a relation is organized, and contain the keys. For example,

in **Categories** and **Products** data sources the key columns are **CategoryID**. It should be noted that in this example, the names of key columns are the same, but this is not a prerequisite. The key data column in the data source **Categories** is called **CategoryID**, and the data source **Products** - **CategoryNumber**. Organizing the relation between data sources **Categories** and **Products** by the key columns **CategoryID**, where the data source **Categories** is the master data source, and **Products** is a detail data source. The relation between data sources will have the form as shown in the picture below (partially):

As can be seen, after the organization of a relation, to each entry from the data source **Categories** will be matched to entries from the data source **Products**. In this example, entry Beverages is matched to entries Chai and Chang; entry Condiments is matched to Aniseed Syrup, Chef Anton's Cajun Seasoning, Chef Anton's Gumbo Mix, Grandma's Boysenberry Spread; entry Dairy Products is matched to Queso Carbales and Queso Manchego La Pastora.

5.1.3.1 Filtering

In Stimulsoft Reports it is possible to filter data using relations between data sources. Let's review data filtering via a relation (in the example we use data source Products). If you want to filter data by the category name, i.e. by the entries in the data column **CategoryName** of the data source **Categories**, then, with established relation between data sources **Categories** and **Products**, to add a filter to the expression: **Products.RelationName.CategoryName** == "category name" by which filtering will occur. The picture below shows a window of data filtering via the relation between data sources:
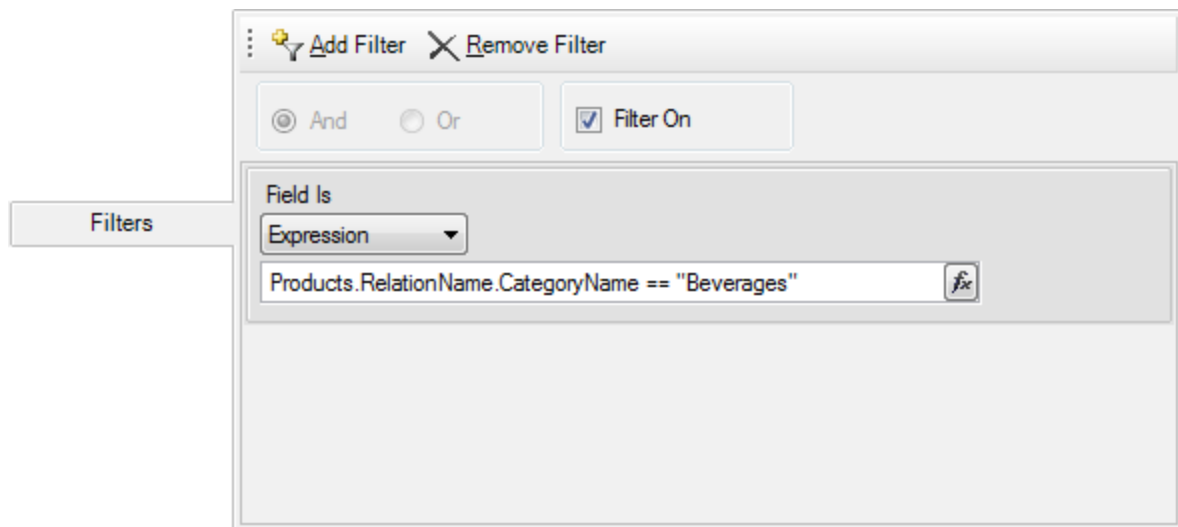
where Products is a data source name; RelationName is a name of the relation between data sources, i.e. reference to another data source vie the relation; CategoryName is a data column in the data source.

Now, when rendering a report, the report generator filters data from the data source **Products** and displays the data that belong to the category **Beverages**. The picture below shows a page of the rendered report:

5.1.3.2 Sorting

When sorting data it can be used not only columns in the specified data source but the columns in the source, which can be accessed vie the relation. Let's review data sorting using a relation (in the example we use data source Products). If you want to sort by

category name, i.e. entries in the data column **CategoryName** of the data source **Categories**, then, with established relation between data sources **Categories** and **Products**, to add sorting to the expression: **Products.RelationName.CategoryName**. You should also select sorting direction. In this example we set the **Ascending** sorting direction. The picture below shows a window of data sorting via the relation between data sources:
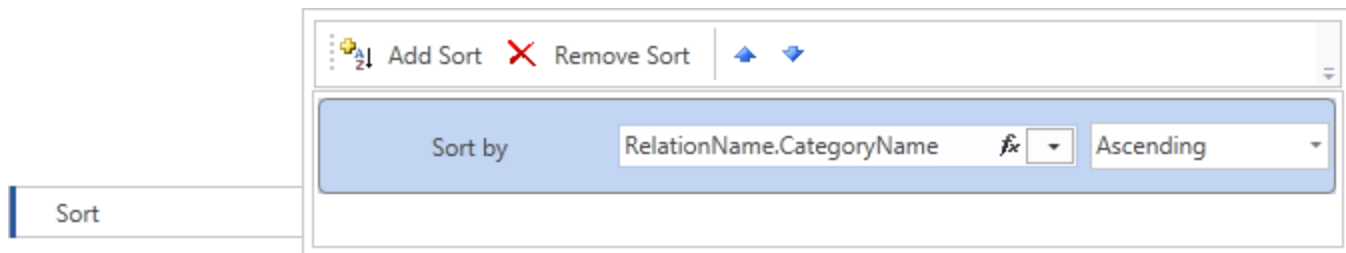
Now, when rendering a report, the report generator will sort data from the data source **Products** by names of the categories in alphabetical order from A to Z. The picture below shows a page of the rendered report:

ProductName	UnitPrice	UnitsInStock
Côte de Blaye	263,5	17
Chartreuse verte	18	69
Steeleye Stout	18	20
Guaraná Fantástica	4,5	20
Sasquatch Ale	14	111
Rhönbräu Klosterbier	7,75	125
Lakkalikööri	18	57
Outback Lager	15	15
Ipoh Coffee	46	17
Laughing Lumberjack Lager	14	52
Chang	19	17
Chai	18	39
Original Frankfurter grüne Soße	13	32
Sirop d'érable	28,5	113
Chef Anton's Gumbo Mix	21,35	0
Northwoods Cranberry Sauce	40	6
Grandma's Boysenberry Spread	25	120
Chef Anton's Cajun Seasoning	22	53
Aniseed Syrup	10	13
Louisiana Hot Spiced Okra	17	4
Vegie-spread	43,9	24
Louisiana Fiery Hot Pepper Sauce	21,05	76
Gula Malacca	19,45	27
Genen Shouyu	15,5	39
Sir Rodney's Scones	10	3
Maxilaku	20	10
Pavlova	17,45	29
Tarte au sucre	49,3	17
Sir Rodney's Marmalade	81	40
Teatime Chocolate Biscuits	9,2	25
Chocolade	12,75	15
Zaanse koeken	9,5	36
Valkoinen suklaa	16,25	65

Beverages — (Côte de Blaye ... Chai)

Condiments — (Original Frankfurter grüne Soße ... Genen Shouyu)

Confections — (Sir Rodney's Scones ... Valkoinen suklaa)

5.1.3.3   Showing Information

Stimulsoft Reports tools can display data from a bound data source. For example, data from columns are displayed in a report: **ProductName**, **UnitPrice**, **UnitsInStock** of the data source **Products**. The picture below shows the a page of the report:

ProductName	UnitPrice	UnitsIn Stock
Chai	18	39
Chang	19	17
Aniseed Syrup	10	13
Chef Anton's Cajun Seasoning	22	53
Chef Anton's Gumbo Mix	21,35	0
Grandma's Boysenberry Spread	25	120
Uncle Bob's Organic Dried Pears	30	15
Northwoods Cranberry Sauce	40	6
Mishi Kobe Niku	97	29
Ikura	31	31
Queso Cabrales	21	22
Queso Manchego La Pastora	38	86
Konbu	6	24
Tofu	23,25	35

If you want to display a category name instead of a product one, and the data column with the names of categories is not present in the data source **Products**, then it can be done using a relation between data sources. To do this, you should change the expression **Products.ProductName** in the text component to the expression **Products.RelationName.CategoryName**. Using the relationship between data sources, the report generator, when report rendering, will take the names of categories from the column **CategoryName** of the data source **Categories**, and substitute them instead of the expression. The picture below shows the a page of the rendered report displaying category names instead of the product name:

ProductName	UnitPrice	UnitsIn Stock
Beverages	18	39
Beverages	19	17
Condiments	10	13
Condiments	22	53
Condiments	21,35	0
Condiments	25	120
Produce	30	15
Condiments	40	6
Meat/Poultry	97	29
Seafood	31	31
Dairy Products	21	22
Dairy Products	38	86
Seafood	6	24
Produce	23,25	35

As can be seen in the picture above, instead of the product names, the category names to which products are related are output.

5.1.3.4 Master-Detail Report

"From the detail via a relation to the master data source" scheme was used in the previous chapters (filtering, sorting, and showing information). When you render a Master-Detail reports a different scheme "from master to detail" is used, i.e. the relation works in reverse order. For example, in the report template DataBand1 is placed in the report template. This band contains a text component with reference to a data column, which contains the categories names. Then, when rendering a report, you will see a list of categories. The picture below shows a report page with the names of categories:

Beverages
Condiments
Confections
Dairy Products
Grains/Cereals
Meat/Poultry
Produce
Seafood

Suppose you want to compare each category from the list to the list of products. To do this, follow these steps:
✔ Add **DataBand2** to the report template;
✔ Specify a data source that contains a list of products and the relation between data sources;
✔ Select the Master component;
✔ Put a text component with reference to a data column from the selected data source in the **DataBand2**. For example, on a data column that contains the name of the product.

And then, when rendering a report, each **Master** entry will be compared to a list of **Detail** entries. The picture below shows a diagram of a **Master-Detail** report:

5.1.3.5   Creating Relation

It is possible to create a relation between data sources in the data dictionary. In order to do this select the item **New Relation** in the context menu of a data source or from the menu **Actions**. The picture below shows a **New Relation** dialog:

As can be seen there are nine fields, which define the relation parameters:

1 In the field **Name in Source** the name of a relation is specified. By this name the relation will be found from the original data (for example in the **DataSet**). If the relation between data sources will be created on the basis of a relation in the DataSet, then this name will coincide with the field **Name**. This field is required to be filled.

2 Filed Name is used to specify the name of a relation which is used to refer to this relation in the report. This field is required to be filled.

3 In the field **Alias** a hint for the relation will be specified and displayed to the user. This field is mandatory.

4 Filed **Parent DataSource** indicates the parent data source for the relation. This field is required to be filled.

5 Filed **Child Data Source** indicates a detail data source for this event. This field is required to be filled.

6 This field displays the selected column from the parent data source. In order to create a relation, you should select the column by which the relationship will be arranged.

7 This field displays the selected column from the child data source. In order to create a relation, you should select the column by which the relationship will be arranged.

8 The **Active Relation** parameter sets the mode of using the current relation by default, for example, when creating a new data transformation.

Information
The editor of relations has the built-in control. In case of issues with creating a relation, the user will see an error message. In this case, you cannot click the Ok button, until the issues are fixed.

5.1.3.5.1 Limitations in Creating Relations

When creating or using relations between data sources, the following restrictions are:

❯ Selected data sources (parent and child) must be of the same type, i.e. types relations should be identical. If the types relations are different, then you can use the **CashAllData** property.

❯ **Name** must be present and correct, in terms of **C#** or **VB.NET** compiler. If the name is reserved in the source, you must add the **@** symbol before the relation name. For example, **@relation**.

❯ Column-keys must comply with all rules of creation a relation to **ADO.NET**:

   ✓ Their number must be the same;

   ✓ Their types must match, so if the primary column-key of the **String** type, then the child column-key must be of the **String** type;

   ✓ Keys must be specified, so the relation cannot be created without keys.

5.1.4   Variables

In Stimulsoft Reports you can use Variables in the report. The Variable is the opportunity to place and use of any value when rendering reports. The values can be of different types: string, date, time, number, array, collection, range, etc. All variables are stored in the data dictionary. Before using a variable in the report, it is necessary to add it to the data dictionary. To add a variable you can select the New variable... in the New

Item of the data dictionary. The picture below shows the New Item menu:
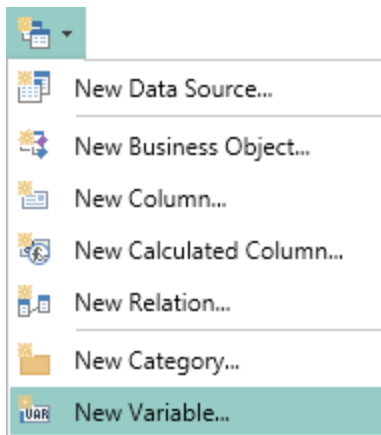
Also, you can create a new variable by selecting New Variable... in the context menu of the Dictionary:
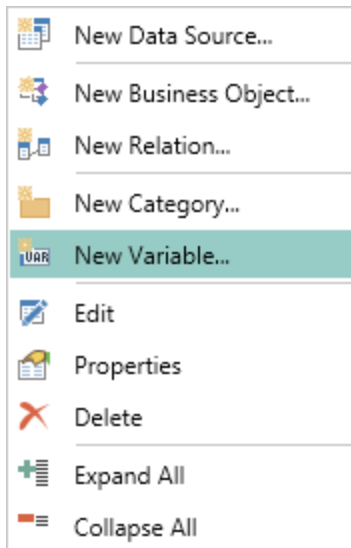
After selecting this item, the New Variable dialog will be called. In this dialog you can set the parameters of the variable. The picture below shows the New Variable dialog:

1 The Name field specifies the name of the variable used in the report.

2 The alias, name of the variable that is displayed to the user, you can specify it in the Alias field.

3 In the Description field, you can specify a description of the variable.

4 In the Type field you can change the type of data that will be placed in a variable, and the type of the variable. This field is represented by two fields with drop-down lists. The first list is a list of all available data types divided into categories:
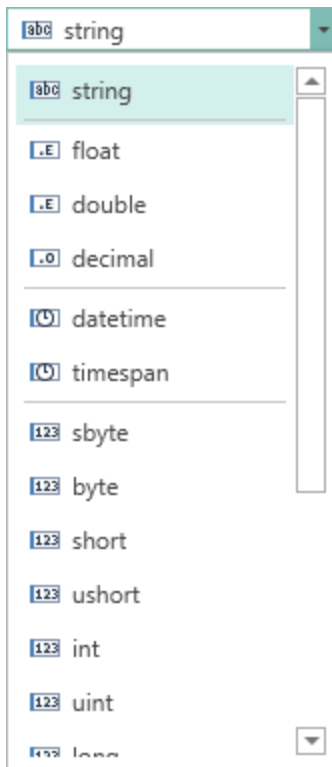
As can be seen from the picture, the integer is selected. The second list contains a list of variables. Depending on the type of a variable, some additional fields of parameters can be displayed. The list of types of variable fields is presented in the second list of the Type field (see. picture above). The picture below shows a list of types of a variable:
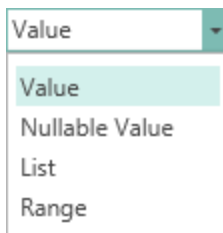
As can be seen from the picture, the variable can be of the following types - Value, Nullable Value, List, Range. Next, consider all types of a variable and the Request from User option in detail.
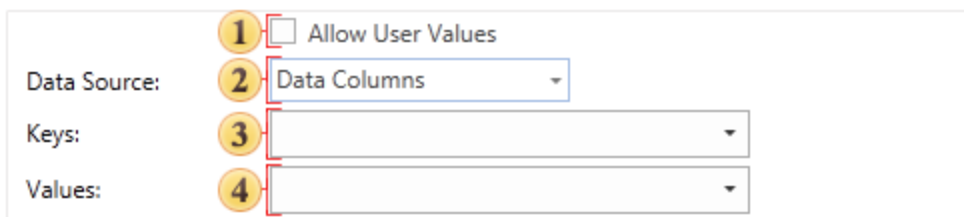
5 The Read Only parameter sets the read-only mode. In this case the value stored in a variable is returned and the user cannot change it. If the value is initialized as an expression then, at the time of treatment to our variable, the expression will be calculated each time.

6 The Request from User parameter establishes a mode under which the returned value can be changed by the user. It should be noted that, if the Request from User is set to true, an additional panel will be displayed. This panel has variable settings that determine the possibility of interaction with the user. In addition, the New Variable dialog can be modified.

7 The Allow using as SQL parameter parameter gives an opportunity to use a variable as a parameter in the query when selecting data.

> **i** Information: When editing a variable, the Save a Copy button will be displayed in the window. When you click on this button, a copy of the edited variable with the Copy postfix in the variable name, will be created.

5.1.4.1   Panel Request From User

The **Request from user** panel contains parameters controls. These parameters determine the possible involvement of the user when using the variable in the report. Some options may present or absent, depending on the value of the **Data Source** field. The picture below shows the **Request from user** panel, if in the **Data Source** field the **Data Columns** value is selected:

1 The **Allow User Values** parameter. Provides an opportunity to set the dialogue mode, i.e. using this variable in a report the user may input values.

2 The **Data Source** field. Contains a drop-down list of values. Depending on the selected value: **Items** or **Data Columns**, on this panel will be fields either **Items**, or **Keys** and **Values**.

3 The **Keys** field. using the ✓, the data column is selected. The entries of the column will be keys.

4 The **Values** field. using the ✓, the data column is selected. The entries of the column will be values.

If the **Data Source** is set to **Items**, then on the **Request from user** panel other options will be located. The picture below shows the **Request from user** panel:
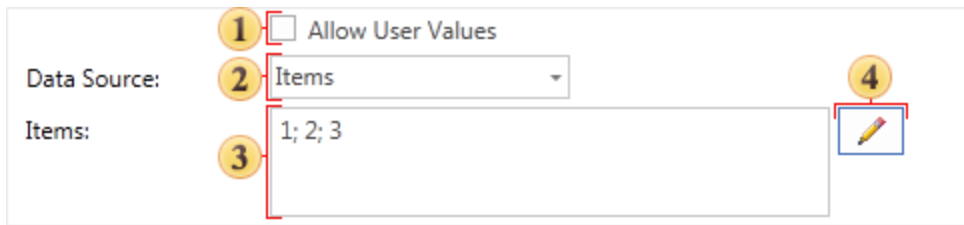
1 The **Allow User Values** parameter. Used to set the dialogue mode, i.e. using this variable in a report the user may input values.

2 The **Data Source** contains a drop-down list of values. Depending on the selected value: **Items** or **Data Columns**, on this panel will be fields either **Items**, or **Keys** and **Values**.

3 The **Items** field. Displays a list of created variable items. If the items are not created, then this field will be blank. It should be noted that the order of items in the list depends on their priority on the list panel in the **Items** dialog, the higher the item is the left its position is in the list, and vice versa.

4 The **Editor** button. Calls the **Items** dialog, where you can create new items, remove existing or edit them.

5.1.4.1.1 Items Dialog

In the **Items** dialog you can create, delete, edit items (values, expressions). This window is invoked when clicking the **Editor** in the **Variables** dialog. The picture below shows the **Items** dialog:
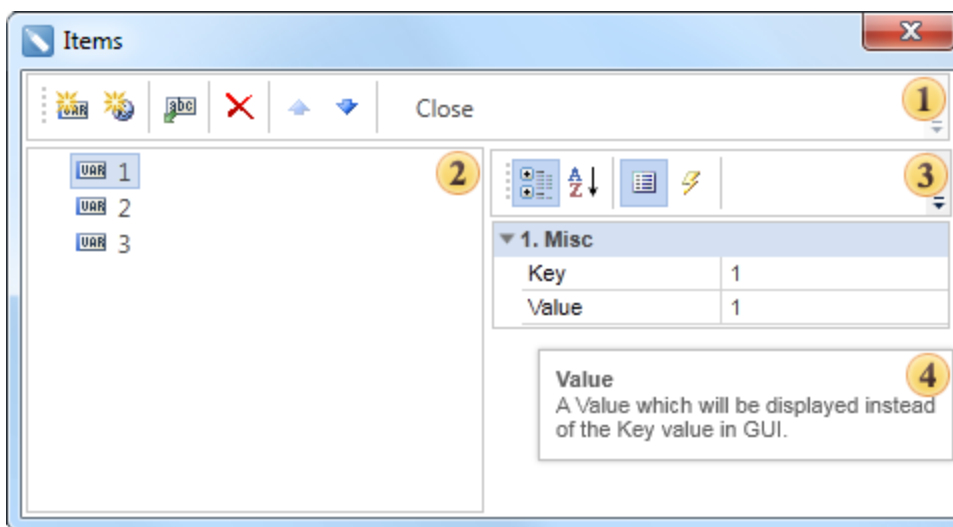
1 Control Panel. This panel contains buttons to control items.

2 In the Toolbox displays a list of created items (values, expressions). Keep in mind that

the order of items in the list affects sequence of items in the **Items** field on the **Request from User** panel.

③ The properties panel. In this panel the properties of the selected item are displayed. The item has two properties: **Key** and **Value**.

④ The panel displays the description of the selected property.

Control Panel

As mentioned above, on this panel (see the picture above) the buttons to control items are placed.

① The **New Value** button. Used to create a new type of the value;

② The **New Expression** button. Creates a new type of an expression;

③ The **Select Columns** button. Calls a dialog where you can specify data columns as keys and values;

④ The **Remove** button. Removes the selected item.

⑤ The **Navigation** buttons. Used to move selected item up or down in the toolbox.

⑥ The **Close** button. Closes the Items dialog saving changes.

5.1.4.1.2 Dependent Variables

When you create a report with parameters, you can use the dependent variables. In this case, one variable will be independent, and the rest ones will depend on it or will represent a hierarchy. Each subsequent variable is dependent on the previous one. To become dependent, the variable must have the checkbox Dependent Value is enabled (it is located on the panel Request From User when you choose a data source Data Column). After you enable the checkbox two fields will be displayed: the Variable and Dependent Column. In the first field, select the variable that will be the main one from which this variable will depend. In the second field select the data column, which will be in relation with the main variable.

This possibility (relations between variables) is useful when using parameters in reports, for example, in Master-Detail reports. Suppose we have a list of categories, each category includes several products, and each product has detailed description. In this case, using the report parameters, the variable by a product and by product information will contain a huge list of values (completely full list of products and descriptions), and, if it is necessary to select a particular product or information on it, this will take much time. If the relations between variables is missing, then the list of category values will contain 8 categories of products - 77 records, and detailed data to several hundreds. It will be almost impossible to find a product or information on it. The images below show examples of lists of values without the relations between the variables:

The dependent variables provides an opportunity to reduce the list of variables. In other words, you can establish a connection among variables. This will lead to filtering the list of values depending on the value of the main variable. For example, depending on the selected category, a list of values of a variable by product is created, and, depending on the selected product, a list of detailed information is created. For example, the category Condiments will be selected, then the list of products will be filtered and will look like this:
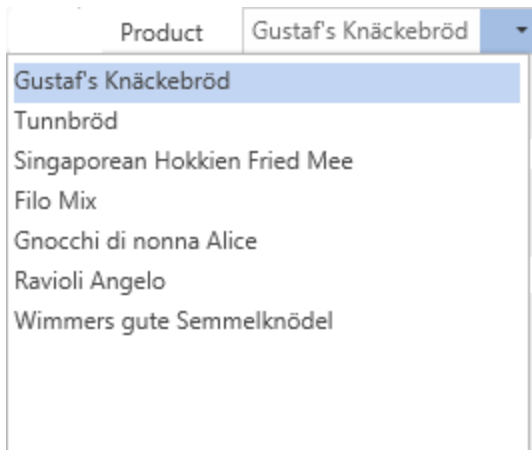
Now select the product Genen Shouyu, and then the list of detailed information will be like this:
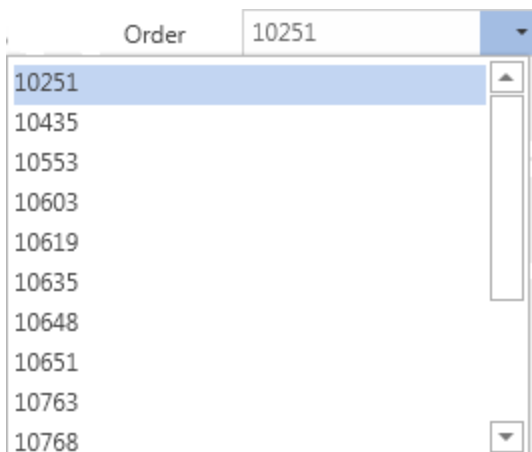
Consider creating and using variables in the report. Create two variables, one of which will contain a list of categories, a second is list of products. And the list of products will depend on the selected category. For example, on the base of data sources from our Demo application.

❯ Create variables Category and Product, of the type Value with initialization of data integer. In the main variable (Category), choose the keys Categories.CategoryID, and the values Categories.CategoryName.

❗ Notice: The key is a unique identifier of a record (row) in the data source. In this case, CategoryID will be a column that contains keys, and ProductID - for products. The connection is organized by keys between the data sources. It is important to understand that different product keys may be related to the same category key.

❯ In the dependent variable define keys Products.ProductID, and the values Products.ProductName. Select the checkbox Dependent Value, select Category as the main variable and data column Products.CategoryID as the dependent column. We go to the tab Preview, as shown in the picture below. It shows two parameters. In the first list the category is selected, and the second list (products) is created depending on the selected category:
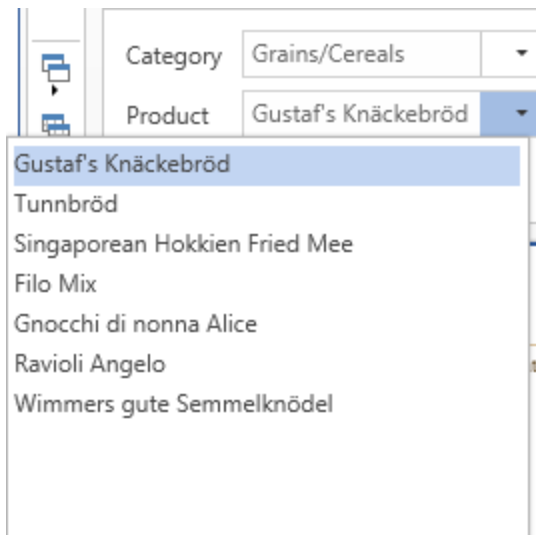
As can be seen from the picture above, the second variable (a list of values) displays not complete list of stored values, but only those values that belong to the selected category.

❯ Add a third variable in the data dictionary. The variable will be named All, of the type Value with initialization data bool.

❯ Now use the dependent variable in the report. Suppose we have a Master-Detail Report, where each category has a few products. Add filters with expressions on Data bands in the report template in order to choose a certain product or products of a certain category:

✔ The first filter is on the data band Master. (this is the band with which a list of categories is created in the report). It is necessary to filter categories, depending on the selected report parameter, so the expression looks like Category == Categories.CategoryID.

✔ Next, add a second filter on the data band Detail (this is the band with which a list of products is created in the report). The filter will have the expression Product == Products.ProductID.

❯ Switch to the tab Preview. In the report parameters select a category, then a product, apply settings to filter report data:

As can be seen from the picture above, the category Grains/Cereals was chosen (note, the key of the category is 5) and the product Gustaf's Knackebrod (product key = 22). In other words, all categories with the key 5 and all the products with the key 22 are displayed.

❯ In order to display a complete list of products related to the category, it is necessary to use the third variable, All. Therefore, you should change the filter expression on the Data band with which to create a list of products (Product == Products.ProductID || All). In this case, depending on the value of the third variable (enabled/disabled) filtering will be done. If the checkbox is disabled, the filter will occur by the product keys (the report shows the product which key matches). If the checkbox is enabled all the products of the selected category will be shown:

As can be seen from the picture above, one category (key = 5) is displayed, and all products related to it, with different keys.

The example that was reviewed above is a single-level dependency. Now consider a more complex example of a two-level dependency. Leave the category, products related to them, and add detailed data by each product. To do this, create the variable Order of the Value type with initialization of data integer. Next, enable the checkbox Request From User, select the data source as the data column.

✔ The column with keys OrderDetails.OrderID, with values OrderDetails.UnitPrice.
✔ Next, set relations with the products. Select Product as a main variable. The dependent column is OrderDetails.ProductID.
✔ Now, in the report template, add the Data band with detailed information on the products. In this example, select Order Details as the data source for the Data band. The Master component will be the Data band with the products. Also indicate the relationship between the data sources.
✔ Add a filter with the expression Order == Order_Details.OrderID in the Data band, which contains detailed information on products.
✔ Go to the tab Preview.

In the report, select a category, and the list of products is filtered. Select the product, and then the list of detailed data for the selected product is filtered. Select a detailed

value, click the button Apply:

If you need to display all the detailed information on the selected product, you should change the filter expression in the Data band with detailed data by products. The expression will be with Variable3 and will look Order == Order_Details.OrderID || All. Now, you can simply specify a category, select a product and get all the detailed information on it:

As can be seen from the picture above one category, one product and all the details by the product were printed. It is also worth noting that the number of nesting levels is not limited.

5.1.4.2   New Variable

The variable of the first type provides the ability to place a simple value of any available data type or expression. Consider the example of creating such a variable. Call the **New Variable...** command. The dialog box in which to define the parameters of the variable will be opened. The Value variable is set by default. The picture below shows the **New Variable** dialog:

① The **Init by** field has a menu with the drop-down list. Depending on the selected item in this menu the type of the value in a variable is defined: Value or Expression, i.e. the method of initializing a variable as a value or expression is selected. In this example, the variable is initialized as a Value.

② This field specifies the value to be stored in a variable. Please note that this field may be missing. If, for example, the Expression is selected in the Init by field, then this field is absent, and the Expression field present instead. In this case, in the Expression field you should specify an expression that will be stored in a variable. In this example, the variable is equal to 2.

After pressing the **Ok** button the variable named **Variable1** will be created. Consider the example of using variable of the type **Value** in the report. Suppose there is a report that contains information about employees (see the picture above).

EmployeeID	LastName	City	Country
1	Davolio	Seattle	USA
2	Fuller	Tacoma	USA
3	Leverling	Kirkland	USA
4	Peacock	Redmond	USA
5	Buchanan	London	UK
6	Suyama	London	UK
7	King	London	UK
8	Callahan	Seattle	USA
9	Dodsworth	London	UK

Add a filter with the expression **Employees.EmployeeID == UNN** in the **DataBand**. Now, when rendering a report, the information about employees whose **EmployeeID** is equal to the value stored in a variable will be output. In this example, **EmployeeID = 2**. The picture below shows a report with the condition of filtering:

EmployeeID	LastName	City	Country
2	Fuller	Tacoma	USA

5.1.4.3 Nullable Value

The **Nullable Value** variable provides the ability to place simple values and values equal to **null**. If it is necessary to return a **null** value in the report, then when using a variable of another type, the report compilation error occurs. The picture below shows the **New Variable** dialog of the **Nullable Value**:

❶ The **Init by** field has a menu with the drop-down list. Depending on the selected item in this menu the type of the value in a variable is defined: Value or Expression, i.e. the method of initializing a variable as a value or expression is selected. In this example, the variable is initialized as a Value.

❷ This field specifies the value to be stored in a variable. Please note that this field may be missing. If, for example, the Expression is selected in the Init by field, then this field is absent, and the Expression field present instead. In this case, in the Expression field you

should specify an expression that will be stored in a variable. In this example, the variable is equal to 2.

5.1.4.4 List

The **List** variable provides the ability to place a list of values of any available data type. In contrast to the **Value** variable, in this case, when report rendering, the variable contains a list of values. The picture below shows the **New Variable** dialog with the selected **List** type:

After clicking OK, a variable named **UNN** and the stored list of values from 0 to 8 will be created. Consider using a variable created in the report. Suppose there is a report that contains numbers, names and descriptions of categories. The picture below shows a report page:

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

If you want to show some of the categories then use already created variable in the report. To do this, add a filter in the **DataBand** with the expression **UNN.Contains(Categories.CategoryID)**, where **UNN** is the variable name. When rendering a report, by default, all categories are displayed. All values in the list of stored values of the variable are selected. Also,values, for example **Grains/Cereals** and keys, for example [**5**] are displayed in the variable list . The picture below shows a list of variable values:
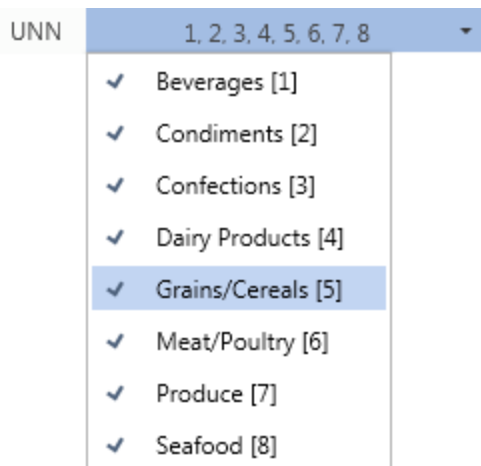
Because the **Allow User Values** parameter is not enabled, in this example, the user can only select values, stored in the variable, but cannot use their own values. Suppose the values such as **Beverages [1]**, **Confections [3]**, **Produce [7]** will be selected. Then, after clicking the **Submit** button, the generator will build a report, considering the filtering conditions and display entries **1**,**3**,**7**. Below is a report using a variable is shown:

5.1.4.5 Range

If using a variable of this type in the report, you can work with ranges of values. In this case, the variable will store a range of values. The picture below shows the New Variable dialog of the **Range** type:

❶ The **Init by** field has a menu with the drop-down list. Depending on the selected item in this menu the type of the value in a variable is defined: Value or Expression, i.e. the method of initializing a variable as a value or expression is selected. In this example, the variable is initialized as a Value.

❷ The **From** field. Specifies the starting value of the range. The value in this field is included into the values range. In our case the date **01/01/2008; 00:00:01** is specified.
❸ The **To** field. Specifies the ending value of the range. The value in this field is included into the values range. In our case the date **12/31/2008; 23:59:59** is specified.

After clicking **OK**, the variable will be created. Here is an example of this type of the variable in the report. Suppose there is a report that contains information about orders: country, name and date of delivery. The picture below shows a report page:

ShipCountry	ShipName	ShippedDate
Germany	Die Wandernde Kuh	12/15/2008 1:00:00 PM
Germany	Die Wandernde Kuh	5/23/2010 1:00:00 PM
Germany	Die Wandernde Kuh	10/23/2009 1:00:00 PM
Germany	Die Wandernde Kuh	11/2/2008 1:00:00 PM
Germany	Die Wandernde Kuh	5/28/2009 1:00:00 PM
Germany	Die Wandernde Kuh	9/18/2009 1:00:00 PM
Germany	Die Wandernde Kuh	9/27/2009 1:00:00 PM
Germany	Die Wandernde Kuh	10/17/2008 1:00:00 PM
Germany	Die Wandernde Kuh	12/27/2008 1:00:00 PM
Germany	Drachenblut Delikatessen	6/4/2010 1:00:00 PM

If you want to display information about orders, which were processed in 2008, then use the variable created in the report. To do this, add a filter in the DataBand with the expression **Orders.ShippedDate** > **Variable1.FromDate** & & **Orders.ShippedDate** < **Variable1.ToDate**. When rendering a report, you will see only the information about orders that were processed in 2008. Below is a report with orders in 2008:

ShipCountry	ShipName	ShippedDate
Germany	Die Wandernde Kuh	12/15/2008 1:00:00 PM
Germany	Die Wandernde Kuh	11/2/2008 1:00:00 PM
Germany	Die Wandernde Kuh	10/17/2008 1:00:00 PM
Germany	Die Wandernde Kuh	12/27/2008 1:00:00 PM

It is worth noting that when referring to the start/end range value, if the **DateTime** data type is used, then to avoid additional changes, you can address to the **VariableName.FromDate** (or **VariableName.FromTime** if the **TimeSpan** data type is used) and **VariableName.ToDate** (or **VariableName.ToTime** if the **TimeSpan** data type is used).

5.1.4.6   Three Modes of Variable Functioning

Depending on the selected parameters the variable in the report can be operated in the following modes: autonomous, user (with selecting values), user (with inputting values). Let us consider these modes in more detail.

Autonomous

This mode will be applied if the **Request from User** parameter is disabled, i.e. using a variable in the report, no action will require from the user. Create a variable that will store the value 2 of the integer type with the name **UNN**. Use this variable in the report. The picture below shows an example of the rendered report:

CategoryID	CategoryName
1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood

Add a filter in the **DataBand**, where specify the expression **Categories.CategoryID == UNN** as the filtering condition. Now when rendering a report, the report generator will consider the filtering condition and display only those entries which values in the column **CategoryID** be equal to the values, stored in the variable. In this case, it is the entry Condiments. The picture below shows an example of a report using a variable to filter data:

CategoryID	CategoryName
2	Condiments

In this case, when rendering a report, no action will require from the user.

User (with selecting values)

This mode of operation of the variable will be used if the **Request from User parameter** is enabled and the **Allow Users Values** is disabled. If using this variable in the report, there may need some actions from the user for selecting values from a variable list. Create the variable **UNN**, which will store a list of items from 1 to 8. Use

this variable in the report. The picture below shows an example of the rendered report:

Add a filter in the **DataBand**, where the expression **Categories.CategoryID == UNN** is a filtering condition. Now, when report rendering, the value from the list will be selected in the viewer window. The picture below shows a list of variable values:

After selecting the value, click the **Submit** button to apply the selected value or the **Reset** button to reset the initial value in the list. The picture below shows the variable panel in the report:

When clicking the Submit button, the report generator will filter data and display these data, which **CategoryID** is equal to the selected value. The picture below shows an example of a report with the selected value **4**:

The **Reset** button resets the current value and sets the first top value from the variable list.

User (with inputting values)

This mode of the variable will be applied if the **Request from User** and **Allow Users Values** is enabled. When using this mode, selecting or entering values in the variable field will require from the user. Create a variable type of **List** with the name **UNN**, and specify the column **CategoryID** as keys and values. The picture below shows an example the rendered report:

Add a filter in the **DataBand**, where as the filter condition, specify the expression **UNN.Contains(Categories.CategoryID)**. Now, when rendering a report, it is necessary to edit the list of values of the variable (remove unwanted items, or change the key in the item field, or create a new item) in the viewer window. The picture below shows an edited list of the variable:
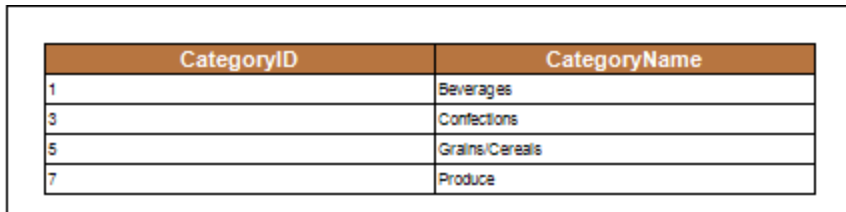
❶ The **New Item** button. Creates a new item with the field in which to specify a key;
❷ The **Remove** buttons. Remove the item to which they belong. Each item in the list has such a button.
❸ The **Remove All** button. Removes all items from the list;
❹ The **Close** button. Closes this menu saving items and input keys.

After that, click the **Submit** button. Now the report generator will filter data and display the data which the **CategoryID** is equal to keys specified in the fields in the list of the variable values. The picture below shows the filtered report:

CategoryID	CategoryName
1	Beverages
3	Confections
5	Grains/Cereals
7	Produce

The **Reset** button, in this case, resets the current list of values to the original one.

5.1.5   Connection

The **Connection** object of the data dictionary describes the report parameters that are used to retrieve data from the database. Stimulsoft Reports supports many types of connection object. All types can be divided into two groups: built-in ones that are included into the report generator, and external data adapters that can be downloaded from our website Database Packs. These packs can be used only for the reporting tools of the product line Stimulsoft Reports.

Built-in data adapters

To create a new built-in connection it is necessary to call the **Select Type Of Connection** dialog. This window can be opened from the **Dictionary** tab, selecting **New Connection...**, as well as from the **New Data Source** dialog by clicking the **New Connection...** After selecting the connection type, press the **OK** button. Depending on the type of connection a dialogue form will be displayed. If you choose an XML connection type, then the following dialog will appear (see the picture below - New Xml Data. Fill the Name, the path to the XSD schema and XML data. Also specify the connection alias.
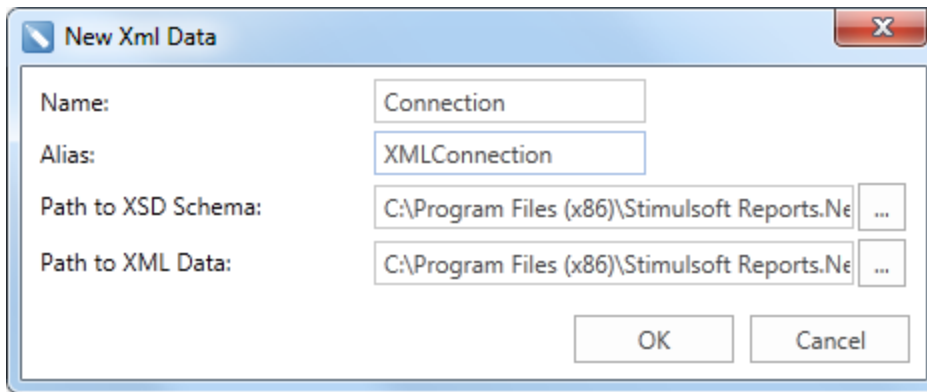
If to choose any other type of connection, the dialog box will be a **New** *type* **connection**, in which set the connection **Name** and **String**. Also specify the connection **Alias**. The picture below shows the **New OleDB Connection** dialog box:
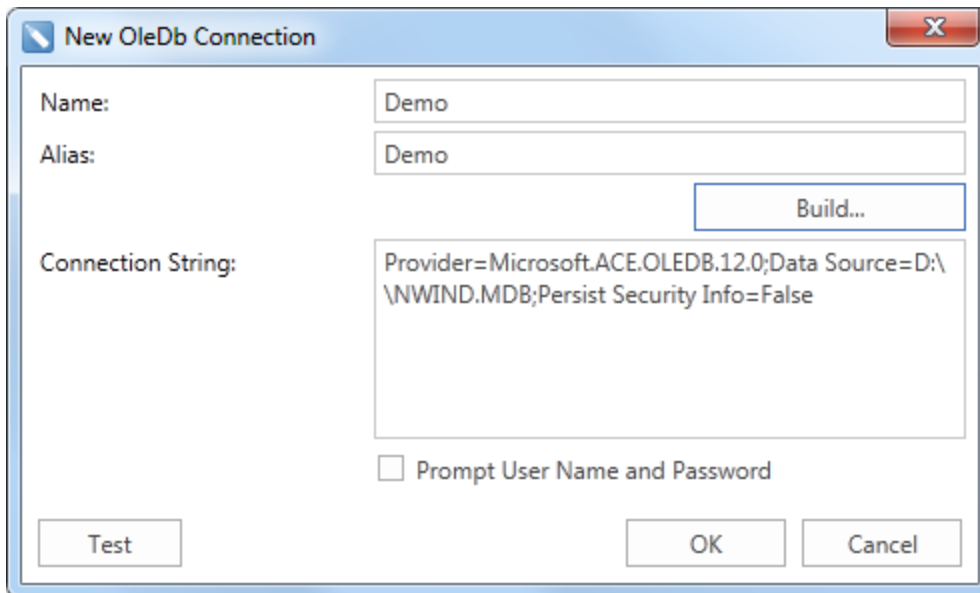
There is a list below with built-in connection types:

❯ **SQL** - this connection describes the parameters to access the Microsoft SQL Server database;

❯ **OleDB** - connection describes the parameters to access databases via the OleDB driver;

❯ **ODBC** connection describes the parameters to access databases via the ODBC driver

❯ **Xml** connection describes the parameters to access XML files;

❯ **MS Access** connection describes the parameters to access the MS Access database;

The picture below shows the **Select Type Of Connection** window:
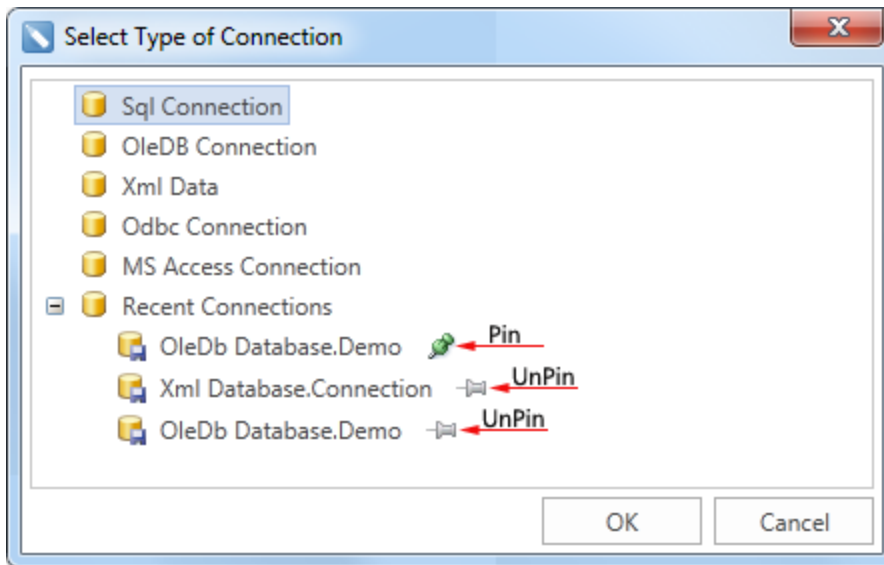
Recent Connection tab

Also in the **Select Type Of Connection** window we can find a **Recent Connections** folder, which contains previously established connections. At the same time it can contain up to 15 connections. When creating subsequent connections, the first connection will be overwritten and so on. If you need the connection never be overwritten, set write protection for it, you should click the Pin icon. In order to remove the write protection, you must click UnPin icon (see the picture above). When selecting a connection from the **Recent Connections** folder, the next dialog box is **New xml Data** when choosing previously created **xml** connection, or **New** *type* **connection**, when any other type is chosen, with already filled fields. If necessary, empty fields may be edited.

External adapters

In addition to the basic types of connections, there are also external data adapters that provide connection to the following databases:

- ✓ **Firebird**;
- ✓ **IBM Db2**;
- ✓ **MySQL Connector.Net**;
- ✓ **MySQL CoreLab**;
- ✓ **Oracle**;
- ✓ **Oracle Data Provider for .NET**;
- ✓ **PostgreSQL**;

✓ **PostgreSQL CoreLab**;
✓ **Sybase Advantage Database Server**;
✓ **Sybase Adaptive Server Enterprise**;
✓ **SqlCe**;
✓ **SQLite**;
✓ **VistaDB**;
✓ **Uni Direct**;
✓ **dot Connect Universal**;
✓ **Informix**;
✓ **EffiProz**.

Consider the example of creating a connection to an external data adapter. Download the external data adapter from our website. In our example, we downloaded the MySQL Connector.Net adapter. Unpack the archive into a temporary directory and run the project. Add references to assemblies **Stimulsoft.Report.dll**, **Stimulsoft.Controls.dll**, **Stimulsoft.Base.dll** and **Stimulsoft.Editor.dll** in the running project and compile the project. Copy the compiled **dll** files to the **bin** folder, and in the beginning of the program add the following code:

StiOptions.Services.DataAdapters.Add(new
Stimulsoft.Report.Dictionary.StiTestXAdapterService());
StiOptions.Services.DataBases.Add(new
Stimulsoft.Report.Dictionary.StiTestXDatabase());

To attach an assembly file to **Designer.exe**, place this assembly file in the same directory in which the **Designer.exe** is located. Furthermore, it should provided an access to a data provider assembly. Thereafter, in the **Select Type Of Connection** dialog a new type of connection will be available, in our case, **MySQL Connector.Net**. There are no restrictions on the number of connections created for various types of data sources in report generator.

5.1.6   Panel Setup

The panel (see the picture below) contains controls that provide an opportunity to change auxiliary parameters of the data dictionary.

1 If the option **Create Field on Double Click** is enabled, then when double clicking the data column data in the report data dictionary, the report template in the DataBand will create a text component with reference to this data column;

2 The parameter **Create Label** is used to create two text components (one with the signature, the a second with reference to the data column) when dragging a data column into the report. If this option is disabled, then, when dragging, only one text component with reference to a data column will be created;

3 In order to show the alias instead of the name, enable the option **Use Aliases**. If this option is disabled, it will display a name of the element.

5.1.7   System Variables

Stimulsoft Reports offers to use system variables in expressions. System variables are variables which provide information about the current status of a report. The following system variables are available:

➤ **Column** – Returns the current column number (starts from 1);

➤ **Line** – Returns the current line number (starts from 1). Used for numbering lines in reports. Numbering for each group goes separately;

➤ **LineThrough** – Returns the current line number (starts from 1). Unlike the **Line** variable it returns lines from the beginning of the report, without report groupings;

➤ **LineABC -** Returns the alphabetical analog of the current line number;

➤ **LineRoman -** Returns the current line number in Roman numerals;

➤ **GroupLine** - Returns the current group line number (starts from 1);

➤ **PageNumber** – Returns the current page number (starts from 1). Used for numbering pages;

➤ **PageNumberThrough** - Returns the current page number (starts from 1). When the **PageNumberThrough** is used, the **ResetPageNumber** property is ignored and numbering starts from the beginning of a report.

➤ **PageNofM** – Returns a localized string, showing "Page N of M" where N is the current page number and M is the **TotalPageCount** of a report:

Page **{PageNumber}** of **{TotalPageCount}**

This variable is a combination of system variables **PageNumber** and **TotalPageCount**, i.e. will output the sequence number of a page in respect of the total number of pages.

❯ **PageNofMThrough -** Returns a localized string, showing "Page N of M" where N is the current page number and M is the TotalPageCount of a report. When the **PageNofMThrough** property, the **ResetPageNumber** property is ignored and numbering starts from the beginning of a report.
❯ **TotalPageCount** – Returns the number of pages in a report;
❯ **TotalPageCountThrough -** Returns the number of pages in a report;
❯ **IsFirstPage** - Returns true, if, in the current moment, the first page of a report is printed;
❯ **IsFirstPageThrough -** Returns true, if, in the current moment, the first page of a report is printed. When calculating the **IsFirstPageThrough**, all **ResetPageNumber** properties are ignored and numbering starts from the beginning of a report. For correct calculation of a variable it is required to execute two passes.;
❯ **IsLastPage** - Returns true, if, in the current moment, the last page of a report is printed. For correct calculation of a variable it is required to execute two passes;
❯ **IsLastPageThrough -** Returns true, if, in the current moment, the last page of a report is printed. When calculating the **IsLastPageThrough**, all **ResetPageNumber** properties are ignored and numbering starts from the beginning of report. For correct calculation of a variable it is required to execute two passes.;
❯ **ReportAlias** - Returns the alias of a report. You can change the **ReportAlias** with help of the **ReportAlias** property of a report;
❯ **ReportAuthor** - Returns the author of a report. You can change ReportAuthor with help of the ReportAuthor property of a report;
❯ **ReportChanged** -The Date when a report was changed;
❯ **ReportCreated** - The Date when a report was created;
❯ **ReportDescription** - Returns the description of a report. You can change the ReportName with help of the ReportDescription property of a report;
❯ **ReportName** - Returns the name of a report. You can change the ReportName with help of the ReportName property of a report;
❯ **Time** – Returns the current time;
❯ **Today** – Returns the current date;

5.1.8   Business Object

A **Business object** is an object of the data class that can be used to represent data in

various structures: tables, lists, arrays, etc. In order to create a description of the business object in the data dictionary, you need to select **New Business Object...** in the context menu of the data dictionary or in the menu **New Item**. After selecting this command the first dialog box of New Business Object will be opened. The picture below shows the first dialog box New Business Object:
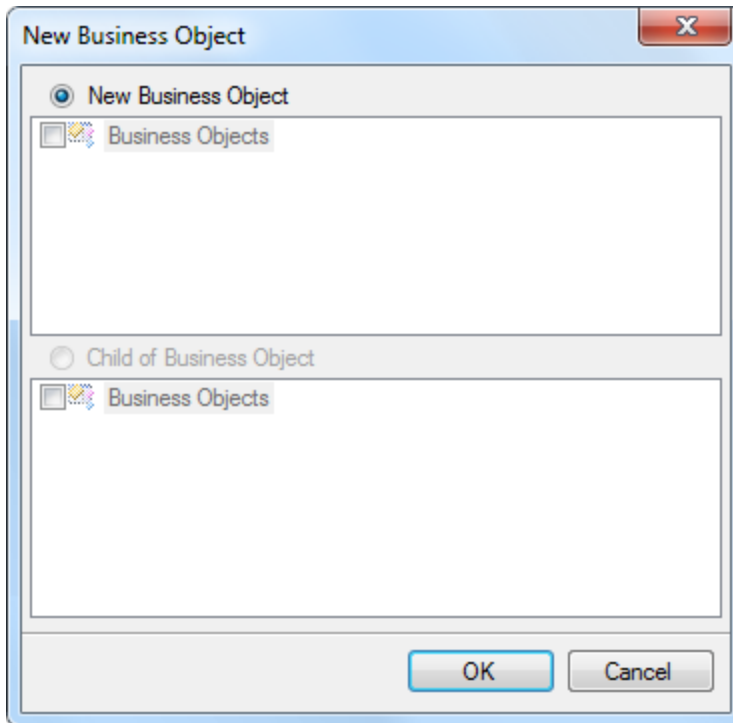
It should be noted that a child business object can be created for each business object. To do this, select the business object and call the command **New Business Object...** Then, the first dialogue box of New Business Object will be called, in which the option Child of Business Object will be checked. After you click OK in that dialog box, the second dialog box will be opened. There you should specify the parameters of the new business object. The picture below shows the second dialog box of **New Business Object**:
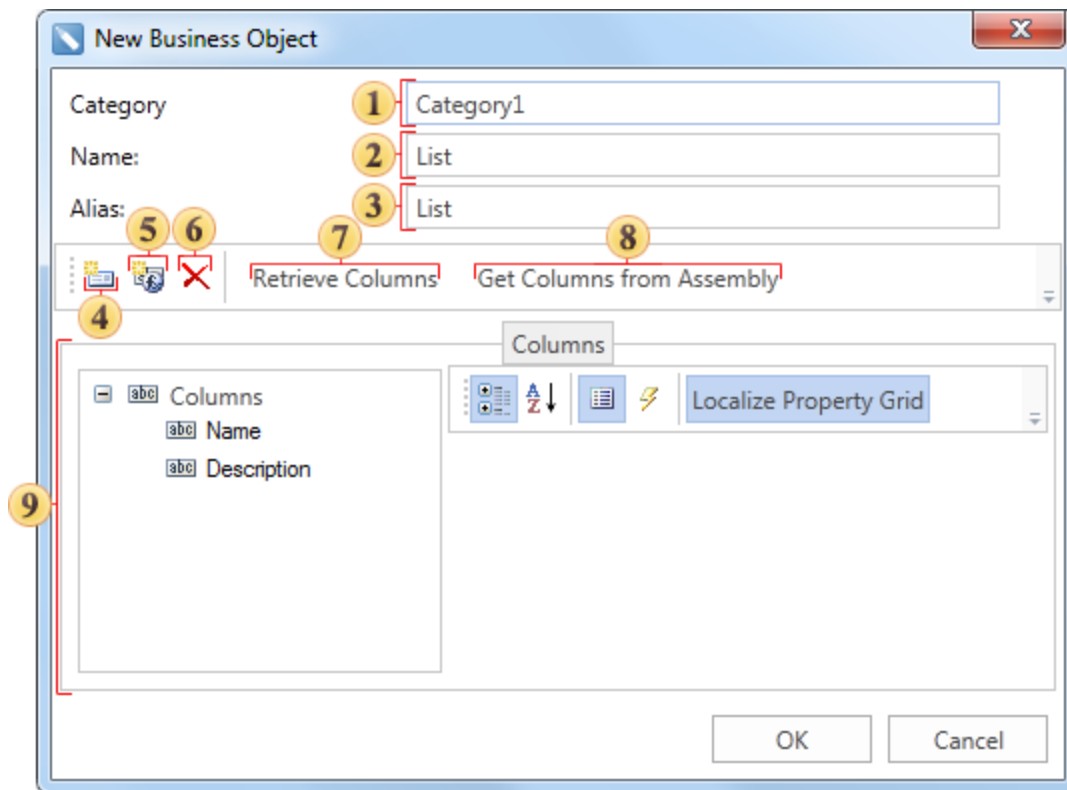
① In the field **Category** you can specify the name of the category. If this field is filled, then the category of business objects in the report dictionary will be created. If the field is left blank, the category will not be created. When you create a child business object this field is not editable.

② The field **Name** is used to specify the business object. This field must be filled and, in this case, the name List is used.

③ The field **Alias** specifies the alias of the business object. If it will not be changed by the user, then, by default, the alias is the same as the name of the business object. In our case, it is List.

④ The button **New Column**. When you click it, a new data column will be created in the business object. It should be noted that the data column created this way is the virtual one, and does not contain actual data.

⑤ When you click the button **New Calculated Column**, a new calculated column will be inserted into the business object.

⑥ When you click the button **Delete**, the selected data column will be deleted. If the tab Columns is selected, it will remove all the columns, which are located in the tab.

⑦ The button **Retrieve Columns** is used to get a data column from the business object.

⑧ The button **Get Columns from Assembly** will open the dialog Open Assembly, in which you select an assembly file. After selecting the file, press the button Open and data columns (if they are present there) will be extracted from that file.

9 The panel **Columns** has three fields. These fields show a list of columns, their properties and description

! **Note**. The Business object created this way does not contain actual data. Therefore, when rendering a report using this business object the error will occur. The Business object with the real data is generated and passed from the code.

5.1.9 Resources

> **YouTube**
>
> Please watch video tutorials for working with resources.

Resources are files that can be embedded in a report template. The following can be added as resources:
> Data files - CSV, Excel, JSON, XML, DBF;
> Image files - SVG, JPEG, JPG, PNG, BMP, GIF and other image files.
> Report templates (*.mrt, *.mrz) and built reports (*.mdc, *.mdz).
> RTF files.

You should do the following to embed a file into a report:
> Select the **New Resource...** command in the **New Item** menu or in the context menu of the report **Dictionary**.
> Drag the file from the explorer to the data dictionary. When dragging a data file, you should select the **New Resource** item.

After selecting this command, the menu for creating a new resource will be displayed:
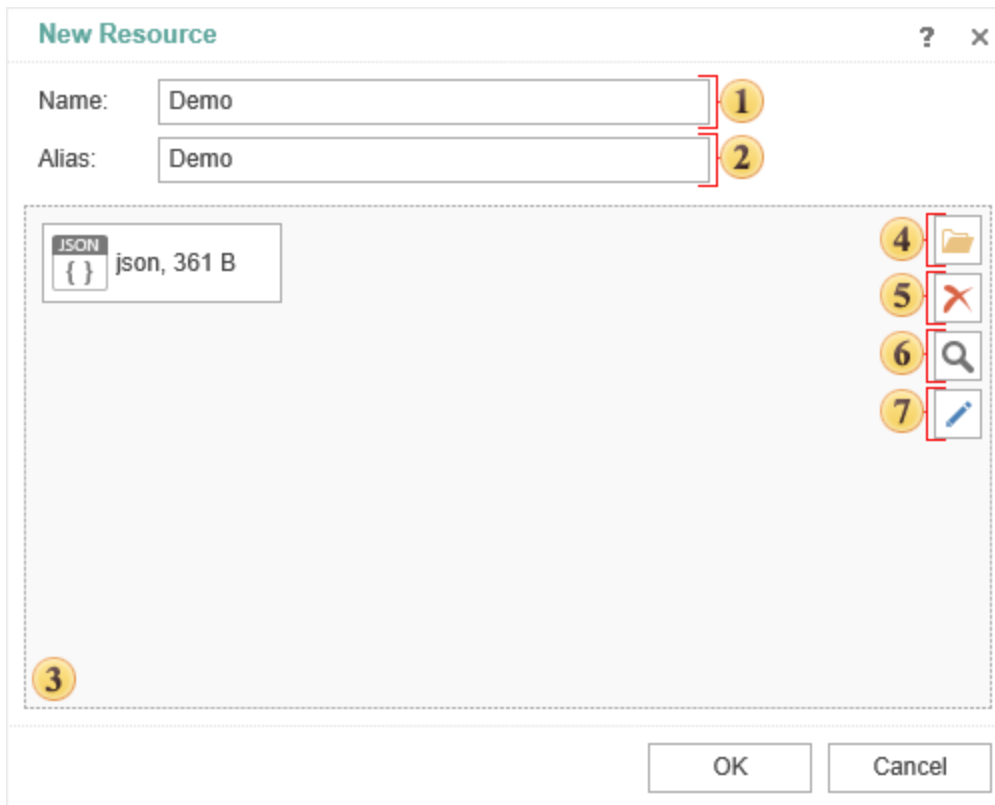
1 This field specifies the **Name** of the resource;

2 This field specifies the **Alias** of the resource;

3 This field displays the selected file which will be loaded as a resource;

4 The button is used to call the explorer to select the file you want to upload to the report;

5 The button is used to delete the selected file;

6 The button is used to view the selected file;

7 The button is used to call the text editor to change the selected file. However, the command is available only for files that can be edited with the text editor. For example, if you select an Excel file, this command will not be available.

Information

When you embed a big file into a report with data or images, and when you add multiple resources, the size of the *.mrt file can be significantly increased.

Saving a resource from a report

To save a resource from the report designer, you should select the resource in the Data Dictionary and click **Save** in the context menu. In this case, a dialog will be called in which it is necessary to specify the location for saving the file. Then, click the **Save** button and the resource will be saved in the format of the source file. In addition, if a *.mrt file was added to the resource, then the *.mrt file will also be saved when this resource is saved.

Resource based data source

When designing reports, data files (CSV, Excel, JSON, XML, DBase) are often used. Based on these files, you can create data sources in the data dictionary that will be used to create report templates. In this case, the data sources will not contain real data but only a description of the methods, parameters and methods to access to real data. The transfer of data from the file to the data sources, and accordingly the filling of the actual data of the report, occurs when rendering of this report.

In this case, you should always consider the specified path in the data source to CSV, Excel, JSON, XML, DBase files, and, if necessary, edit them. Also, if you want to transfer the report template to another user, you should provide a data file to correctly render this report.

In such cases, when you create reports, you may add data files (CSV, Excel, JSON, XML, DBF) to the report resources.

After the data file is added to the report as a resource and based on it, you can create a data source:
❯ Select the **New Data Source...** in the **New Item** menu or in the **Data Dictionary** context menu, define the appropriate source type by specifying the path to the resource or simply by selecting it from the resource gallery.
❯ Select the resource in the data dictionary and select **New Data Source [Resource Name]** from the context menu.

Below is the menu for creating an Excel data source:

① This field specifies the **name** of the data source.

② This field specifies the data source **alias**.

③ This field specifies the path to the Excel file that contains the data. In this case, a link to the resource in the report is specified. The link can be specified manually using the template **resource://file name** or the link will be generated automatically when the resource is selected from the resource list.

④ The parameter of using the first line in the Excel file as a header for the data. If it is enabled, the first line will be the header for the data.

⑤ A resource gallery, on the base of which you can create a data source of a certain type.

After clicking **Ok**, the generated data source can be used to create reports.

Images from resources

Images in reports can be obtained from various resources - uploaded directly, from a file, from a data source, by a hyperlink, etc. When you send a report to another person or move the report (or images) to another medium, you will have to send (along with the report) images, editing the path to these images. Except the cases when the image is uploaded directly to the **Image** component. However, each time you load an image into the Image component, the size of the report file is increased by the size of the image file.

Therefore, if the same image is used in several **Image** components or in a watermark

for various report pages, it is better upload this image into a resource. Then, in the Image or Watermark component, you need to specify a link to this resource. Also the added image in the resources will be displayed in the image gallery of the Image or Watermark component. If the image is uploaded to the resource, the size of the report file will grow only by the size of the image file, and when sending to another person (or when the report is moved to another media), no additional editing of the Image components is required.

After adding an image to the **Resource**, it can be used as a watermark of the report or in the **Image** component.

Sub-reports from Resources

The Sub-report component is used to display another report on the same report page within this component. This other report is the report which is nested one can be located on another page in this report template or in another report template file. Using the Sub-report component, you can also display the rendered report.

If the report you want to display in the Sub-report component is another file (*.mrt, *.mrz, *.mdc, *.mdz), you can add it to the report resources. After adding to the resources, you can:
❯ Drag the resource to the report page. In this case, the Sub-report component with a link to this resource will be created.
❯ Add the Sub-report component to the report template. When editing this component, you should specify a link to the resource.
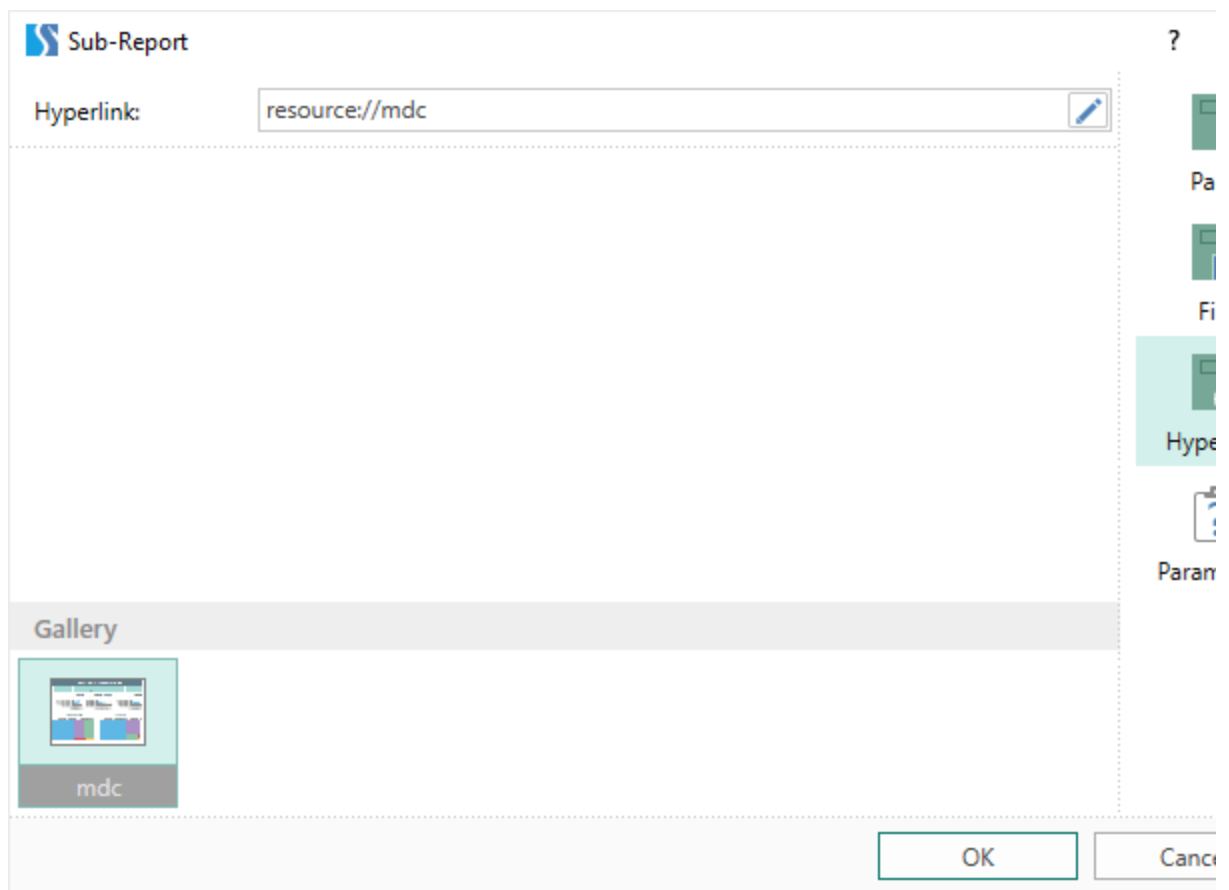
Also, you can pass a parameter in the Sub-report component editor. For example, to filter data in a nested report. However, this is only relevant for the not rendered report

(*.mrt, *.mrz).

Rich text from resources

Sometimes you need to use Rich text in your reports. There is a special component to display this text in the report designer – Rich Text. You can add the Rich text to the report with next ways:

❯ Enter text in the Rich Text editor. In this case, you will have to edit the text formatting.

❯ Specify a file or hyperlink from where the text will be obtained. In this case, when you move a report or file, you may have to edit the path to the source text.

Therefore, one of the options is to add a Rich text file to the report resources. In order to output the Rich text from resources:

❯ You need to drag the resource to the report template;

❯ In the Rich Text editor, you need to specify a link to the resource or simply select a resource from the gallery.

Also, if necessary, the text obtained from the resources can be edited in the Rich Text editor.

6 Report Designer

The report designer is a standalone application that is a part of Stimulsoft Reports product and is used to create and modify reports. The UI of this component provides the user with a great set of tools, components, and tools to develop reports, visually design and preview them.

6.1 Ribbon UI

Ribbon interface 2013, based on the tabs (similar to **Microsoft Office 2013**). The tabs are grouped instruments, thereby reducing the number of toolbars. Consider this type

of interface in more detail.

6.1.1   Main menu

The basic **Ribbon** 2013 interface is a **Ribbon** 2013 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.
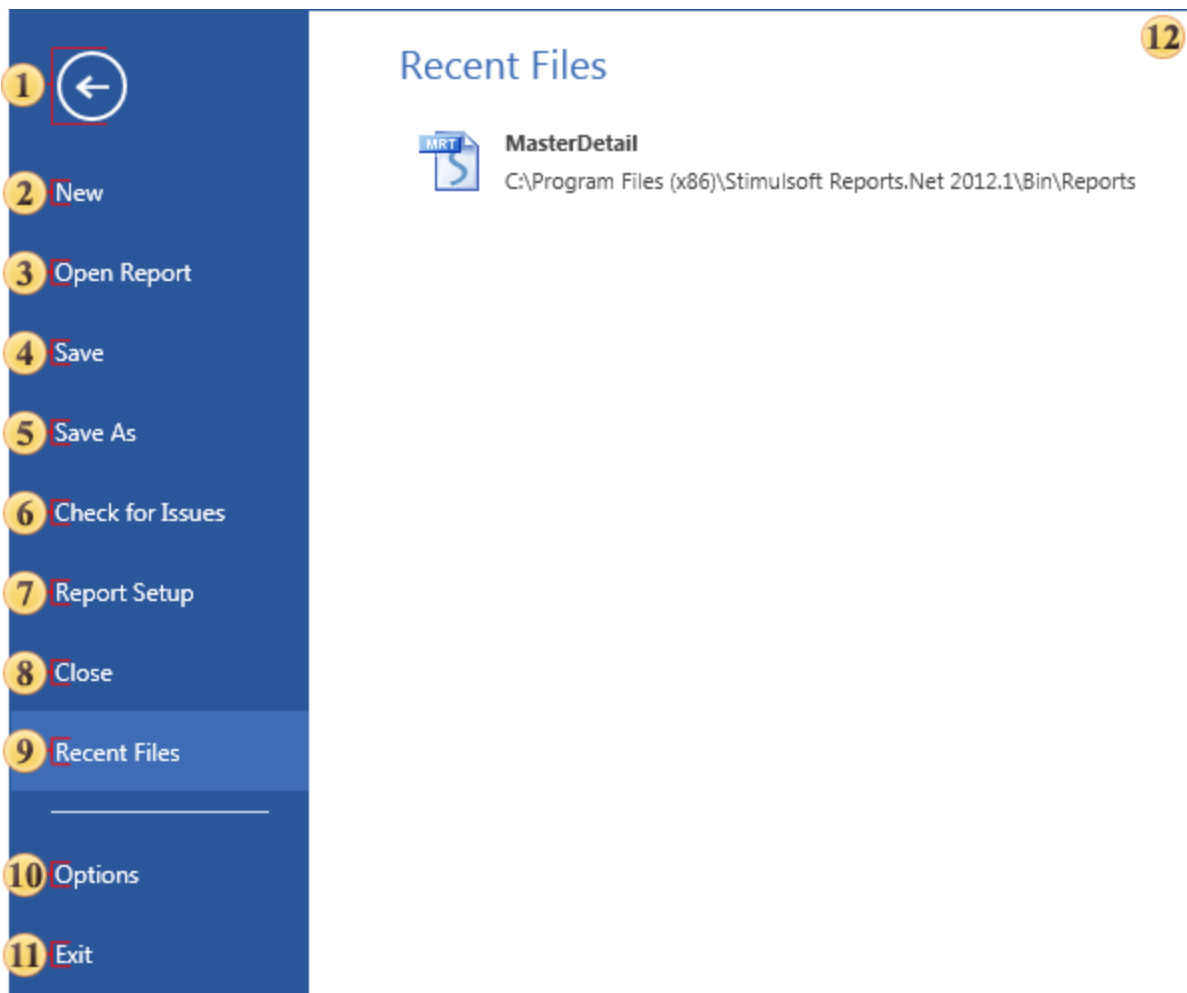
① Exit button from the main menu

② The New menu item contains submenu where a list of new report components is available for creation is shown.

③ The Open Report menu item. When calling this menu item, a dialog for opening a report will appear.

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 Call the Report Checker tool.

⑦ Call the Report Setup window of report options.

⑧ Close a report that is opened in the report designer.

⑨ The Recent Documents menu item contains submenu where recently opened reports are shown.

⑩ The Options menu item calls a window for designer parameters settings.

11 The Exit button closes a report designer.

12 The panel shows a submenu of selected menu item or selected group.

6.1.1.1   Import

YouTube
Watch our video tutorials on <u>using Importing Tool</u>. Subscribe to the <u>Stimulsoft channel</u> and be the first who watches new video tutorials. Leave your questions and suggestions in the comments to the video.

Stimulsoft Reports provides the ability to import reports from other reporting tools. Import of the report can be done in the report designer or with the help of the import utility. To import reports in the report designer, you should go to the File menu -> Open -> Import and specify the reporting tool you to import reports from:
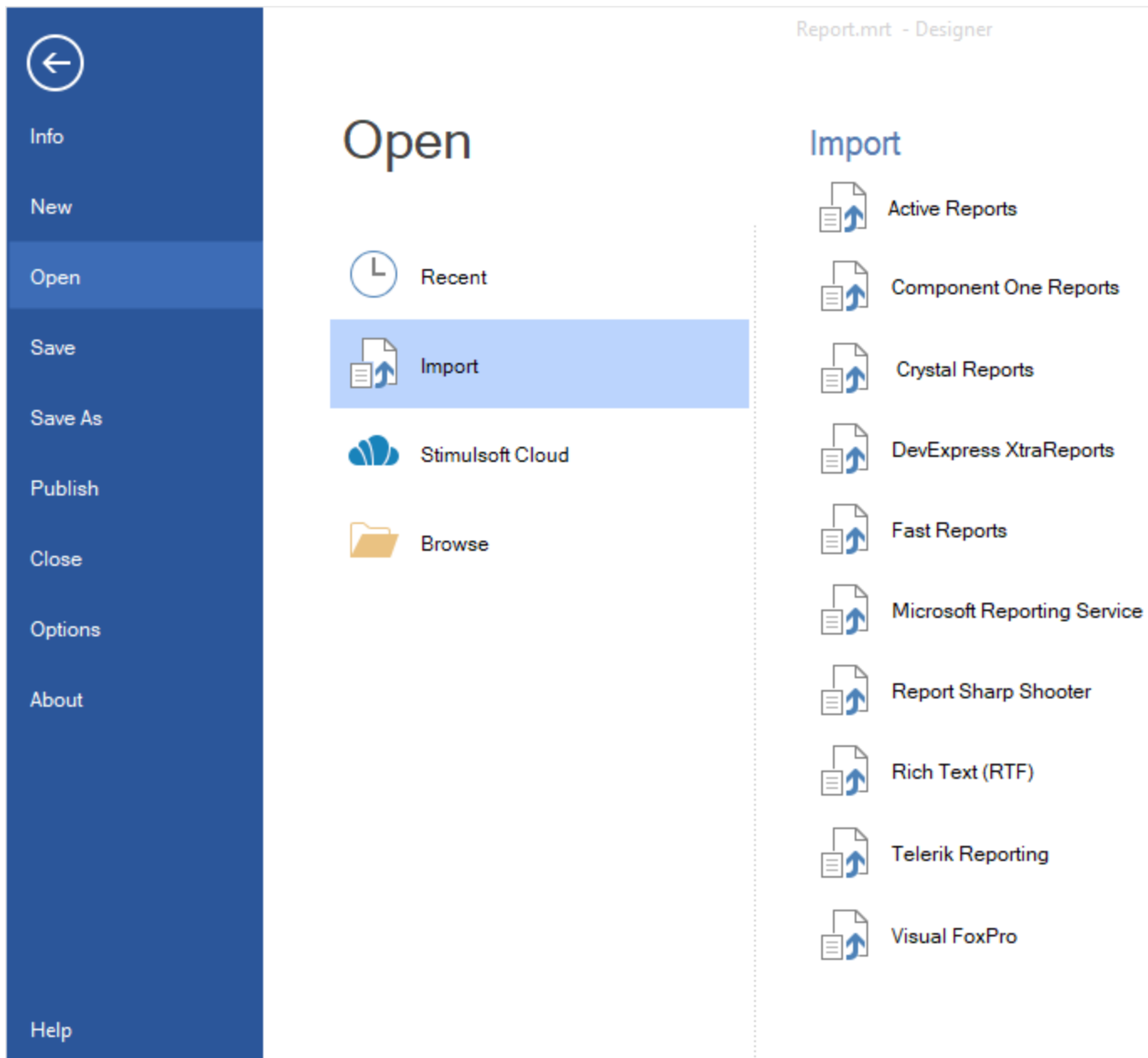
Report imports, with the exception of the import from Crystal Reports and XtraReports (DevExpress), are built-in. In this case, the report can be opened directly from the designer. Reports from Crystal Reports and XtraReports (DevExpress) should be converted using the import utility, and then opened in the Stimulsoft Reports designer.

Information

In some report designers, you may not find the Import option by clicking the **Open** button. In this case, you can import reports using the import utility, and then use these reports in Stimulsoft Reports.

�֍ Run the Import Utility

With the help of import utilities, you can convert reports from other reporting tools. Every file type has its own import utility. To run the import utility, follow the steps below:

Step 1: Open the web browser and go to https://github.com/stimulsoft/Importing.Tools

Step 2: Download the archive with the import projects and unpack it.

Step 3: Use the development environment, such as Visual Studio, to compile the project.

Step 4: Select the reports that you want to convert to Stimulsoft reports and specify the location where the converted report should be saved.

Step 5: Click the Convert button. The result of the conversion will be displayed to the user.
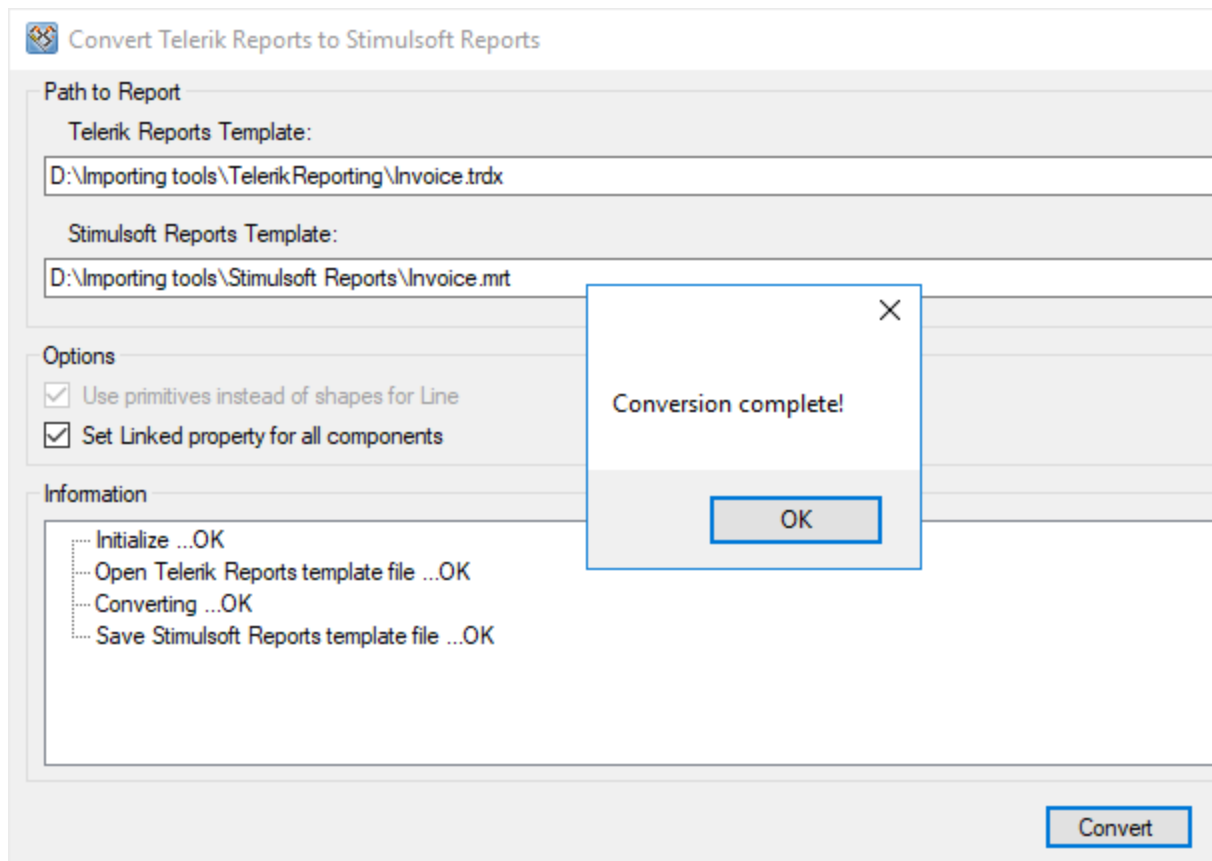
Use converted reports in Stimulsoft products.

Import Report from Crystal Reports

The utility converts the Crystal Reports templates (.rpt-files) to the Stimulsoft Reports report templates format (.mrt-files). The tool is supplied as the C# source code only and requires referencing of some Crystal Reports runtime libraries in order to be built successfully in Visual Studio 2010, .NET Framework 4.0 or higher. Please download the archive from the link below, unzip it and open in the Visual Studio. The project will be built successfully, once all the required dll libraries are referenced and found in Visual Studio.

The project was created in a way that all the required assemblies would be automatically taken from the GAC (Global Assembly Cache). If * .dll libraries of Stimulsoft Reports are not in the GAC, they will be added to the project from NuGet automatically. If you do not have an Internet connection, you should manually add **Stimulsoft.Base.dll** and **Stimulsoft.Report.dll** to the project.

The Crystal Reports report templates' file format is a proprietary format. Therefore, the tool requires some Crystal Reports special managed assemblies. The tool interacts with these assemblies via some special Crystal Reports interfaces for the special Visual Studio managed dlls.

These assemblies are not always installed in the system together with Crystal Reports, usually the additional and an official installation of these assemblies is required in order for them to work correctly with the import tool.

For example, for Crystal Reports 2013 the Support Pack (developer version for VS: Updates & Runtime) is required and needs to be installed first, and only after that the import tool will be built successfully.

The current Crystal Reports version requires the additional installation of the 'SAP Crystal Reports runtime engine' (32 bit or 64 bit). The automatic installer will copy the required assemblies to the GAC. But this installer must be downloaded separately, it is not a part of the standard Crystal Reports installation package.

The project uses the following Crystal Reports assemblies:
- CrystalDecisions.CrystalReports.Engine
- CrystalDecisions.ReportAppServer.DataDefModel
- CrystalDecisions.ReportAppServer.ReportDefModel
- CrystalDecisions.Shared
- CrystalDecisions.Web
- CrystalDecisions.Windows.Forms

These assemblies are not included with the tool. The packages will not work if they are just referenced and copied to the project without the proper installation by the Crystal Reports' official installer first.
Please find the explanation of the required installations:

Operational system	Platform Target, CPU	Installation package

		requirements
Windows x32	Any CPU	'SAP Crystal Reports runtime engine 32 bit'.
Windows x64	Any CPU	'SAP Crystal Reports runtime engine 64 bit'.
Windows x64 + runtime engine x32bit	X86	not required
Windows x64 + runtime engine x32bit	Any CPU	'SAP Crystal Reports runtime engine 64 bit'.

The above mentioned installers can be downloaded using the following links:
http://www.crystalreports.com/crvs/confirm/
http://downloads.businessobjects.com/akdlm/cr4vs2010/
CRforVS_redist_install_32bit_13_0_20.zip
http://downloads.businessobjects.com/akdlm/cr4vs2010/
CRforVS_redist_install_64bit_13_0_20.zip

Please read more about the requirements of those additional installations in the official reply from the Crystal Reports:
https://archive.sap.com/discussions/thread/3675145

Run Crystal Reports on client machine without install runtime package?
No, the only way to make your app work is to run one of the redist packages on the user's PC. We don't support nor do we have a way to manually deploy the runtime. Too many Registry entries and registering of the dll's in order to do this manually.

Parameters of Import Utility

❯ Use primitives instead of shapes for the Line and the Box
If the flag is not enabled then the Line and the Box components will be converted to ordinary primitives (shapes, VerticalLine/HorizontalLine, and Rectangle/RoundedRectangle). If the flag is enabled then the Line and the Box components will be converted to special primitives (VerticalLinePrimitive/HorizontalLinePrimitive and RectanglePrimitive/RoundedRectanglePrimitive). When viewing/printing reports, there are no big differences between graphic and special primitives. Graphic primitives are exported as images when exporting. So it is easier to work with special primitives. But, due to Crystal Reports peculiarity, special primitives cannot work correctly on complex

reports. This is why there is the ability to select the option.

❯ Use functions for Formula Fields

In each Formula Field either expression or a data string can be placed. Each Formula Field is converted into the variable in the data dictionary. If the Use functions for Formula Fields flag is enabled, then the Function flag is set in the variable. In other words, when report rendering, Stimulsoft Reports will use the value of a variable as an expression and will try to calculate the value of this expression. If the Use functions for Formula Fields flag is not enabled, then the value of a variable will be used as the data string.

Problems with conversion

One of the main problems in conversion is that not all object properties are available when working with managed dll. The second problem is the different reporting tools structures, such as data structures, work with bands etc. Therefore, it is not always possible to convert a report automatically, and it is required to correct a report manually.

❯ DataBase:

Crystal Reports often uses their internal libraries when working with data bases. It is possible to get only some properties from .NET and it is impossible to get ConnectionString. So, not all data bases can be identified. By default, for not identified data bases, the StiOleDbDatabase type and ConnectionString template without specifying the provider is used.

❯ Data Bases:

In CrystalReports, each report/sub-report has its own data dictionary, and the data base will be described differently in subreports. In Stimulsoft Reports, the common data dictionary is used. So, all dictionaries are united after conversion. If the data base is repeated then it is not included into the common dictionary.

❯ Image:

Sizes and locations can be indicated for images but, if it is saved in the report template, then it is impossible to get the content of an image.

❯ FormulaField:

Expressions and formulas can be placed in these fields. On the current moment, parsing and syntax of these expressions are written "as is". So in many cases further manual correction is required.

{Crystal Reports allows using expressions and formulas in FormulaFields. On the current moment parsing and syntax conversion cannot be done, expressions are written 'as is'. Therefore, in many cases, it is required further manual correction of expressions.} The section describes the tools for converting third party formats to internal Stimulsoft Reports formats. You may download the tools described below from the Tools section at http://www.stimulsoft.com/en/downloads

Information

Report SharpShooter v2.0 + applies different internal file format of the report template than the older versions. The importing utility is made for the new format. Older report templates are converted partially. It is better to re-save old reports in the new format and then import them.

6.1.1.2 Menu Item New

The command **New** contains a submenu, which displays the methods and options for creating reports. The picture below shows the submenu of the command **New**:

Create a New Report

Blank Report   Standard Report   Master-Detail Report

Label Report   Chart   Cross-Tab

① The **Blank Report**. A blank report is opened in the report designer.

② The **Standard Report** wizard. Used to create reports as a list.

③ The **Master-Detail Report** wizard. Used to create a Master-Detail reports.

④ The **Label Report** wizard. Used to create Label reports.

⑤ The **Chart** wizard. Used create reports with charts.

⑥ The **Cross-Tab** wizard. Used to create Cross-Tab reports.

6.1.1.3   Dialog Report Setup

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** dialog:
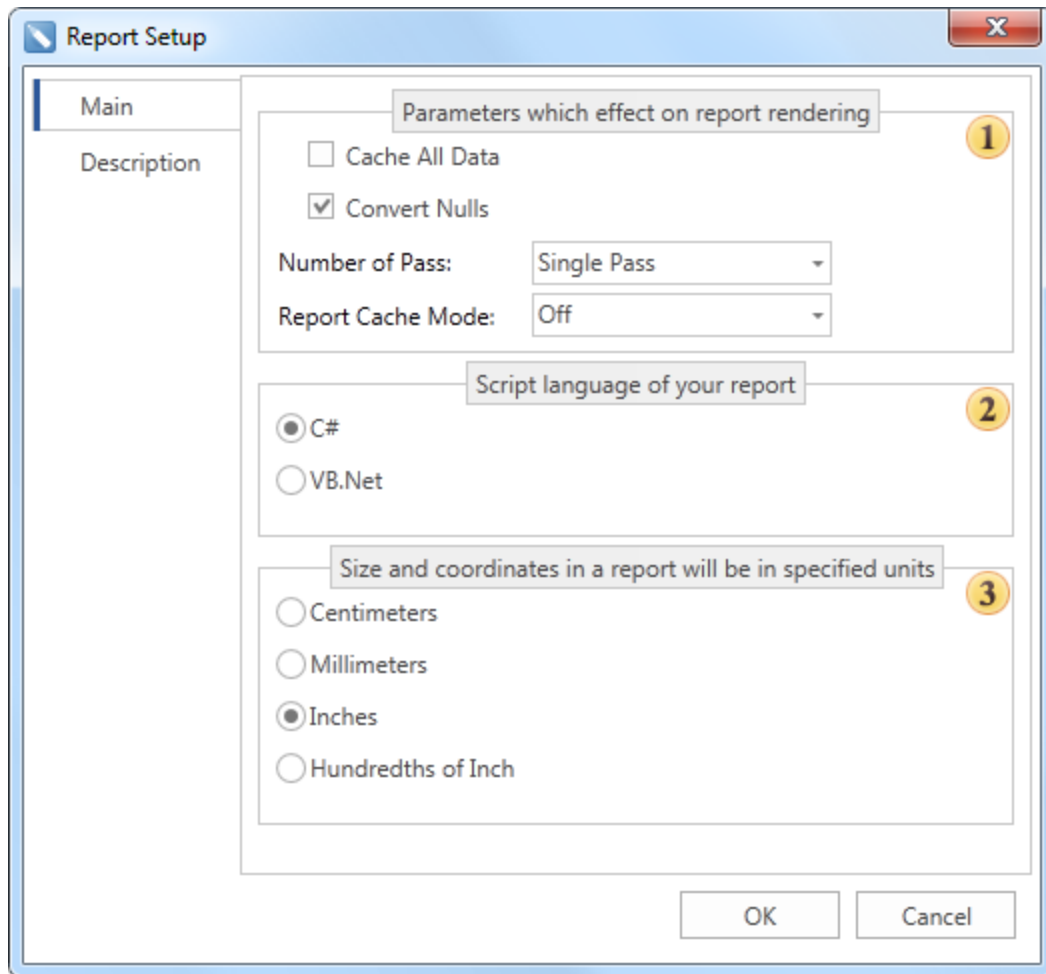
As can be seen from the picture above, the editor of the report parameters contains two tabs: **Main** and **Description**. The **Main** tab is represented by three groups, which define the most important parameters of the report:

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**.

6.1.1.4 Dialog 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:
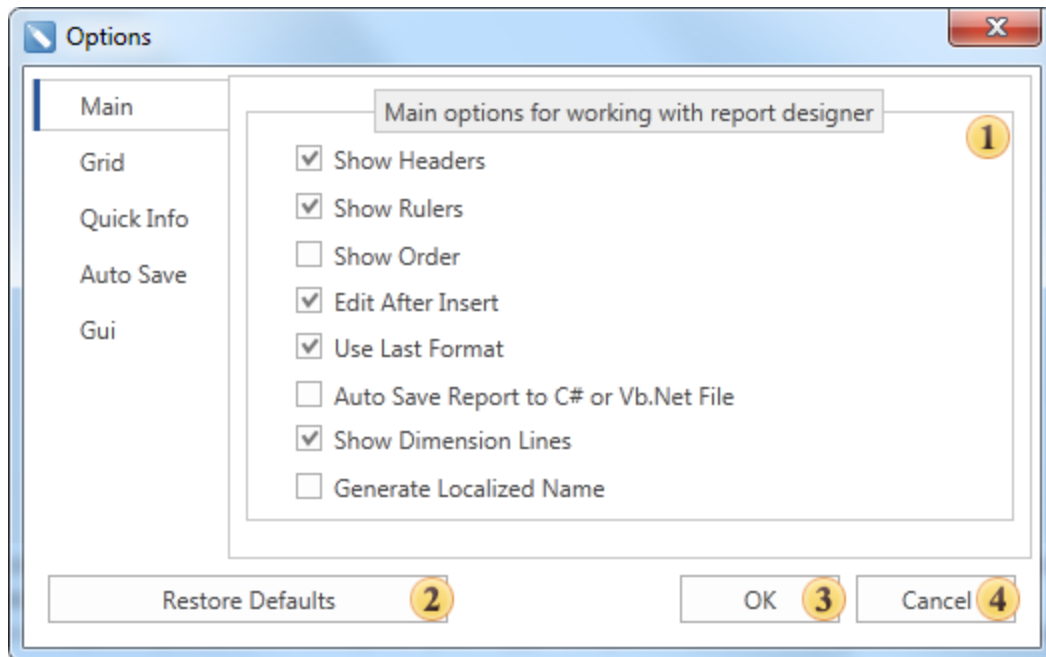
As can be seen from the picture above, the **Options** dialog box contains five tabs: **Main**, **Grid**, **Quick Info**, **Auto Save**, **Gui**. The **Main** tab is represented by one ❶, which has the basic options of a designer such as:

❯ The Show Headers option enables/disables displaying headers of components of the report.

❯ The **Show Rulers** option enables/disables displaying rulers.

❯ The **Show Order** option enables/disables displaying the order number of the report component.

❯ The **Edit After Insert** option enables/disables invoking the editor after creating a component in the report.

❯ The **Use Last Format** option enables/disables using the latest format of a component.

❯ 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.

❯ The **Show Dimension Lines** option enables/disables the dimension lines.

❯ 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.

Grid

Grid options

☑ Align to Grid
☐ Show Grid

①

Grid drawing options

◉ Lines
◯ Dots

②

Grid size

③

Inches:	0.1	in
Hundredths of Inch:	10	hi
Centimeters:	0.2	cm
Millimeters:	2	mm
Pixels:	8	px

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.

Quick Info

Options of Quick Info

◉ None
◯ Show Components Names
◯ Show Content
◯ Show Fields
◯ Show Fields Only
◯ Show Events

①

☑ Display Over Components

As can be seen from the picture above, this tab contains a single group ①, which defines the following parameters:

❯ The Options of Quick Info option specifies what information you want to display.

❯ 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.
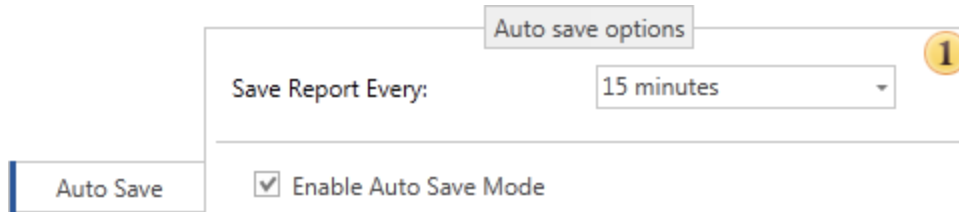
This tab contains a single group ①, which contains the following options:

❯ The **Save Report Every** option determines the time interval after which an auto-save event occurs.

❯ 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.
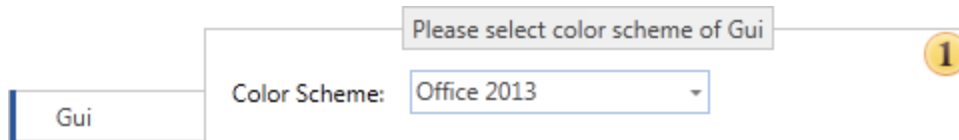
This tab contains a single group ① and one **Color Scheme** parameter, which is required to change the type of interface and/or color theme.

6.1.1.5 Recent Documents

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.
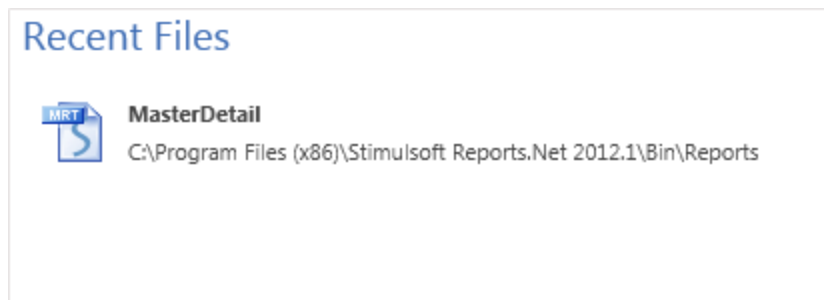
6.1.1.6 **Share**

When designing reports, it often becomes necessary to provide access to it to other users. This can be done in various ways:

i Save a report template or a rendered report, by forwarding these files to other users. However, in this case, Stimulsoft Reports installed will be required to view the reports. In addition, each time you change the report or retrieve new data, you will have to forward these files.

i Export the rendered report to a format like PDF, Excel, HTML, etc. In this case, Stimulsoft Reports installed will not be required, but you will have to generate new documents every time you change the report and forward these files to other users.

i Set up remote access to the report and send the link to users. In this case, Stimulsoft Reports installed will not be required, and you will not have to generate documents every time the report is modified. It will be enough to refresh the page in the browser.

How it works
❯ The report from the report designer is saved in Stimulsoft Cloud;
❯ A link to the report is forwarded to the user. You can also embed the report access code in your HTML page or get a QR code with an access link.
❯ The user opens the link in the browser, browses the report or downloads it in a format like PDF, Excel, HTML, etc.

In order to use the remote access, it is necessary to:
❯ Use the report designer.
❯ Have access to the Internet, both from the report designer side (to make changes in the report), and from the user side (to download the report from the Cloud).
❯ Have a Stimulsoft account for the designer of reports. If you do not have an account, you may register it for free.

The remote access can be set up form the **Share** dialog, which can be called by selecting the **Share** command from the **File** menu.

Information

If the report template was opened not from Stimulsoft Cloud (when you select the **Share** command from the **File** menu), the report must be saved to the account of a user.

If you are not logged in, after selecting the **Share** command from the **File** menu, the login form will be displayed. If you do not have an account, click the account registration command.

Below is the menu for setting access to the report:

① **No Access** sets the ability to view only from the report designer or the cloud service, the report cannot be viewed by the link.
② **Public Access** allows you the remote viewing of the report the link for any user. Also this mode is used when embedding a report to the HTML page.
③ This parameter sets the time and date after which access will be denied. If this option

is disabled, then there is no validity period, access will always be enabled.

④ Access to the report can be provided in the following ways:

❯ Select **Link to Share** to get a link only to this item. Also, in this case, the field contains the **Copy** button (when you click it, the item will be copied to the clipboard), and the **Update** button (when you click it, a new link to the item will be generated).

❯ Select **Embed Code** to get the code for the HTML page with a link to this element. Also in this case, the field contains the **Copy** button (when you click it, the embed code is copied to the clipboard).

❯ Select **QR Code** to display the QR code for reading. When this code is read, the link to the item will be automatically received.

Step-by-step instructions to set up share to a report

Step 1: Run the report designer;

Step 2: Create or open a report to which you want to configure share.

Step 3: Select **Share** in the **File** menu.

Step 4: If the report was not opened from [Stimulsoft Cloud](#), specify the storage location in the workspace of your account and click the **Save** button in the **Save As...** dialog;

Step 5: Set the **Public Access** in the **Share** menu.

Step 6: Enable the **End date** parameter, and specify the time which is the date when the public access to the report is terminated;

Step 7: Select the type of file in which the report will be submitted. Please note that viewing a report without downloading is possible only if the **Document file** type is selected. If you select a different type, for example PDF, then when you click on the link, you will download the PDF file for local viewing.

Step 8: Copy the share link to the report;

Step 9: Click the **Ok** button in the **Share** menu;

Step 10: Forward the link to the user.

When clicking on the link, the web browser will run. The report will be displayed in the web browser or if a different result type is selected, the report will be downloaded using the browser.

Information

The report can be viewed by the link only if the Result Type is set as the Document file. In other cases, the report will be downloaded using the web browser.

However, when viewing the report, you can always download it in the required format without changing the access settings and without sending a new link. Add **/Result** Type to the link.

For example, if the link for viewing the report is https://stimulsoft.com/s/55af6 , and this report should be downloaded as a PDF document, then, in the address bar, add **/pdf**. So the link will look like https://stimulsoft.com/s/55af6/pdf.

6.1.2 Tabs

A tab is a part of the interface on the toolbar. The report designer has three tabs: **Home**, **Page**, **Layout**, **Insert**. Consider these tabs and the main instruments located on them in detail.

6.1.2.1 Tab Home

This is a basic tab of the report designer. Main commands for customizing report components are placed on this tab.

6.1.2.1.1 Group Clipboard

This group allows working with the Clipboard of the report designer.
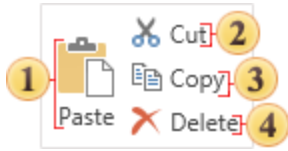
❶ Paste components from the Clipboard on the current page of a report.
❷ Cut the selected components from the current page to the Clipboard.
❸ Copy the selected components on the current page to the Clipboard.
❹ Delete selected components on the current page.

6.1.2.1.2 Group Font

This group is used to output text with the specified font type, color etc.
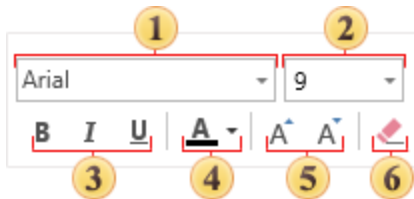
❶ Select the font type of the text components on the current page.
❷ Select font size of the text components on the current page.
❸ Sets the font style as Bold, Italic, Underlined.
❹ Set the font color of the text components on the current page.
❺ Changes (increases/decreases) the font size.
❻ Delete the content of all selected text components.

6.1.2.1.3 Group Borders

This group contains the commands to setup border components.

All commands can be applied to selected components on the current page.

1 Sets or removes borders from all sides of a component.
2 Sets or removes borders from each side of a component.
3 Sets a border color of a component.
4 Sets the shadow of a component.
5 Sets a background color of a component.
6 Sets a type of the border line.

6.1.2.1.4 Group Text Format

The group to control text formatting.
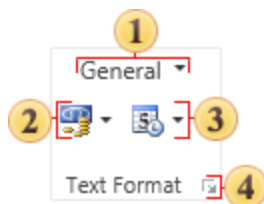
1 Select text format.
2 Select symbols of currency.
3 Set one of the predefined formats to output a text.
4 Call a form of formats editing.

6.1.2.1.5 Group Styles

This group is used to control styles and conditions which are used to automatically design components in a report.
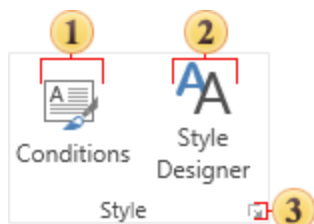
1 Opens a window of the Conditions Editor for selected components.
2 Opens a window of Styles Editor.
3 Calls a form of styles editing.

6.1.2.1.6 Group Alignment

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, center vertically and bottom the content of a component.
2 The angle of the text rotation. This command can be applied only to the text component.
3 Align left, center, right or justify the content of a component.
4 Used for the **WordWrap** property of the text component.

6.1.2.2   Tab Page

This tab is used to control page parameters.
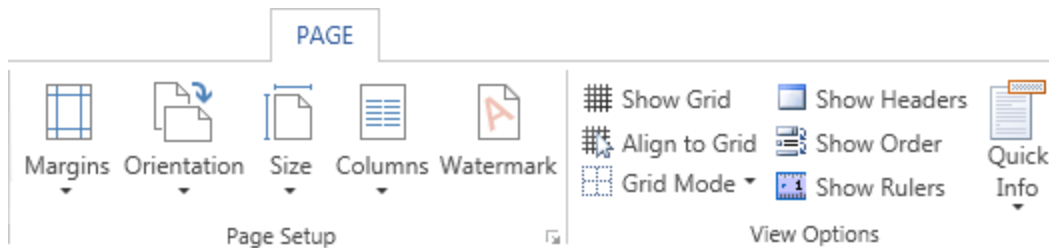
6.1.2.2.1 Group Page Setup

This group contains elements to control basic parameters of a page. These are page margins, orientation, page size, columns.
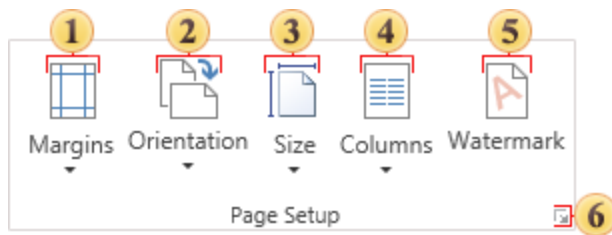
Changes can be applied to the currently selected page in the report designer.

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 Set a watermark on a page.
6 Invokes the Page Setup dialog window.

6.1.2.2.2 Group Watermark Text

This group is used to customize the watermark text.
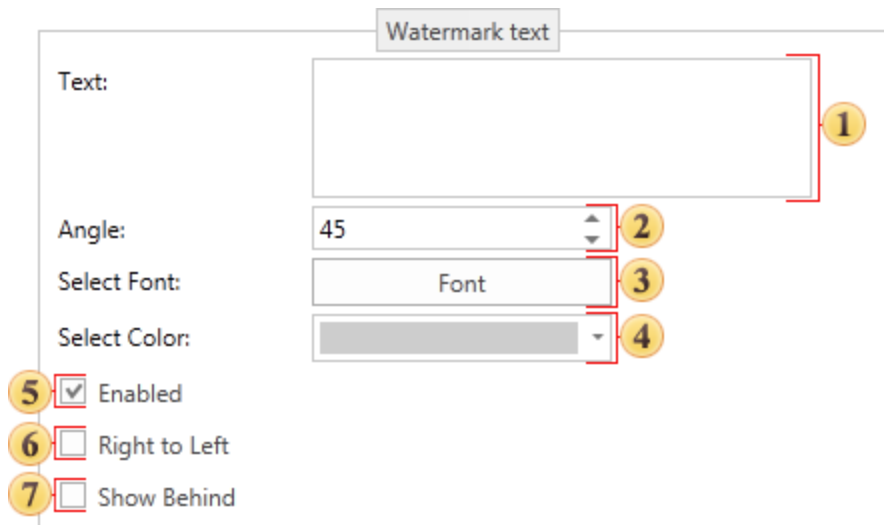
1 Watermark text
2 Set a rotation angle of the watermark text
3 Select a font type to output a text of the watermark
4 Watermark text color.
5 Enabling/disabling watermark.
6 Set the direction of the watermark output.
7 Show the watermark behind the text

6.1.2.2.3 Group Watermark Image

This group is used to customize the watermark image.
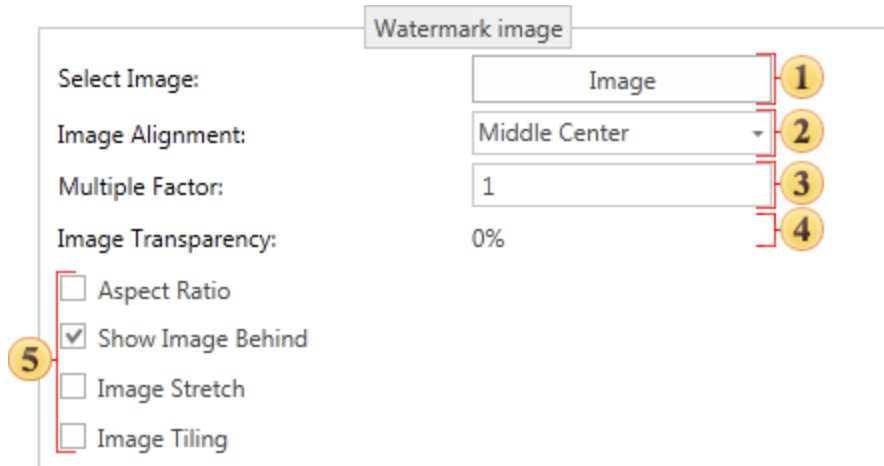
Load an image for the watermark
Align the watermark image
Sets the number of watermarks
Set the image transparency
Other watermark options

6.1.2.2.4 Group Viewing Options

This group contains settings for different parameters of showing a report and working with the report designer.
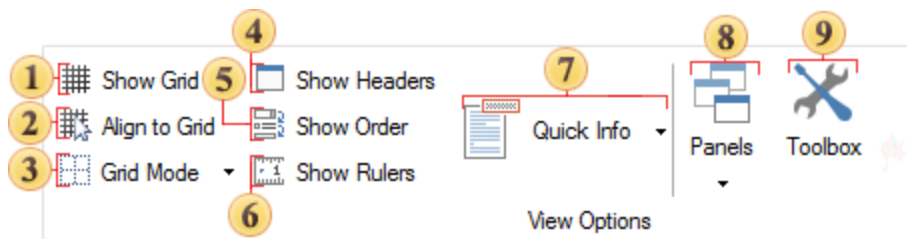
Control showing grid on a page.
Control automatic alignment of components by the grid.
Select grid type.
Show/hide headers of bands.
Show/hide the order of placing components on a page.

6 Show/hide rulers on a page.

7 Control Quick Info.

8 In this menu, you can enable/disable the following panels: **Properties**, **Data Dictionary**, and **Report Tree**.

9 This button enables/disables displaying the Toolbox.

! **Notice**. This group is not always present on the tab Page. In the WinRT report designer and Mobile, the group is missing.

6.1.2.3 Tab Layout

This tab is used to control placing different components on a page and also to specify styles of docking components.
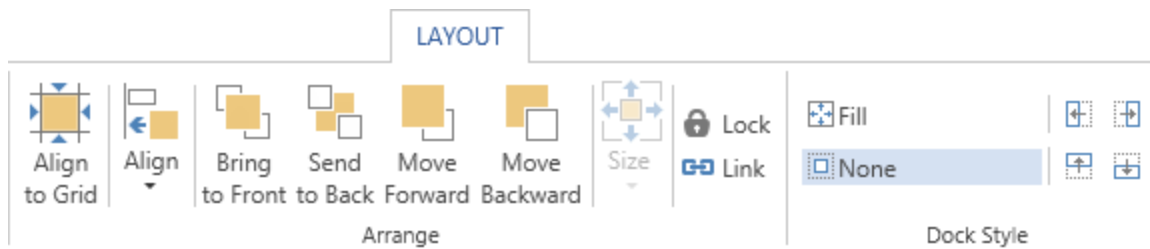
6.1.2.3.1 Group Arrange

The group contains a lot of commands to change position of components on a page. The picture below shows this group.
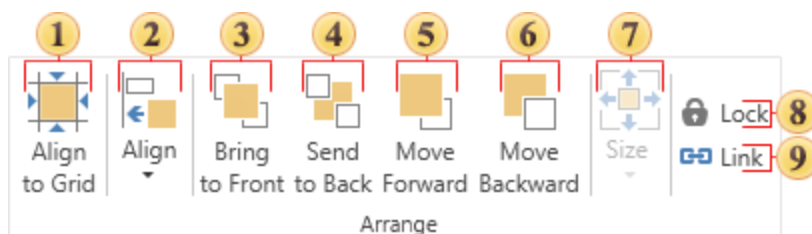
1 Align all selected components to the page grid.

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.
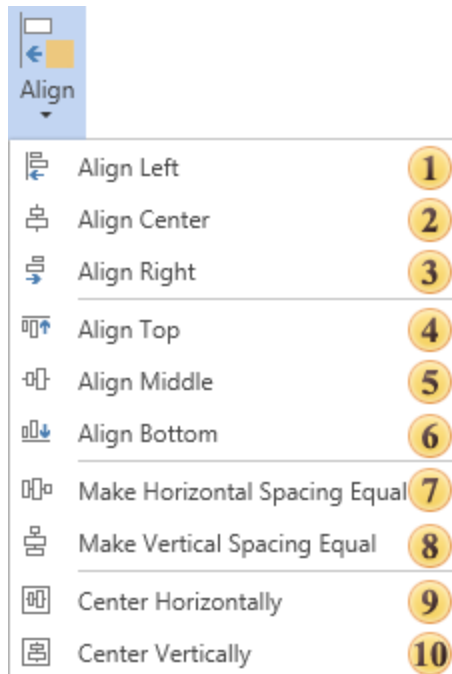
1 Align all selected components to their common left margin.

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.
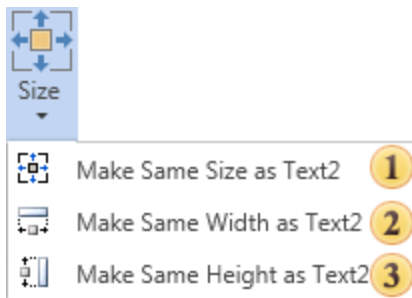
1 Make the same size of components as the size of the first selected component.
2 Make the same width of components as the size of the first selected component.
3 Make the same height of components as the size of the first selected component.

6.1.2.3.2 Group Dock Style

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.

6.1.2.4 Tab Insert

The tab **Insert** contains the main components for creating reports. This tab is present in the designer WinRT and Mobile, and is analogous to the **Toolbox** in other designers.
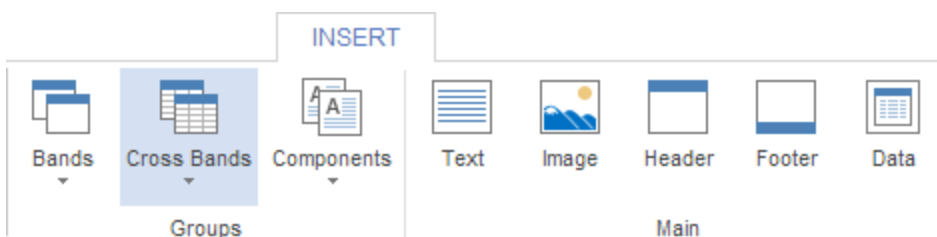
6.1.3 Status Bar

The status bar is placed under the designer window. The picture below shows a status bar of the **Standard** UI.

The bar contains 4 sections:

1 Units. This field shows current units in a report. It is possible to change them
2 Report Checker checks the report on issues
3 The field shows a name of the currently selected component
4 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
5 Page Zooming control.

6.2 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.

6.2.1 Overview

When creating a new report in the **New Report** dialog you should choose a way to create a report. The picture below shows the **Create a New Report** dialog:

Create a New Report

Blank Report   Standard Report   Master-Detail Report

Label Report   Chart   Cross-Tab

As can be seen from the picture above, there are several ways of creating a report: select a **Blank Report**, and manually create a report template, or create a report using the report wizards.

❯ The **Blank Report** icon can be used to create a blank report and the user should put components manually.

❯ The **Standard Report** wizard is used to create reports as a list.

❯ The **Master-Detail Report** wizard is used to create a **Master-Detail** reports.

❯ The **Label Report** wizard is used to create Label reports.

❯ The **Chart** wizard is used to create reports with charts.

❯ 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:
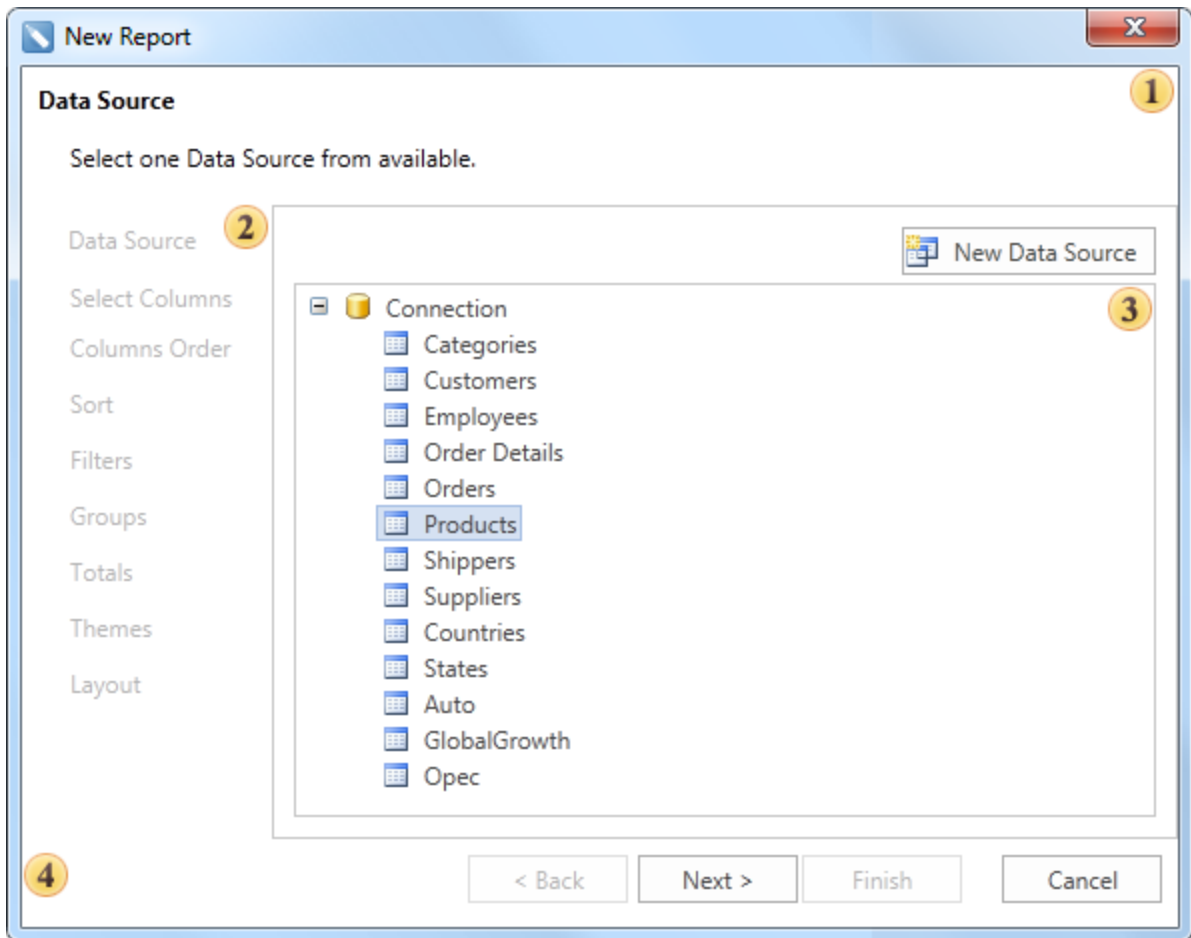
1. The **Description Panel**. This panel shows description of each steps to be done.
2. The **Steps Panel**. Shows steps of creating reports using a report wizard.
3. The **Selection Parameters Panel**. This panel shows report parameters. On each step of report creation its own options are available.
4. The **Control Panel**. Contains buttons to control the **Report Wizard**.

6.2.1.1   Wizard Standard Report

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:
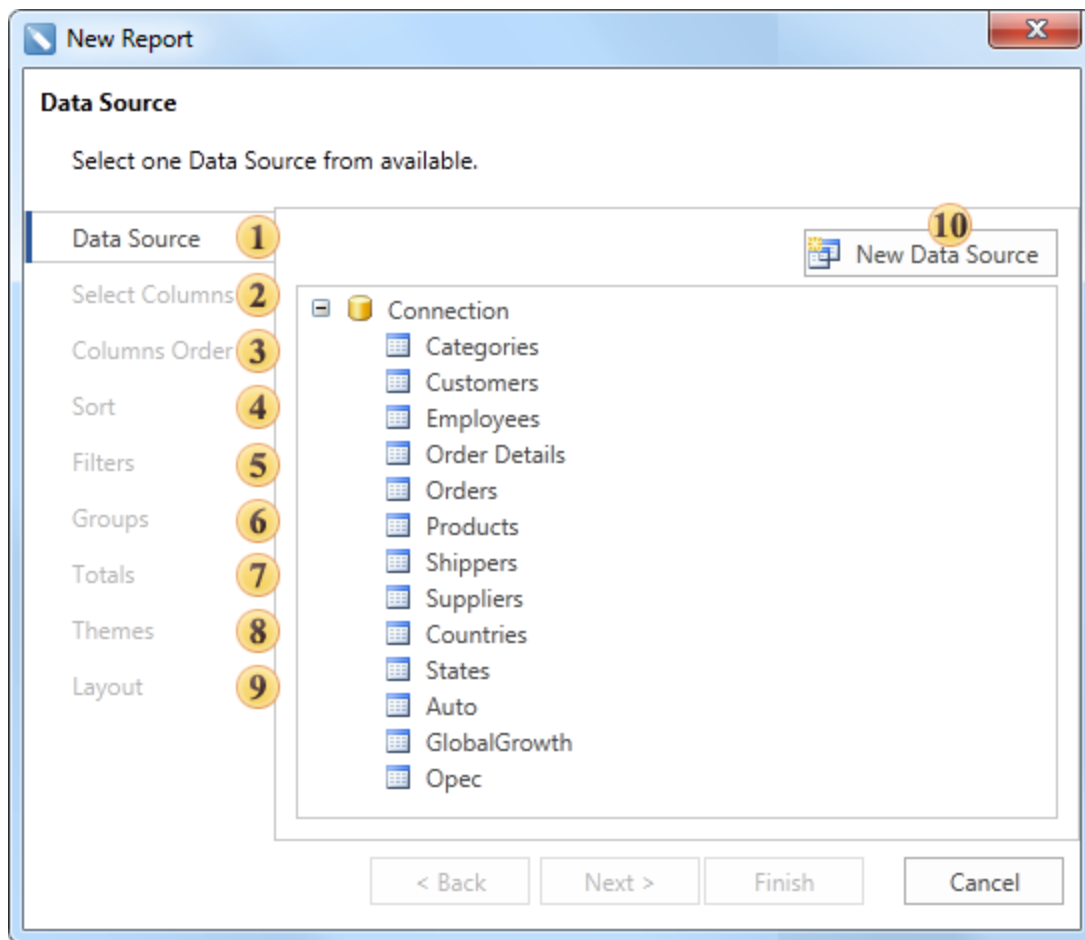
① **Data Source**. On this step the data source is defined. This step is obligatory.

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**:
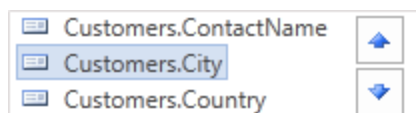
④ **Sort**. On this step, it is possible to specify elements and sorting direction. The picture below shows a sample of the **Selection Parameters Panel** of sorting:

5 **Filters**. On this step, it is possible to set the conditions of filtering. The picture below shows a sample of selection filtering 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 **Themes**. This step defines the report style.

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 **New Data Source** button is used to create a new data source.

6.2.1.2   Wizard Master-Detail Report

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:
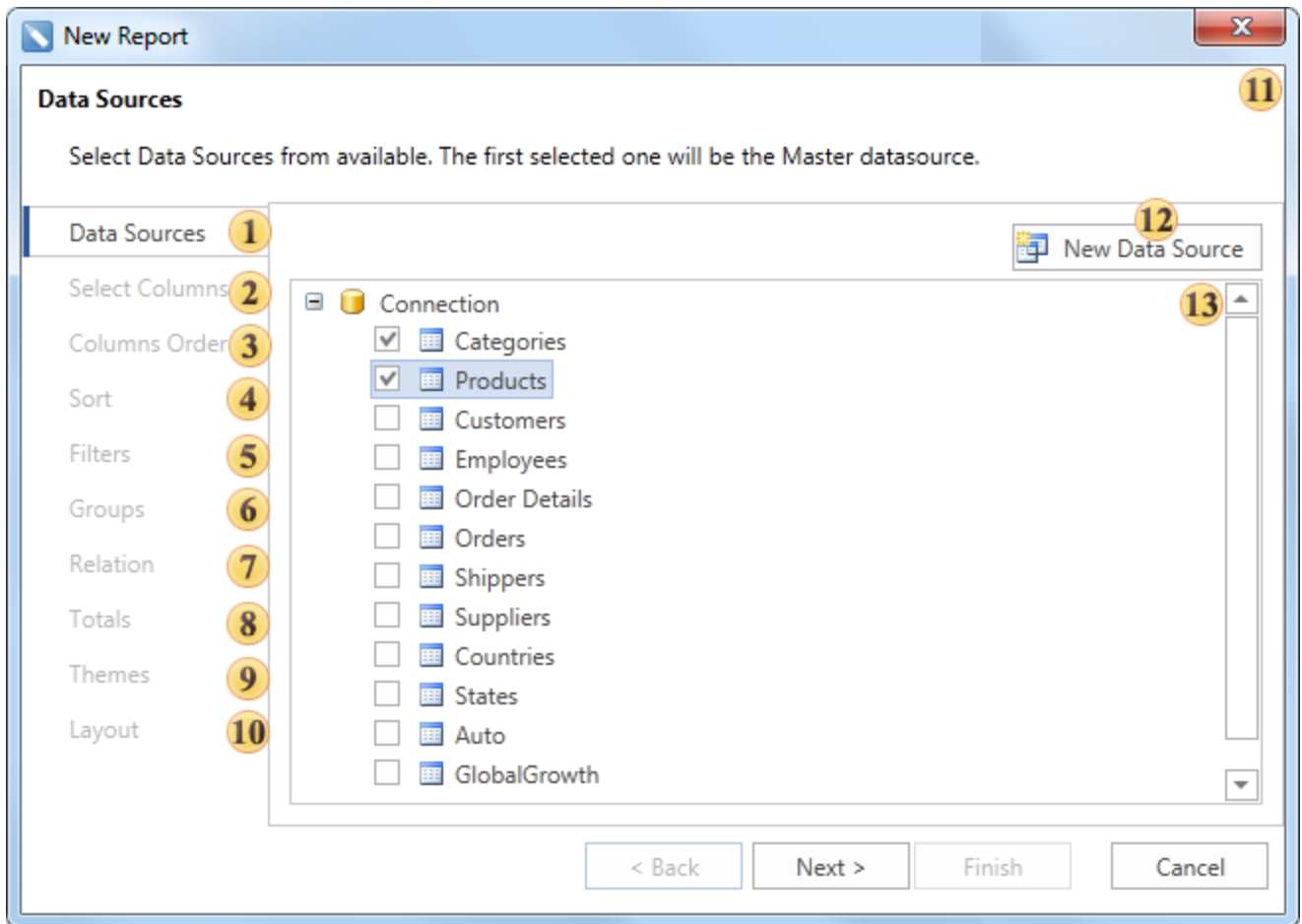
① **Data Source**. On this step the data source is defined. This step is obligatory. For creating the **Master-Detail Report**, the report template should have no less than one **Master** band and one **Detail** band.

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**:

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 **Themes**. This step defines the report style.

¹⁰ **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**.

¹¹ The **Description Panel**. Shows description for the current step.

¹² The **New Data Source** button is used to create a new data source.

¹³ The **Selection Parameters Panel** shows options, actions, settings available on this step.

6.2.1.3   Wizard Label Report

The **Label Report** wizard is used to create reports which have labels. The picture below shows a window of the **Label Report** wizard:
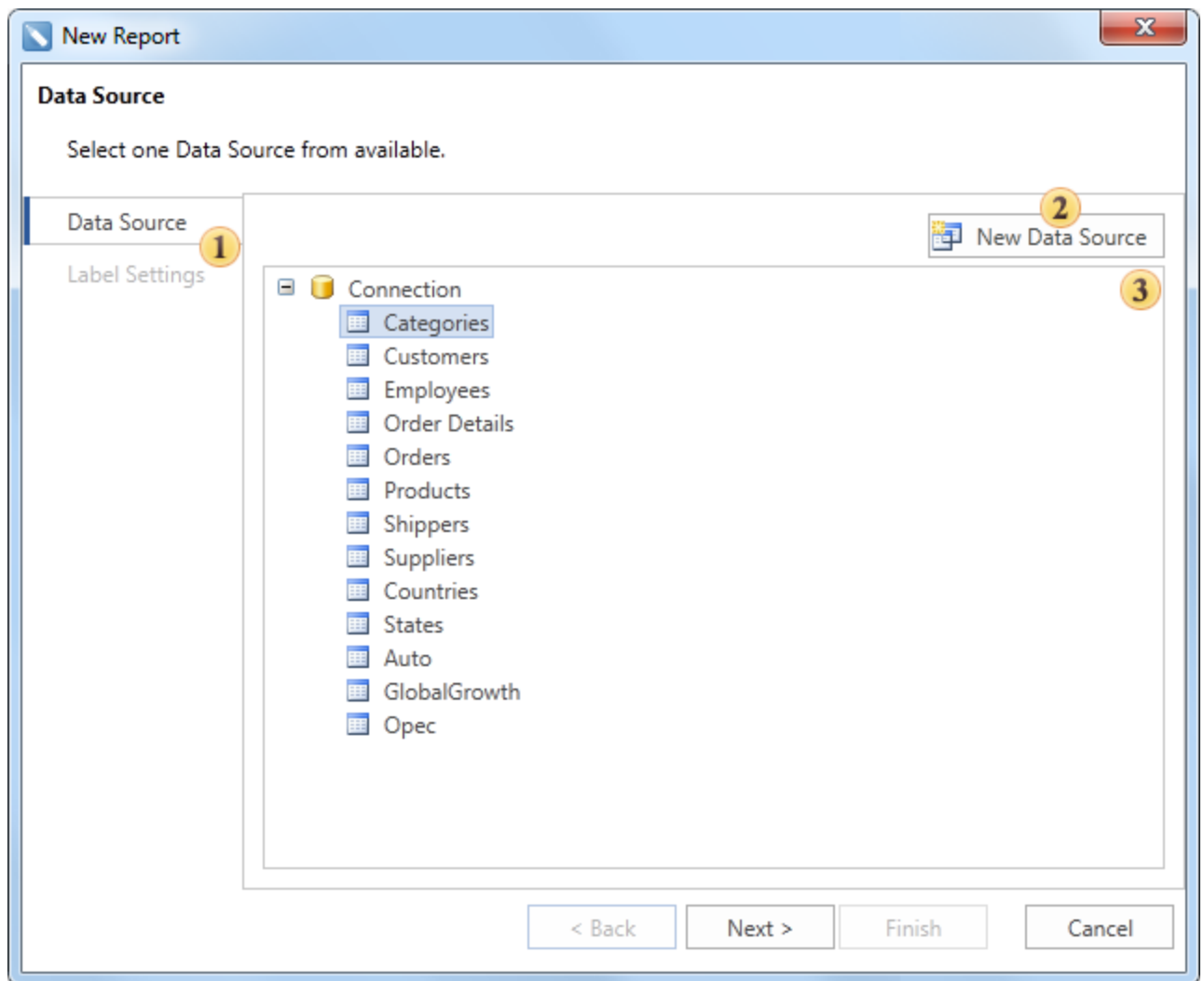
❶ The **Description Panel**. Shows description for the current step.

❷ The **Steps Panel** shows step of report creation.

❸ The **Selection Parameters Panel** shows options, actions, settings available on this step.

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.

6.2.1.4   Wizard Chart

The **Chart** wizard is used to create reports with charts. The picture below shows a window of the **Chart** wizard.
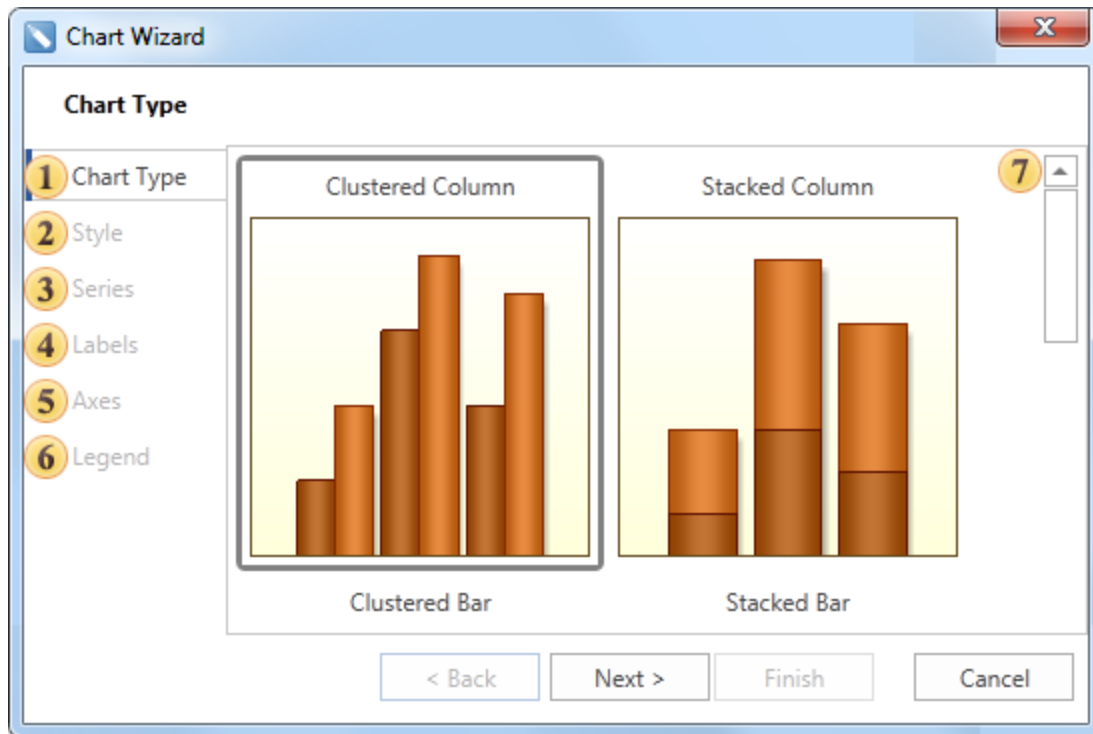
1 **Chart Type**. Select the chart type.

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 a rotation **Angle**.

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.

6.2.1.5 Wizard Cross-Tab

The **Cross-Tab** wizard is used to create reports with cross-tab. The picture below shows the window of the **Cross-Tab** wizard.
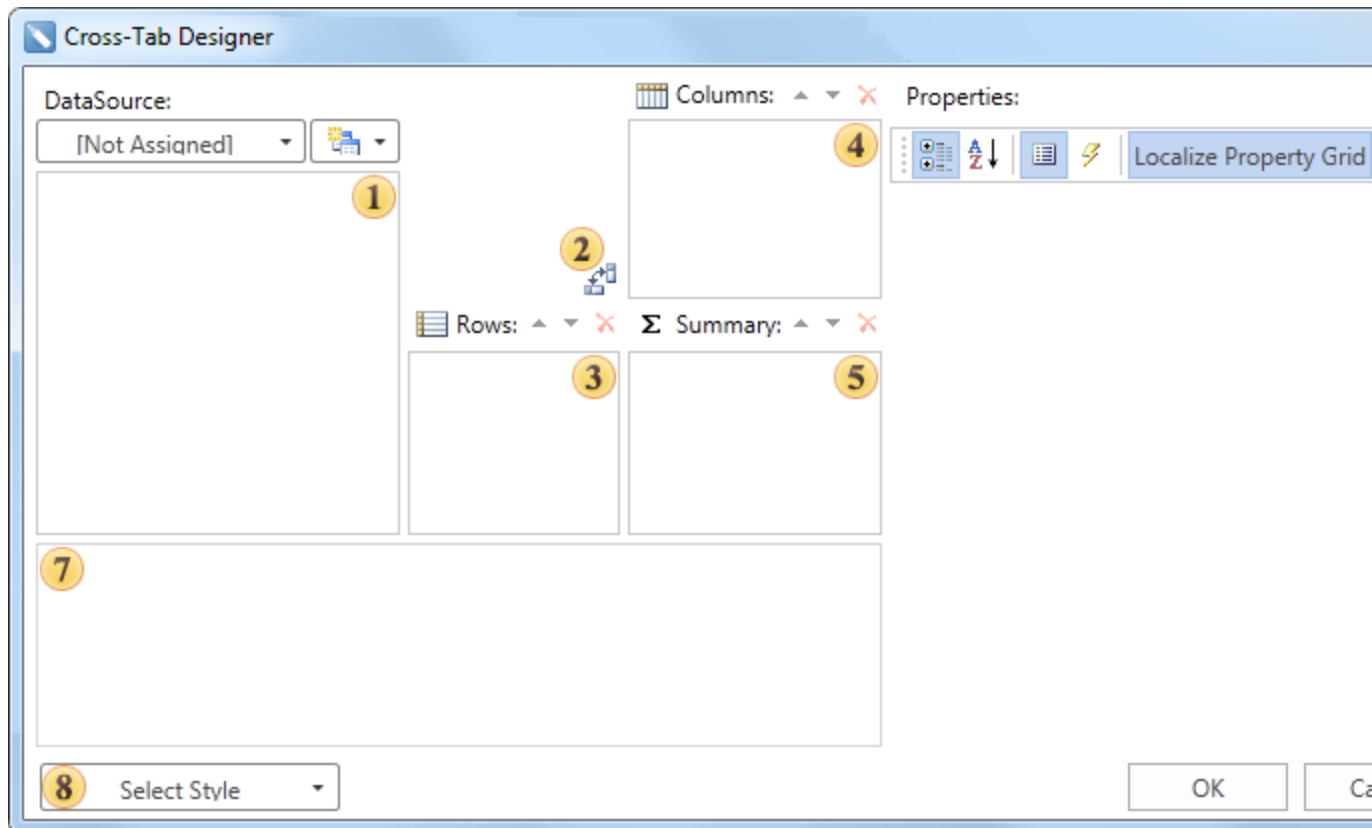
1 **Data Source Panel**. In the **Data Source** field it is necessary to select the data source. Then data source columns will be shown on the panel of the data source.

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 **Select Style** button is used to select style of the cross table appearance.

6.3   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**.

6.3.1 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:
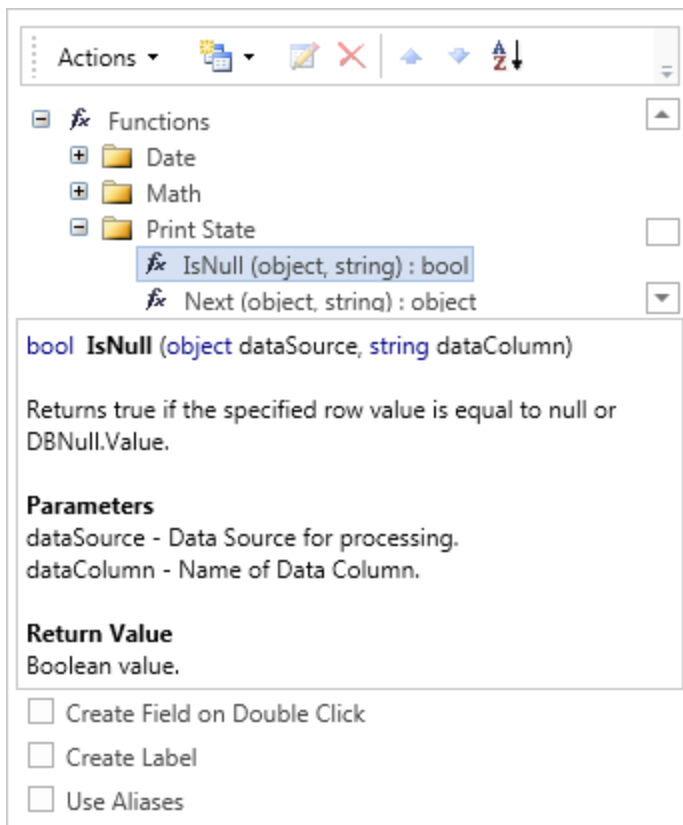
The Dictionary panel includes: **Dictionary ToolBar**, **Data Tree**, **Description Panel**, **Dictionary Setting Panel**.

❯ The **Dictionary ToolBar** is a set of tools and commands to work with the Dictionary. The picture below shows the Dictionary ToolBar:
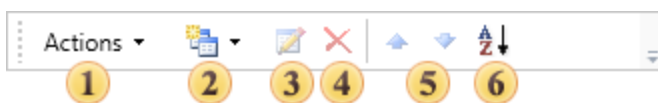
❶ The **Actions** menu is a set of commands to work with vocabulary such as: creating,

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.

❯ 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:
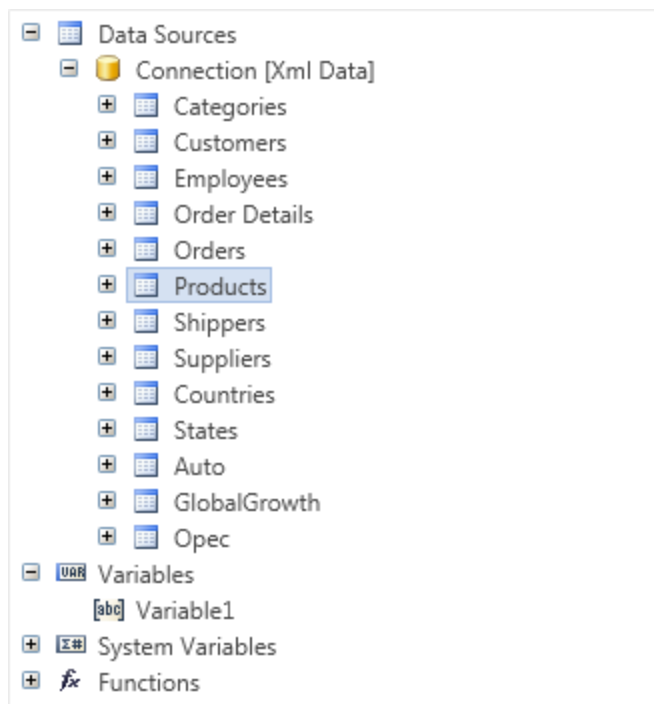
❯ The **Description Panel** displays a short description of the selected system variable or function. The picture below shows an example of the Description Panel with the description of the Line system variable:

Line

Returns the current line number (starts from 1).

❯ The **Dictionary Setting Panel** is a panel with three options to optimize the work with the dictionary and its contents.

(1) ☐ Create Field on Double Click
(2) ☐ Create Label
(3) ☐ Use Aliases

❶ 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.
❷ The **Create Label** option attaches the column data header when it is dragged on the report template.
❸ The **Use Aliases** option provides an opportunity to show Aliases of components instead of the Name in the report template.

6.3.2   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:

☐ 🔲 Report
  ☐ 📄 Page1
      ☐ Panel1
      📊 SubReport1
    ☐ 🔲 DataProducts
        🔤 DataProducts_ProductName
        🔤 DataProducts_UnitPrice
        🔤 DataProducts_UnitsInStock
  ☐ 📄 subReport_1
    ☐ 🔲 DataBand1
        🔤 Text1
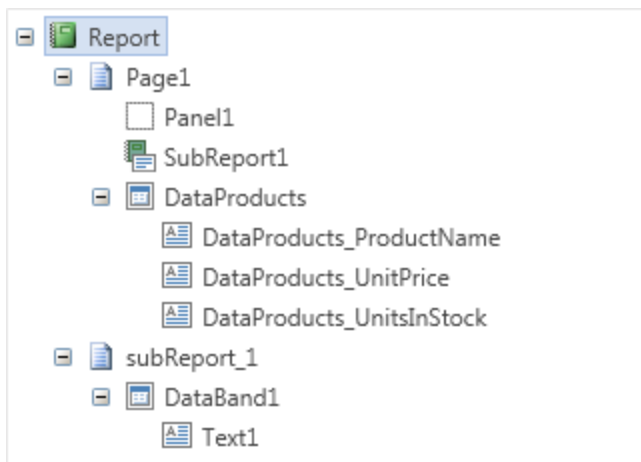
As can be seen on the picture above, hierarchy is represented on the principle of

"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".

6.3.3 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):
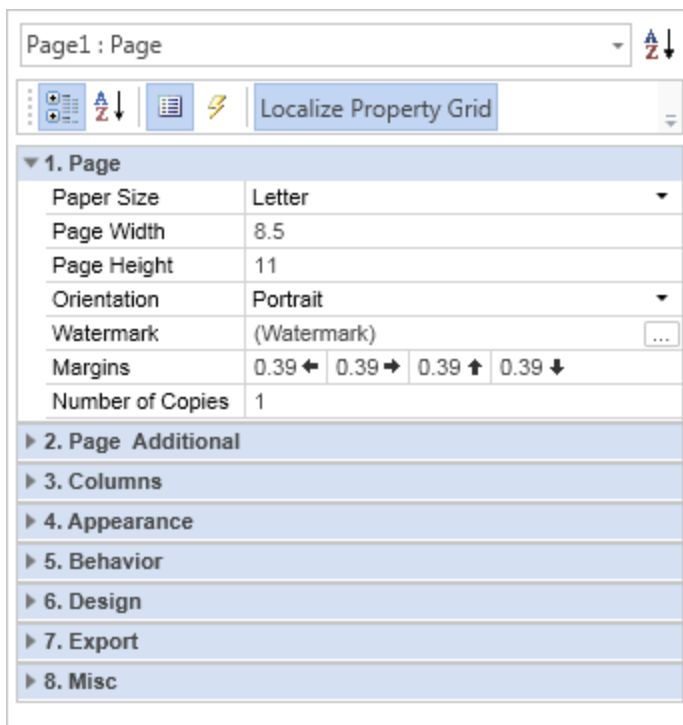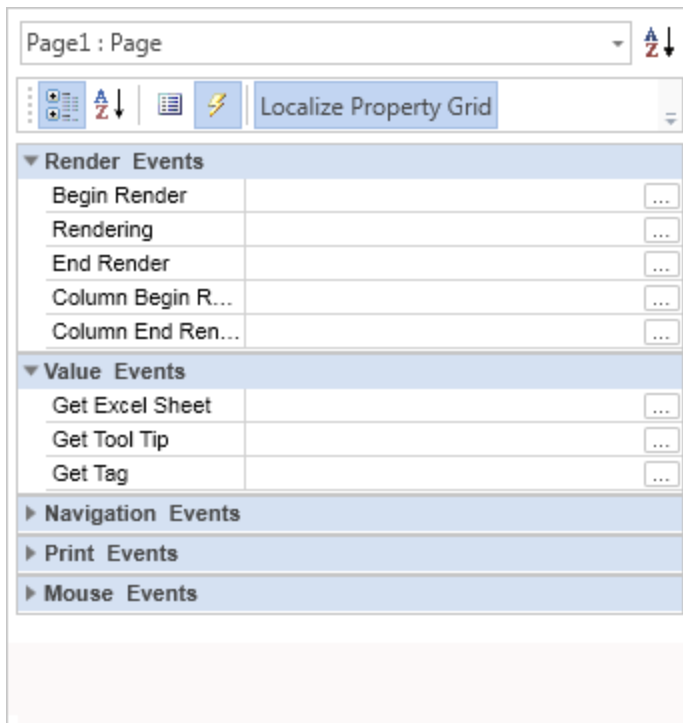
The **Properties** panel includes: **Drop-Down List of Components**, **Properties ToolBar**, **Properties** or **Events Table**, **Description Panel**.

> 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**:
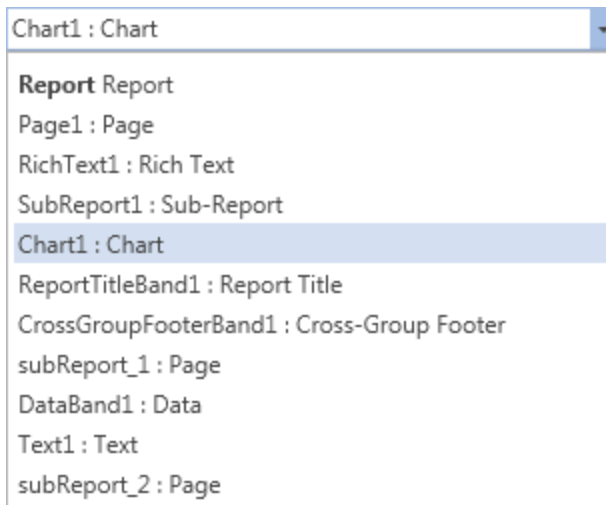
As can be seen on the picture above, the list is presented in two columns. The first column displays the **Name** of a component and the second one shows its type. For example, the string "**Text7: Text**", "**Text7**" is a **Name**, "**Text**" this is a type. If to select a

component in this list then, on the Properties panel, properties and events of the selected component will be shown.

❯ The **Properties ToolBar** is designed to control the **Properties** panel. The picture below shows the **Properties ToolBar**:
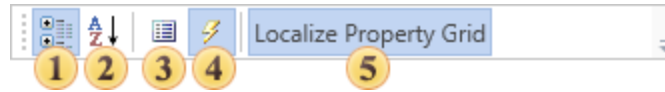
1 The **Categorized** sorting button is used to show a list of properties or events sorted by category.

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**.

5 The button for changing the property panel localization. If it is enabled then the properties panel will have a default localization.

❯ The **Properties Tab** or **Events** are a 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):

In the context menu of the panel Properties there is a command Localize Property Grid. If this option is enabled (checked), the translation of the panel Properties will be done. If this command is disabled (not checked), the names of the properties, events, values and description of the properties will stay in default English.

6.4 Keyboard Shortcuts

The report 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).

Buttons	Description
Ctrl+B	Makes letters **bold** for the selected text component
Ctrl+I	Makes letters *Italic* for the selected text component
Ctrl+U	Makes letters <u>Underlined</u> 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...
Ctrl+Tab	Switching pages in the report template
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
F3	Select the **"Text Editor"** tool
F6	Select the **"Copy Style"** tool
F8	Select the **"Hand"** 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

6.5   Report Checker

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:
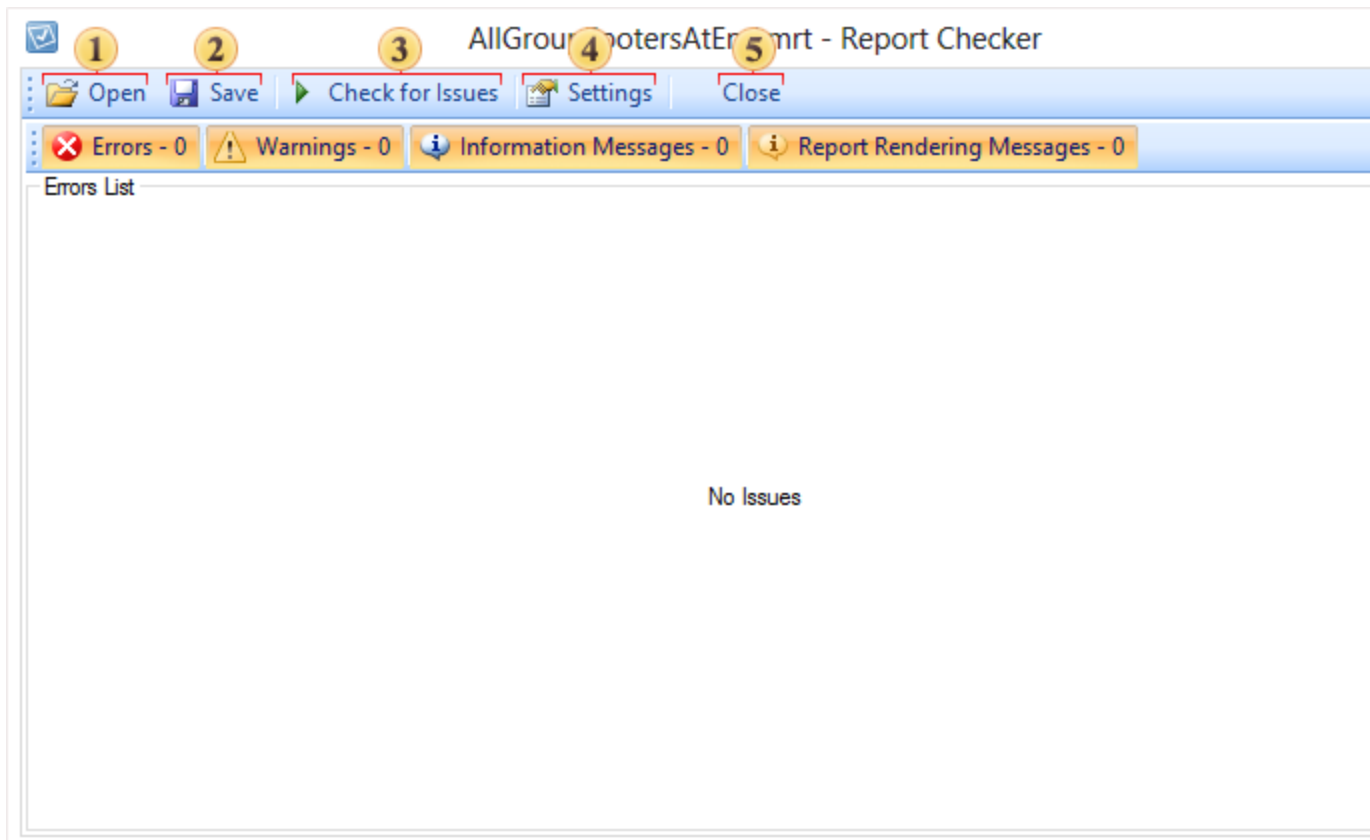
1 The button **Open**. Clicking this button, the user will see a dialog box to select a previously saved report and loading it to the Report Checker.

2 The button **Save** saves changes in the report, that was opened in the Report Checker.

3 The button **Check for Issues** starts the process of checking the report.

4 The button **Settings** opens the window of settings of the Report Checker. The picture below shows the Settings window:
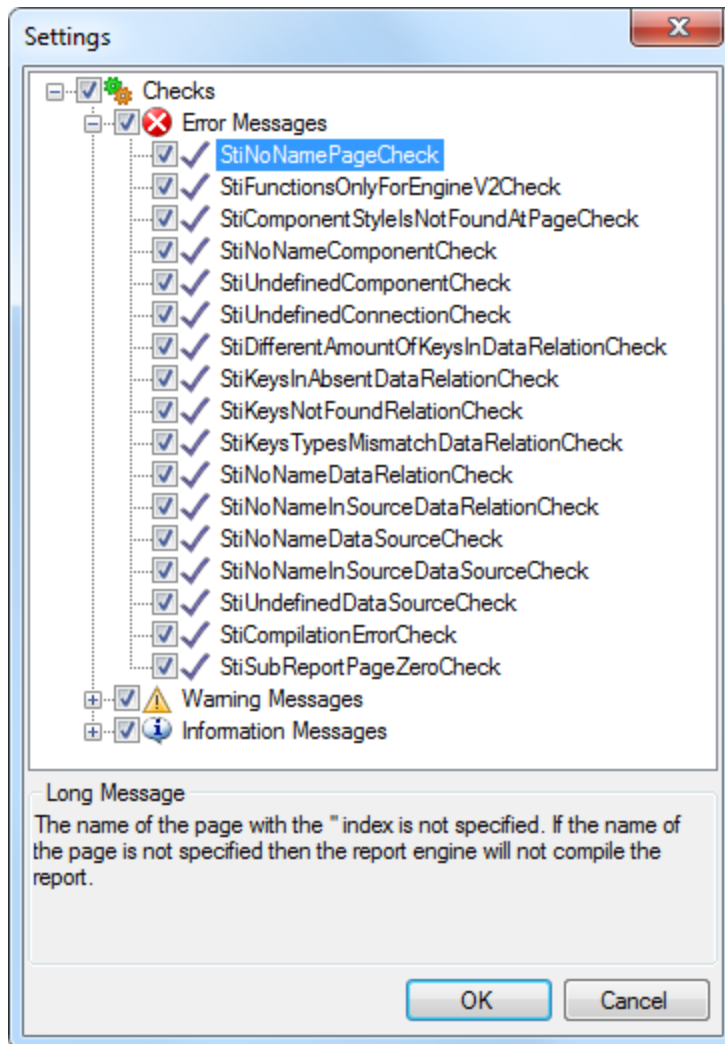
In this window, you can mark messages and warnings you want notifications to be displayed.

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**.

6.6 Globalization Editor

When designing reports, there can be situations when users who view rendered reports have different language cultures. In this case, they can make the required number of copies of the report, each of which is localized in a specific language. However, when editing one report template, you will have to do changes in all of its copies. Thus, every change in the report template will increase the time spent on preparing the report and significantly increases the likelihood of errors in copies of this report.

Our report generator provides the ability to localize the report depending on the selected report culture. The **Globalization Strings** tool is used for this. You can define a list of cultures with which the elements of the report will be translated. The items for which you can configure localization include:

❯ Report properties: Report Alias, Report Author, Report Description;
❯ Text component, text in cells, Rich text;
❯ Each cell of the Table and the Cross-tab components;
❯ Variables in the report;
❯ The text fields of the Chart component (labels, legends, rows, charts, and also you can override the values of the text properties before and the text after these chart items).

You should know that for each text component, as well as for each cell in the **Table** and **Cross-tab**, you can override several properties of this component. For example, if the report uses the text component **Text1**, then:

❯ In the **Text1.Hyperlink** property, you can specify a hyperlink (or expression) when you select a specific culture. For example, you can specify a hyperlink (or expression) on a localized page of your website.
❯ In the **Text1.Tag** property, you can specify a tag (or expression) for this text component when you select a specific culture. The tags in the report are used to refer to a particular report component.
❯ In the **Text1.Text** property, you can specify the text (or expression) of the text component that will be processed when the report is rendered and displayed to the user when a particular culture is selected.
❯ In the **Text1.Tooltip** property, you can specify the tooltip (or expression) of this text component when you select a specific culture.

If a property is not filled, then when you select a specific culture, the result will be empty. For example, if you do not specify anything for a particular culture in the **Text1.Text** property, then when you select this culture, the text component will be printed without any content.

Notice

The report culture does not depend on the localized GUI of the report designer. The culture of the report depends on the value of the **Culture** property. The list of values for this property depends on the list of cultures supported by the operating system. By

default, the report uses the current culture of the operating system.

To call the **Globalization Editor**, you should go to the report properties and click in the report template area.
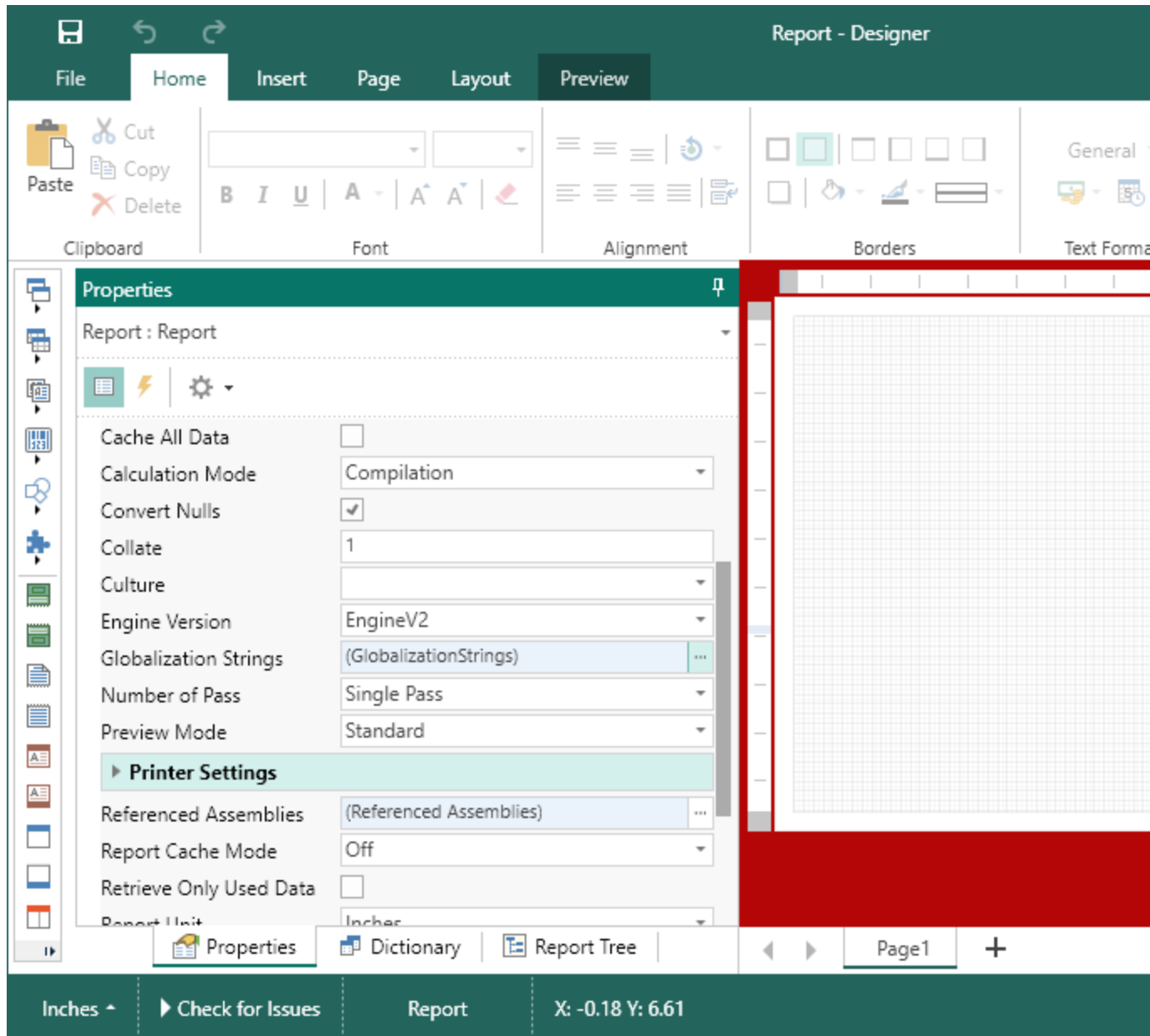
And on the properties panel, click the **Browse** button on the Globalization property.

Below is the view of the Globalization Editor.
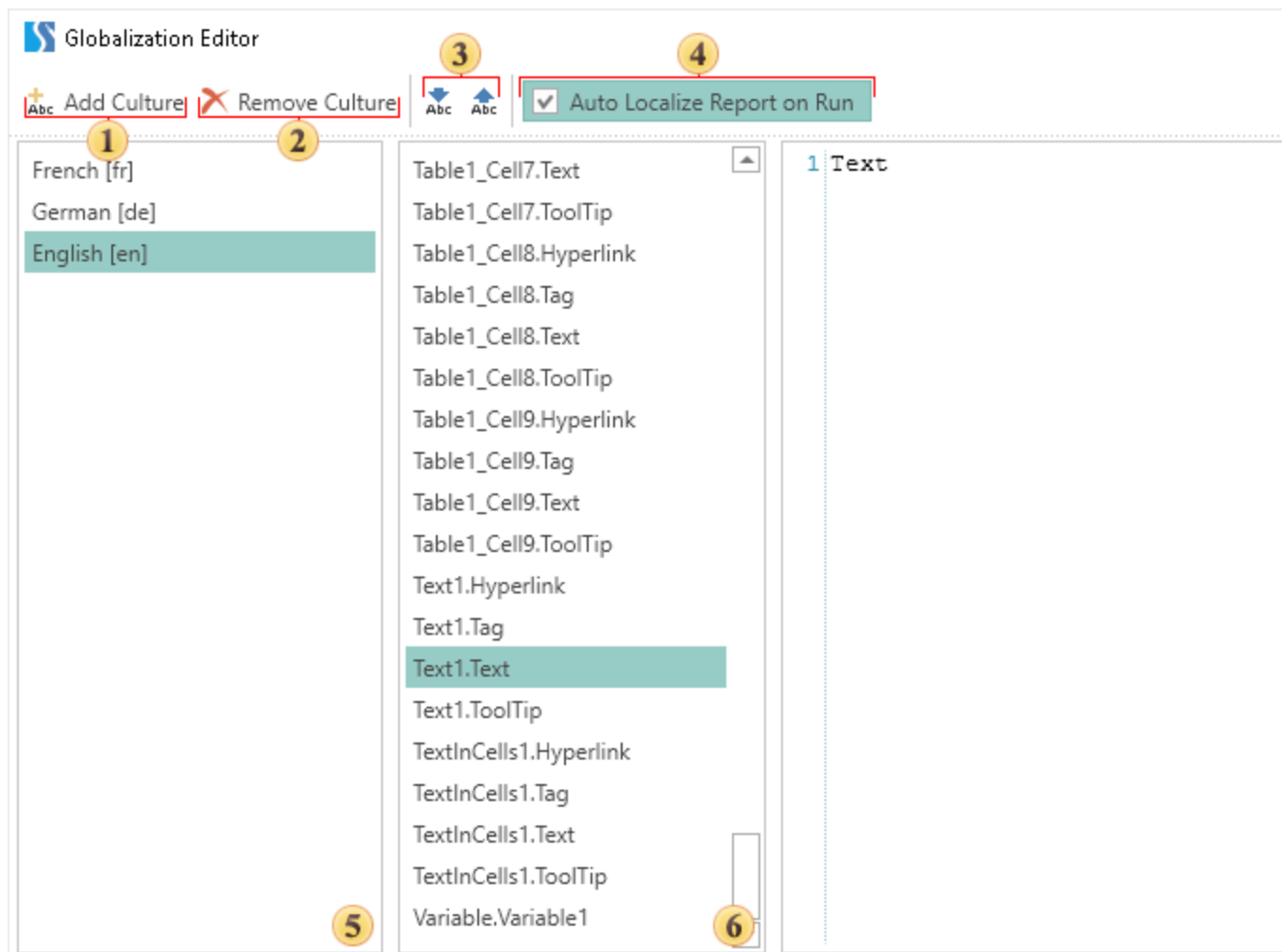
1 Click this button to add a new culture. The added cultures will be displayed in the list of cultures.

2 Select the culture in the list and click this button to remove the culture from the list.

3 The buttons to control cultures.
> Get the culture settings from the report, in this case, for the items of the selected culture, the values that are used in the report will be specified.
> Transfer culture settings to a report, in this case, the values from the selected culture will be specified for the report items.

4 If the **Auto Localize Report on Run** option is enabled, then, when rendering reports, the report engine will check the report culture and whether they are presented in the list. If identical cultures are found, then expressions of the report items will be

replaced.

5 The list of cultures, setting which, the localization of the report items will occur (i.e. replacing expressions that are specified in a particular culture).

6 The list of report items, which localization can be configured.

7 An expression of the item that will be assigned to the selected report item when you select a specific culture.

In order to automatically localize the report, you should specify the report culture after specifying the list of cultures and their settings. To do this, select the required value in the **Culture** property of the report. Then, when rendering the report, the report engine will check the report culture and their presence in the list of cultures. If identical cultures are found, the expressions of the report items will be replaced.

6.7   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:
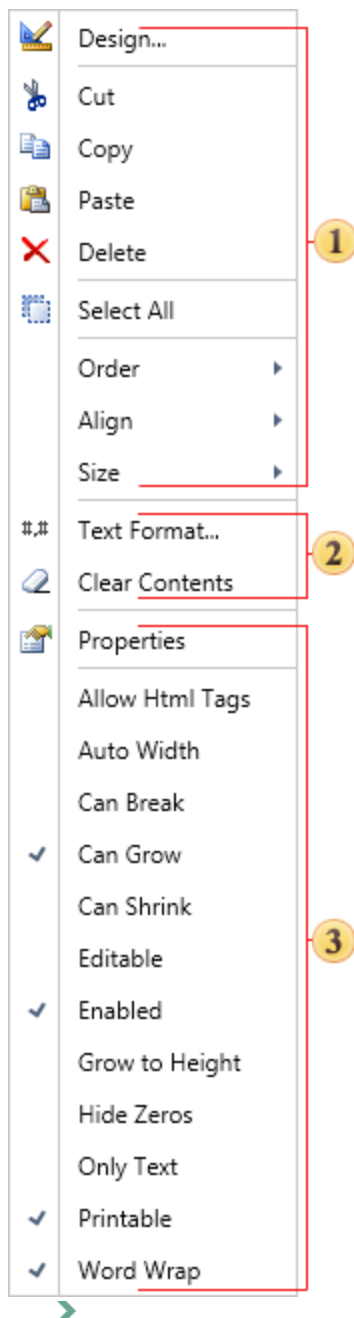
The context menu consists of three groups:

1 General commands. These are static commands, which can be applied to any component of the designer:

❯ 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.

❯ The **Cut** command cuts the selected component to the clipboard.

❯ The **Copy** command copies the selected component to the clipboard.

❯ The **Paste** command pastes from the copied or cut component from the clipboard.

❯ The **Delete** command deletes the selected component.

❯ 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.

Size	▶	⬚	Make Same Size as DataProductsCategoryID
		⬚	Make Same Width as DataProductsCategoryID
		⬚	Make Same Height as DataProductsCategoryID
		⬚	Make Horizontal Spacing Equal
		⬚	Make Vertical Spacing Equal
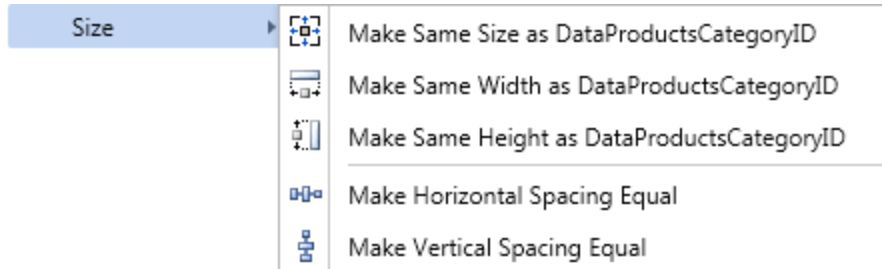
If you select all the components on the page, i.e. choose the **Select all** command in the context menu, or press the **Ctrl+A** key combination, then the prototype of the size for all components will be the size of the component that is placed on a higher level and higher than other components in the 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.

- Report
  - Page1
    - SubReport1
    - Chart1
    - Text2
    - DataProducts_SupplierID
    - ReportTitleBand1
    - CrossGroupFooterBand1
    - DataProducts
      - DataProducts_ProductName
      - DataProducts_CategoryID
      - DataProducts_QuantityPerUnit
      - RichText1
  - subReport_1
    - DataBand1
      - Text1
  - subReport_2
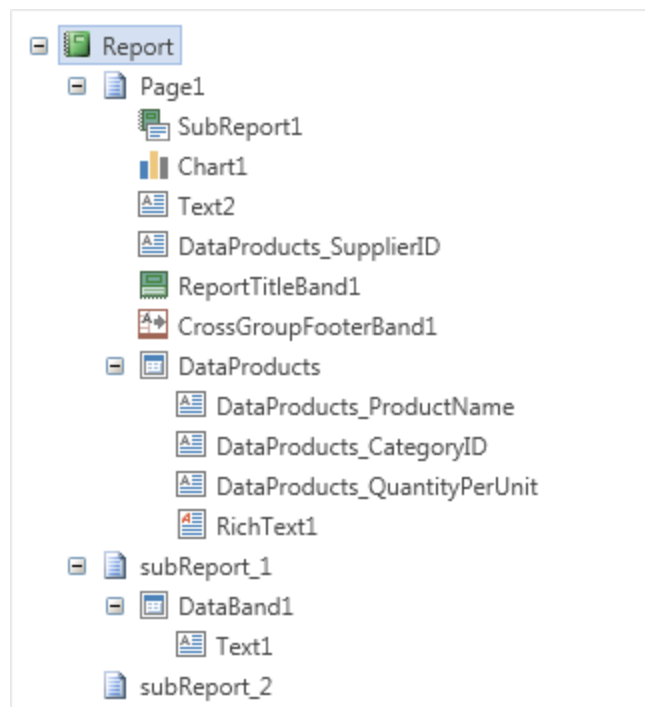
As seen on the picture above, the **PageHeaderBand** is located higher, so, in this case, its size is a sample of sizes for the remaining components. If the **PageHeaderBand** is absent, then, as a sample of sizes, the **PageFooterBand** will be taken. If the **PageFooterBand** is

absent, then, as a sample of sizes, the **Text1** will be taken.

❯ The **Select All** command selects all components on the current page.
❯ 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:
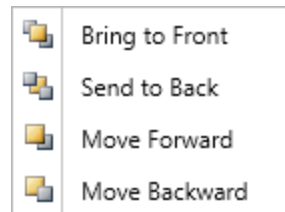
❯ The **Align** command invokes the submenu, in what it is possible to select the **Align to Grid** command. The picture below shows the **Align** submenu:
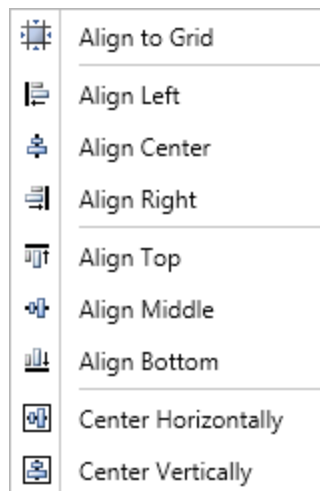
② Commands which are specific for the component.
③ It is possible to enable/disable different properties, without closing the context menu of the selected component.

6.8   Previewing Reports

Report Designer allows previewing a report before printing, exporting, sending via Email 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.

⏽ Notice. In the report designers for **WinRT** and **Mobile**, switching to the preview windows can be done by selecting the tab **Preview**.
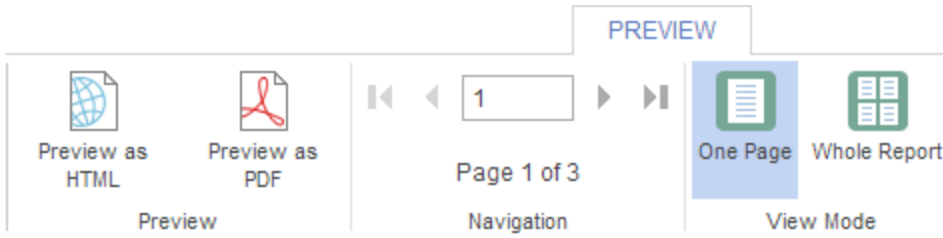
6.9 Toolbox

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**:

Category Shortcut · Category Components · Category Select Edit Mode
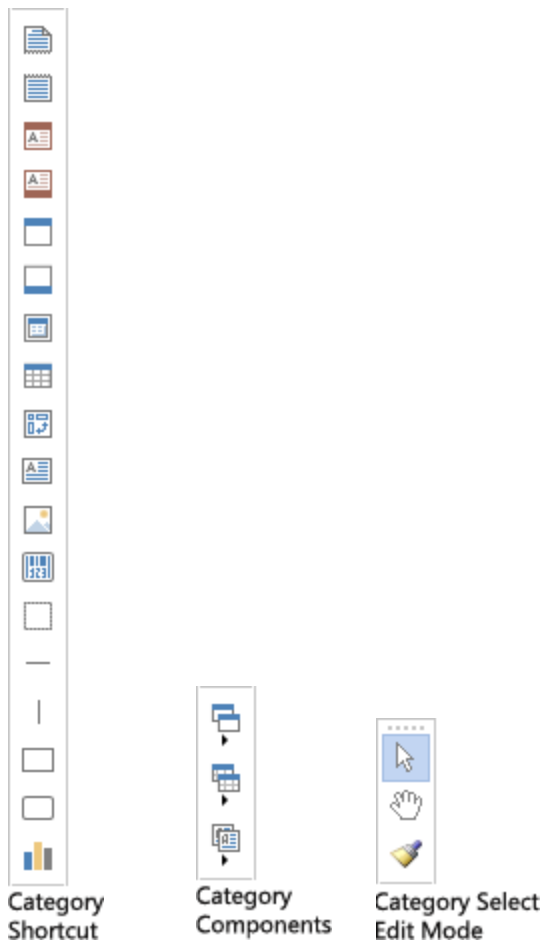
The Toolbox is located on the left side in the designer window and looks like a vertical panel. If necessary, the Toolbox bar can be shown or hidden. In Ribbon UI hiding or showing the **Toolbox** can be done by pressing the **Toolbox** button, which is located in the **View** tab. If the button is pressed, 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.

❗ **Notice**. The Toolbox is not always displayed on the left side. In some report designers, such as WinRT and Mobile, the development tools are located on the tab Insert.

6.9.1   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:

➤ 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.

➤ 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 Shortcut

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:
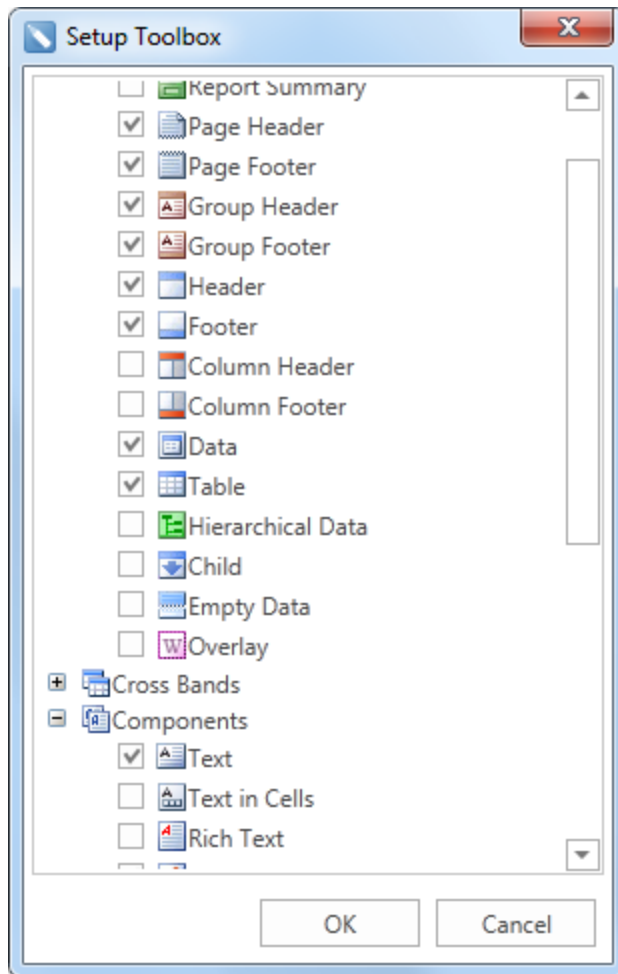
Components which are marked with the "check" in the **Setup Toolbox** window will appear on tools in the shortcut category. Accordingly, the components that are not marked will not be displayed.

6.10   Wizard Components Placement

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**.

6.11   Wizard Drag and Drop

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:
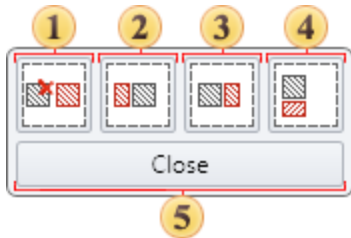
1 Substitute an expression in the text component which is placed in the report template, for an expression of the text component which is being dragged;

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.

6.12   Report Culture

The report contains a variety of data. Some of them are always presented in the same form, such as text, and appearance of other entries, for example a date and time, depend on the regional settings. By default, regional settings in the report correspond to the regional settings of the current culture in the operating system. If you want the data in the report be displayed the same way regardless the current culture in the operating system, you should apply a certain culture to this report. In order to apply the culture to the report, the report has the Culture property. In the field of the property, you should specify the code of the culture (the code format looks xx-XX, for example

en-GB). After that, the report generator will set the specified culture for the application and appearance of rendered components will fit this culture. Below is an example of the same report with different cultures.

Russian (Russia) (ru-RU)

ProductName	UnitPrice	OrderDate
Queso Cabrales	18,00p.	03.08.2008 23:00:00
Singaporean Hokkien Fried Mee	18,00p.	03.08.2008 23:00:00
Mozzarella di Giovanni	18,00p.	03.08.2008 23:00:00

Arabic (Libya) (ar-LY)

ProductName	UnitPrice	OrderDate
Queso Cabrales	18.000.د.ل	03/08/2008 11:00:00 م
Singaporean Hokkien Fried Mee	18.000.د.ل	03/08/2008 11:00:00 م
Mozzarella di Giovanni	18.000.د.ل	03/08/2008 11:00:00 م

English (United Kingdom) (en-GB)

ProductName	UnitPrice	OrderDate
Queso Cabrales	£18.00	03/08/2008 23:00:00
Singaporean Hokkien Fried Mee	£18.00	03/08/2008 23:00:00
Mozzarella di Giovanni	£18.00	03/08/2008 23:00:00

Note that the first column contains a text that does not depend on the report culture. The second is currency and the third is date-time columns. They depend on the culture. Therefore, when the culture is changed the data entry is changed too.

❗ Notice: It is virtually impossible to remember all codes of cultures. Therefore, for convenience, in the filed of the Culture property you may find a drop down list with the list of cultures that are available in the operating system on the computer.

If necessary, the components be independent from the specified culture, i.e. displayed the same way regardless an applied report culture, this component should have specific parameters, i.e. to apply the formatting of the text component. For example, it is necessary to have the prices on goods be displayed always in the same format. Below is a report with different cultures.

Russian (ru-RU)

ProductName	UnitPrice 1	UnitPrice 2
Sasquatch Ale	14,00p.	14,00p.
Steeleye Stout	18,00p.	18,00p.
Inlagd Sill	19,00p.	19,00p.

English (United States) (en-US)

ProductName	UnitPrice 1	UnitPrice 2
Sasquatch Ale	$14.00	$14.00
Steeleye Stout	$18.00	$18.00
Inlagd Sill	$19.00	$19.00

As can be seen from the picture above, the price in this report depends on the culture applied to the report, which is not quite true. In order the price has always been fixed, set the currency USD for the column UnitPrice 2. Now, regardless of the culture report, the price in this column will always be the same.

English (United States) (en-US)

ProductName	UnitPrice 1	UnitPrice 2
Sasquatch Ale	$14.00	$14.00
Steeleye Stout	$18.00	$18.00
Inlagd Sill	$19.00	$19.00

Russian (ru-RU)

ProductName	UnitPrice 1	UnitPrice 2
Sasquatch Ale	14,00p.	$14.00
Steeleye Stout	18,00p.	$18.00
Inlagd Sill	19,00p.	$19.00

As seen from the picture above, when applying the culture Russian (ru-RU), the currency of the second column is not changed while the first depends on the used culture.

Notice: If the selected report culture is not supported by the operating system, the

current culture of the operating system will be applied.

6.13   Publish

Publishing reports means saving them as separate projects or files to simplify and speed up the process of embedding these reports into an application on different platforms. The report is published using a wizard that can be called by clicking the **Publish** button on the Ribbon panel of the designer, or by selecting this command from the **File** menu:

After calling the wizard, you need to specify the platform for which the report will be published.

Information

Depending on the selected platform, the number of parameters may vary.

✖ Publishing Settings

As it is already mentioned, the number of parameters can vary depending on the selected platform. Consider the parameters of the wizard when publishing report for the JavaScript as an example.

① The option for selecting the type of the report deployment:

> **Project**. The report will be saved as a project to run it in the development environment or embed it into the application.

> **Standalone**. The report will be saved as a separate file (or files). For example, for the JavaScript platform, this will be an HTML page, and if you select the WinForms platform, then this will be the executable (exe) file.

2 The option for selecting a framework type. You can select a JavaScript application without using a framework, or select the Node.js framework.

3 The option to load a report from:
> **File**;
> **String**;
> **Hyperlink**.

Information

On some platforms, you can also load a report from:
> **Stream**,
> **Bite Array**,
> **Resource**,
> **Class**,
> **Assembly**.

---

4 The option for selecting an action with a report, after it is published:

> **Show**. The project will be created for viewing the report. When you run the project, the report viewer is called with this report. Also, when you select a Web platform for publishing, you can enable the report to be displayed in the full browser window.

> **Export**. The project will be created to convert the report. When you run the project, the report will be converted to the selected format. You should also specify the type of document to which the report will be converted.

> **Design**. The project will be created to edit the report. When you run the project, the report designer with this report will be called.

Information

If there are data sources and parameters (variables) in the report, then when you select any action, you should specify the data connection parameters:

> **Use Connection from Report**. If the connection is present in the report, then it will be used when the project is run.

> **Replace Connection String**. Provides the ability to specify a new connection string to the data storage.

i If the report uses file data sources (XML or JSON), then, instead of the

---

Replace Connection String option, the **Replace Path to Data** parameter will be present. Using it you can specify a new path to the data files.

❯ **Register Data from Code**. Select this option if you want to use data from XML, JSON sources or from Business objects. If you select this item, you can also enable the following options:

i **Synchronize Report Dictionary**. Use this option to synchronize the registered data in the data storage and in the data dictionary of the report.

i **Use Only for Report Preview**. Select this option to use the data only for preview.

In addition, the data dictionary can contain variables. When you select the **Show** or **Export** action, you can define a value for each variable:

❯ **Use Value from Report**. The value of the variable will remain as the default.
❯ **Replace Value from Code**.
❯ **Request from User**. Use the value entered by the user.

---

5 Options that depend on the selected action. In this case, the **Show** action is selected, so the **Display the Viewer in Full Browser Window** option is available.

6 **Include License Key**. If this option is not enabled, the report will be displayed with the Trial watermark. If you enable this option, you can connect the license key in one of the following ways:

❯ **String**;
❯ **File**.

7 **Include Localization**. This option is relevant only for the **Show** and **Design** actions. When this option is enabled, select the interface of the viewer localization if the **Show** action is selected, or the designer, if the **Design** action is selected.

8 Include UI Theme. This option is relevant only for the **Show** and **Design** actions. When this option is enabled, you can specify the theme of the layout of the viewer interface, if the **Show** action is selected, or the designer, if the **Design** action is selected.

9 **Use Compressed Scripts**. If you enable this option on, the size of the scripts will significantly decrease but when you run the application it will take time to unpack them.

10 The **Hide Options** button is used to expand and collapse the options bar in the publish wizard.

11 The **Save Project Package** button. When you click this button, a dialog box will be displayed to specify the location of the project or standalone application. Note, when

saving a project, it will be saved as a zip archive.

12 The **Close** button can be used to close the Publish wizard.

13 The field in which the current project code is displayed. Also in this field, you can find the **Copy** button, with which you can copy the code to the clipboard.

> **Information**
>
> On some platforms, the **Get Stimulsoft Libraries from NuGet** option may be present. In this case, when the project is run, if there are no Stimulsoft libraries in it, they will be automatically loaded from the NuGet repository.
>
> For the Java platform you can find the **Get Stimulsoft Libraries from Maven** option.

Publishing report step by step

Step 1: Run the report designer.

Step 2: Create a report or open it.

Step 3: Save the last changes.

Step 4: Call the Publish wizard by clicking the **Publish** button on the Ribbon panel or by clicking **Publish** from the **File** menu.

Step 5: Select the platform for which the report will be published. The following platforms are available ASP.NET, ASP.NET MVC, .NET Core, WinForms, WPF, JavaScript, PHP, Java.

Step 6: Specify the publishing settings for the selected platform.

Step 7: Click the **Save Project Package** button and specify the location where the project should be saved.

Step 8: Unpack the archive, if the package is saved as a project. Open the .sln file with Visual Studio or another development environment.

Step 9: Make changes in the project code, if necessary.

Step 10: Run the project.

7 Viewing Reports

7.1 Report Viewer Structure

On the picture below you may find the basic elements of the report viewer.

① In this panel contains menus which have the basic control commands of the report

viewer.

2 The basic commands to control the report are represented on the toolbar.

3 The tree of bookmarks of the output report. Using these bookmarks it is possible to jump by structure elements of a report.

4 The area where the report is shown.

5 The report thumbnails panel. Decreased copies of report pages are shown on this panel. The panel is used to quickly navigate throughout a report.

6 Find Panel.

7 The toolbar to scroll up or down in reports pages.

8 The toolbar to select the mode of report showing.

9 The toolbar to zoom reports.

7.2   Basic Toolbar of Report Viewer

The main toolbar locates commands to control the report. Below is the structure of the toolbar with the description of each command:

1 Print a report. 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 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.

6 Opens the dialog for changing basic parameters of the rendered report.

7 Show/hide the reports thumbnails.

8 Enable the search panel.

9 Open the report designer and show the current page for editing.

10 Run the full screen mode of report showing.

11 Change zoom of the report to display only one full page. More than one page by the width can be output.

12 Change zoom of the report to fit the page width to the screen width.

13 Closes report viewer.

7.3   Page Navigation

On the picture below you can see the toolbar that is used for report navigation.

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 one.
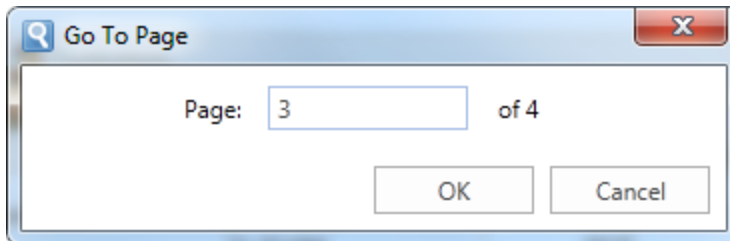
4 Set the next page of a report as the current one.

5 Set the last page of a report as the current page.

7.4   Displaying Mode

The viewer for WinForms supports three modes of viewing pages:

> ❑ **Single Page**
> ⊟ **Continuous**
> ⊟ **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.

Each mode has its own advantages.

Single page. In this mode the current page of a report is shown in the window of the viewer. The picture below shows how this mode works.
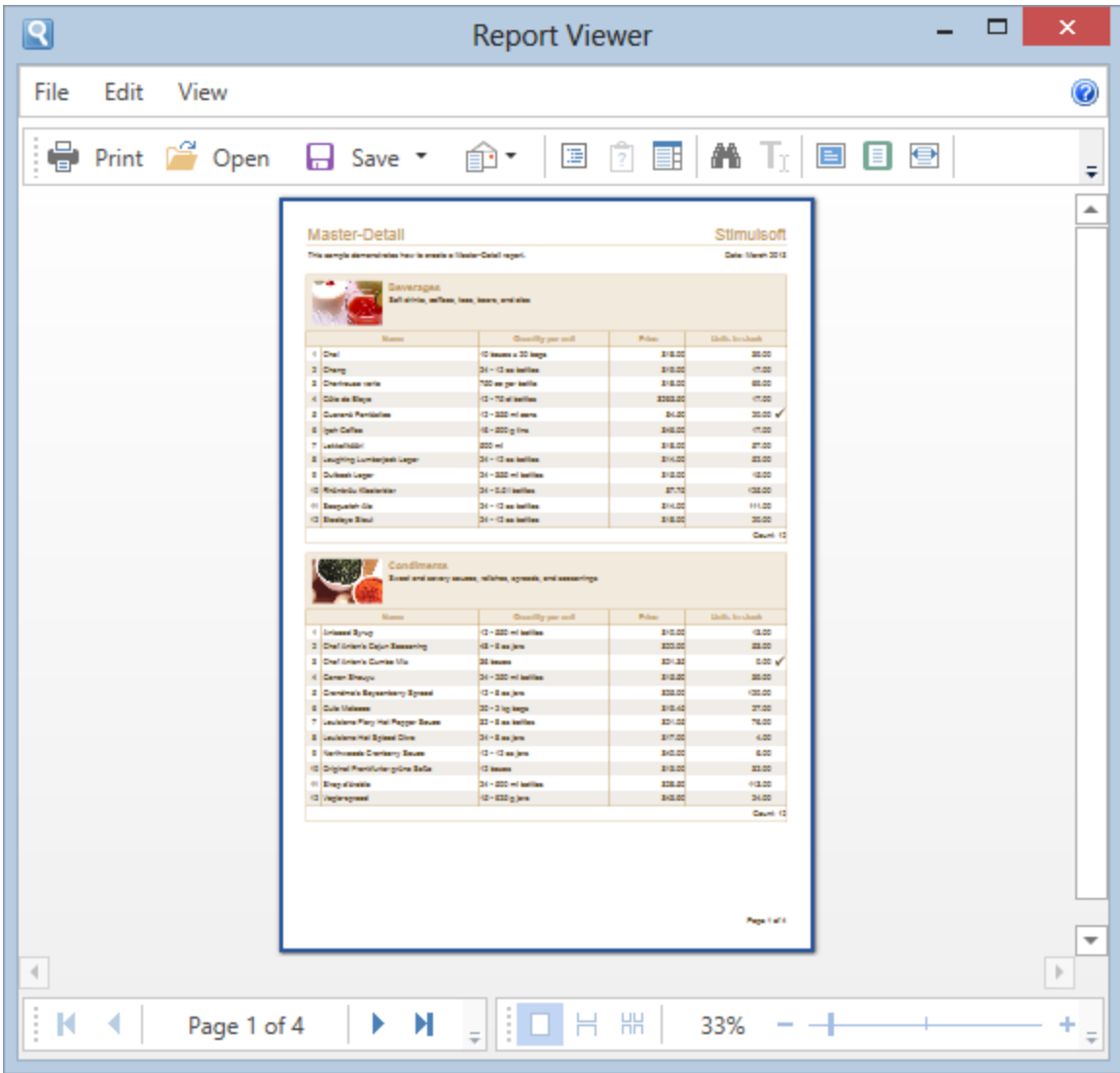
Continuous. In this mode all pages are placed into one vertical line. The picture below shows how this mode works.

Multiple Pages. In this mode as many pages in the selected zoom as they can fill the window of the viewer are shown. The picture below shows how this mode works.
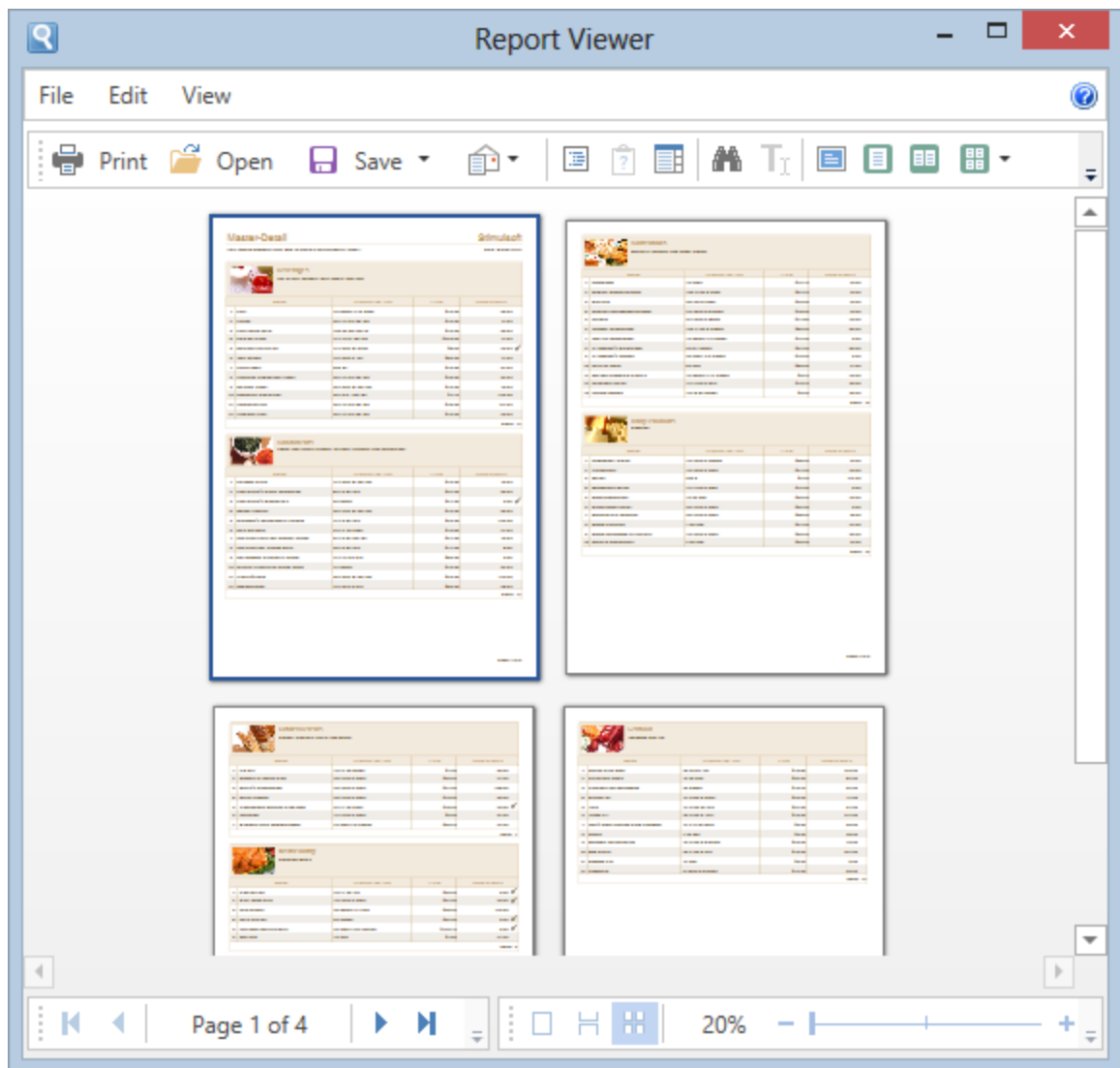
7.5   Search Panel

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.
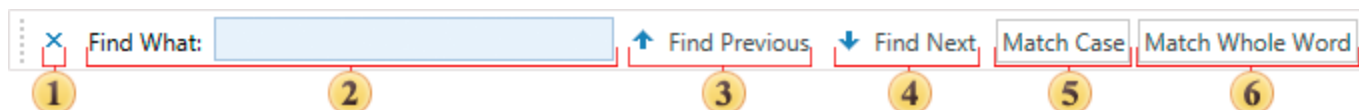
1 Close the search panel.
2 The field to put a text that should be found.

③ The button to run search.

④ The button to run search.

⑤ If the flag is set, then search will be repeated considering the case.

⑥ If the flag is set, then search will be done considering the whole word.

7.6 Keyboard Shortcuts

The list of keyboard shortcuts in the report viewer is as follows:

Shortcut	Actions
Ctrl+P	Print a report
Ctrl+O	Close a report
Ctrl+Shift+N	Add a new page to the report
Ctrl+Shift+D	Delete the current page of a report
Ctrl+Shift+E	Edit the current page of a report in the report designer
Ctrl+Shift+S	Change report parameters
Ctrl+B	Enable/disable tree of bookmarks
Ctrl+T	Enable/disable thumbnails
Ctrl+F	Search
Ctrl+E	Edit components which support editing
F2	Run the full screen mode of view a report
F3	Set zoom of a report view - one page
F4	Set zoom of a report view - two pages
F5	Set zoom of a report view - by page width
Ctrl+G	Jump to page
Shift+F2	Enable the page view mode - one page
Shift+F3	Enable the page view mode - continues
Shift+F4	Enable the page view mode - some pages

7.7 Sending Report via E-Mail

Any rendered report can be sent via **Email**. Send a report via Email following the instruction below.

▶ The report is exported as a file. The file format is defined by the user in the menu **Send Email**;

▶ Then create a new message and attach a file to the Email;

▶ A dialog of the Email client is open by default, i.e. the wizard for sending Emails is invoked.

The picture below shows the menu **Send Email**.
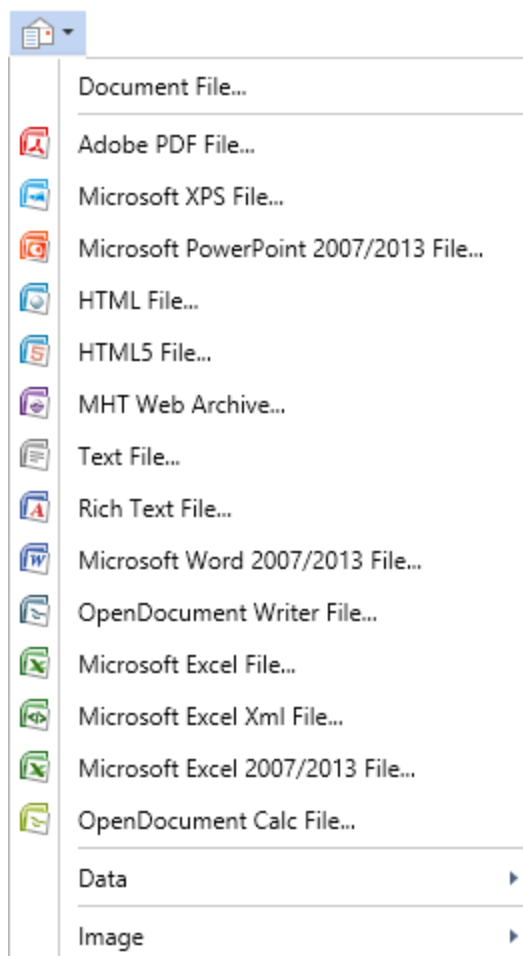
7.8 Dot-Matrix Viewer

The **Dot-Matrix** viewer is designed to preview the report before printing it on dot matrix printer. The Dot matrix printer is used to print only the text and characters of

pseudographics. Accordingly the viewer displays only the text and borders of objects as pseudographics characters.

7.8.1 Toolbar

The picture below shows the toolbar of the **Dot-matrix** viewer:
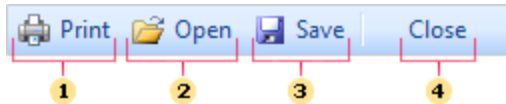
1 Prints the report. After activation of this command the Print dialog will be displayed, where you will be asked to select printing options.
2 Opens a previously saved text file.
3 Saves the rendered report to a text file.
4 Closes the **Dot-matrix** viewer dialog box.

7.8.2 Bar Options

The **Options** bar is grouped and each group is located on a separate tab. The picture below shows the options bar:

1 The **Kill Space Lines** option removes empty rows in the text.

2 The **Kill Space Graph Lines** option deletes the rows that contain only the "vertical line" pseudographics characters.

3 The **Put Feed Page Code** option inserts the FormFeed symbol on the bottom of each page.

4 The **Draw Border** option draws the borders of the objects of the selected type.

5 The **Cut Long Lines** option cuts long lines of the text that is out of bounds of the text component.

6 - **8** options are the parameters of the border and define its type:

6   **Simple** border is drawn with + - | symbols and will be saved and printed in any encoding;

7 **Unicode-Single** single lines of pseudographics are used;

8 **Unicode-Double** double lines of pseudographics are used;
Pseudographics characters are not present in each encoding.

9 - **10** options. When exporting to text all the coordinates and sizes of objects are recalculated. Zoom **X** and Zoom **Y** coefficients control this conversion.

By default, Zoom **X** = 100%, Zoom **Y** = 100%. With these values of the parameter, the A4 page is converted to text with sizes of 80 characters by width and 62 rows by height. This corresponds to using the **Pica** font of the printer (80 characters per line) and the line spacing **1,0**. The following values are frequently used:

➤ Zoom **X** = 100% corresponds to using the **Pica** font of the printer (80 characters per line);

➤ Zoom X = 120% corresponds to using the **Elite** font of the printer (96 characters per line);

➤ Zoom X = 170% corresponds to using the condensed font of the printer (136 characters per line);

➤ Zoom Y = 100% corresponds to the using the line spacing 1,0.

9 **Zoom X: -** by the width of the page.
10 **Zoom Y: -** by the height of the page.
11 **Encoding** - encoding of the text.
12 **Auto Refresh** automatically updates the rendered report if there are any changes in the parameters.
13 The **Refresh** button is used to manually update the report.

7.9   Special Viewing Options in Web

Recommendations on Placing Components on Page

How the **StiWebViewer** helps to view a report? To view a report the **StiWebViewer** exports it to the HTML format. This HTML text is output in the part of the **StiWebViewer** that is used to show reports. The HTML file is formed as one big table. The output is done in the HTML format do there are some limitations when report rendering. Stimulsoft Reports stores all objects separately but not as a table. When converting a report to the HTML format the objects edges may be intersected. Such intersections may lead to incorrect output of a report in the browser, though the report generator tries to output a report correctly with overlapping objects. Therefore, it is better do not overlap objects. Examples of components overlapping are shown on the picture below.
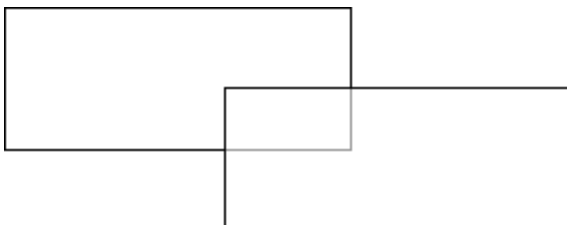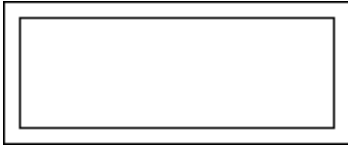
When report rendering, it is better use the grid. It allows placing objects by the grid and getting correct viewing a report in the browser.

Using Graphic Objects in Report
Stimulsoft Reports offers full set of graphic objects. The following graphic objects are used in web:
- Images;
- Charts;
- Graphic primitives (the Shape component);
- Bar-codes;
- RTF text;
- CheckBox.

The Vertical Line, Horizontal Line, Rectangle components are not graphic objects.
Also it is important to consider that vector images (WMF, EMF, EMF+) are not supported by the HTML format. So they will be converted to images in the pixel format.

❗ **Notice!** All text components which contain text are rotated (the value of the **Angle** property is not 0) and converted to images. Besides, if the **ExportAsImage** property is set to **true** then the text components will also be converted to the image.

All components are joined with one rule - all of them will be converted as images. The HTML format does not allow passing an image in its body and the report generator uses the cache of a page or the cache of a session for saving images. When huge amount of calling to a report and multiple images in a report, there can be huge amount of objects in the page cache or in the session cache. And these objects will take additional server memory. Therefore, it is better do not use many graphic objects. Using the **ServerTimeOut** property can be used to set the time of objects caching in the page cache or in the session cache.

❗ **Notice!** HTML supports some formats of showing images (JPEG, PNG, BMP, and GIF). It is possible to set the image type using the **ImageFormat** property of the StiWebViewer component. Every type of an image has its own advantages and disadvantages.

Displaying Images Placed on Server
If an image that should be output is static and can be saved on the server then it is recommended to use the **ImagerUrl** property of the **Image** component for showing images. When using this property the report generator does not save the image in the cache of a page or the cache of a session but puts a link on this image. So the report generator saves nothing in the cache of a page or the cache of a session and the server memory is not used for this.

Printing Reports
It is difficult to print a report from the browser. Stimulsoft Reports has three methods of printing:
- Converting a report to the PDF file and passing it to the end-user for printing.
- Printing a report with preview in the pop-up window.

> Printing without preview.

The first method is the best way. It allows printing a report more precisely. But it is required to have installed Adobe Acrobat to print a report to the PDF format. Often this requirement is a big disadvantage. When printing reports with preview the report generator creates a new pop-up window. A report in the HTML format is output in this window. The end-user may format this report and print it. In printing report without preview the report generator prints a report without preview. When choosing the method of printing characteristics of each method should be considered.
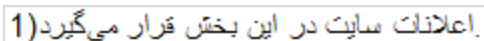
❗ **Notice!** The StiWebViewer component cannot control page parameters (page size, page orientation, page margins) when printing using the 2 and 3 method. All parameters are controlled with the browser.

8 Right To Left
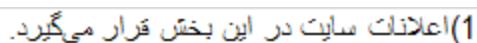
By default, components are output from left to right. The **Right to Left** property allows changing the mode of showing report items.

8.1 Text Component

How the text will be output depends on the **RightToLeft** property. If it is set to **false**, then a text (all symbols except letters) is output from left to right. The picture below shows a text sample in Arabic that is output from left to right:

If the **RightToLeft** property is set to **true**, then a text is output from right to left. The picture below shows a text sample in Arabic that is output from right to left:

In any case a text written in a right-to-left language will be output right to left.

8.2 Text In Cells Component

A text in cells is placed symbol-by-symbol (one symbol or a space - one cell). How the text will be output depends on the **RightToLeft** property. If it is set to **false**, then a text

is output from left to right. The picture below shows a text sample in Arabic that is output from left to right:

If the **RightToLeft** property is set to **true**, then a text is output from left to right. The picture below shows a text sample in Arabic that is output from right to left:
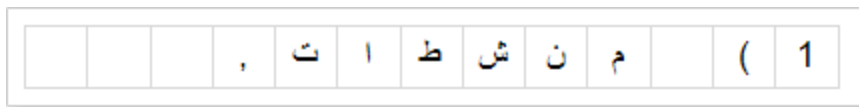
The **RightToLeft** property of the **Text in Cells** component works the same way with all languages. So a text characters and sy6mbols will be output from left to right or from right to right depending on the value of this property. The picture below shows a text output in "left to right" (the first picture) and right to left (second picture) modes:
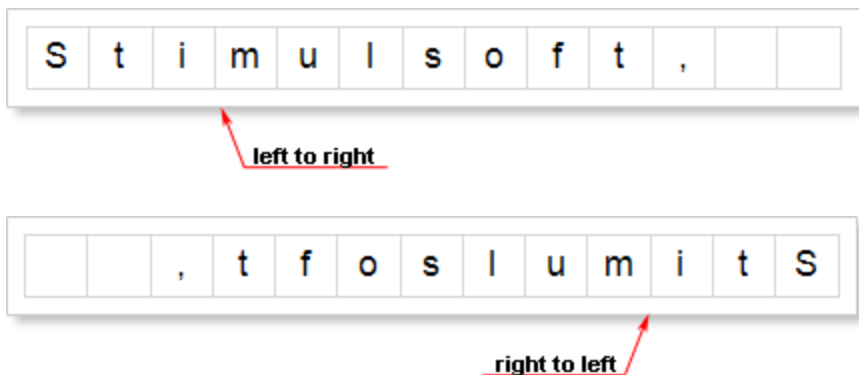
The **RightToLeft** property depends on the **Continuous Text** property. If the **Continuous Text** property is set to **true**, then the **RightToLeft** property will not work. In other words, a text will be output from left to right regardless the **RightToLeft** property. If the **Continuous Text** property is set to **false**, then the text direction will depend on the value of the **RightToLeft** property.

8.3   Cross Table Component

The cross table component has the **RightToLeft** property, that allows showing a cross-table in the right-to-left mode. If the **RightToLeft** property is set to **false**, then the

cross table is rendered in the "left-to-right" mode. The picture below shows a cross table sample with the **RightToLeft** property set to **false**:

Products / 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 the **RightToLeft** property of a cross table is set to **true**, then the cross table is output in the "right-to-left" mode. The picture below shows a cross table sample with the **RightToLeft** property set to **true**:

Total	Seafood	Produce	Meat/Poultry	Grains/Cereals	Dairy Products	Confections	Condiments	Beverages	CategoryName / Country
168	42	20		38		29	24	15	Australia
20								20	Brazil
266			136			17	113		Canada
100	100								Denmark
132						75		57	Finland
246	62				98			86	France
355	10	26		22		140	32	125	Germany
80				57	23				Italy
162	55	39	29				39		Japan
51						51			Netherlands
164					164				Norway
70				26			27	17	Singapore
108					108				Spain
389	224			165					Sweden
143						74	13	56	UK
665	208	15					259	183	USA
3119	701	100	165	308	393	386	507	559	Total

By default, the **RightToLeft** property of the cross table is set to **false**, this means that the cross table is output from left to right.

8.4   Chart Component

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.):
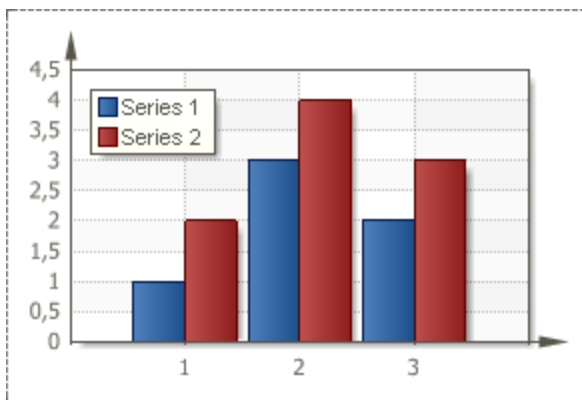
If the **Reverse Horizontal** property is set to **true**, then the chart will appear in the opposite direction horizontally. The picture below shows an example of a chart, with the Reverse Horizontal property is set to true (As one can see, the values of the x-axis have right to left direction.):
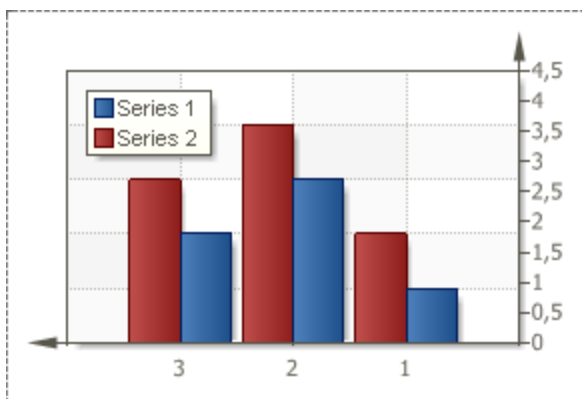
By default, the **Reverse Horizontal** property is set to **false**.

8.5 Columns on Page

Stimulsoft Reports prints bands until there is a free space on a page. Then, instead of creating a new page, the reporting tool adds a new column on the right. Then it prints data from the top of a page. This happens until all data are printed and page will be exhausted. The columns direction is always from top to bottom, and a mode of showing columns can be different. there are two modes: "left to right" and "right to left". The mode of showing columns on a page depends on the value of the **RightToLeft** property of a page. If the **RightToLeft** property is set to **false**, then columns will be output in the "left to right" mode. If this property of a page is set to **true**, then columns will be output in the "right to left" mode. The picture below shows columns on a page output in two modes:

Company	Phone	Company	Phone
1.Alfreds Futterkiste	030-0074321	43.Laughing Bacchus Wine Cellars	(604) 555-3392
2.Ana Trujillo Emparedados y helados	(5) 555-4729	44.Lazy K Kountry Store	(509) 555-7969
3.Antonio Moreno Taqueria	(5) 555-3932	45.Lehmanns Marktstand	069-0245984
4.Around the Horn	(171) 555-7788	46.Let's Stop N Shop	(415) 555-5938
5.Berglunds snabbköp	0921-12 34 65	47.LILA-Supermercado	(9) 331-6954
6.Blauer See Delikatessen	0621-08460	48.LINO-Delicatese	(8) 34-56-12
7.Blondesddsl père et fils	88.60.15.31	49.Lonesome Pine Restaurant	(503) 555-9573
8.Bólido Comidas preparadas	(91) 555 22 82	50.Magazzini Alimentari Riuniti	035-640230
9.Bon app'	91.24.45.40	51.Maison Dewey	(02) 201 24 67
10.Bottom-Dollar Markets	(604) 555-4729	52.Mère Paillarde	(514) 555-8054
11.B's Beverages	(171) 555-1212	53.Morgenstern Gesundkost	0342-023176
12.Cactus Comidas para llevar	(1) 135-5555	54.North/South	(171) 555-7733
13.Centro comercial Moctezuma	(5) 555-3392	55.Océano Atlántico Ltda.	(1) 135-5333
14.Chop-suey Chinese	0452-076545	56.Old World Delicatessen	(907) 555-7584
15.Comércio Mineiro	(11) 555-7647	57.Ottilies Käseladen	0221-0644327
16.Consolidated Holdings	(171) 555-2282	58.Paris spécialités	(1) 42.34.22.66
17.Die Wandernde Kuh	0711-020361	59.Pericles Comidas clásicas	(5) 552-3745
18.Drachenblut Delikatessen	0241-039123	60.Piccolo und mehr	6562-9722
19.Du monde entier	40.67.88.88	61.Princesa Isabel Vinhos	(1) 356-5634
20.Eastern Connection	(171) 555-0297	62.Que Delícia	(21) 555-4252
21.Ernst Handel	7675-3425	63.Queen Cozinha	(11) 555-1189
22.Familia Arquibaldo	(11) 555-9857	64.QUICK-Stop	0372-035188
23.FISSA Fabrica Inter. Salchichas S.A.	(91) 555 94 44	65.Rancho grande	(1) 123-5555
24.Folies gourmandes	20.16.10.16	66.Rattlesnake Canyon Grocery	(505) 555-5939
25.Folk och fä HB	0695-34 67 21	67.Reggiani Caseifici	0522-556721
26.France restauration	40.32.21.21	68.Ricardo Adocicados	(21) 555-3412
27.Franchi S.p.A.	011-4988260	69.Richter Supermarkt	0897-034214
28.Frankenversand	089-0877310	70.Romero y tomillo	(91) 745 6200
29.Furia Bacalhau e Frutos do Mar	(1) 354-2534	71.Santé Gourmet	07-98 92 35
30.Galería del gastrónomo	(93) 203 4560	72.Save-a-lot Markets	(208) 555-8097
31.Godos Cocina Típica	(95) 555 82 82	73.Seven Seas Imports	(171) 555-1717
32.Gourmet Lanchonetes	(11) 555-9482	74.Simons bistro	31 12 34 56
33.Great Lakes Food Market	(503) 555-7555	75.Spécialités du monde	(1) 47.55.60.10
34.GROSELLA-Restaurante	(2) 283-2951	76.Split Rail Beer & Ale	(307) 555-4680
35.Hanari Carnes	(21) 555-0091	77.Suprêmes délices	(071) 23 67 22 20
36.HILARION-Abastos	(5) 555-1340	78.The Big Cheese	(503) 555-3612
37.Hungry Coyote Import Store	(503) 555-6874	79.The Cracker Box	(406) 555-5834
38.Hungry Owl All-Night Grocers	2967 542	80.Toms Spezialitäten	0251-031259
39.Island Trading	(198) 555-8888	81.Tortuga Restaurante	(5) 555-2933
40.Königlich Essen	0555-09876	82.Tradição Hipermercados	(11) 555-2167
41.La corne d'abondance	30.59.84.10	83.Trail's Head Gourmet Provisioners	(206) 555-8257
42.La maison d'Asie	61.77.61.10	84.Vaffeljernet	86 21 32 43

left to right

right to left

8.6 Columns in Data Band

"Down Then Right" direction

In this direction the reporting tool tries equally to distribute all rows between columns. Then, after distribution rows between columns, the first column is output. And the column is not output to the end of a page, but until the number of elements that are distributed for this column. Then the second column is output. So the data take as much space on the page as it is required. So data will be distributed approximately equally among all the columns. And all data will be presented on a sheet in a convenient form. The mode of showing columns depends on the value of the **RightToLeft** property of

the **DataBand**. If the **RightToLeft** property is set to **false**, then columns on the report page will be displayed from left to right. If the **RightToLeft** property is set to **true**, then the column on the report page will be displayed from right to left. The picture below shows examples of two modes of showing columns on report pages:

left to right

right to left

"Right Then Down" direction

In this direction lines are sequentially output in the **Data Band**. By default the mode of output is left to right. Row are displayed - one line in one column. When all rows are displayed in columns in the **Data Band**, a new Data Band is created and it again displays all the rows in columns. So, the data will take as much space on the page as it is required. The mode of showing columns depends on the value of the **RightToLeft** property of the **DataBand**. If the **RightToLeft** property is set to **false**, then columns on the report page will be displayed from left to right. If the **RightToLeft** property is set to **true**, then the column on the report page will be displayed from right to left. The picture below shows examples of two modes of showing columns on report pages:

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 B	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
64.QUICK-Stop	65.Rancho grande	66.Rattlesnake Canyon Grocery
67.Reggiani Caseifici	68.Ricardo Adocicados	69.Richter Supermarkt
70.Romero y tomillo	71.Santé Gourmet	72.Save-a-lot Markets
73.Seven Seas Imports	74.Simons bistro	75.Spécialités du monde
76.Split Rail Beer & Ale	77.Suprêmes délices	78.The Big Cheese
79.The Cracker Box	80.Toms Spezialitäten	81.Tortuga Restaurante
82.Tradição Hipermercados	83.Trail's Head Gourmet Provisiones	84.Vaffeljernet
85.Victuailles en stock	86.Vins et alcools Chevalier	87.Wartian Herkku
88.Wellington Importadora	89.White Clover Markets	90.Wilman Kala
91.Wolski Zajazd		

left to right

Company	Company	Company
3.Antonio Moreno Taquería	2.Ana Trujillo Emparedados y helados	1.Alfreds Futterkiste
6.Blauer See Delikatessen	5.Berglunds snabbköp	4.Around the Horn
9.Bon app'	8.Bólido Comidas preparadas	7.Blondesddsl père et fils
12.Cactus Comidas para llevar	11.B's Beverages	10.Bottom-Dollar Markets
15.Comércio Mineiro	14.Chop-suey Chinese	13.Centro comercial Moctezuma
18.Drachenblut Delikatessen	17.Die Wandernde Kuh	16.Consolidated Holdings
21.Ernst Handel	20.Eastern Connection	19.Du monde entier
24.Folies gourmandes	23.FISSA Fabrica Inter. Salchichas B	22.Familia Arquibaldo
27.Franchi S.p.A.	26.France restauration	25.Folk och fä HB
30.Galería del gastrónomo	29.Furia Bacalhau e Frutos do Mar	28.Frankenversand
33.Great Lakes Food Market	32.Gourmet Lanchonetes	31.Godos Cocina Típica
36.HILARION-Abastos	35.Hanari Carnes	34.GROSELLA-Restaurante
39.Island Trading	38.Hungry Owl All-Night Grocers	37.Hungry Coyote Import Store
42.La maison d'Asie	41.La corne d'abondance	40.Königlich Essen
45.Lehmanns Marktstand	44.Lazy K Kountry Store	43.Laughing Bacchus Wine Cellars
48.LINO-Delicateses	47.LILA-Supermercado	46.Let's Stop N Shop
51.Maison Dewey	50.Magazzini Alimentari Riuniti	49.Lonesome Pine Restaurant
54.North/South	53.Morgenstern Gesundkost	52.Mère Paillarde
57.Ottilies Käseladen	56.Old World Delicatessen	55.Océano Atlántico Ltda.
60.Piccolo und mehr	59.Pericles Comidas clásicas	58.Paris spécialités
63.Queen Cozinha	62.Que Delícia	61.Princesa Isabel Vinhos
66.Rattlesnake Canyon Grocery	65.Rancho grande	64.QUICK-Stop
69.Richter Supermarkt	68.Ricardo Adocicados	67.Reggiani Caseifici
72.Save-a-lot Markets	71.Santé Gourmet	70.Romero y tomillo
75.Spécialités du monde	74.Simons bistro	73.Seven Seas Imports
78.The Big Cheese	77.Suprêmes délices	76.Split Rail Beer & Ale
81.Tortuga Restaurante	80.Toms Spezialitäten	79.The Cracker Box
84.Vaffeljernet	83.Trail's Head Gourmet Provisiones	82.Tradição Hipermercados
87.Wartian Herkku	86.Vins et alcools Chevalier	85.Victuailles en stock
90.Wilman Kala	89.White Clover Markets	88.Wellington Importadora
		91.Wolski Zajazd

right to left

9 Exports

This section describes principles of saving rendered reports to different formats, basic characteristics of methods for export, export optimization guidelines data structure which are used in export methods. Stimulsoft Reports supports great many export formats to save rendered reports. Many clients think that there are too many formats. But when you need to get file of definite format type, write only one string of code and the format is not PDF, HTML or RTF, only Stimulsoft Reports may help. We do not think that too many export formats in our report generator is disadvantage and continually work on adding new formats. The more exports the better, as they say.

9.1 Available File Formats

A list of supported file formats is represented in the table below. All exports are joined into groups.

Export Name
PDF (Portable Document Format)
XPS (XML Paper Specification)
HTML (HyperText Markup Language)
HTML5 (HyperText Markup Language)
MHTML (MIME HTML)
TXT (Text File)
RTF (Rich Text)
Microsoft Word 2007/2010
ODT (Open Document Text)
Microsoft Excel
Microsoft Excel Xml
Microsoft Excel 2007/2010
Microsoft Power Point 2007/2010
ODS (Open Document Spreadsheet)

CSV (Comma Separated Values)
DBF (DataBase File)
XML (eXtensible Markup Language)
DIF (Data Interchange Format)
SYLK (Symbolic Link)
BMP (Bitmap)
GIF (Graphics Interchange Format)
PNG (Portable Network Graphics)
TIFF (Tagged Image File Format)
JPEG (Joint Photographic Experts Group)
PCX (PCExchange)
WMF (Windows MetaFile)
SVG (Scalable Vector Graphics)

9.2   Common Export Settings

These chapters describe export settings which are not unique and are met in a few exports. Therefore, to prevent describing them again and again, they are joined in this section.

9.2.1   Image Quality

Image quality is the compression degree of JPEG. If the compression is low then an image is of good quality and has big file size. If the compression is high then an image is of bad quality and has small file size. In Stimulsoft Reports an image quality can vary from 0.0 (the lowest quality) to 1.0 (highest quality). If an image quality is 1.0 it does not mean that the image is saved without compression. The JPEG algorithm always compresses an image. The 1.0 quality means that an image quality will be the same as the quality of an original document but the file size will be smaller than the original. The 0.0 quality means that the image has slightest similarity to the original document. In practice, the 0.9 quality has not great distinction from the 1.0 quality but the image with lower than the 0.1 quality looks bad. By default, in Stimulsoft Reports the image quality

is 0.75.

9.2.2   Image Resolution

Raster images such as scanned photos consist of small cells called pixels. Image resolution depends on the pixel size and is measured in pixels per inch, ppi, and sometimes in dots per inch, dpi. The higher resolution the more pixels the image contains and, accordingly, the more size of the image. In Stimulsoft Reports it is possible to set any image resolution. But when increasing the resolution in 2 times the image size will increase in 4 times. Also it is not good to set the image resolution more than maximal resolution of an output device. For example, devices may have the following resolution:
- matrix printer - 72dpi
- monitor screen - 96dpi
- laser printers - 300dpi or 600dpi
- high-end printers - 1200dpi and higher.

By default the resolution is 100dpi.

9.2.3   Image Comparer

Sometimes repetitive image can be met in a report, for example, company logos on the header of each page. If do not process such duplicates then a report after export will have big size. Some formats allows exporting only one image and then refer to it from different parts of a document. In Stimulsoft Reports, there is a special class that calculates check sums and searches and processes duplicates. Image processing may slow down the process of exporting, so it is possible to disable this feature. Each export has its own property to enable or disable image comparison. By default this property is always enabled.

9.2.4   Convert Digits to Arabic

Arabs do not use Arabic digits. They use Hindi digits and Arabic digits are auxiliary (the same as Roman digits for us). But, in any case, all digits are written from left to right. This property indicates whether it is necessary to convert Roman digits (ASCII 0030h-0039h) to Arabic digits (Unicode 0660h-0669h or 06F0h-06F9h, depending on the ArabicDigitsType property). In each types of export the digits conversion can be set by their own property.

9.2.5   Arabic Digits Type

Arabic digits have two variants of drawing: Standard and Eastern. The property allows selecting the type of Arabic digits that will be used in export: Standard or Eastern; by default the Standard type is used. In each export the type of Arabic digits is enabled or disabled by its own property.

9.2.6   Divide Segment Pages

Stimulsoft Reports allows creating segmented pages. These are pages which horizontal and/or vertical size are increased in some times. Some applications, such as MS Excel, allows working with pages of any size, because breaking into small segments can is processed with the spreadsheet itself. Other applications, such as MS Word, cannot break pages into small segments. For such applications segmented pages are broken into separate pages on the stage of selected export; if property, for example, for Word2007, **StiOptions.Export.Word2007.DivideSegmentPages** , is set to **false** then pages are passed "as is" without breaking into segments. Each type of export has its own property for breaking segmented pages.

9.2.7   Remove Empty Space at Bottom

Many exports uses the table mode of export. In this mode data is converted into one table. If, in the initial report, there is an empty space on the bottom of a page then the table is broken. The following property allows removing empty space at the bottom of a page and resulting table is not broken. If it is necessary to save the initial view of a document then it is necessary to set this property to **false**. In each type of exports their own property is used.

9.2.8   Use One Page Header and Footer

When exporting to Excel then all report is converted in one table. Headers and footers of a page break this table. This property leaves only the first header and the last header of a page. All other headers and footers are removed. If it is necessary to save the initial view of a document then it is necessary to set this property to **false**. For each type of exports their own property is used.

9.3   Formats with Fixed Page Layout

Stimulsoft Reports supports three exports with fixed page layout. What is the fixed page layout? This means that all elements of a page can be placed at any part of a page. In this case, if to change a position of one element then other components position will not be changed. These are formats to **PDF** (Portable Document Format), **Microsoft Power Point 2007/2010** and **XPS** (XML Paper Specification).

9.3.1   PDF

PDF (Portable Document Format) – is a file format created by Adobe Systems for document exchange used to create electronic editions using the Adobe Acrobat package. The PDF format is a file text format that is used to publish documents on any platform and OS. The PDF document contains one or more pages. Each page may contain any components: text, graphic and illustrations, information, that provides navigation across the document.

> **Information**
>
> Export to PDF is based on the "Adobe Portable Document Format, Version 1.3, second edition", using some elements of latest format specifications.

PDF uses various compression methods to reduce the size of the file. To compress the text, the LZW ("Flate") algorithm is used. To compress pictures and omages, the JPEG or LZW algorithms are used. The JPEG algorithm is the compression with loss. It is recommended for full-color illustrations and images. The LZW algorithm is a lossless compression, it is recommended for illustrations and images with a small number of colors, for example graphics, chart, and schemes. To ensure the independence from the PDF fonts, the file contains a description for each font used in the document. The description includes the name, size, style and other font options. In the view document mode, if the font described in the document is available, then it is used. If the font is not available, it is replaced with a similar one with the same size and other characteristics. Fonts can be embedded in a document. This greatly increases the size of the file, but ensures that the document will be correctly displayed on any computer.

Export options in PDF

① The checkbox **All** enables processing of all report pages.

② The checkbox **Current Page** enables processing only the current (selected) report page.

③ The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

④ The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file.

⑤ The **Image Resolution Mode** parameter. Depending on the values of this parameter, a certain resolution will be applied to the images in the report:

> ❯ **Exactly** - all images after conversion will have the resolution specified in the Image resolution option;
>
> ❯ **No more than**. If the original resolution of the image is less than specified in the Image resolution parameter, then the resolution of the image after the conversion of the report will be equal to the original one. If the original resolution is greater than the one specified in the Image Resolution settings, the image resolution will correspond to the value of the Image Resolution settings.
>
> ❯ **Auto**. The image after the report is converted will have the original resolution.

6 The **Image Compression** Method allows defining the mode of image compression in the PDF file. The following modes are available:

> ❯ JPEG - compression with loss;
> ❯ Flate - compression without loss;
> ❯ Simple - monochrome mode without dithering;
> ❯ Ordered - monochrome image with dithering;
> ❯ FloydSt. - the most precise monochrome mode with dithering.

7 The option **Allow Editable** provides the ability to enable the mode in which, after exporting, it will be possible to modify components with the Editable property enabled. If No is set, then you can edit all components, unless it is not limited with safety parameters. If you select Yes then you can only edit components with the Editable property enabled.

Information

Please note that restrictions on editing a Word document do not use encryption algorithms strong to cracking. Therefore, for the security of the document it is recommended to use a digital signature and security group.

---

8 The **Image Quality** will be available only if you select the compression method JPEG. This option allows you to change the image quality. Keep in mind that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

9 The flag **Embedded Fonts** provides the ability to embed the font files into the created PDF file. If this option is enabled, then when you export a report, the files of all the fonts used in the report will be included in a PDF file, and fonts in the resulting file will be displayed correctly in any PDF viewer. If the property is disabled, then to display the file correctly all the fonts used in the report must be installed on the computer.

Information

When editing a text in the rendered report, the font may be different from the standard. Therefore, when the editing is performed by some other font, then this font will be embedded in the PDF file. This may lead to a significant increase of the size of the PDF.

10 The flag **Export Rich Text as Image** as Image enables/disables the conversion of the RTF text into the image. If the option is disabled, the Rich Text is decomposed into simpler primitives supported by the PDF format. The Rich Text with complex formatting (embedded images, tables) cannot always be converted correctly. In this case it is recommended to enable this option.

Information

When you enable this option, the file size may increase significantly.

11 The flag **PDF/A Compliance** enables/disables support for standard long-term archiving of electronic documents. Compliance ensures that the document will have the same look in later versions of Adobe Acrobat. Enabling this option will also automatically include the options Embed Fonts and use Unicode.
12 The Document Security.

9.3.1.1 Embedded Fonts

By default all embedded fonts are optimized. Characters which are not used in a report are excluded. It allows decreasing the size of a file. But, for correct work of the editable field, the font should be complete. Therefore, for fonts, which are used in editable fields, optimization is not done. This increases the output file size. If Asian languages are used, the file size can be 15-20mb.

9.3.1.1.1 Font Styles

There is one peculiarity of the export: all fonts for embedding to PDF files should be

installed in the system. And for each font style a single font file should be installed.

For example, for the Arial font 4 files should be installed:
> "arial.ttf" - the regular style,
> "ariali.ttf" - the italic style,
> "arialb.ttf" - the bold style,
> "arialbi.ttf" - the bold-italic style;

This font is embedded correctly and all styles of this font can be output.

The Lucida Console font is usually represented with one file that contains the regular style (other styles are generated by the system). Therefore, when embedding such a font to the PDF file, only **regular** style will be output, instead of all styles of this font.

9.3.1.2 Digital Signature

Digital signature is a requisite of an electronic document used to protect this document from falsification. This document is a result of cryptographic conversion of information using the **closed key** of the electronic signature and allows identifying the owner of the certificate of the key of the signature. Digital signatures are often used to implement electronic signatures.

9.3.1.2.1 Keys

Key is secret information is the secret information that is used by the cryptographic algorithm when creating and checking the digital signature. Usually for digital signature the pair of keys is used:
> Private key this key is known only for the owner;
> Public key this key is available for all users of cryptographic system.

In Digital Signature algorithms the signature is signed on the secret key of a user and is checked on the public key. So anyone may check what user put this signature. Keys are bound with specific certificates.

9.3.1.2.2 Public Key Certificate

Public key certificate EDS is a digital document confirming the correspondence between a public key and information identifying the owner of the key. It contains information about the owner of the key information about the public key, its purpose and scope, the

name of the certification authority and so on. Each certificate can be also connected with a private key. Storage of certificates is called a certificate authority. Certificate store often contains numerous certificates, possibly derived from different CAs. Certificates in the repository into folders (categories) that have their own hierarchy. To access any certificate must specify the name of the repository (category), in which it is located. To create a digital signature private key is required. Certificates that contain private keys that are usually located in the two repositories - the repository of the current user or local computer store. To select the storage you want to use the property **Use Local Machine Certificates**. By default it is set to **false**, and the search is made in the certificate store of the current user (CERT_SYSTEM_STORE_CURRENT_USER). If it is set to **true**, the search will be made in the local computer store (CERT_SYSTEM_STORE_LOCAL_MACHINE).

9.3.1.2.3 Choosing Certificate

There are two ways to create the digital signature:
> using the interface of the system library of cryptograph;
> directly by specifying the string - certificate identifier.

In the first case it is necessary to set the **Get Certificate From CryptoUI** property to **true**. When exporting, the menu for selecting certificate from the current storage of certificates will be displayed. It is necessary to select one certificate from the list of available ones.

❗ **Important:** In web applications this way cannot be used, because the menu of selecting a certificate is displayed on a computer on what the export is in process, in other words on the server. So the user cannot see and cannot do anything with it. In other words the export endlessly waits when the certificate will be selected.

In the second way, it is necessary to use the **SubjectNameString** property and write in it the string - certificate identifier. Identifier is the name of the certificate owner (full string) or a part of the name (substring).

9.3.1.2.4 Placing Digital Signature Identifier

By default the digital signature identifier is placed on the top of the first page of a document in the right corner, on margins. If it is required to set another position of the digital signature identifier, then it is necessary to place the text box with the description of the digital signature, and to set the **Tag** property to "**PdfDigitalSignature**".

9.3.1.3 Encryption

A PDF document can be encoded to protect the content from unauthorized access. A user may set the following parameters of encryption. The Document Security is a set of parameters with which you can protect documents from unauthorized access to them:

> In the field **User Password**, specify the password required to open the document. If you set the password, access to the opening file is limited and will occur only if you specify the correct password. If no password is specified, i.e. the field is left blank, then the file may be opened without restrictions.

> In the field **Owner Password**, specify the owner password to access the file. If you specify a password, access to the file operations, such as printing, copying etc will be available only after entering a password. If no password is specified, i.e., the field is left blank, the file operations will be available without restriction.

> The flag **Allow Print Document** enables/disables the restricted access to the printing operation. If this option is disabled, specifying the owner password is required to perform this operation. If enabled, then printing will be available for everyone who opens the document.

> The flag **Allow Modify Contents** enables/disables access to editing the text in the report. If this option is disabled, specifying the owner password is required to perform this operation. If enabled, then editing will be available for everyone who opens the document.

> The flag **Allow Copy Text and Graphics** enables/disables access to copying the information. If this option is disabled, specifying the owner password is required to perform this operation. If enabled, then copying will be available for everyone who opens the document.

> The flag **Allow Add or Modify Text Annotations** enables limited access to work with the annotations in the document. If this option is disabled, specifying the owner password is required to perform this operation. If enabled, then this operation will be available for everyone who opens the document.

> The flag **Encryption Key Length** allows selecting the length of the encryption key. The longer the length is, the more difficult is to decrypt the document, and, therefore, the safety of the document is higher.

9.3.1.3.1 Passwords and Access Permission

According to the PDF specification, it is possible to set the access and two passwords: the public password and the owner's password. If there are no passwords and everything is allowed to do with the document, then the document is not encrypted. If

even one password is set or access is not allowed, then the document is encrypted.

The public password allows opening and viewing documents, and also some actions are allowed:
➤ edit document;
➤ copy text and graphics from the document;
➤ add and change commentaries;
➤ print document.

The owner password provides access to the document, including password changing and access permission.

If the owner's password is set, and the public password is not set, then, when opening a document, the password is not requested.

9.3.1.3.2 Key Length

The PDF Reference defines both 40-bit, 128-bit and 256-bit encryption. By default 40-bit key is used.
256-bit and 128-bit keys is more secure the 40-bit key. But is some countries the key length of encryption is limited.

Quote from PDF Reference:
"A PDF document can be encrypted to protect its contents from unauthorized access. The encryption of data in a PDF file is based on the use of an encryption key computed by the security handler. Different security handlers can compute the key in a variety of ways, more or less cryptographically secure. In particular, PDF's standard encryption handler limits the key to 5 bytes (40 bits) in length, in accordance with U.S. cryptographic export requirements in effect at the time of initial publication of the PDF 1.3 specification."

9.3.1.4 Compatibility of Different Versions

The information below shows the compatibility of Adobe Acrobat versions.

Adobe Acrobat 5:
➤ the PageScaling option from the file is ignored. By default the option in parameters of Adobe Acrobat is set to "None" but "Fit to printable area" value is used.

Adobe Acrobat 5 & 6:

❯ when editing Adobe Acrobat does not recognize the Unicode - only Latin characters are output (Latin-1 encoding), other characters are output as dots;

❯ if the "**UseUnicode**" option in export parameters is enabled, then it is necessary to embed fonts (the "**Embedded Fonts**" option), otherwise the will be output incorrectly.

Adobe Acrobat 7:

❯ it is necessary to embed fonts to the PDF file. Otherwise, when editing, any font will be replaced on the default font (usually on Tahoma).

Adobe Acrobat 7 Reader:

❯ there are some problems with 7.0.5 - 7.0.9 versions. In these versions the field is not included into the editing mode, if there are non Latin characters present in the text field (different from Latin-1).

Adobe Acrobat 9

❯ Support for 256-bit encryption. In earlier versions, files with 256-bit encryption algorithm will not be opened.

Adobe Acrobat X

❯ Support for 256-bit encryption with improved internal calculations, and hence with a more crypto-stable algorithm.

9.3.2   XPS

XPS (XML Paper Specification) is the open graphic format of fixed page layout on the base XML (more precisely XAML-based) used to store printed output as electronic documents. This format was developed by Microsoft as alternative to the PDF format.The XPS document format consists of structured XML markup that defines the layout of a document and the visual appearance of each page, along with rendering rules for distributing, archiving, rendering, processing and printing the documents. The markup language for XPS is a subset of XAML that allows including vector graphic elements, using XAML to mark up the WPF-primitives. The XPS is a ZIP-archive that contains the files which make up the document. The archive includes page mark up (one file per each page of a document), text, embedded fonts, raster images, 2D vector graphics and other information.

Export options in XPS

1 The checkbox **All** enables processing of all report pages.

2 The checkbox **Current Page** enables processing only the current (selected) report page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file.

5 The **Image Quality** allows changing the image quality. Keep in mind that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

6 The flag **Export Rich Text as Image** as Image enables/disables the conversion of the RTF text into the image. If the option is disabled, the Rich Text is decomposed into simpler primitives supported by the PDF format. The Rich Text with complex formatting (embedded images, tables) cannot always be converted correctly. In this case it is recommended to enable this option.

! **Notice:** When you enable this option, the file size may increase significantly.

7 The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.3.3   Microsoft Power Point 2007/2010

Microsoft PowerPoint is a presentation program developed by Microsoft. It is a part of the Microsoft Office suite. PowerPoint presentations consist of a number of individual pages or "slides". Slides may contain text, graphics, movies, and other objects, which may be arranged on the slide. The presentation can be printed, displayed on a PC, or navigated through at the command of the presenter. In Stimulsoft Reports each report page corresponds to one slide.

Export options in Microsoft Power Point

1 The checkbox **All** enables processing of all report pages.

2 The checkbox **Current Page** enables processing only the current (selected) report page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is,

the greater is the size of the finished file.

5 The **Image Quality** allows changing the image quality. Keep in mind that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

6 The flag **Export Rich Text as Image** as Image enables/disables the conversion of the RTF text into the image. If the option is disabled, the Rich Text is decomposed into simpler primitives supported by the PDF format. The Rich Text with complex formatting (embedded images, tables) cannot always be converted correctly. In this case it is recommended to enable this option.

Notice: When you enable this option, the file size may increase significantly.

7 The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.4 Web Documents

There are two formats **HTML** (HyperText Markup Language), **HTML5** and **MHTML** (MIME HTML) are described in this chapter. The first and second formats are used for web page layout. The second format is a web page archive format used to bind resources together with the HTML code into a single file.

Export options in Web
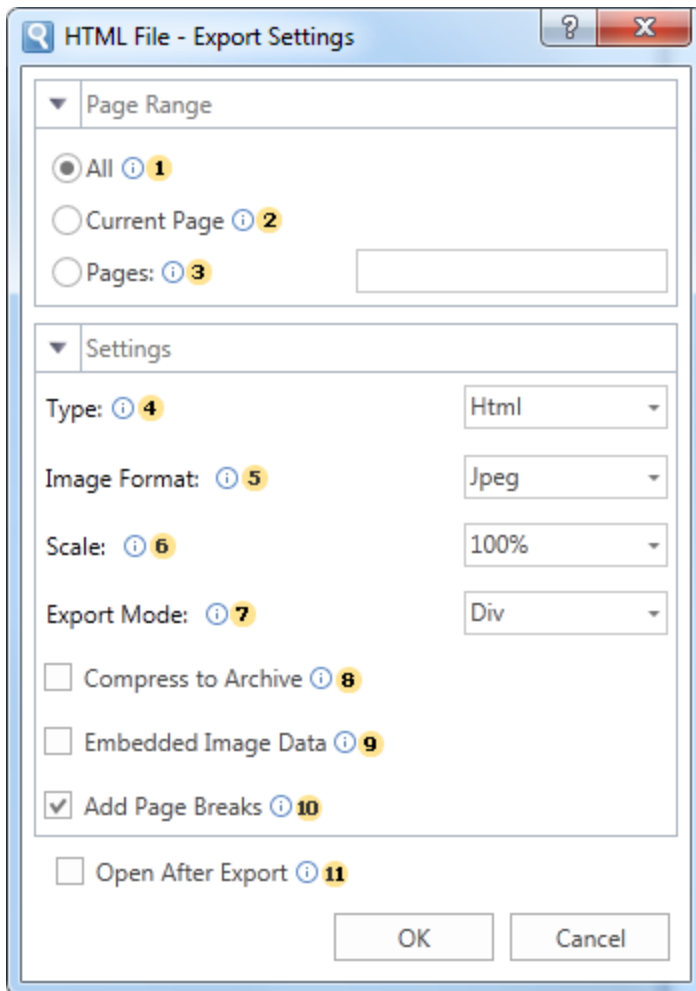
1 The checkbox **All** enables processing of all report pages.

2 The checkbox **Current Page** enables processing only the current (selected) report page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The option **Type** provides the ability to determine a type of the file the report will be converted into.

! **Notice**. If **Html5** is selected the following additional options are available:

> **Continuous Page**, which provides the ability to set the location of pages in the report as a vertical strip;

> The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file;

❯ The **Image Quality** allows changing the image quality. Keep in mind that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

5 With the **Image Format** it is possible to specify the format of images, which will be transformed into the image of the report.

6 The option **Scale** provides the ability to determine the size (scale) of report pages and items of the report after the export.

7 The option **Export Mode** provides the ability to determine the markup for the HTML page. The page layout is possible using tags div, span or table.

8 The flag **Compress to Archive** provides the ability, when exporting to HTML, to get the zip file after conversion. If this flag is on, the report processing occurs first, and then all the files and folders will be packed in a zip archive.

9 The flag **Embedded Image Data** provides the ability to embed images directly into the HTML file. In this case, it is necessary to consider that the correct displaying of this file depends on the browser being used. Not all browsers support the option to view the HTML file with embedded pictures.

10 The flag **Add Page Breaks** enables/disables the visual separator of report pages. If, for example, a few pages of the report are exported to a HTML page, it is not always possible to identify the beginning of the report page. To do this, you should select this option, then it will be, the beginning of the report page will be indicated by the appropriate delimiter.

11 The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.4.1 HTML

HTML (HyperText Markup Language) is the predominant markup language for Web pages. The majority of web pages are created using the HTML language. The HTML language is interpreted by browser and shown as a document. HTML is a tag language of the document layout. It provides a means to describe the structure of text-based information in a document by denoting certain text as links, headings, paragraphs, lists, etc. Elements are the basic structure for HTML markup. Elements have two basic properties: attributes and content. Each attribute and each element's content has certain restrictions that must be followed for a HTML document to be considered valid. An element usually has a start tag (e.g. <element-name>) and an end tag (e.g. </element-name>).

9.4.1.1 Export Modes

There are three mode of export to HTML:

❯ **Div** - in this mode all objects of a report are converted to the **div** block element; the

report is converted precisely, except for vertical text alignment;
> **Span** is the same as the Div mode but the span element is used;
> **Table** - in this mode all objects of a report are converted to the **table** block element; in this mode the vertical text alignment is correct but, if the WordWrap is disabled then the problem may occur with long lines of text.

9.4.1.2 Export Images in HTML Format

Also it is possible to specify how to export images of a document. Images with transparency can be saved to the PNG format. It is important to remember that some browsers (for example Internet Explorer 6) do not support images with transparency.

9.4.1.3 Compatibility of Different Versions

The following minimal web-browsers versions are required for correct HTML export:

> Internet Explorer 6.0 and higher;
> FireFox 1.5 and higher;
> Opera 7.5 and higher.

9.4.1.4 Exporting Text Components

When exporting reports to the **HTML** format, it is necessary to take the following features of this format into consideration:
> if a text does not fit a table cell horizontally, then a browser automatically carries a text to the next page;
> if a text does not fit a table cell vertically, then a browser automatically increases height of a table cell.

Such a behavior of a text can be obtained in the **Net** and **WPF** viewers (**Win**-viewers) by setting **WordWrap** and **CanGrow** properties of a text component to **true**. In the **HTML** format (and in the **Web** viewer correspondingly), no matter what is the value of these two properties, the text component will be shown the same way. For example, put 2 text components on a report template. Insert long text to the first component and a short one to the second. Set **WordWrap** and **CanGrow** properties to **false**. The picture below shows a report template:
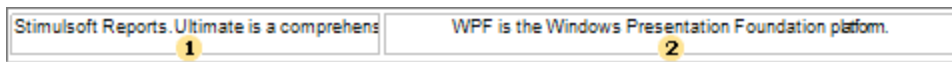
After rendering a report in the **Win**-viewer, a report will look like on a picture below:
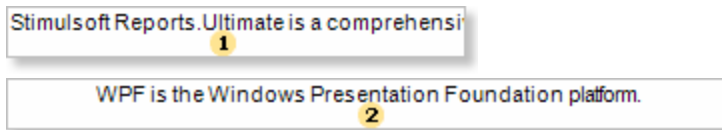
As seen on the picture, a text in the first text component did not fit and was cut, in the second text component the text fits a text component and shown without changes. Now set the **WordWrap** property to **true** for both components. After rendering, a report will look in the **Win** viewer like on the picture below:
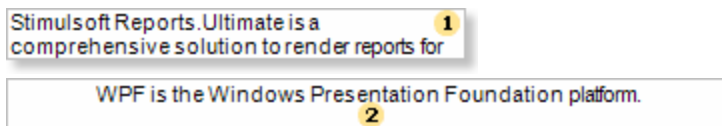
As seen on the picture, a text in the first text component is wrapped to the second row. But the component is not grown by height, so the text does not fit this component and was cut. In the second component the text fit this component and shown without changes. In both ways the text in the **HTML** format in the **Web** will look the following way:
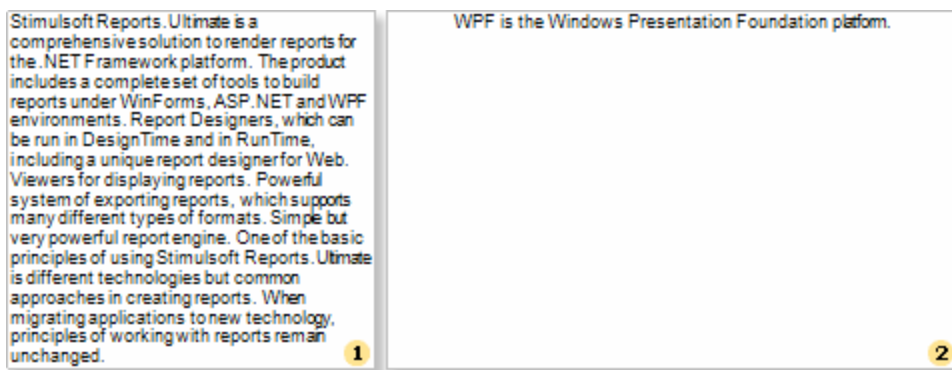
If to set the **Can Grow** properties of these texts components to **true**, then the report will look the same in the **Win** viewer and **Web** viewer:

Stimulsoft Reports.Ultimate is a comprehensive solution to render reports for the .NET Framework platform. The product includes a complete set of tools to build reports under WinForms, ASP.NET and WPF environments. Report Designers, which can be run in DesignTime and in RunTime, including a unique report designer for Web. Viewers for displaying reports. Powerful system of exporting reports, which supports many different types of formats. Simple but very powerful report engine. One of the basic principles of using Stimulsoft Reports.Ultimate is different technologies but common approaches in creating reports. When migrating applications to new technology, principles of working with reports remain unchanged. **1**

WPF is the Windows Presentation Foundation platform. **2**

9.4.2 HTML5

HTML5 is a language for structuring and presenting content for the World Wide Web, and is a core technology of the Internet originally proposed by Opera Software.[1] It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML4 as of 1997)[2] and as of December 2011 is still under development. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). HTML5 is intended to subsume not only HTML 4, but XHTML 1 and DOM Level 2 HTML as well.

9.4.3 MHT

MHTML (MIME HTML) is a web page archive format used to bind resources which are typically represented by external links (such as images, Flash animations, Java applets, audio files) together with HTML code into a single file. This file is a web archive and has the «**.mht**» extension. The content of a file is written as an Email message using the MIME standard: in the beginning of a file the HTML file is written. Then all resources in the base64 encoding with headers are written. Internet Explorer, Opera, Microsoft Word can work with the MHTML format.

9.5 Text Formats

This chapter describes exports formats of text files. In other words the files which are used to create text documents.

9.5.1   TXT

Text file (TXT) is a kind of computer file that is structured as a sequence of lines. A text file exists within a computer file system. The end of a text file is often denoted by placing one or more special characters, known as an end-of-file marker, after the last line in a text file.

Text files are commonly used for storage of information.

Export options in TXT:
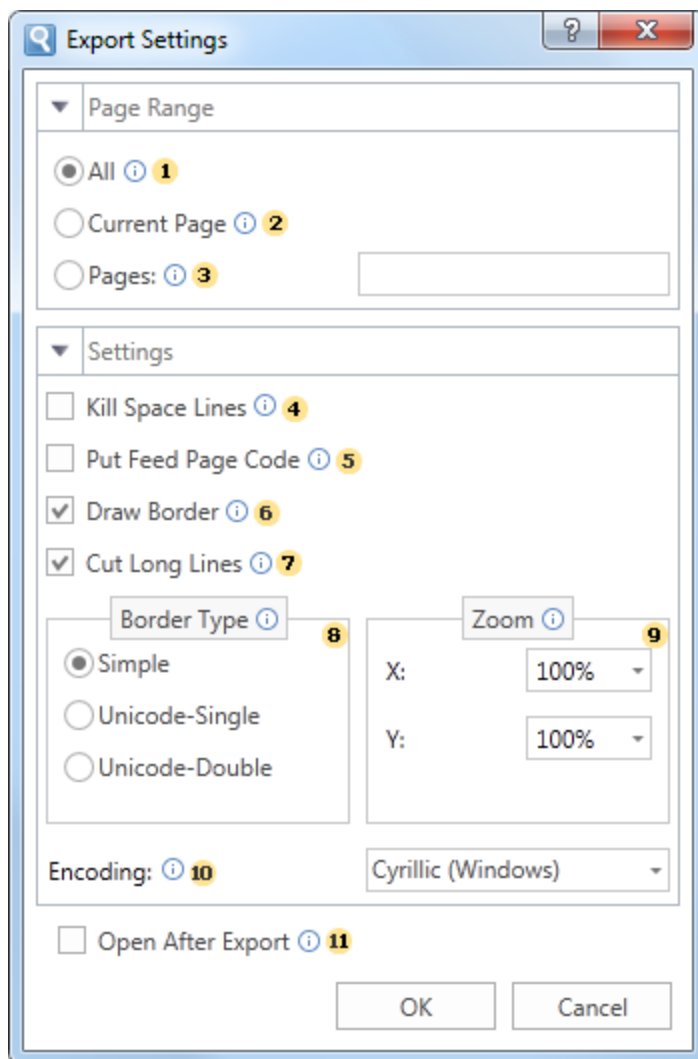
1 The checkbox **All** enables processing of all report pages.
2 The checkbox **Current Page** enables processing only the current (selected) report page.

③ The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

④ The checkbox **Kill Space Lines** provides the ability to delete blank lines in the document. If there are blank lines in a report, setting this flag will make the report more compact, but it should be taken into consideration that removing those lines can disrupt the formatting of other report elements.

⑤ The checkbox **Put Feed Page Code** provides an opportunity to select the end of the page with a special character.

⑥ The checkbox **Draw Border** enables/disables drawing borders of components with graphic symbols.

⑦ The checkbox **Cut Long Lines** provides the ability to cut lines by the margins of the component. If this option is enabled, the line length is limited to the margins of the component. If this option is disabled, the line will be displayed in its full length.

⑧ The option **Border Type** is used to enable a certain type of borders of components. The options are:

➤ **Simple** - drawing the borders of components with characters +, -, |.

➤ **Unicode-Single** - drawing the borders of components with box-drawing characters.

➤ **Unicode-Double** - drawing the borders of components with double box-drawing characters.

⑨ The option **Zoom** provides the ability to set the report zoom horizontally and vertically.

⑩ The option **Encoding** provides the ability to set the text encoding of the report after exporting.

⑪ The checkbox **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.5.1.1 Border Types

The border in the text mode can be drawn using simple symbols or using pseudographics. Using the **BorderType** property it is possible to choose the mode of border drawing. It may have the following modes:

➤ Simple - drawing a border using simple symbols such as "+", "-", and "|";

➤ UnicodeSingle - drawing a border using the symbols of pseudographics; symbols of solid border are used;

❯ UnicodeDouble - drawing a border using the symbols of pseudographics; symbols of double border are used.

9.5.1.2 Column Width

When exporting to the text format, all coordinates and sizes of objects are recalculated to get the text appearance the same as it is in a report. You can control the conversion, by changing the zoom coefficients of ZoomX and ZoomY. The width of the columns of the output text is proportional to the width of the initial report. If you want to change the column width, it is possible to use the following methods:

❯ change the width of a column: it is necessary to specify the column width in characters in the **Tag** text box, the width will be set only for those lines which contain this text box;

❯ column width can be set globally via the **ColumnWidths** static property; in this case, the width of the columns is indicated starting from the left column, through the separator (a semicolon), for example, "10, 12, 45, 10, 10, 5, 20, 50 "; zero width of columns is ignored.

9.5.1.3 New Export Mode

The old/new export mode is set using the **UseOldExportMode** property. The new mode is created on the base of the StiMatrix: if the Word Wrap is enabled and a text cannot be placed in a cell then the cell height is increased automatically. By default the new mode is enabled.

9.5.2 RTF

Rich Text Format (RTF) is a free document file format developed by Microsoft for cross-platform document interchange. The first version of the RTF standard appeared in 1987. Since that time format specification was changed and added. RTF-documents are supported by many text editors.

Export options in RTF:

1 The checkbox **All** enables processing of all report pages.

2 The checkbox **Current Page** enables processing only the current (selected) report page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file.

5 The **Image Quality** allows changing the image quality. Keep in mind that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

6 The checkbox **Export Mode** provides the ability to define the presentation of the report data after export. If you select **Table**, then, after exporting, the entire report will look like a table, where each report component is a table cell. All components are located in different cells with relations created between them. If the **Frame** is selected, then, after export, each component will be a single frame, but without relations between them.

⁊ The checkbox **Use Page Headers and Footers** is used to define the Page Header and Footer as the header and footer of the Word document. If this option is not set, then, after exporting, page header and footer will be a table cell or an individual frame. In case of editing a report they may change its location. If this option is enabled, the data bands will be output as objects a header and footer in the Word document.

❗ **Notice:** If the checkbox **Use Page Headers and Footers** is on, it should be taken into consideration that, in this case, the height of the lines will be minimum allowable.

⁸ The checkbox **Remove Empty Space at Bottom of the Page** is used to display data one after the other while minimizing empty space at the bottom of the page. If this option is enabled, then, if empty space is available, the part of data from the next page will be moved to the empty space. If this option is disabled, the empty space is ignored and the report will be displayed in the viewer or in the tab Preview.

⁹ The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.5.2.1 Export Modes

The export to the **RTF** format has 2 basic modes:

❯ **Frames** - in this mode all objects of a report are converted to the **frame** rtf-objects; the report is converted precisely, but it is difficult to edit such a document.

❯ **Table** - in this mode all objects of a report are converted to the single table; when converting, objects can be changed, but this document can be easily edited, and, therefore, this mode is more frequently used and this mode is the default mode of this export.

Also there are 2 modes of operation, which are obsolete and retained for compatibility:

❯ **WinWord** - similar to Frames, but all the objects passed as "frame" objects of MS-Word.

❯ **TabbedText** - the same as the export the Text format, the position of a text is set using blank spaces and tabulations.

9.5.2.1.1 Table Mode

In this mode the whole report is converted into a single table. When exporting the report is converted into a single table. The document is easily editable but some objects can be changed.

Depending on the value of the **Use Page Headers and Footers** property the report is exported as follow:

❯ value is set to **false** - the report is exported "as is" and will look the same as in preview;

❯ value is set to **true** - the report is additionally processed, all changes are described in the text below.

The list of document changes:

❯ PageHeaders and PageFooters are exported as MS-Word objects. So they are cut from the table and other bands are converted into a single page. It is very convenient because it is easy to correct the document, for example, to put or edit text in cells, change the cell size; all data are moved, and headers and footers of a page stay on their place. (**Notice**: the header and the footer are exported from the first page of a report, others are ignored; in addition the improvement was done: now the header is searched on the second page; if the property PrintOn of this header is set to ExceptFirstPage, then everything is exported correctly (using the RTF tags) - the header will not be output on the first page.

❯ If the Header of the PrintOnAllPages property is enabled, then it is exported as the table header, and is correctly output on each page.

❯ The height or rows in not exported (the "not set" mode; by default the "precise" mode is set).

❯ If the **Tag** field is not empty, then the content of the **Tag** field is exported. The **Text** field is not exported. The following expression can be used to change MS-Word commands:

Name	Description
#PageNumber#	The number of the current page (PAGE)
#TotalPageCount#	Total number of pages in the document (NUMPAGES)
#PageRef Bookmark#	The number of pages on what the bookmark is placed (PAGEREF)

For example, the following expression can be written in the **Tag** field:

Page #PageNumber# of #TotalPageCount#

When exporting, #PageNumber# and #TotalPageCount# will be substituted on the "Page number" field and "Total Page" field. And they will be automatically changed.

The following string-commands can be written in the **Tag** field:

Name	Description
rtfparagraph	The TextBox, RichTextBox and Image content is output as simple text, in the table break. It is supposed that this is the only component in the row of text, so other components in this row are ignored.
rtfnewpage	The page break is put before the text box

Also it is possible to export a separate sheets of a template to separate sections of the document with the headers/footers. To do this, use the **ExcelSheet** property. in this case all pages of a report with the same value of the ExcelSheet property are combined in groups, then each group is exported as a separate section of the document with its headers/footers. By default, this property is not filled, and the report is exported as a single partition.

9.5.2.2 Issues

MS-Word: if to set top and bottom margin of one of cells in the table row, the same margin will be set in all cells of the row. Therefore, if to set the top and bottom margins of the text box, then, after exporting, the same margin will be set for the row of the table and the text will be moved. In OpenOffice this works without problems.

9.5.2.3 **Compatibility of Different Versions**

The RTF format is based on the RTF specification version 1.4 from 9/1995. The RTF files can be opened in Microsoft Word starting with the 97 (Office 97) version. In Microsoft Word 95 the RTF will have the following limitation:

> does not support vertical alignment in cells;
> does not some parameters of a page;
> some colors and not shown correctly.

9.5.3 Word 2007/2010

Microsoft Word is a text processing software produces by Microsoft. It is a component of the Microsoft Office system. The first version was released for IBM PC's running DOS in 1983. Later there was a release for Apple Macintosh (1984), SCO UNIX, and Microsoft Windows (1989). Microsoft Word is the most popular text processors. Starting with first versions MS Word could write files in binary code with the «.doc» extension. The Word specification was secret and only in 2008 was published. The latest version of **Word 2007/2010** "uses by default" the XML based format: Microsoft Office Open XML. For a new format the «**.docx**» file extension is used. This format is a zip-archive that contains a text as XML, graphics, and other data. When exporting, a report is converted into one table. Such a document is easy to edit.

Export options in *Word*

❶ The checkbox **All** enables processing of all report pages.

❷ The checkbox **Current Page** enables processing only the current (selected) report page.

❸ The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

❹ The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file.

❺ The **Image Quality** allows changing the image quality. Remember that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

❻ The parameter **Restrict Editing** provides the ability to restrict editing the Word document. The available modes are: **No** – without editing; **Yes**- editing is not allowed; **Except Editable Fields** - editing is allowed only for editable fields in the report. In this case, the Editable property of components must be set to true.

❼ The checkbox **Use Page Headers and Footers** is used to define the Page Header and Footer as the header and footer of the Word document. If this option is not set, then, after exporting, page header and footer will be a table cell or an individual frame. In case of editing a report they may change its location. If this option is enabled, the data bands will be output as objects a header and

footer in the Word document.

❗ **Notice:** If the checkbox **Use Page Headers and Footers** is on, it should be taken into consideration that, in this case, the height of the lines will be minimum allowable.

8 The checkbox **Remove Empty Space at Bottom of the Page** is used to display data one after the other while minimizing empty space at the bottom of the page. If this option is enabled, then, if empty space is available, the part of data from the next page will be moved to the empty space. If this option is disabled, the empty space is ignored and the report will be displayed in the viewer or in the tab Preview.

9 The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.5.3.1   Headers and Footers

Depending on the value of the **Use Page Headers and Footers** property a report is exported in the following way:
- the value is false - a report is exported "as is" and looks as in preview;
- the value is true - a report is additionally processed. All changes are described below.

The list of changes of the document:
- PageHeaders and PageFooters are exported as MS-Word objects. So they are cut from a table and all other bands are exported as one table. It is very convenient, if it is necessary to elaborate the document (add rows or edit a text in cells, change cell size); in this case all data are moved but headers and footers stay on their place. (Notice: a header and a footer of the first page are taken, others are ignored).
- Row height is not exported (the "not set" mode; by default - the "precise" mode).

9.5.3.1.1 Page Numbering

If the Tag is not empty then the content of the Tag property is exported. The Text field is not exported. Also the string may contain the following expressions, which are changed on MS-Word commands:

#PageNumber#	The number of the current page (PAGE)
#TotalPageCount#	Total number of pages in a document (NUMPAGES)

For example, in the Tag property the following expression can be written:

Page #PageNumber# of #TotalPageCount#

When exporting #PageNumber# and #TotalPageCount# will be replaced on "PageNumber" field and "TotalPageCount" field and will be automatically changed together with text.

9.5.4   ODT

Open Document Text (**ODT**) is the open document for storing documents of the OpenOffice Writer, which is included into the OpenOffice.org package. OpenOffice.org is the open package of office applications created as alternative to Microsoft Office. OpenOffice.org was one of the first what supported the new open OpenDocument. Works on Microsoft Windows and UNIX systems: GNU/Linux, Mac OS X, FreeBSD, Solaris, Irix. OpenDocument Format (ODF) is the open file format for storing office documents, including text documents, spreadsheets, images, data bases, presentations. This format is based on the XML format.

OpenOffice Writer is the text processor and visual HTML editor, included into the OpenOffice. It is open software (LGPL license). Writer is similar to Microsoft Word and has approximately the same functionality. Writer allows saving documents in different formats including Microsoft Word, RTF, XHTML, and OASIS Open Document Format. Starting with the OpenOffice version 2.0, the OpenDocument Format is the default format for saving documents. File have the «.odt» extension. When exporting the report is converted into a single table. The document is easily editable but some objects can be changed.

Export options in ODT

1 The checkbox **All** enables processing of all report pages.

2 The checkbox **Current Page** enables processing only the current (selected) report page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file.

5 The **Image Quality** allows changing the image quality. Keep in mind that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

6 The checkbox **Use Page Headers and Footers** is used to define the Page Header and Footer as the header and footer of the Word document. If this option is not set, then, after exporting, page header and footer will be a table cell or an individual frame. In case of editing a report they may change its location. If this option is enabled, the data bands will be output as objects a header and footer in the Word document.

! **Notice:** If the checkbox **Use Page Headers and Footers** is on, it should be taken into consideration that, in this case, the height of the lines will be minimum allowable.

7 The checkbox **Remove Empty Space at Bottom of the Page** is used to display data one after the other while minimizing empty space at the bottom of the page. If this option is enabled, then, if empty space is available, the part of data from the next page will be moved to the empty space. If this option is disabled, the empty space is ignored and the report will be displayed in the viewer or in the tab Preview.

8 The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.6 Spreadsheets

This group of exports create spreadsheets. They are exports to both different formats of Microsoft Excel and to OpenOffice Calc.

Export options in Excel

1 The checkbox **All** enables processing of all report pages.

2 The checkbox **Current Page** enables processing only the current (selected) report page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The option **Type** provides the ability to determine a type of the file the report will be converted into.

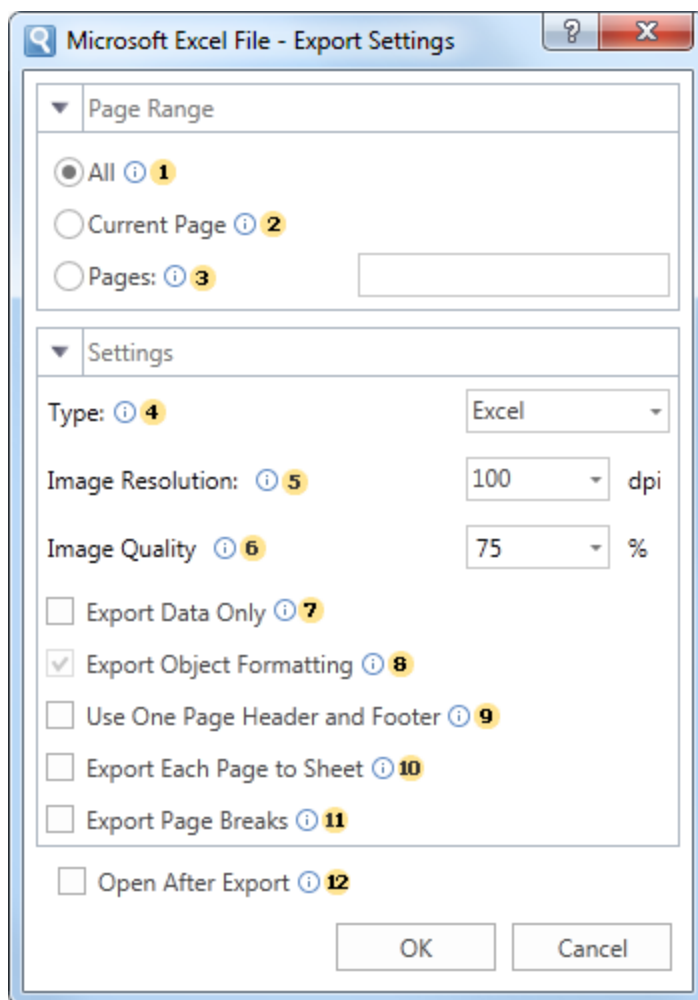**5** The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file.

6 The **Image Quality** allows changing the image quality. Keep in mind that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

7 The checkbox **Export Data Only** enables/disables the export of data only. If this option is enabled, information from the Data bands (component table, Hierarchical band) will be exported. Only these bands are processed. The rest bands and components are ignored. If this option is disabled, the entire report will be exported.

8 The checkbox **Export Object Formatting** is available only when you export the data. It provides the opportunity to apply formatting to them. If this option is enabled, the data will be exported with formatting applied in the report. If this option is disabled, the data formatting will be lost.

9 The checkbox **Use One Page Header and Footer** is used to get rid of repeats of headers and footers on the report pages. By default the page header and footer in the report are located on each page. The report in export to Excel is printed on a sizeless page. The page is able to grow in height as long as there are data. In this case, when you view the document in Excel, page headers and footers are output on the top and bottom of each report page. For example, if the report consists of 15 pages (in the Excel document it will all be placed on a single sheet), the page header and footer page will be output 15 times (each time on the top and bottom of the report page). To avoid this, you should enable this option, and then the page header will be displayed only on the top of the Excel sheet, and the page footer - in the end.

! **Notice:** Enabling this option may have residual effects. For example, if the page header or footer has borders, then, when this option is enabled, these borders may be shown. It is recommended, before rendering the report, to enable the parameter of the report page, Unlimited Height. In this case, the report will be rendered on a sizeless single page. The page header and footer will be printed only once on the Excel sheet.

10 The checkbox **Export Each Page to Sheet** is used to export each report page on a separate Excel sheet. If this option is enabled, then each report page will be located on a separate sheet in Excel. If this option is disabled, the entire report will be printed on a single sheet of Excel.

11 The checkbox **Export Page Breaks** is used to display the borders of the report pages on the Excel sheet. In other words, if the report contains 10 pages, all of them are placed on one sheet after export. Enable this option to define the borders of pages. If this option is disabled, all report pages will be printed, and, if no other delimiters present, will be output in one sizeless page.

The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.6.1 Excel

Microsoft Excel is a spreadsheet application written and distributed by Microsoft for Microsoft Windows. It allows using calculation, graphing tools, pivot tables and a macro programming language called VBA. So, it is the most popular table processor available for these platforms since version 5 in 1993.

Microsoft Excel up until Excel 2007 version used a proprietary binary file format called Binary Interchange File Format (BIFF) and **.xls** file extension. Specification was closed but since 2008 it was published. Besides, most of Microsoft Excel can read CSV, DBF, SYLK, DIF, and other formats.

9.6.1.1 Excel Sheets

By default a report is exported as one table to one sheet of Excel. Maximal number of rows on a sheet is unlimited. It depends on the Excel version and is set using the **MaximumSheetHeight** static property (by default 65534, for Excel XP and Excel 2003). If the number of rows is more than default then odd rows will be carried on the next sheet.

Also it is possible to export each page of a report on a single sheet of Excel. To do this it is possible to set the **ExportEachPageToSheet** property to **true**.

Besides the forced Excel sheets creation they can be created using the **ExcelSheet** page property to what any value can be assigned. If some sheets has the same **ExcelSheet** value then they are joined and exported as one sheet. In this case the name of a sheet is a name of a value.

9.6.1.2 Compatibility of Different Versions

The **XLS** format is based on the BIFF8 specification. Full support of this format is realized starting with the Excel 9.0 (Excel 2000).

Excel 8.0 (Excel 97):
> does not support correct color;
> does not fully support the **Right to Left** mode.

Excel 7.0 (Excel 95) and earlier versions:
> does not support vertical alignment in a cell;
> does not support integrated cells;
> does not support some other parameters.

9.6.2   Excel XML

For storing documents as the basic Microsoft Excel format, right up to the Excel 2007 version, used its own binary format of files (BIFF) and the file extension was «.xls». In **Excel 2003** additionally, a new format based on XML (XMLSS) was used. This opened format is convenient for developers and is data oriented. The basic disadvantage of the format is impossibility to embed raster images.

9.6.3   Excel 2007/2010

For storing documents as the basic Microsoft Excel format, right up to the Excel 2007 version, used its own binary format of files (BIFF) and the file extension was «.xls». In **Excel 2007/2010**, the basic format is the Microsoft Office Open XML format and stores document in files with the «**.xlsx**» extension. The Excel 2007 is compatible with binary formats such as CSV, DBF, SYLK, DIF, and others.

9.6.3.1   Sheets in Excel

By default a report is exported as one table to one Excel sheet. Maximal number of rows on a sheet in limited. It depend on the version of Excel and is set using the **MaximumSheetHeight** static property (by default 1048574 for Excel 2007). If rows are too many then redundant rows will be output on the next sheet. Also it is possible to export each page of a report to the single sheet Excel. To do this, it is necessary to set the **ExportEachPageToSheet** property to **true**.

Each page of a report has the **ExcelSheet** report property to what any expression may be assigned. Numbers of pages with the same value in the "ExcelSheet" are combined and exported to a single sheet of Excel. The name of the sheet becomes the value of the expression.

9.6.4 ODS

Open Document Spreadsheet (**ODS**) is the opened format to store OpenOffice Calc spreadsheet documents, that is included into the OpenOffice.org package. OpenOffice.org is a free package of office applications developed as alternative to Microsoft Office. The OpenDocument is one of the first what started to support the opened format. it works on Microsoft Windows and UNIX-like systems: GNU/Linux, Mac OS X, FreeBSD, Solaris, Irix. OpenDocument Format (ODF) — an open document file format for storing and exchanging editable documents including text documents (such as notes, reports, and books), spreadsheets, drawings, databases, presentations. The format is based on the XML-format. The standard was jointly developed by public and various organizations and is available to all and can be used without restrictions. OpenOffice Calc is the table processor that is included into the OpenOffice and is a free software (LGPL license). Calc is similar to the Microsoft Excel spreadsheet and functionality of these processors is approximately equal. Calc allows you to saving documents to various formats, including Microsoft Excel, CSV, HTML, SXC, DBF, DIF, UOF, SLK, SDC. Starting with version OpenOffice 2.0, for document storage format by default OpenDocument Format, files are saved with the extension «. Ods». Starting with the OpenOffice version 2.0 for storing documents, by default, the OpenDocument Format is used. Files are stored with the «.ods» extension.

❶ The checkbox **All** enables processing of all report pages.
❷ The checkbox **Current Page** enables processing only the current (selected) report

page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The **Image Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file.

5 The **Image Quality** allows changing the image quality. Keep in mind that if you change this option the size of the finished file will increase. The higher the quality is, the larger is the size of the finished file.

6 The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.7   Data

This is a group of file formats which are used to store table data.

Export options in Data
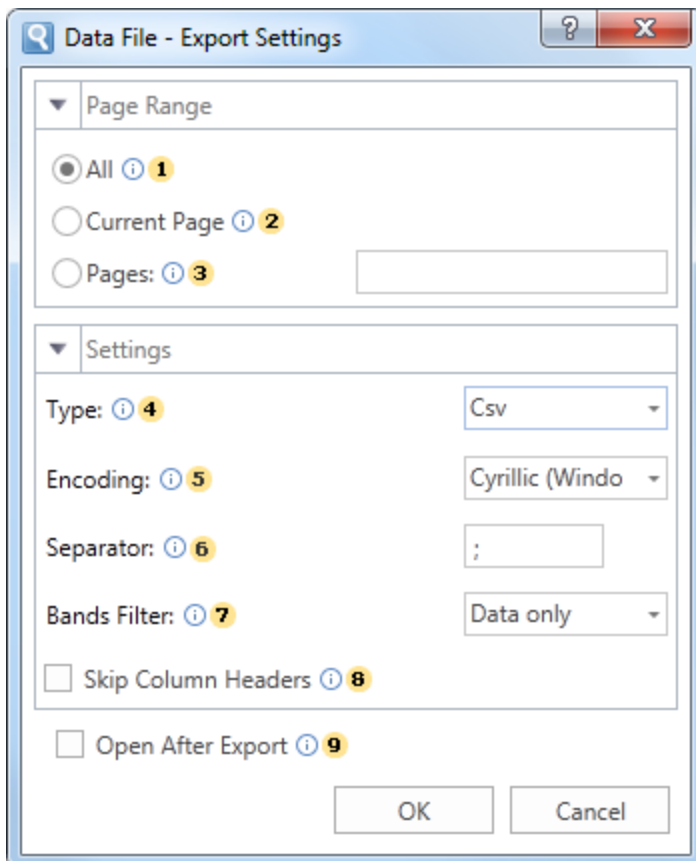
1 The checkbox **All** enables processing of all report pages.

2 The checkbox **Current Page** enables processing only the current (selected) report page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The parameter **Type** provides the ability to determine a type of the file the report will be converted into.

❗ **Notice:** Depending on the file type, parameters, and their number may vary. For example, when you select a format DIF or Sylk, the following options will be available:

❯ The option **Only Data Only** enables/disables the mode of exporting data only. If this option is enabled, information will be exported from the Data bands (the component table, Hierarchical band). Only these bands are processed, the rest are ignored. If this option is disabled, the entire report will be exported;

❯ The option **Use Default System Encoding** allows you to use the system encoding by default. Different encoding can be applied depending on the installed system. If this option is disabled, you must set the encoding by the standard.

5 The parameter **Encoding** is used to define file encoding.

6 The parameter **Separator** specifies delimiter between the data in the CSV file.

7 The parameter **Bands Filter** is used to apply a filtering condition in the export. The following options are available:

❯ **Data Only** - in this case only Data bands will be processed (the Table component, Hierarchical band);

❯ **Data and Headers/Footers** - Data bands will be processed (the Table component, Hierarchical band), and their headers/footers, if any;

❯ **All Bands** - all bands of the report will be processed.

8 The checkbox **Skip Column Headers** enables/disables the column headers. If the option is enabled, then column headers will not be displayed. If this option is disabled, then column headers (if present in the report) will be displayed.

9 The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.7.1   CSV

CSV (Comma Separated Values) is a text format that is used to represent table data. Each string of the file is one row of the table. The values of each column are separated by the delimiter that depends on regional settings. The values that contain reserved characters (such as a comma or a new string) are framed with the double quotes (") symbol; if double quotes are found in the value they are represented as two double

quotes in the file.

❗ **Notice:** Only those data (components) can be exported to the CSV format which are placed on data bands. If the SkipColumnHeaders property is set to false then, additionally, column headers are exported as the first row.

9.7.1.1   Controlling Exports

The Tag property of each textbox in a Data band can be specified with the following elements that control the export:
- Export Type : "FieldName"
- Column: "FieldName" "DataRow"

Several elements should be separated with the semicolon.

The "Export Type" element indicates for which export the field name is set. The values can be used: "dbf", "csv", "xml", "default". The "FieldName" element indicates the field name in the file. The own name can be specified to each type of export. If the name for each export is not specified then the name for the "default" type is taken. For example:

 DBF : "Describe" ; CSV : "Description" ; default: "Default name"

The "Column" element indicates that additional field is added to exported data. The "FieldName" element indicated the name of a new field. The "DataRow" element indicates the content of a new field and can be an expression. For example:

 Column: "SortField" "{Products.Categories.CategoryName}"

9.7.2   DBF

DBF (DataBase File) is the format to store data and it is used as the standard way to store and pass information. The DBF file consist of a header section for describing the structure of the data in the file. There are several variations on the .dbf file structure.

❗ **Notice:** Only data can be exported to the DBF format, in other words only the components, which are placed on data bands.

9.7.2.1 **Controlling Exports**

The following elements can be specified in the Tag property to control export:
> DataType [: FieldLength [: DecimalPartLength]]
> ExportType : "FieldName"
> Column: "FieldName" "DataString"
Several elements should be separated with the semicolon. The "DataType" element should be only one and should be placed first, other elements – if necessary.

Values of the "DataType" element are shown in the table below. If the data type is not set, then the **string** data type is taken by default. The "FieldLength" element sets fixed width of a data field. If the field width is not set, then the width is taken from the table. For the **string** type the default width is the longest string. The "DecimalPartLength" element sets the number of characters after comma. If it is not set, then the default number is taken.

Data type	DBF data type (default size)	Description
int	Numeric (15 : 0)	Numeric
long	Numeric (25 : 0)	Numeric
float	Numeric (15 : 5)	Decimal
double	Numeric (20 : 10)	Decimal
string	Character (auto)	Text
date	Date (8)	Date

Sample of using elements are shown in the table below.

Type	Description
string : 25	set the column width (25 characters) and cuts all long strings
float	converts decimal digit with the length 15 characters, 5 characters after comma
float :10	converts decimal digit with the length 10 characters, 5 characters after comma

float :10 : 2	converts decimal digit with the length 10 characters, 2 characters after comma
int :10 : 2	converts integer digit with the length 10 characters; the second parameter is ignored

❗ **Notice:** If the integer part of a digit is long and cannot be placed into the specified field, then it is cut, so data are lost. For example, if the write «-12345,678» in the «float:8:3» field, then the «2345,678» will be output.

The "ExportType" element indicates for which export the field name is set. The values can be used: "dbf", "csv", "xml", "default". The "FieldName" element indicates the field name in the file (for the DBF the is automatically cut up to 10 characters). The own name can be specified to each type of export. If the name for each export is not specified then the name for the "default" type is taken. For example:

DBF : "Describe" ; XML : "Description" ; default: "Default name"

The "Column" element indicates that the additional field is added to the exported data. The "FieldName" element indicates the name of a new field. The "DataRow" element indicates the content of a new field and can be expression. For example

Column: "SortField" "{Products.Categories.CategoryName}"

9.7.3   XML

XML (eXtensible Markup Language) is a text format that is used to store structured data (in exchange for existed files of data bases), for exchange of information between programs and also to create on its base the special markup languages (for example, XHTML), sometimes called dictionaries. XML is the hierarchical structure that is used to store any data. Visually this structure can be represented as the tree. XML supports Unicode and other encoding.

❗ **Notice:** Only those data (components) are exported to the XML format which are placed on data bands.

9.7.3.1 Controlling Exports

The following elements can be specified in the Tag property to control export to XML:
> DataType
> ExportType : "FieldName"
> Column: "FieldName" "DataRow"

Several elements should be separated with the semicolon. The "DataType" element should be only one and should be placed first, other elements – if necessary.

Values of the "DataType" element are shown in the table below. If the data type is not set, then the **string** data type is taken by default.

Data type	Description
int	Numeric
long	Numeric
float	Decimal
double	Decimal
string	Text
date	Date

The "ExportType" element indicates for which export the field name is set. The values can be used: "dbf", "csv", "xml", "default". The "FieldName" element indicates the field name in the file. The own name can be specified to each type of export. If the name for each export is not specified then the name for the "default" type is taken. For example:

DBF : "Describe" ; XML : "Description" ; default: "Default name"

The "Column" element indicates that additional field is added to the exported data. The "FieldName" element indicates the name of a new field. The "DataRow" element indicates the content of a new field and can be expression. For example:

 Column: "SortField" "{Products.Categories.CategoryName}"

9.7.4 DIF

DIF (Data Interchange Format) is a text format that is used to exchange sheets between spreadsheets processors (Microsoft Excel, OpenOffice.org Calc, Gnumeric, StarCalc, Lotus 1-2-3, FileMaker, dBase, Framework, Multiplan, etc). The only limitation of this format is that the DIF format may contain only one sheet in one book.

9.7.5 SYLK

SYLK (Symbolic Link) format- this text format is used to exchange data between applications, specifically spreadsheets. Files of SYLK have «**.slk**» extension. Microsoft does not publish a SYLK specification, therefore work with this format in different applications can be different.

❗ **Notice:** A SYLK file can be written in Unicode and read by some applications but anyway many applications which do support Unicode writes SYLK files into ANSI but not Unicode. Therefore, symbols which do not have representation in the system code page will be written as ('?') symbols.

9.8   Images

Export groups to graphic formats. All graphic formats can be divided in to types: bitmapped images and vector formats. Notice. On the current moment the export of monochrome image is supported only to **BMP**, **GIF**, **PCX**, **PNG**, **TIFF** format. So the **DitheringType** property works only for these exports.

Export options in Image
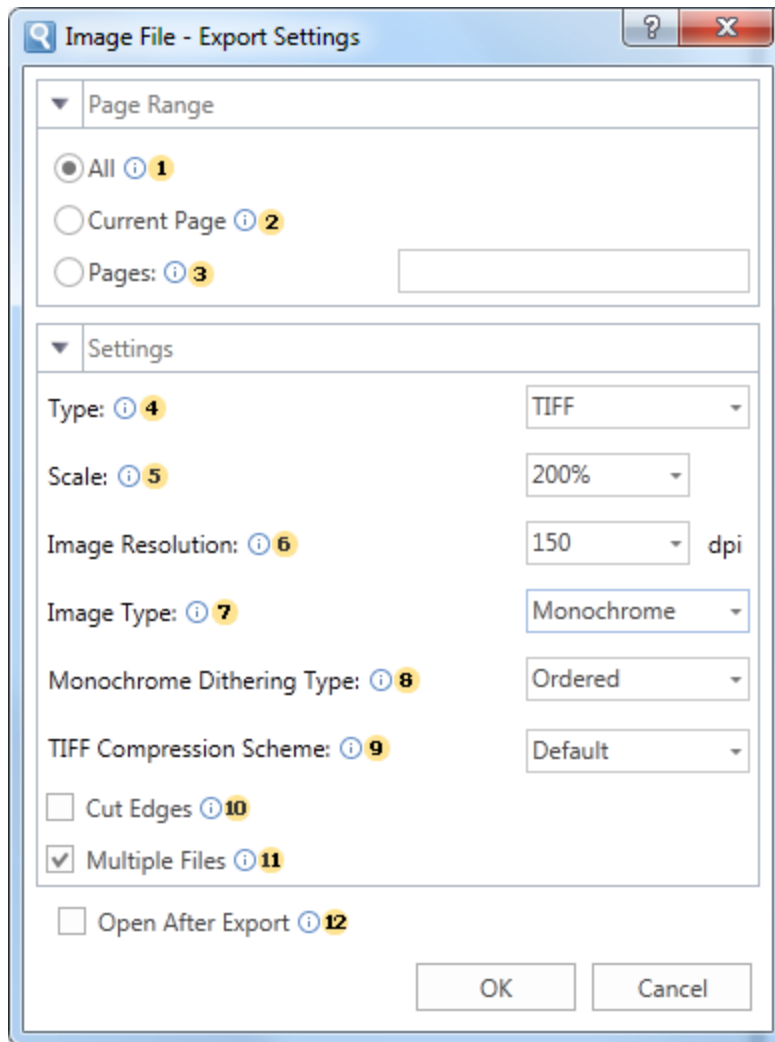
1 The checkbox **All** enables processing of all report pages.

2 The checkbox **Current Page** enables processing only the current (selected) report page.

3 The checkbox **Pages** has the field. This field specifies the number of pages to be processed. You can specify a single page, several pages (using a comma as the separator) and also specify a range by defining the start page and end page range separated with "-". For example, 1,3,5-12.

4 The option **Type** provides the ability to determine a type of the file the report will be converted into.

5 The option **Scale** allows you to increase/decrease the size of the report after export. It should also be taken into consideration that the smaller the scale is selected, the greater is the number of pixels per inch, and vice versa.

6 The Image **Resolution** is used to change DPI (image property PPI (Pixels Per Inch)). The greater the number of pixels per inch is, the greater is the quality of the image. It

should be noted that the value of this parameter affects the size of the finished file. The higher the value is, the greater is the size of the finished file.

7 The option **Image Type** provides the ability to define the color scheme of the image.

❯ **Color** - an image after export will fully comply with the image in the report;

❯ **Gray** - an image after export will be gray.

❯ **Monochrome** - images will be strictly black and white. At the same time, it should be taken into consideration that monochrome images have three modes None, Ordered and FloydSt.

8 The option **Monochrome Dithering Type** allows you to determine the type monochrome color mixing: None - no dithering, Ordered, FloydSt. - with dithering.

9 The option **TIFF Compression Scheme** provides the ability to define a compression scheme for TIFF files.

10 The checkbox **Cut Edges** provides the ability to display a report without page edges. If this is enabled, then when you export the report the page edges will be cut off. If this option is disabled, the report page will be displayed with the specified edges.

11 The checkbox **Multiple Files** is available when exporting to TIFF. By default, each report page is a separate image. When exporting to TIFF you can put multiple images in a single file by disabling the option. You need a special viewer to view the TIFF file that contains multiple images.

12 The flag **Open After Export** enables/disables the automatic opening of the created document (after completion of exports), the default program for these file types.

9.8.1   BMP

BMP (Bitmap) is an image file format used to store bitmap digital images. Initially the format could store only DDB (Device Dependent Bitmap) but today the BMP format stores device-independent rasters (DIB - Device Independent Bitmap). Color depth in this format varies from 1 to 48 bits per pixel. The maximal image size is 65535×65535 pixels. An image can be compressed but often is stored in uncompressed and has big size of the file. Many programs work with the BMP format because its support is integrated into Windows and OS/2.

9.8.2   GIF

GIF (Graphics Interchange Format) is a format to store graphic images. The GIF format can store compressed images, supports up to 8 bits per pixel, allowing a single image to reference a palette of up to 256 distinct colors. The GIF format was introduced by

CompuServe in 1987 and has since come into widespread usage on the World Wide Web. In 1989 the format was modified (GIF89a), and transparency and animation was added. GIF uses LZW-compression. It allows reducing the file size without degrading the visual quality (logos, schemes). GIF is widely used in World Wide Web.

9.8.3 PNG

PNG (Portable Network Graphics) - is a bitmapped image format that employs lossless data compression. PNG was created to improve and replace more simple GIF format, and to replace more complicated TIFF format. In compare with the GIF format, the PNG format supports RGB images without color losses, supports alpha channels, and uses DEFLATE (open algorithm of compression), that provides higher compression of multicolored files. The PNG format is usually used in World Wide Web and for graphic editing.

9.8.4 TIFF

TIFF (Tagged Image File Format) is a file format for storing images. Originally, the TIFF format was created by the Aldus company in cooperation with Microsoft for using with PostScript. TIFF became popular for storing high-color-depth images, and is used for scanning, fax, to identify text, polygraphy and widely used in graphic applications. This format is flexible. It allows saving photos in different color spaces, and to use different algorithms of file compression, and to store a few images in one file.

9.8.5 JPEG

JPEG (Joint Photographic Experts Group) is a format to store images. This format was created by C-Cube Microsystems as effective method to store high-color-depth images. For example, scanned photos with smooth variations of tone and color. Algorithm of compression with losing information is used in the JPEG format. This means that some visual quality is lost in the process and cannot be restored. It is necessary to get the highest coefficient of compression. Unpacked JPEG images are rarely have the same quality as original image but differences are insignificant. Compression ratio is usually set in conventional units, for example from 1 to 100. 100 is the best quality and 1 is the worst quality. The better quality the bigger file size.

9.8.6 PCX

PCX is a format to store images. This format was used in the ZSoft PC Paintbrush graphic editor (one of the most popular programs) for MS-DOS, text processors and Microsoft Word and Ventura Publisher. This is not so popular format analogue of BMP but is supported with such graphics editors as Adobe Photoshop, Corel Draw and others. The algorithm of compression is very quick but is not effective for compression of photos and other detailed computer graphics. Today this format is not displaced with formats which supports better compression. These formats are GIF, JPEG, and PNG.

9.8.7 EMF

WMF (Windows MetaFile) is a universal graphics file format on Microsoft Windows systems. This format was created by Microsoft and is an integral part of Windows because this file stores a list of function calls that have to be issued to the Windows graphics layer GDI in order to display an image on screen.

WMF is a 16-bit format. This format was introduced in Windows 3.0. A 32-bit version is called Enhanced Metafile **EMF** (Enhanced Metafile). The EMF format supports many new commands, supports work with the GDI+ library, and also is used as a graphic language for drivers of printers.

9.8.8 SVG

SVG (Scalable Vector Graphics) is an XML-based file format for describing two-dimensional vector graphics, both static and dynamic. The **SVG** specification is an open standard. **SVG** supports scripting and animation. The vector image is composed of a fixed set of shapes.
SVG allows three types of graphic objects:
➤ Vector graphics;
➤ Raster graphics;
➤ Text.

The Images below shows the difference between exporting Bitmap format and SVG (vector) format.

Bitmap Formats

SVG Format

9.8.9   Compressed SVG

In addition to the **SVG** file format, there is a **compressed SVG** (with file extension **SVGZ**), which applies industry-standard, nonproprietary "gzip" compression (an open-source variant of Zip compression) to **SVG** files. Compressed SVG files are typically 50 to 80 percent smaller than SVG files. **SVG** files are compact and can be used to provide high-quality graphics on the Web.

9.8.10   Dither

Dither is an intentionally applied form of noise, when processing digit signals. It is used in most often surfaces in the fields of digital audio and video. The following image shows (from left to right) original image and the result of export to monochrome image. There are three modes of **DitheringType**: **Ordered**, **FloydSteinberg**, **None**.
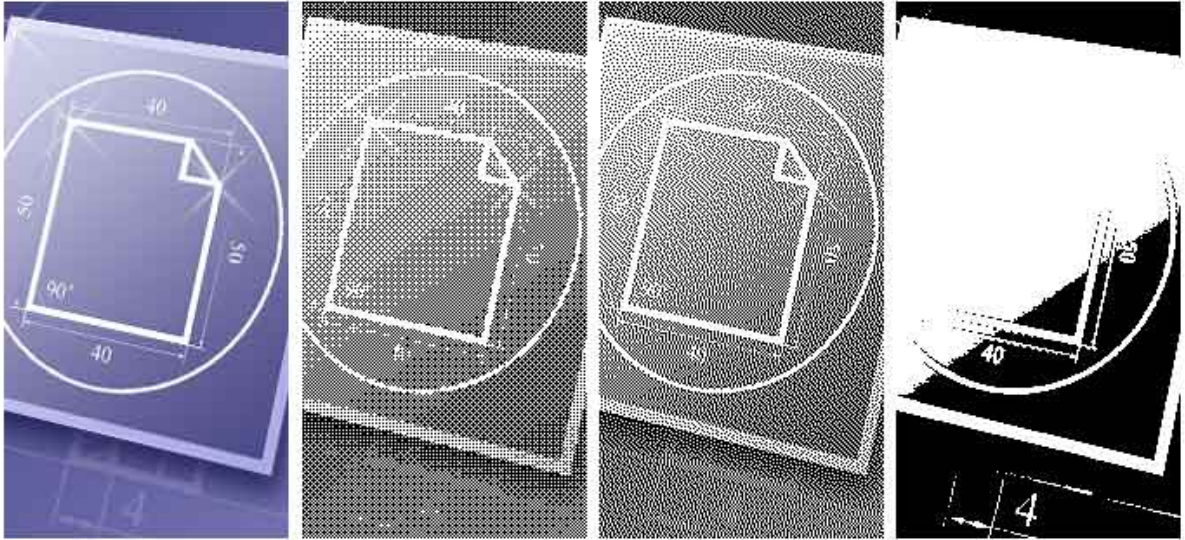
> ❗ **Notice:** On the current moment the export of monochrome image is supported only to the PCX format. So the DitheringType property works only for this export. Different images may look differently in these modes. The **FloydSteinberg** is the best mode to output an image but the file size is too big.

9.9   How to Create Report for Export?

Many exports have the table mode. In this mode the whole report is converted into one table. Creating correct templates from the source code allows making the table look much better, decrease the size of the file, increase the speed of working with export. Therefore, when using the table mode of export it is important to follow some recommendations:

➤ use the "Align to Grid" button of the designer. This will decrease the number of rows and columns in the output file; also this allows avoiding very small gaps between components (some formats "do not like" table with very small columns);

➤ put components on the data band at the same level (see the picture below); this will decrease the number of rows and columns in the output file;
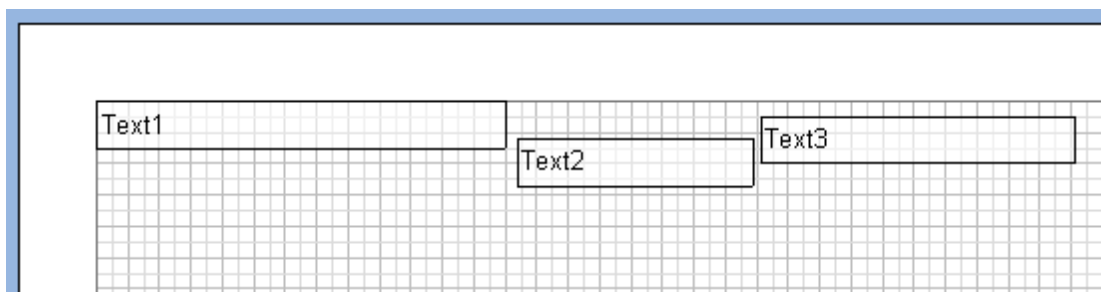
For example: put three components in the designer. They should be placed without gaps. See the picture below:
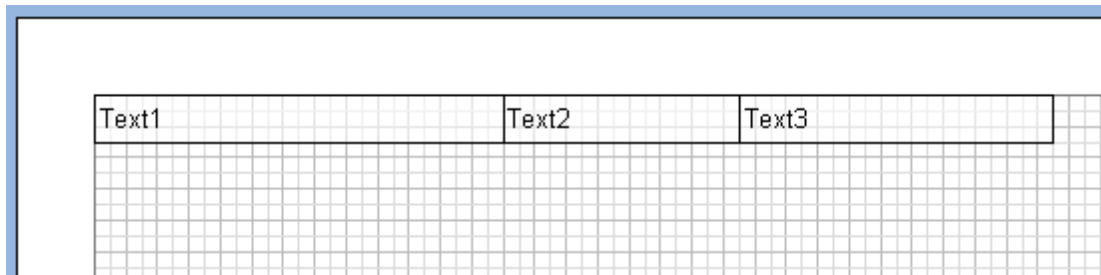
As a result we get a simple table: one row and three columns.

Put three components as seen on the picture below.

As a result we get the Excel table: five rows and three cells (see the picture below). It is not convenient to edit such a table, the file size, time of export, and required memory are increased in some times.


The **Excel** sheet consists of cells that are formed at the intersection of rows and columns. All items (text, images, and other data) are arranged in cells and can take only an integer number of cells, both by width and height. Therefore, when the location of components, column width and row height is adjusted so that the margins of components coincide with the boundaries of columns/rows:
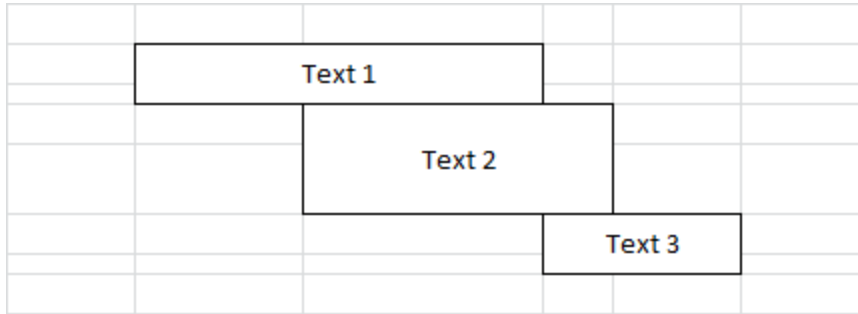
When you export a report, the column width and row height is calculated automatically, so as to place all components using as the smaller number of columns and rows as possible. If all components are arranged in columns/rows, the number of result columns/rows in the Excel file will match the number of columns/rows in the report components. If the template structure is more complex, for example components as headers are not placed in the columns, then additional columns/rows will be added the Excel file. Consider the following example:
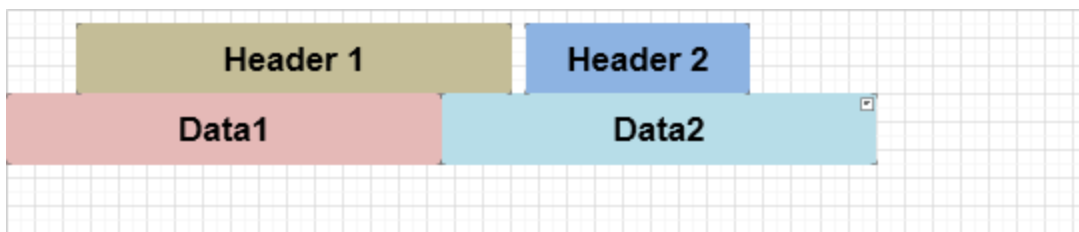
As can be seen from the picture above the text components in the report template are located on different levels (rows) and not in the same columns. In this case, when you export a report to Excel, the result will be as follows:
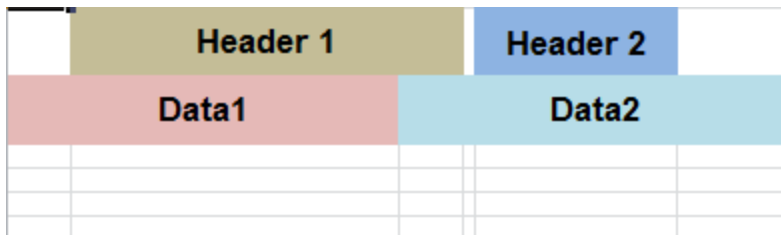
As can be seen from the picture above you add more columns/rows.

> do not use the **Autowidth** property. This property increases the number of columns in the exported file which is proportionally to number of records.

On the left picture the number of columns is 14, and this case is equal in number of data rows. If to disable the **AutoWidth** property then only one column will be output (see the right picture). Accordingly, the file size of a report, shown of the right picture, is some times smaller then the file of the report shown on the left picture and the export works faster.

❗ **Notice:** Number of columns is very important for the text editors. For example, MS Word allows no more than 64 columns; if the table has more than 64 columns then the document is output incorrectly.