# Table of Contents

## Part I Introduction

1. Activation .......................................................................................................................... 2
2. Technical Support ................................................................................................................. 8
3. Web Links and Online Resources ...................................................................................... 9

## Part II Getting Started

1. Simple List Report .............................................................................................................. 12
2. Master-Detail Report .......................................................................................................... 18
3. Report with Groups ........................................................................................................... 25
4. Report with Columns on Page ............................................................................................ 31
5. Report with Columns in Data Band .................................................................................... 38
6. Report with Chart on Page ................................................................................................. 48
7. Report with Chart in Data Band .......................................................................................... 55
8. Report with Cross-Tab on Page .......................................................................................... 65
9. Cross-Tab Report in Data Band .......................................................................................... 72
10. Hierarchical Report ......................................................................................................... 85
11. Report with Sub-Report .................................................................................................... 92
13. Report with Sub-Reports in Data Band ............................................................................ 106
14. Master-Detail Report and Sub-Reports ............................................................................ 114
15. Report with Empty Band ................................................................................................. 123
18. Report without Bands ....................................................................................................... 147
19. Report with Multiple Pages in Template .......................................................................... 150
20. Report with Segmented Pages .......................................................................................... 162
22. Report with Primitives in Band ....................................................................................... 175
23. Report with Cross-Primitives ........................................................................................... 183
24. Drill-Down Report ........................................................................................................... 190
25. Report with Dynamic Data Sorting in Preview ............................................................... 196
27. Report with Table Component ........................................................................................ 214
28. Master-Detail Report with Table .................................................................................... 219
29. Anchors in Report ............................................................................................................ 225
30. Invoice Report .................................................................................................................. 238
31. Invoice Report With Parameters ..................................................................................... 246
Part III  Report Internals

1  Expressions .................................................................................................................. 254
   Text Expressions ........................................................................................................... 254
   Calculating Values in Expressions ............................................................................... 255
   Multi-line Expressions ................................................................................................. 256
   Using Dictionary Variables ......................................................................................... 257
   Using Data Fields ........................................................................................................ 258
   Using Component Properties ....................................................................................... 260
   Using Functions in Expressions ................................................................................... 261
   Conditional Expressions .............................................................................................. 262
   Using Aliases in Expressions ....................................................................................... 263

2  Appearance .................................................................................................................... 265
   Background Brushes .................................................................................................... 265
   Fonts and Font Brushes ............................................................................................... 266
   Borders ......................................................................................................................... 273
      Simple Borders .......................................................................................................... 273
      Advanced Borders .................................................................................................... 277
      Conditional Borders ................................................................................................. 278
   Horizontal Alignment .................................................................................................. 278
      Horizontal Text Alignment ....................................................................................... 279
      Horizontal Image Alignment .................................................................................... 279
   Vertical Alignment ....................................................................................................... 280
      Vertical Text Alignment ............................................................................................ 280
      Vertical Image Alignment ......................................................................................... 281
   Styles ............................................................................................................................. 282
      Alternate Row Styles ................................................................................................ 284
      UseParentStyles Property ....................................................................................... 284
   Style Designer .............................................................................................................. 285
      Creating Collection of Styles .................................................................................. 289
      Conditions ................................................................................................................. 294

3  Conditional Formatting ................................................................................................ 299
   Value Condition ............................................................................................................ 301
   Operators ...................................................................................................................... 302
   Expression Condition .................................................................................................. 304
   Multi Part Conditions ................................................................................................. 305
   Defining Formatting ..................................................................................................... 306
   Conditional Formatting and Text Components .......................................................... 307
   Conditional Formatting and Cross-Tables .................................................................... 308
   Visual Styles Menu ....................................................................................................... 308
      Font Name .................................................................................................................. 309
      Font Size .................................................................................................................... 311
      Font Bold .................................................................................................................... 313
      Font italic ..................................................................................................................... 315
      Font Underlined ......................................................................................................... 317
      Text Color ................................................................................................................ 319
      Back Color ............................................................................................................... 321
      Borders ...................................................................................................................... 323
      Enabling Component ............................................................................................... 325
      Assigning Expression ............................................................................................... 327
   Data Bar Condition ....................................................................................................... 329
   Color Scale Condition .................................................................................................. 333
   Icon Set Condition ....................................................................................................... 337

© 2003-2020 Stimulsoft
# Contents

## 4 Output Text Parameters

- Text Editor .......................................................... 341
- Multiline Text ...................................................... 344
- Trimming in the End of Text Line .......................... 345
- Prevent Showing Incompletely Visible Lines .......... 346
- Lines of Underlining ............................................ 346
- Maximal Number of Lines ..................................... 347
- Text Rotation ....................................................... 348
- Processing Duplicates ........................................... 348
- Ignoring Null Values ............................................ 351
- ReportTo Property ................................................ 352
- Shrink Font To Fit Property .................................... 353
- Shrink Font to Fit Minimum Size Property ............. 355
- Output Text Only without Taking Expressions into Consideration ................................................. 356
- Expression Processing in the End of Report Rendering ................................................................. 357
- Zip code .............................................................. 357

## 5 Text Formatting

- Standard Formatting ............................................. 358
- Numerical Formatting ........................................... 360
- Currency Formatting ............................................ 363
- Date Formatting .................................................... 367
- Time Formatting ................................................... 370
- Percentage Data Formatting ................................. 372
- Boolean Values Formatting ..................................... 375
- Custom Formatting ............................................... 375
- Formatting in Text ................................................ 378

## 6 HTML Tags

- HTML <font> Tag .................................................. 379
- <color Attribute> .................................................. 381
- <face Attribute> ................................................... 382
- <size Attribute> .................................................... 383
- HTML Tags to Change Font Style ......................... 385
- <b> Tag ............................................................... 385
- <i> Tag ................................................................. 386
- <em> Tag .............................................................. 386
- <u> Tag ................................................................. 387
- <s> Tag ................................................................. 387
- <sup> Tag ............................................................. 387
- <sub> Tag .............................................................. 388
- <strong> Tag ......................................................... 388
- <p> Tag ................................................................. 389
- <br> Tag ............................................................... 389
- <ol> Tag .............................................................. 389
- <ul> Tag .............................................................. 390
- <background-color> Tag ...................................... 391
- <text-align> Tag .................................................... 391
- <letter-spacing> Tag .............................................. 392
- <word-spacing> Tag .............................................. 392
- <line-height> Tag ................................................ 392
- Special Characters .............................................. 393

## 7 Rich Text

- Rich Text Editor .................................................. 397
- Expressions in Rich Text ...................................... 398
8 Graphic Information Output ................................................................. 402
  Loading Images .................................................................................. 404
  Image Stretching ................................................................................ 406
  Resources of Images .......................................................................... 407
9 Autosize ............................................................................................. 409
  Automatically Resizing Text Component ............................................. 410
  Automatically Resizing Panels ............................................................ 411
  Automatically Resizing Bands ............................................................. 412
  Binding Bottom Border of Component ............................................... 414
  Automatically Shifting Components .................................................. 416
10 Barcodes ............................................................................................ 418
  Barcode Editor .................................................................................... 420
  Barcode Size ........................................................................................ 424
  Linear Barcodes .................................................................................. 425
    EAN/UPC Based ............................................................................... 425
      What is EAN.UCC System? ............................................................. 426
      UPC-A .......................................................................................... 426
      UPC-E .......................................................................................... 427
      EAN-13 ....................................................................................... 428
      EAN-8 ........................................................................................ 429
      Add-On Symbols .......................................................................... 430
      EAN-128 ..................................................................................... 431
      ITF-14 ......................................................................................... 432
      JAN-13 ....................................................................................... 433
      JAN-8 ........................................................................................ 433
      ISBN-10 ..................................................................................... 434
      ISBN-13 ..................................................................................... 435
  Other Barcodes .................................................................................... 435
    Pharmacode ...................................................................................... 435
    Plessey ........................................................................................... 436
    MSI ................................................................................................. 437
    2of5 ............................................................................................... 438
    FIM ................................................................................................. 439
    Codabar .......................................................................................... 440
    Postnet ............................................................................................ 441
    Australia Post 4-state ...................................................................... 441
    Royal TPG Post KIX 4-State ............................................................. 442
    Royal Mail 4-state .......................................................................... 443
    Code11 ............................................................................................ 444
    Code39 ........................................................................................... 445
    Code93 ........................................................................................... 446
    Code128 ......................................................................................... 447
    Barcode Comparison Table ............................................................... 448
  2D Barcodes ........................................................................................ 451
    PDF417 ........................................................................................... 451
    Datamatrix ....................................................................................... 453
    QR Code .......................................................................................... 456
11 Report ................................................................................................ 457
  Report Structure .................................................................................. 458
  Report Rendering ............................................................................... 459
12 Pages .................................................................................................. 460
<table>
<thead>
<tr>
<th>Contents</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print On Previous Page Property</td>
<td>460</td>
</tr>
<tr>
<td>Margins</td>
<td>463</td>
</tr>
<tr>
<td>13 Bands</td>
<td>466</td>
</tr>
<tr>
<td>Band Types</td>
<td>466</td>
</tr>
<tr>
<td>Standard Bands</td>
<td>467</td>
</tr>
<tr>
<td>Cross-bands</td>
<td>470</td>
</tr>
<tr>
<td>Rendering Order of Bands</td>
<td>471</td>
</tr>
<tr>
<td>14 Creating Lists</td>
<td>489</td>
</tr>
<tr>
<td>Data Band</td>
<td>489</td>
</tr>
<tr>
<td>Data Source of Data Band</td>
<td>490</td>
</tr>
<tr>
<td>List Output</td>
<td>494</td>
</tr>
<tr>
<td>List with Header</td>
<td>496</td>
</tr>
<tr>
<td>List with Footer</td>
<td>497</td>
</tr>
<tr>
<td>KeepHeaderTogether Property</td>
<td>498</td>
</tr>
<tr>
<td>KeepFooterTogether Property</td>
<td>499</td>
</tr>
<tr>
<td>Enumeration in Lists</td>
<td>500</td>
</tr>
<tr>
<td>Selecting Rows One After Another</td>
<td>502</td>
</tr>
<tr>
<td>Events and Data Band</td>
<td>505</td>
</tr>
<tr>
<td>Data Sorting</td>
<td>506</td>
</tr>
<tr>
<td>Data Filtering</td>
<td>507</td>
</tr>
<tr>
<td>Lists One After Another</td>
<td>513</td>
</tr>
<tr>
<td>PrintOn Property</td>
<td>515</td>
</tr>
<tr>
<td>PrintOnEvenOddPages Property</td>
<td>515</td>
</tr>
<tr>
<td>PrintOnAllPages Property</td>
<td>517</td>
</tr>
<tr>
<td>PrintAtBottom Property</td>
<td>518</td>
</tr>
<tr>
<td>Drag and Drop From Dictionary</td>
<td>519</td>
</tr>
<tr>
<td>Check Box</td>
<td>521</td>
</tr>
<tr>
<td>15 Creating Master-Detail Lists</td>
<td>522</td>
</tr>
<tr>
<td>MasterComponent Property</td>
<td>524</td>
</tr>
<tr>
<td>DataRelation Property</td>
<td>525</td>
</tr>
<tr>
<td>Relation</td>
<td>526</td>
</tr>
<tr>
<td>Multilevel Nesting</td>
<td>529</td>
</tr>
<tr>
<td>KeepDetails Property</td>
<td>530</td>
</tr>
<tr>
<td>Rows Numbering in Master-Detail Reports</td>
<td>532</td>
</tr>
<tr>
<td>Through Lines Numbering in Master-Detail Reports</td>
<td>534</td>
</tr>
<tr>
<td>Headers, Footers and Master-Detail Reports</td>
<td>535</td>
</tr>
<tr>
<td>PrintIfDetailEmpty Property</td>
<td>536</td>
</tr>
<tr>
<td>16 Groups</td>
<td>537</td>
</tr>
<tr>
<td>Grouping Conditions</td>
<td>538</td>
</tr>
<tr>
<td>Group Header band</td>
<td>539</td>
</tr>
<tr>
<td>Group Footer band</td>
<td>540</td>
</tr>
<tr>
<td>Data Sorting in Group</td>
<td>541</td>
</tr>
<tr>
<td>GroupFooter</td>
<td>543</td>
</tr>
<tr>
<td>KeepGroupTogether Property</td>
<td>544</td>
</tr>
<tr>
<td>KeepGroupHeaderTogether Property</td>
<td>545</td>
</tr>
<tr>
<td>KeepGroupFooterTogether Property</td>
<td>546</td>
</tr>
<tr>
<td>Events and Group Header band</td>
<td>547</td>
</tr>
<tr>
<td>Group Without Group Header</td>
<td>547</td>
</tr>
<tr>
<td>Nested Groups</td>
<td>548</td>
</tr>
<tr>
<td>Groups Without Group Footer</td>
<td>549</td>
</tr>
<tr>
<td>LineThrough System Variable</td>
<td>550</td>
</tr>
<tr>
<td>Numbering Rows in Group</td>
<td>550</td>
</tr>
<tr>
<td>GroupLine System Variable</td>
<td>551</td>
</tr>
</tbody>
</table>
VI Reports and Dashboards

Combining Groups and Master-Detail Reports ............................................................... 552

17 Page Bands ............................................................................................................. 552
Page Header Band .................................................................................................... 553
Page Footer Band .................................................................................................... 554
PrintOnEvenOddPages Property ........................................................................... 557

18 Report Bands ....................................................................................................... 559
Report Title band .................................................................................................... 559
Report Summary band ............................................................................................ 559
ReportTitleBand Property ..................................................................................... 560
KeepReportSummaryTogether Property ................................................................ 561
Print At Bottom Property ...................................................................................... 562
Print If Empty Property ........................................................................................ 563

19 Columns .............................................................................................................. 564
Columns on Page .................................................................................................... 565
Columns on Data Band ............................................................................................ 567
AcrossThenDown Mode ........................................................................................ 568
DownThenAcross Mode ........................................................................................... 570
Minimal Number of Rows in Column .................................................................... 573
Column Header Band ............................................................................................... 577
PrintIfEmpty Property ........................................................................................... 576
Column Footer Band ............................................................................................... 577
PrintIfEmpty Property ........................................................................................... 578
Header and Footer Combinations ......................................................................... 578
AcrossThenDown Column Mode ........................................................................... 579
DownThenAcross Column Mode ........................................................................... 579

20 Page and Column Break ....................................................................................... 580
Page Break ............................................................................................................... 580
Column Break ......................................................................................................... 583
Break if Less Than Property .................................................................................... 586
Skip First Property ................................................................................................ 586

21 Pagination ........................................................................................................... 587
Page Number ............................................................................................................ 587
Total Page Count .................................................................................................... 588
Page NoFm ................................................................................................................. 588
ResetPageNumber Property ................................................................................... 589
Sequentially Numbered Pages ................................................................................ 591

22 Breaking Component ......................................................................................... 593
Breaking Bands ....................................................................................................... 594
Breaking Text .......................................................................................................... 595
Breaking Panels ...................................................................................................... 597
Breaking RichText .................................................................................................. 599
Breaking Images .................................................................................................... 600
Auto-break ............................................................................................................... 601
Breaking and Page Bands ....................................................................................... 602

23 Hierarchical Band ............................................................................................... 602
Data Output .............................................................................................................. 602
KeyDataColumn Property ...................................................................................... 603
MasterKeyDataColumn Property ......................................................................... 604
ParentValue Property ............................................................................................. 604
Indent Property ...................................................................................................... 606

24 Child Band .......................................................................................................... 608

© 2003-2020 Stimulsoft
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi Line Header</td>
<td>609</td>
</tr>
<tr>
<td>Child Band and Data</td>
<td>611</td>
</tr>
<tr>
<td>KeepChildTogether Property</td>
<td>612</td>
</tr>
<tr>
<td><strong>25 Empty Band</strong></td>
<td></td>
</tr>
<tr>
<td>Empty Band Modes</td>
<td>613</td>
</tr>
<tr>
<td><strong>26 Watermarks</strong></td>
<td>614</td>
</tr>
<tr>
<td>Watermark Property</td>
<td>616</td>
</tr>
<tr>
<td>Overlay Band</td>
<td>617</td>
</tr>
<tr>
<td>Vertical Alignment Property</td>
<td>619</td>
</tr>
<tr>
<td>Direct Allocation on Page</td>
<td>621</td>
</tr>
<tr>
<td><strong>27 Panels</strong></td>
<td>622</td>
</tr>
<tr>
<td>Placing Bands on Panel</td>
<td>623</td>
</tr>
<tr>
<td><strong>Placing Panels</strong></td>
<td>623</td>
</tr>
<tr>
<td>Placing Panels on Page</td>
<td>623</td>
</tr>
<tr>
<td>Placing Panels on Band</td>
<td>624</td>
</tr>
<tr>
<td>Placing Panels on Panel</td>
<td>624</td>
</tr>
<tr>
<td>Side-by-Side Reports</td>
<td>625</td>
</tr>
<tr>
<td>Multiple Tables on One Page</td>
<td>627</td>
</tr>
<tr>
<td>Cloning</td>
<td>628</td>
</tr>
<tr>
<td><strong>28 Cross-Tab</strong></td>
<td>629</td>
</tr>
<tr>
<td>Data Source Property</td>
<td>630</td>
</tr>
<tr>
<td>Cross Table Items</td>
<td>631</td>
</tr>
<tr>
<td>Columns</td>
<td>631</td>
</tr>
<tr>
<td>Rows</td>
<td>633</td>
</tr>
<tr>
<td>Summary Cells</td>
<td>636</td>
</tr>
<tr>
<td><strong>Cross-Tab Editor</strong></td>
<td>639</td>
</tr>
<tr>
<td>Cross-Tab Tab</td>
<td>641</td>
</tr>
<tr>
<td>Styles Tab</td>
<td>643</td>
</tr>
<tr>
<td><strong>Data Summary Types</strong></td>
<td>644</td>
</tr>
<tr>
<td>Sort Direction</td>
<td>644</td>
</tr>
<tr>
<td>Conditions</td>
<td>645</td>
</tr>
<tr>
<td>Showing Totals</td>
<td>647</td>
</tr>
<tr>
<td>Processing Values for Summary</td>
<td>649</td>
</tr>
<tr>
<td>Word Wrap</td>
<td>649</td>
</tr>
<tr>
<td><strong>29 Charts</strong></td>
<td>650</td>
</tr>
<tr>
<td>Charts Editor</td>
<td>657</td>
</tr>
<tr>
<td>Tab Chart</td>
<td>659</td>
</tr>
<tr>
<td>Tab Series</td>
<td>661</td>
</tr>
<tr>
<td>Tab Area</td>
<td>662</td>
</tr>
<tr>
<td>Tab Labels</td>
<td>664</td>
</tr>
<tr>
<td>Tab Styles</td>
<td>665</td>
</tr>
<tr>
<td>Wizard</td>
<td>667</td>
</tr>
<tr>
<td><strong>Area</strong></td>
<td>674</td>
</tr>
<tr>
<td>Doughnut</td>
<td>675</td>
</tr>
<tr>
<td>Series Labels</td>
<td>675</td>
</tr>
<tr>
<td>Pie</td>
<td>676</td>
</tr>
<tr>
<td>Series Labels</td>
<td>677</td>
</tr>
<tr>
<td>CutPeList Property</td>
<td>679</td>
</tr>
<tr>
<td><strong>Legend</strong></td>
<td>681</td>
</tr>
<tr>
<td>Title Property</td>
<td>681</td>
</tr>
<tr>
<td>HorizontalAlignment Property</td>
<td>681</td>
</tr>
<tr>
<td>VerticalAlignment Property</td>
<td>683</td>
</tr>
</tbody>
</table>

© 2003-2020 Stimulsoft
### Series Labels

<table>
<thead>
<tr>
<th>Property</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Series Labels Appearance</td>
<td>725</td>
</tr>
<tr>
<td>Brush Property</td>
<td>725</td>
</tr>
<tr>
<td>Font Property</td>
<td>727</td>
</tr>
<tr>
<td>LabelColor Property</td>
<td>729</td>
</tr>
<tr>
<td>UseSeriesColor Property</td>
<td>729</td>
</tr>
<tr>
<td>Angle Property</td>
<td>730</td>
</tr>
<tr>
<td>Draw Border Property</td>
<td>731</td>
</tr>
<tr>
<td>BorderColor Property</td>
<td>732</td>
</tr>
<tr>
<td>AntiAliasing Property</td>
<td>733</td>
</tr>
<tr>
<td>Format Property</td>
<td>733</td>
</tr>
<tr>
<td>ValueType Property</td>
<td>736</td>
</tr>
<tr>
<td>ValueSeparator</td>
<td>739</td>
</tr>
<tr>
<td>PreventIntersection Property</td>
<td>739</td>
</tr>
<tr>
<td>ShowOnZeroValues Property</td>
<td>740</td>
</tr>
<tr>
<td>Step Property</td>
<td>742</td>
</tr>
<tr>
<td>TextBefore and TextAfter Properties</td>
<td>743</td>
</tr>
<tr>
<td>Visible Property</td>
<td>743</td>
</tr>
</tbody>
</table>
Part VI

1 Data Dictionary .................................................................................................................. 1290
Part VIII

Reports and Dashboards

Control Panel .................................................................................................................. 1291
  Actions Menu ........................................................................................................... 1292
  New Item Menu ....................................................................................................... 1293
Data Sources .................................................................................................................. 1295
  Queries ..................................................................................................................... 1297
    Parameters .............................................................................................................. 1301
  Creating Data Source ............................................................................................... 1306
  Creating and Editing Data Columns ......................................................................... 1313
  Calculated Data Column ......................................................................................... 1315
  Data From Other Data Source .................................................................................. 1317
  Data From Cross-Tab ............................................................................................... 1320
  Custom Data Sources ............................................................................................... 1323
Relation ........................................................................................................................ 1325
  Filtering .................................................................................................................... 1326
  Sorting ....................................................................................................................... 1327
  Showing Information ............................................................................................... 1329
  Master-Detail Report .............................................................................................. 1331
  Creating Relation .................................................................................................... 1332
    Limitations in Creating Relations ........................................................................... 1334
Variables ....................................................................................................................... 1334
  Panel Request From User ........................................................................................ 1338
    Items Dialog ........................................................................................................ 1339
    Dependent Variables ........................................................................................... 1340
  New Variable ........................................................................................................... 1348
  Nullable Value ........................................................................................................ 1350
  List ........................................................................................................................... 1351
  Range ......................................................................................................................... 1353
  Three Modes of Variable Functioning .................................................................... 1355
Panel Setup .................................................................................................................. 1358
  System Variables .................................................................................................... 1358
  Business Object ...................................................................................................... 1360
Resources ..................................................................................................................... 1363

Part VII Right To Left .................................................................................................. 1370

  1 Text Component ................................................................................................... 1370
  2 Text In Cells Component ...................................................................................... 1370
  3 Cross Table Component ....................................................................................... 1371
  4 Chart Component ................................................................................................ 1373
  5 Columns on Page .................................................................................................. 1373
  6 Columns in Data Band .......................................................................................... 1375

Part VIII Viewer ......................................................................................................... 1379

  1 Reports ................................................................................................................ 1380
    Dot-Matrix ............................................................................................................. 1386
    Special Viewing Options in Web ............................................................................ 1389
  2 Dashboards ........................................................................................................... 1392

Part IX Exports ............................................................................................................. 1397

  1 Available File Formats ......................................................................................... 1398
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Common Export Settings</td>
<td>1399</td>
</tr>
<tr>
<td>Image Quality</td>
<td>1399</td>
</tr>
<tr>
<td>Image Resolution</td>
<td>1399</td>
</tr>
<tr>
<td>Image Comparer</td>
<td>1400</td>
</tr>
<tr>
<td>Convert Digits to Arabic</td>
<td>1400</td>
</tr>
<tr>
<td>Arabic Digits Type</td>
<td>1400</td>
</tr>
<tr>
<td>Divide Segment Pages</td>
<td>1401</td>
</tr>
<tr>
<td>Remove Empty Space at Bottom</td>
<td>1401</td>
</tr>
<tr>
<td>Use One Page Header and Footer</td>
<td>1401</td>
</tr>
<tr>
<td>3 Formats with Fixed Page Layout</td>
<td>1401</td>
</tr>
<tr>
<td>PDF</td>
<td>1402</td>
</tr>
<tr>
<td>Embedded Fonts</td>
<td>1405</td>
</tr>
<tr>
<td>Font Styles</td>
<td>1405</td>
</tr>
<tr>
<td>Digital Signature</td>
<td>1406</td>
</tr>
<tr>
<td>Keys</td>
<td>1406</td>
</tr>
<tr>
<td>Public Key Certificate</td>
<td>1406</td>
</tr>
<tr>
<td>Choosing Certificate</td>
<td>1407</td>
</tr>
<tr>
<td>Pacing Digital Signature Identifier</td>
<td>1407</td>
</tr>
<tr>
<td>Encryption</td>
<td>1408</td>
</tr>
<tr>
<td>Passwords and Access Permission</td>
<td>1408</td>
</tr>
<tr>
<td>Key Length</td>
<td>1409</td>
</tr>
<tr>
<td>Compatibility of Different Versions</td>
<td>1409</td>
</tr>
<tr>
<td>XPS</td>
<td>1410</td>
</tr>
<tr>
<td>Microsoft Power Point 2007/2010</td>
<td>1412</td>
</tr>
<tr>
<td>4 Web Documents</td>
<td>1413</td>
</tr>
<tr>
<td>HTML</td>
<td>1415</td>
</tr>
<tr>
<td>Export Modes</td>
<td>1415</td>
</tr>
<tr>
<td>Export Images in HTML Format</td>
<td>1416</td>
</tr>
<tr>
<td>Compatibility of Different Versions</td>
<td>1416</td>
</tr>
<tr>
<td>Exporting Text Components</td>
<td>1416</td>
</tr>
<tr>
<td>HTML5</td>
<td>1418</td>
</tr>
<tr>
<td>MHT</td>
<td>1418</td>
</tr>
<tr>
<td>5 Text Formats</td>
<td>1418</td>
</tr>
<tr>
<td>TXT</td>
<td>1419</td>
</tr>
<tr>
<td>Border Types</td>
<td>1420</td>
</tr>
<tr>
<td>Column Width</td>
<td>1421</td>
</tr>
<tr>
<td>New Export Mode</td>
<td>1421</td>
</tr>
<tr>
<td>RTF</td>
<td>1421</td>
</tr>
<tr>
<td>Export Modes</td>
<td>1423</td>
</tr>
<tr>
<td>Table Mode</td>
<td>1423</td>
</tr>
<tr>
<td>Issues</td>
<td>1425</td>
</tr>
<tr>
<td>Compatibility of Different Versions</td>
<td>1425</td>
</tr>
<tr>
<td>Word 2007/2010</td>
<td>1426</td>
</tr>
<tr>
<td>Headers and Footers</td>
<td>1428</td>
</tr>
<tr>
<td>Page Numbering</td>
<td>1428</td>
</tr>
<tr>
<td>ODT</td>
<td>1429</td>
</tr>
<tr>
<td>6 Spreadsheets</td>
<td>1431</td>
</tr>
<tr>
<td>Excel</td>
<td>1433</td>
</tr>
<tr>
<td>Excel Sheets</td>
<td>1433</td>
</tr>
<tr>
<td>Compatibility of Different Versions</td>
<td>1433</td>
</tr>
<tr>
<td>Excel XML</td>
<td>1434</td>
</tr>
<tr>
<td>Excel 2007/2010</td>
<td>1434</td>
</tr>
</tbody>
</table>
Sheets in Excel................................................................. 1434
ODS ............................................................................. 1435

7 Data ........................................................................... 1436
CSV .............................................................................. 1437
Controlling Exports ....................................................... 1438
DBF .............................................................................. 1438
Controlling Exports ....................................................... 1439
XML ............................................................................ 1440
Controlling Exports ....................................................... 1441
DIF .............................................................................. 1442
SYLK ........................................................................... 1442

8 Images ........................................................................ 1442
BMP ............................................................................ 1444
GIF ............................................................................. 1444
PNG ............................................................................ 1445
TIFF ............................................................................ 1445
JPEG .......................................................................... 1445
PCX ............................................................................ 1446
EMF ............................................................................ 1446
SVG ............................................................................ 1446
Compressed SVG ........................................................... 1447
Dither .......................................................................... 1447

9 How to Create Report for Export? ...................................... 1448

10 Export Dashboard ........................................................ 1451

Index 0
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:
- Technical Support
- Trial License Limitations
- Information in Web and References
- Evaluate Demo Version
- Features - Stimulsoft Reports Product Line
  - Features - Stimulsoft Reports.Fx Product Line

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

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

Work with Data:
- Control Panel

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

Viewer
- Report Viewer
- Dot-matrix Viewer
- Special Viewing Options in Web
- Dashboard Viewer

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

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

1.1 Activation

Trial version
The free trial of Stimulsoft software is a full-featured version. It has a few limitations, which are as follows:
- The evaluation period is limited in 30 days which starts from the date of the account registration;
- The Trial watermark is printed on each report page or the dashboard panel.

Information

It is impossible to run the designer without logging into the user account. You can create a free account to start the report creation.

This chapter will cover the following:
- Registering a user account on the website;
- Report Designer Activation;
- Logging in to the account using a proxy server;
- Contacts.
Registering a user account on the website

Step 1: Go to the Stimulsoft website at https://stimulsoft.com.

Step 2: Find the Sign Up button on top of the start page and click it;

Step 3: Fill in the required fields - first name, last name, email address, and account password.

Step 4: Confirm “I'm not a robot”;

Step 5: Read the license agreement. Check the box that you have read and accepted the license agreement.

Step 6: Click the Sign Up button, if you agree and accept the privacy policy and terms of use.
You can register a user account from the report designer. In addition, you can log in using your Google account.
**Report Designer Activation**
The report designer will be activated when you log in to a licensed user account. If, after authentication, the Trial watermark is present on the report pages and dashboards, you probably do not have a license for this product.

**Purchasing or renewing a report designer**

**Step 1:** Click on the **Account** menu in the upper right corner of the report designer and select the **Subscription** item;

**Step 2:** In the **Subscriptions** dialog, hover over the subscription you want to renew and click the **Renew** button;

**Step 3:** After that, you will be redirected to the **Online store**;

**Step 4:** Select the subscription option you need and click the **Request Quote** button;

**Step 5:** Fill in the required fields and click the **Get Quote** button;

**Step 6:** Click the **Purchase** button in the PDF file you get;

**Step 7:** Make the payment.

**Purchasing a new subscription or updating it from the personal account**

**Step 1:** Log in to your personal account and click the **Purchase** button for a specific product;
**Step 2:** After that, you will be redirected to the [Online store](#);

**Step 3:** Select the subscription type you need and click the **Request Quote** button;

**Step 4:** Fill in the required fields and click the **Get Quote** button;

**Step 5:** Click the **Purchase** button in the PDF file you get;

**Step 6:** Make the payment.

Logging in to the account using a proxy server

**Step 1:** Run the report designer;

**Step 2:** Click the **Settings** button in the login window;

**Step 3:** Enter the proxy server address, port, username and password in the appropriate
fields;

**Step 4:** Click the **Save** button;

![Image of Save button and proxy details]

**Step 5:** Log in to the designer using your account credentials.

In case of any other situation 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:**
  - +375-29-2125706

- **Skype:**
  - stimulsoft
1.2 **Technical Support**

Licensed users and users who are evaluating the software (pre-sale) may get technical support.

---

For technical questions, please send requests to [support@stimulsoft.com](mailto:support@stimulsoft.com)

For licensing, subscription, payment questions, send your questions to [sales@stimulsoft.com](mailto:sales@stimulsoft.com)

For other inquiries, please use the e-mail address: [info@stimulsoft.com](mailto:info@stimulsoft.com)

If you have issues 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 licensed user and you contact us for technical support, use the same email address you used when you purchased our product. Otherwise, it won’t be easy to identify you as a licensed client. 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;
- The 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
  - Description of products can be found at https://stimulsoft.com/en/products
  - The latest downloads of the product you may find at https://www.stimulsoft.com/en/downloads/reports
  - You can read the latest news about the company at https://stimulsoft.com/en/blog/news

- Besides, you can download packages of products Stimulsoft from other resources:
You can evaluate our reporting tools online - https://demo.stimulsoft.com

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

A vast number of video lessons are available:
  - on our YouTube channel https://www.youtube.com/user/StimulsoftVideos
  - on our website https://www.stimulsoft.com/en/videos

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

Find us in social networks and messengers:

https://twitter.com/stimulsoft

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

https://www.facebook.com/Stimulsoft

https://t.me/stimulsoft

HTTPS://WWW.STIMULSOFT.COM/EN/RSS

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

+375-29-2125706

WhatsApp - Stimulsoft

Skype - Stimulsoft

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

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

2 **Getting Started**

This section describes step-by-step approaches in designing reports and dashboards.

Creating a report or dashboard ready for printing consists of the following steps:

- Creating a [report](#) or [dashboard](#) template (structure) in the report designer;
- Viewing the [report](#) or [dashboard](#) template in the report viewer or on the Preview tab.
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:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Customers</th>
<th>...</th>
</tr>
</thead>
</table>

6. Put text components with expressions in the DataBand. Where expression is a reference to the data field. For example, put two text components with expressions: {Customers.CompanyName} and {Customers.City};

7. Edit Text and TextBox component:
   7.1. Drag and drop the text component in the DataBand;
   7.2. Change parameters of the text font: size, type, color;
   7.3. Align the text component by width and height;
   7.4. Change the background of the text component;
   7.5. Align text in the text component;
7.6. Change the value of properties of the text component. For example, set the Word Wrap property to true, if you need a text to be wrapped;

7.7. Enable Borders for the text component, if required.

7.8. Change the border color.

The picture below shows a report template with the list:

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

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
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:
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:
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,
4. Put two **Data Bands** on a page of a report template.

5. Edit **Data Band 1** and **Data Band 2**:
   5.1. Align them by height;
   5.2. Change values of required properties. For example, if to set the **PrintIfDetailEmpty** property of the **Data Band 1** 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 **Data Band 2** 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 **Data Band**;
   5.4. Enable **Borders** of the band, if required;

6. Define data sources for **Data Bands**, a define the **Master** component. In our tutorial, the **Master** component is the **Data Band 1**. This means that in the **Data Setup** window of the lower **Data Band 2**, the **Data Band 1** will be specified as the Master component in the **Master Component** tab;

7. Fill the **Data Relation** property of the **Data Band**, that is the **Detail** components. In our case this **Data Band 2**:

8. Put text components with expressions on **Data Bands**. Where expression is a reference to the data field. For example, put a text component with the expression `{Customers.CompanyName}` on the **Data Band 1**. Put a text component with `{Products.ProductName}` and `{Products.UnitPrice}` expressions in the **Data Band 2**;

9. Edit **Text** and **Text Box** component:
   9.1. Drag and drop the text component in **Data Bands**;
   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.

[Image]

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:

[Image]

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**:

![Simple List Report Template](image)

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**:

![Style Designer](image)

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:
   1. Create **New Connection**;
   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:
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:
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:
   1. Create **New Connection**;
   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.

![DataBand example](image)

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:

```
Data Source   Customers   ...
```

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}`.

![DataBand with text components](image)

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:

![Report Template with Edited Text Component](image)

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:

![Report Sample with Two Columns](image)

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:

![Report sample with header and footer](image)

**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**:

![Style Designer](image)
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.

![DataBand example](image)

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

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Customers</th>
<th>...</th>
</tr>
</thead>
</table>

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**:

![Column options example](image)

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.
<table>
<thead>
<tr>
<th>1. Aachen</th>
<th>24. Elgin</th>
<th>47. Madrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>11. Bracke</td>
<td>34. Lander</td>
<td>57. Montreal</td>
</tr>
<tr>
<td>18. Campinas</td>
<td>41. London</td>
<td>64. Paris</td>
</tr>
<tr>
<td>22. Cowes</td>
<td>45. Lueé</td>
<td>68. Reims</td>
</tr>
<tr>
<td>23. Cunevalde</td>
<td>46. Lyon</td>
<td>69. Resende</td>
</tr>
</tbody>
</table>
10. Go back to the report template;
11. If needed, add other bands to the report template, for example, **ColumnHeaderBand** and **ColumnFooterBand**.
12. Edit these bands:
   12.1. Align them by height;
   12.2. Change values of properties, if required;
   12.3. Change the background of bands;
   12.4. Enable **Borders**, if required;
   12.5. Set the border color.

13. Put text components with expressions in these bands. Where expression of the text component in the **ColumnHeaderView** is the column name and the expression of the text component in the **ColumnFooterView** is the data footer.

14. Edit **Text** and **TextBox** component:
   14.1. Drag and drop the text component in **ColumnHeaderView** and **ColumnFooterView**;
   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

<table>
<thead>
<tr>
<th>City</th>
<th>City</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Anchorage</td>
<td>24. Elgin</td>
<td>45. Lille</td>
</tr>
<tr>
<td>4. Århus</td>
<td>25. Bucuresti</td>
<td>46. Lyon</td>
</tr>
<tr>
<td>15. Buenos Aires</td>
<td>36. Lille</td>
<td>57. Montréal</td>
</tr>
<tr>
<td>20. Charleroi</td>
<td>41. London</td>
<td>52. Oulu</td>
</tr>
</tbody>
</table>
### 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:
<table>
<thead>
<tr>
<th>City</th>
<th>City</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aachen</td>
<td>Cowes</td>
<td>London</td>
</tr>
<tr>
<td>Albuquerque</td>
<td>Cuneo</td>
<td>London</td>
</tr>
<tr>
<td>Anchorage</td>
<td>Djem</td>
<td>Lusail</td>
</tr>
<tr>
<td>Århus</td>
<td>Eugene</td>
<td>Lyon</td>
</tr>
<tr>
<td>Barcelona</td>
<td>Frankfurt a.M.</td>
<td>Madrid</td>
</tr>
<tr>
<td>San Quintino</td>
<td>Genève</td>
<td>Madrid</td>
</tr>
<tr>
<td>Bergamo</td>
<td>Graz</td>
<td>Madrid</td>
</tr>
<tr>
<td>Berlin</td>
<td>Helsinki</td>
<td>Mannheim</td>
</tr>
<tr>
<td>Bern</td>
<td>I. de Margarita</td>
<td>Marseille</td>
</tr>
<tr>
<td>Bielefeld</td>
<td>Kiel</td>
<td>México D.F.</td>
</tr>
<tr>
<td>Bielefeld</td>
<td>Kiel</td>
<td>México D.F.</td>
</tr>
<tr>
<td>Brandenburg</td>
<td>Köln</td>
<td>México D.F.</td>
</tr>
<tr>
<td>Brussels</td>
<td>Lander</td>
<td>México D.F.</td>
</tr>
<tr>
<td>Buenos Aires</td>
<td>Lille</td>
<td>Montreal</td>
</tr>
<tr>
<td>Buenos Aires</td>
<td>Lisboa</td>
<td>München</td>
</tr>
<tr>
<td>Butte</td>
<td>Lisboa</td>
<td>Münster</td>
</tr>
<tr>
<td>Campinas</td>
<td>London</td>
<td>Nantes</td>
</tr>
<tr>
<td>Caracas</td>
<td>London</td>
<td>Nantes</td>
</tr>
<tr>
<td>Clermont</td>
<td>London</td>
<td>Oulu</td>
</tr>
<tr>
<td>Cork</td>
<td>London</td>
<td>Paris</td>
</tr>
</tbody>
</table>
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; 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, 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 video file.

Suppose a Chart component is placed on the page of the report, then, for a report, this component will be rendered as a page item. If the Chart component is placed in the DataBand, then, when rendering a report, this component will be rendered as part of the DataBand. Since the Chart component placed in the DataBand, is rendered as a part of the DataBand, and will be printed as many times as the DataBand will be output. An example of designing a report with a chart in the DataBand will be described below. In this example, the chart will graphically display the detailed data of the data source in the DataBand. Follow the steps below 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:

   | Data Source | Categories | ... |
7. Put the Chart component in the DataBand as seen on a picture below:

![Chart in DataBand](image)

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.

![Edited Chart](image)
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, 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:

![Chart Example](image)

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**:
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:

   | Data Source | Products | ... |

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 **DataSource** to the **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, 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 the DataSource field to 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, 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 the DataSource field to the Summary field of the cross-table. For example, add the UnitInStock data column to the Summary field of the cross-table, all entries in this data column will be summary entries in the cross-table;

8. Press the OK button to save your changes and go back to the report template with cross-table.

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

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:
2.9 Cross-Tab Report in Data Band

For better understanding this step-by-step instruction, please watch the video file.

If the Cross-Tab component is placed in the DataBand, then when designing a report, this component will be constructed as part of the DataBand. Because the Cross-Tab component placed in the DataBand is designed as an element of the DataBand, then, when designing a report, this component will be printed as many times as the DataBand. Consider an example of building a report with the Cross-Tab in the DataBand. In this example, Cross-Tab will display the detailed entries in the Master-Detail report. Do the following steps to build a report with the Cross-Tab in the DataBand:

1. Run the designer;
2. Connect data:
   2.1. Create New Connection;
   2.2. Create New Data Source;
3. Create the Relation between data sources. If the Relation is not created and/or the Relation property will be not filled for the Detail data source, then, for each Master entries, all Detail entries will not be output;
4. Put two DataBands on a page of a report template;
5. Edit **DataBand1** and **DataBand2**:
   5.1 Align the **DataBands** vertically;
   5.2 Change the value of the required properties. For example, for the **DataBand1**, which is a **Master** component in the **Master-Detail** report, set the **Print If Detail Empty** property to **true**, if you want the **Master** entries be printed in any case, even if the **Detail** entries are not available. And for the **DataBand2**, which is a **Detail** component in the **Master-Detail** report, set the **CanShrink** property to **true**, if it is necessary for this band to be shrunk;
   5.3 Change the background color of the **DataBand**;
   5.4 If necessary, set **Borders** of the **DataBand**;

6. Specify data sources for **DataBands**, as well as assign the **Master** component. In our example, the **Master** component is the upper **DataBand1**, and hence indicate the **DataBand1** in the **Master Component** tab of the **Data Setup** dialog box of the lower **DataBand2** as the **Master** component;

7. Fill in the **Data Relation** property of the **DataBand**, which is the **Detail** component, in our case, this is the **DataBand2**:

   ![Data Band](image)

   | Data Relation | Categories | ... |

8. Put the text component with an expression. Where the expression is a reference to the data field. For example: the **DataBand1**, that is the **Master** component, put the text component with the **(Categories.CategoryName)** expression;

9. Edit text and text components located in the **DataBand**:
   9.1 Drag the text component to the required place in the **DataBand**;
   9.2 Align the text in a text component;
   9.3 Change the value of the required properties. For example to set the **Word Wrap** property to **true**, if you want the text be wrapped;
   9.4 Set **Borders** of a text component, if required.
   9.5 Change the border color.
10. Put the **Cross-Tab** component in the **DataBand**. In this case, the **Cross-Tab** component will be located on the **DataBand2**, that is the **Detail** component of the report.

![Image of Cross-Tab component in DataBand]

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**:  

   ![Data Source table with columns: Data Source, Products]

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 **DataSource** field to the **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, 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 **DataSource** field to the **Summary** field of the cross-table. For example, add the **UnitInStock** data column to the **Summary** field of the cross-table, entries in this data column will be summary entries in the cross-table;

15. Press the **OK** button 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

<table>
<thead>
<tr>
<th>Product Name</th>
<th>CategoryID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Mutton</td>
<td></td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>13</td>
</tr>
<tr>
<td>Boston Crab Meat</td>
<td>123</td>
</tr>
<tr>
<td>Camembert Pierrot</td>
<td>19</td>
</tr>
<tr>
<td>Camarvon Tigers</td>
<td>42</td>
</tr>
<tr>
<td>Chai</td>
<td>39</td>
</tr>
<tr>
<td>Chang</td>
<td>17</td>
</tr>
<tr>
<td>Chartreuse verte</td>
<td>69</td>
</tr>
<tr>
<td>Chef Anton’s Cajun Seasoning</td>
<td>53</td>
</tr>
<tr>
<td>Chef Anton’s Gumbo Mix</td>
<td></td>
</tr>
<tr>
<td>Chocolade</td>
<td>15</td>
</tr>
<tr>
<td>Côte de Blaye</td>
<td>17</td>
</tr>
<tr>
<td>Escargots de Bourgogne</td>
<td>62</td>
</tr>
<tr>
<td>Filo Mix</td>
<td>38</td>
</tr>
<tr>
<td>Fløtemysost</td>
<td>26</td>
</tr>
<tr>
<td>Geltost</td>
<td>112</td>
</tr>
<tr>
<td>Genen Shoyyu</td>
<td>39</td>
</tr>
<tr>
<td>Gnocchi di nonna Alice</td>
<td>21</td>
</tr>
<tr>
<td>Gorgonzola Telino</td>
<td></td>
</tr>
<tr>
<td>Grandma’s Boysenberry Spread</td>
<td>120</td>
</tr>
<tr>
<td>Gravad lax</td>
<td>11</td>
</tr>
<tr>
<td>Guaraná Fantástica</td>
<td>20</td>
</tr>
<tr>
<td>Gudbrandsdalsost</td>
<td>26</td>
</tr>
<tr>
<td>Guila Malacca</td>
<td>27</td>
</tr>
<tr>
<td>Gumbär Gummibärchen</td>
<td>15</td>
</tr>
<tr>
<td>Gustaf’s Knöckebröd</td>
<td>104</td>
</tr>
<tr>
<td>Ikura</td>
<td>31</td>
</tr>
<tr>
<td>Inlagd Sill</td>
<td>112</td>
</tr>
<tr>
<td>Ipoh Coffee</td>
<td>17</td>
</tr>
<tr>
<td>Jack’s New England Clam Chowder</td>
<td>85</td>
</tr>
<tr>
<td>Konbu</td>
<td>24</td>
</tr>
<tr>
<td>Lakkaliköri</td>
<td>57</td>
</tr>
<tr>
<td>Laughing Lumberjack Lager</td>
<td>52</td>
</tr>
<tr>
<td>Longlife tofu</td>
<td>4</td>
</tr>
<tr>
<td>Louisiana Fiery Hot Pepper Sauce</td>
<td>76</td>
</tr>
<tr>
<td>Louisiana Hot Spiced Otra</td>
<td>4</td>
</tr>
<tr>
<td>Manjimup Dried Apples</td>
<td>20</td>
</tr>
<tr>
<td>Mascarpone Pabolli</td>
<td>9</td>
</tr>
<tr>
<td>Mascotriu</td>
<td>10</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>29</td>
</tr>
<tr>
<td>Mozzarella di Giovanni</td>
<td>14</td>
</tr>
</tbody>
</table>
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:
<table>
<thead>
<tr>
<th>Products</th>
<th>Category ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Mutton</td>
<td>1</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>13</td>
</tr>
<tr>
<td>Boston Crab Meat</td>
<td>123</td>
</tr>
<tr>
<td>Camembert Pierrot</td>
<td>19</td>
</tr>
<tr>
<td>Caravron Tigers</td>
<td>42</td>
</tr>
<tr>
<td>Chai</td>
<td>39</td>
</tr>
<tr>
<td>Chang</td>
<td>17</td>
</tr>
<tr>
<td>Chartreuse verte</td>
<td>69</td>
</tr>
<tr>
<td>Chef Anton’s Cajun Seasoning</td>
<td>53</td>
</tr>
<tr>
<td>Chef Anton’s Gumbo Mix</td>
<td></td>
</tr>
<tr>
<td>Chocolade</td>
<td>15</td>
</tr>
<tr>
<td>Côte de Blaye</td>
<td>17</td>
</tr>
<tr>
<td>Escargots de Bourgogne</td>
<td>62</td>
</tr>
<tr>
<td>Filo Mix</td>
<td>38</td>
</tr>
<tr>
<td>Flotemysost</td>
<td>26</td>
</tr>
<tr>
<td>Geistost</td>
<td>112</td>
</tr>
<tr>
<td>Genen Shouyu</td>
<td>39</td>
</tr>
<tr>
<td>Gnocco di nonna Alice</td>
<td>21</td>
</tr>
<tr>
<td>Gorgonzola Telino</td>
<td></td>
</tr>
<tr>
<td>Grandma’s Boysenberry Spread</td>
<td>120</td>
</tr>
<tr>
<td>Gravad lax</td>
<td>11</td>
</tr>
<tr>
<td>Guaraná Fantástica</td>
<td>20</td>
</tr>
<tr>
<td>Gudbrandsdalscost</td>
<td>26</td>
</tr>
<tr>
<td>Gula Malacca</td>
<td>27</td>
</tr>
<tr>
<td>Gumbár Gummibärcchen</td>
<td>15</td>
</tr>
<tr>
<td>Gustaf’s knobkebrod</td>
<td>104</td>
</tr>
<tr>
<td>Ikurs</td>
<td>31</td>
</tr>
<tr>
<td>Inlagd Sill</td>
<td>112</td>
</tr>
<tr>
<td>Ipoh Coffee</td>
<td>17</td>
</tr>
<tr>
<td>Jack’s New England Clam Chowder</td>
<td>85</td>
</tr>
<tr>
<td>Konbu</td>
<td>24</td>
</tr>
<tr>
<td>Lakkalikööri</td>
<td>57</td>
</tr>
<tr>
<td>Laughing Lumberjack Lager</td>
<td>52</td>
</tr>
<tr>
<td>Longlife tofu</td>
<td>4</td>
</tr>
<tr>
<td>Louisiana Fiery Hot Pepper Sauce</td>
<td>76</td>
</tr>
<tr>
<td>Louisiana Hot Spiced Ostra</td>
<td>4</td>
</tr>
<tr>
<td>Manjimup Dried Apples</td>
<td>20</td>
</tr>
<tr>
<td>Mascarpone Fabbili</td>
<td>9</td>
</tr>
<tr>
<td>Maxitaku</td>
<td>10</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>29</td>
</tr>
<tr>
<td>Mozzarela di Giovanni</td>
<td>14</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Products</th>
<th>CategoryID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductName</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>Alice Mutton</td>
<td></td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>10</td>
</tr>
<tr>
<td>Boston Crab Meat</td>
<td>123</td>
</tr>
<tr>
<td>Camembert Pierrot</td>
<td>19</td>
</tr>
<tr>
<td>Camaron Tigers</td>
<td>42</td>
</tr>
<tr>
<td>Chai</td>
<td>39</td>
</tr>
<tr>
<td>Chang</td>
<td>17</td>
</tr>
<tr>
<td>Chartreuse verte</td>
<td>69</td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>53</td>
</tr>
<tr>
<td>Chef Anton's Gumbo Mix</td>
<td></td>
</tr>
<tr>
<td>Chocolade</td>
<td>15</td>
</tr>
<tr>
<td>Côte de Blaye</td>
<td>17</td>
</tr>
<tr>
<td>Escargots de Bourgogne</td>
<td>62</td>
</tr>
<tr>
<td>Filo Mix</td>
<td>38</td>
</tr>
<tr>
<td>Floramysost</td>
<td>26</td>
</tr>
<tr>
<td>Geistost</td>
<td>112</td>
</tr>
<tr>
<td>Genen Shoyyu</td>
<td>39</td>
</tr>
<tr>
<td>Gnocchi di nonna Alice</td>
<td>21</td>
</tr>
<tr>
<td>Gorgonzola Telino</td>
<td></td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>120</td>
</tr>
<tr>
<td>Gravad lax</td>
<td>11</td>
</tr>
<tr>
<td>Guaraná Fantástica</td>
<td>20</td>
</tr>
<tr>
<td>Gudbrandsdalacost</td>
<td>26</td>
</tr>
<tr>
<td>Guia Malacca</td>
<td>27</td>
</tr>
<tr>
<td>Gumbär Gummiñochen</td>
<td>15</td>
</tr>
<tr>
<td>Gustaf's knobbebrod</td>
<td>104</td>
</tr>
<tr>
<td>Ikuru</td>
<td>31</td>
</tr>
<tr>
<td>Inlagd Sill</td>
<td>112</td>
</tr>
<tr>
<td>Ippoh Coffee</td>
<td>17</td>
</tr>
<tr>
<td>Jack's New England Clam Chowder</td>
<td>85</td>
</tr>
<tr>
<td>Konbu</td>
<td>24</td>
</tr>
<tr>
<td>Lakkalikööri</td>
<td>57</td>
</tr>
<tr>
<td>Laughing Lumberjack Lager</td>
<td>52</td>
</tr>
<tr>
<td>Longlife tofu</td>
<td>4</td>
</tr>
<tr>
<td>Louisiana Fiery Hot Pepper Sauce</td>
<td>76</td>
</tr>
<tr>
<td>Louisiana Hot Spiced Otra</td>
<td>4</td>
</tr>
<tr>
<td>Manjimup Dried Apples</td>
<td>20</td>
</tr>
<tr>
<td>Mascarpone Fabboli</td>
<td>9</td>
</tr>
<tr>
<td>Makiteku</td>
<td>10</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>29</td>
</tr>
<tr>
<td>Mozzarella di Giovanni</td>
<td>14</td>
</tr>
</tbody>
</table>
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 **KeyDataColumn** property, 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](image)

9. Set the **MasterKeyDataColumn** property, 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](image)

10. Set the **Indent** property, 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](image)

11. Set the **ParentValue** property, 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](image)

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.
13. Go back to the report template;
14. If necessary, add other bands into the report template, for example, **HeaderBand**;
15. Edit this band:
   15.1. Align the **HeaderBand** vertically;
   15.2. Set properties of the **HeaderBand**, if necessary;
   15.3. Set the background color of the **HeaderBand**;
   15.4. If necessary, set the **Borders**;
   15.5. Change the border color.

16. Put text components with the expressions. Where expressions in text components in the **HeaderBand** will be the data headers;
17. Edit text and text components:
   17.1. Drag the text component to the required place in the band;
   17.2. Set the font settings: size, style, color;
   17.3. Align the text component vertically and horizontally;
   17.4. Set the background color of the text component;
   17.5. Align the text in a text component;
   17.6. Set the value of properties of a text component, if necessary;
   17.7. If necessary, set **Borders** of a text component;
   17.8. Set the border color.
18. Click the **Preview** button or call **Viewer**, using the **Preview** menu item. After rendering a report, all references to data sources will be replaced with data from these sources:

![Employee City Region Table](image)

**Adding styles**

1. Go back to the report template;
2. Select component. In our case this is the text component;
3. 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, \texttt{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 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 video file.

Do the following steps to create a sub-report:

1. Run the designer;
2. Connect data:
   2.1. Create New Connection;
   2.2. Create New Data Source;
3. Add the Sub-Report component to a report on a page of the report template:

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

5. Go to the sub-report page;
6. Add to the **DataBand** to the sub-report page;

![DataBand](image)

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:

   ![Data Source](image)

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.

![Text and TextBoxes](image)

11. Click the **Preview** button or call **Viewer**, using the **Preview** menu item to see how the report will look like.
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.
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:
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 **Data Bands**:
   7.1. Align the **Data Bands** vertically;
   7.2. Change values of properties of the **Data Bands**.
   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 **Data Band** using the **Data Source** property. For example, set the **Customers** data source for the **Data Band1**, and the **Products** data source for the **Data Band2**:

   ![Data Bands with different data sources]

9. Put text components with expressions in the **Data Bands**. Where an expression is a reference to a data field. For example, put the following expressions to the **Data Band1**: `{Customers.CompanyName}` and `{Customers.City}`, put the following expressions to the **Data Band2**: `{Products.ProductName}` and `{Products.UnitPrice}`;

10. Edit **Text** and **TextBoxes**:
    10.1. Drag the text component to the required place in the **Data Band**;
    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:

   ![Preview of the report with data and text components]
As can be seen from the picture above, the report generator rendered the report, which was located in the nested page and placed it on the report page but not in the Sub-Report component.

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

15. Put a text component with expression where the expression of the text component in the **HeaderBand** will be the page title.
16. Edit the text component:
16.1. Drag the text component to the required place in the band;
16.2. Set the text font: size, style, color;
16.3. Align text component vertically and horizontally;
16.4. Set the background color of the text component;
16.5. Align text in the component;
16.6. Set values of the properties of text components;
16.7. Set **Borders** of a text component.
16.8. Set the border color.

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

**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.

![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:
Do the following steps to create a simple list report:

1. Run the designer;
2. Connect data:
   2.1. Create **New Connection**;
   2.2. Create **New Data Source**;
3. Put the **DataBand** on a page of a report template.
4. Edit **DataBand**:
   4.1. Align the **DataBand** by height;
4.2. Change values of band properties. For example, set the Can Break property to true, if you wish the data band to be broken;
4.3. Change the DataBand background color;
4.4. Enable Borders for the DataBand, if required;
4.5. Change the border color.

5. Define the data source for the DataBand using the Data Source property. For example, define the Categories data source for the DataBand:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Categories</th>
</tr>
</thead>
</table>

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:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Customers</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>Products</td>
<td>...</td>
</tr>
</tbody>
</table>

12. Put text components with expressions in the DataBands. Where an expression is a reference to a data field. For example, put the following expressions to the DataBand1: {Customers.CompanyName} and {Customers.City}. put the following expressions to the DataBand2: {Products.ProductName} and {Products.UnitPrice};

13. Edit Text and TextBoxes:
   13.1. Drag the text component to the required place in the DataBand;
   13.2. Set the text font: size, style, color;
   13.3. Align text component vertically and horizontally;
   13.4. Set the background color of the text component;
   13.5. Align text in the component;
   13.6. Set values of the properties of text components. For example to set the Word Wrap property to true, if you want the text to be wrapped;
   13.7. Set Borders of a text component.
   13.8. Set the border color.

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

<table>
<thead>
<tr>
<th>Alfreds Futtkiste</th>
<th>Berlin</th>
<th>Chai</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ana Trujillo Empanados y helados</td>
<td>México D.F.</td>
<td>Chang</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>México D.F.</td>
<td>Aniseed Syrup</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>London</td>
<td>Chef Anton’s Cajun Seasoning</td>
</tr>
<tr>
<td>Berglunds snäbbkös</td>
<td>Luleå</td>
<td>Chef Anton’s GumboMix</td>
</tr>
<tr>
<td>Blauer See Delikatessen</td>
<td>Mannheim</td>
<td>Grandma’s Boysenberry Spread</td>
</tr>
<tr>
<td>Blondes ddsi père et fils</td>
<td>Strasbourg</td>
<td>Uncle Bob’s Organic Dried Pears</td>
</tr>
<tr>
<td>Bólido Comidas preparadas</td>
<td>Madrid</td>
<td>Northwoods Cranberry Sauce</td>
</tr>
<tr>
<td>Bon app’</td>
<td>Marseille</td>
<td>Mishi Kobe Niku</td>
</tr>
<tr>
<td>Bottom Dollar Markets</td>
<td>Taawassen</td>
<td>Ikura</td>
</tr>
<tr>
<td>B’s Beverages</td>
<td>London</td>
<td>Queso Cabrales</td>
</tr>
<tr>
<td>Cactus Comidas para llevar</td>
<td>Buenos Aires</td>
<td>Queso Manchego La Pastora</td>
</tr>
<tr>
<td>Centro comercial Moctezuma</td>
<td>México D.F.</td>
<td>Konbu</td>
</tr>
<tr>
<td>Chop-suey Chinese</td>
<td>Bern</td>
<td>Tofu</td>
</tr>
<tr>
<td>Comércio Mineiro</td>
<td>São Paulo</td>
<td>Geren Shouyu</td>
</tr>
<tr>
<td>Consolidated Holdings</td>
<td>London</td>
<td>Pavlova</td>
</tr>
<tr>
<td>Dronkbult Delikatessen</td>
<td>Aachen</td>
<td>Alice Mutton</td>
</tr>
<tr>
<td>Du monde entier</td>
<td>Nantes</td>
<td>Camaron Tigers</td>
</tr>
<tr>
<td>Eastern Connection</td>
<td>London</td>
<td>Teatime Chocolate Biscuits</td>
</tr>
<tr>
<td>Ernst Handel</td>
<td>Graz</td>
<td>Sir Rodney’s Marmalade</td>
</tr>
<tr>
<td>Família Arquibaldo</td>
<td>São Paulo</td>
<td>Sir Rodney’s Scones</td>
</tr>
<tr>
<td>FISSA Fabricon Inter Saichicas S.A.</td>
<td>Madrid</td>
<td>Gustaf’s Knäckebröd</td>
</tr>
<tr>
<td>Folies Gourmandes</td>
<td>Lille</td>
<td>Tumbbröd</td>
</tr>
<tr>
<td>Folk och fäHR</td>
<td>Bränke</td>
<td>Guaraná Fantástica</td>
</tr>
<tr>
<td>Frankenversand</td>
<td>München</td>
<td>NuLuCaNuß-Nougat-Cremes</td>
</tr>
<tr>
<td>France restoration</td>
<td>Nantes</td>
<td>Gumbär Gummibärchen</td>
</tr>
<tr>
<td>Franchi S.p.A.</td>
<td>Torino</td>
<td>Scho Obl Schokolade</td>
</tr>
</tbody>
</table>
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:
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**.

![Style Designer](image)

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:
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**:

    ![Data Source](image)

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, 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:
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:
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:

![Report Sample](image)

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 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:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Categories</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>Products</td>
<td>...</td>
</tr>
</tbody>
</table>

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.UnitPrice}** 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:

![DataBand Example](image)

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:
14. Go back to the report template;
15. Add other bands to a report template, for example, add the HeaderBand to the Page2 of a report;
16. Edit the band:
   16.1. Align it by height;
   16.2. Change values of properties, if required;
   16.3. Change the background of the band;
   16.4. Enable Borders, if required;
   16.5. Set the border color.

17. Put a text component with an expression in this band. The expression in the text component is a header in the HeaderBand.
18. Edit text and text components:
   18.1. Drag and drop the text component in the band;
   18.2. Change font options: size, type, color;
   18.3. Align text component by height and width;
18.4. Change the background of the text component;
18.5. Align text in the text component;
18.6. Change values of text component properties, if required;
18.7. Enable **Borders** of the text component, if required;
18.8. Set the border color.

19. Click the **Preview** button or invoke the **Viewer**, 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, shows the ratio of detailed data to the master **Condiments** entry:

Adding styles

1. Go back to the report template;
2. Select the **DataBand**;
3. Change values of **Even style** and **Odd style** properties. If values of these properties are not set, then select the **Edit Styles** in the list of values of these properties and, using **Style Designer**, create a new style. The picture below shows the **Style Designer**.

![Style Designer](image)

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, shows the ratio of detailed data to the **Confections** master entry with different styles even/odd rows of the **DataBand**:
2.17 Drill-Down Report Using External Report

For better understanding this step-by-step instruction, please watch the video file.

Drill-Down report using external report is an interactive report in what detailed data are placed in an external report and the relationship between master and detailed data in reports is organized using the Interaction.Drill-Down Report property. Follow the steps below 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:

   ![Data Source](image)

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:
9. Go back to the report template;
10. Add other bands to a report template, for example, add the **HeaderBand** to the report page;
11. Edit the band:
   11.1. Align it by height;
   11.2. Change values of properties, if required;
   11.3. Change the background of the band;
   11.4. Enable **Borders**, if required;
   11.5. Set the border color.
12. Put a text component with an expression in this band. The expression in the text component is a header in the **HeaderBand**.

13. Edit text and text components:
   13.1. Drag and drop the text component in the band;
   13.2. Change font options: size, type, color;
   13.3. Align text component by height and width;
   13.4. Change the background of the text component;
   13.5. Align text in the text component;
   13.6. Change values of text component properties, if required;
   13.7. Enable **Borders** of the text component, if required;
   13.8. Set the border color.

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

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:
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, 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:
2.18 Report without Bands

For better understanding this step-by-step instruction, please watch the video file.

If it is necessary to display data from only one entry of the data source or data from variables or other data sources that are not lists, the report can be created without the bands. In this case, components are placed directly on a report page.

1. Run the designer;
2. Connect the data:
   2.1. Create a New Connection;
   2.2. Create a New Data Source;
3. Put the Image component with the image on a page;
4. Edit the Image component and an image:
   4.1. Drag and drop the Image component on the report page;
   4.2. Align the Image component by height and width;
   4.3. Set the background color of the Image component;
   4.4. Align the image in the component;
   4.5. Change values of the properties of the Image component. For example to set
the **Print** property to **true**, if you want this component be printed;

4.6. If necessary, set **Borders** of the **Image** component;

4.7. Set the border color.

5. Put **TextBoxes** with the text on a page. In this report, put 15 Text components. The **TextBox1** contains the **{Time}** system variable, which will display the current time and date. **2-8 TextBoxes** contain the row names in the address box, and **9-15 TextBoxes** will include links to data sources;

6. Edit text and text components:

   6.1. Drag and drop the text component in the band;
   6.2. Change font options: size, type, color;
   6.3. Align text component by height and width;
   6.4. Change the background of the text component;
   6.5. Align text in the text component;
   6.6. Change values of text component properties, if required;
   6.7. Enable **Borders** of the text component, if required;
   6.8. Set the border color.

7. Click the **Preview** button or invoke the **Viewer**, 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:
   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 `Data Bands`. 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 `Data Bands`;
   5.2. Change parameters of the text font: size, type, color;
   5.3. Align the text component by width and height;
   5.4. Change the background of the text component;
   5.5. Align text in the text component;
   5.6. Change the value of properties of the text component. For example, set the `Word Wrap` property to `true`, if you need a text to be wrapped;
   5.7. Enable `Borders` for the text component, if required.
   5.8. Change the border color.

6. Add other bands to the report template, for example, the `HeaderBand`;
7. Edit this bands:
   7.1. Align it by height;
   7.2. Change values of properties, if required;
   7.3. Change the background of bands;
   7.4. Enable `Borders`, if required;
   7.5. Set the border color.
8. Put text components with expressions in the 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.
Report on Employees

City: Berlin
Address: Obere Str. 57
Country: Germany
Phone: 030-0074321
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.
Report on Employees

City: Berlin
Address: Obere Str. 57
County: Germany
Phone: 030-0074321
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. 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.

![DataBand](image)

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:

   ![Data Source](image)

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.

![Text and TextBox](image)

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.

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.

![Image](image.png)

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.

![Image](image.png)

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

   ![DataBand](image)

   **Data Source** | **Employees** | ...

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.

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.
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**: 

```
<table>
<thead>
<tr>
<th>Nancy</th>
<th>Seattle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew</td>
<td>Tacoma</td>
</tr>
<tr>
<td>Jane</td>
<td>Kirkland</td>
</tr>
<tr>
<td>Margaret</td>
<td>Redmond</td>
</tr>
<tr>
<td>Steve</td>
<td>London</td>
</tr>
<tr>
<td>Michael</td>
<td>London</td>
</tr>
<tr>
<td>Robert</td>
<td>London</td>
</tr>
<tr>
<td>Laura</td>
<td>Seattle</td>
</tr>
<tr>
<td>Anne</td>
<td>London</td>
</tr>
</tbody>
</table>
```
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.

<table>
<thead>
<tr>
<th>FirstName</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nancy</td>
<td>Seattle</td>
</tr>
<tr>
<td>Andrew</td>
<td>Tacoma</td>
</tr>
<tr>
<td>Janer</td>
<td>Kirkland</td>
</tr>
<tr>
<td>Margaret</td>
<td>Redmond</td>
</tr>
<tr>
<td>Steven</td>
<td>London</td>
</tr>
<tr>
<td>Michael</td>
<td>London</td>
</tr>
<tr>
<td>Robert</td>
<td>London</td>
</tr>
<tr>
<td>Laura</td>
<td>Seattle</td>
</tr>
<tr>
<td>Anne</td>
<td>London</td>
</tr>
</tbody>
</table>

### 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. 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.
9. Go back to the report template.
10. Add the **Shape** component to a report template in the **DataBand** and edit it:
   10.1. Drag and drop the **Shape** component on the page;
   10.2. Change the type of a shape using the **Shape Type** property. Set the **Shape Type** property to **Complex Arrow**;
   10.3. Stretch the **Shape** component horizontally and vertically;
   10.4. Change the value of other properties. For example, set the **Grow to Height** property to **true**.

The picture below shows a report template with the **Shape** component placed on the report page:

11. Click the **Preview** button or invoke the **Viewer**, clicking the **Preview** menu item.
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.
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.
## 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. To design a report with cross-primitives, follow the steps below:

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

<table>
<thead>
<tr>
<th>FirstName</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nancy</td>
<td>Seattle</td>
</tr>
<tr>
<td>Andrew</td>
<td>Tacoma</td>
</tr>
<tr>
<td>Janet</td>
<td>Kirkland</td>
</tr>
<tr>
<td>Margaret</td>
<td>Redmond</td>
</tr>
<tr>
<td>Steven</td>
<td>London</td>
</tr>
<tr>
<td>Michael</td>
<td>London</td>
</tr>
<tr>
<td>Robert</td>
<td>London</td>
</tr>
<tr>
<td>Leura</td>
<td>Seattle</td>
</tr>
<tr>
<td>Anne</td>
<td>London</td>
</tr>
</tbody>
</table>
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 **GroupBoxHeaderBand** and **GroupBoxFooterBand**;
   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.
A Drill-Down report is an interactive report in which 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.

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:
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.
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 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.

![DataBand example](image)

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:

   ![Data Source](image)

   - Data Source: Products

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}`  
   - `{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.

![Text and TextBox example](image)

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.

9. Go back to the report template;
10. If needed, add other bands to the report template, for example, ReportTitleBand and ReportSummaryBand;
11. Edit these bands:
   11.1. Align them by height;
   11.2. Change values of properties, if required;
   11.3. Change the background of bands;
   11.4. Enable Borders, if required;
   11.5. Set the border color.

12. Put text components with expressions in the these bands. The expression in the text component is a title in the ReportTitleBand, and a summary in the ReportSummaryBand.
13. Edit text and text components:
   13.1. Drag and drop the text component in the band;
   13.2. Change font options: size, type, color;
   13.3. Align text component by height and width;
   13.4. Change the background of the text component;
   13.5. Align text in the text component;
   13.6. Change values of text component properties, if required;
13.7. Enable **Borders** of the text component, if required;
13.8. Set the border color.

14. Click the **Preview** button or invoke the **Viewer**, 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.
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 Band1.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.
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, 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

<table>
<thead>
<tr>
<th>Product Name</th>
<th>QuantityPerUnit</th>
<th>UnitsInStock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice’s Mutton</td>
<td>2 - 1 kg tins</td>
<td>0</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>12 - 500 ml bottles</td>
<td>12</td>
</tr>
<tr>
<td>Boston Crab Meat</td>
<td>24 - 4 oz tins</td>
<td>123</td>
</tr>
<tr>
<td>Camembert Pierrot</td>
<td>15 - 300 g rounds</td>
<td>19</td>
</tr>
<tr>
<td>Canned Tigerfish</td>
<td>16 kg pkg.</td>
<td>42</td>
</tr>
<tr>
<td>Chai</td>
<td>10 boxes x 20 bags</td>
<td>39</td>
</tr>
<tr>
<td>Chang</td>
<td>24 - 12 oz bottles</td>
<td>17</td>
</tr>
<tr>
<td>Chairaeuse verre</td>
<td>750 cc per bottle</td>
<td>89</td>
</tr>
<tr>
<td>Chef Anton’s Cajun Seasoning</td>
<td>48 - 6 oz jars</td>
<td>63</td>
</tr>
<tr>
<td>Chef Anton’s Gumbo Mix</td>
<td>36 boxes</td>
<td>0</td>
</tr>
<tr>
<td>Chocolade</td>
<td>12 pkg.s.</td>
<td>15</td>
</tr>
<tr>
<td>Coq de Braye</td>
<td>12 - 75 cl bottles</td>
<td>17</td>
</tr>
<tr>
<td>Escargots de Bourgogne</td>
<td>24 pieces</td>
<td>82</td>
</tr>
<tr>
<td>Filo Mix</td>
<td>15 - 2 kg boxes</td>
<td>26</td>
</tr>
<tr>
<td>Fletimysost</td>
<td>10 - 500 g pkg.s.</td>
<td>26</td>
</tr>
<tr>
<td>Gaiost</td>
<td>500 g</td>
<td>112</td>
</tr>
<tr>
<td>Genes Shoyu</td>
<td>24 - 250 ml bottles</td>
<td>39</td>
</tr>
<tr>
<td>Gnocchi di nona Alice</td>
<td>24 - 250 g pkg.s.</td>
<td>21</td>
</tr>
<tr>
<td>Gorgonzola Talno</td>
<td>12 - 100 g pkg.s.</td>
<td>0</td>
</tr>
<tr>
<td>Grandma’s Boysenberry Spread</td>
<td>12 - 8 oz jars</td>
<td>120</td>
</tr>
<tr>
<td>Grana Padano</td>
<td>12 - 600 g pkg.s.</td>
<td>11</td>
</tr>
<tr>
<td>Guarana Fantastica</td>
<td>12 - 385 ml cans</td>
<td>20</td>
</tr>
<tr>
<td>Gutbrandesdalsost</td>
<td>10 kg pkg.</td>
<td>26</td>
</tr>
<tr>
<td>Guila Malacca</td>
<td>20 - 2 kg bags</td>
<td>27</td>
</tr>
<tr>
<td>Gumßel Gummibärchen</td>
<td>100 - 250 g bags</td>
<td>15</td>
</tr>
<tr>
<td>Swedish Kolossbröd</td>
<td>24 - 500 g pkg.s.</td>
<td>104</td>
</tr>
<tr>
<td>Kurz</td>
<td>12 - 200 ml jars</td>
<td>31</td>
</tr>
<tr>
<td>Majd Sill</td>
<td>24 - 250 g jars</td>
<td>112</td>
</tr>
<tr>
<td>Irish Coffee</td>
<td>16 - 500 g tins</td>
<td>17</td>
</tr>
<tr>
<td>Jack’s New England Clam Chowder</td>
<td>12 - 12 oz cans</td>
<td>95</td>
</tr>
<tr>
<td>Kombu</td>
<td>2kg box</td>
<td>24</td>
</tr>
<tr>
<td>Lakshmi Kori</td>
<td>500 ml</td>
<td>87</td>
</tr>
<tr>
<td>Laughing Lumberjack Lager</td>
<td>24 - 12 oz bottles</td>
<td>52</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Quantity Per Unit</th>
<th>Units in Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Mutton</td>
<td>20 - 1 kg tins</td>
<td>9</td>
</tr>
<tr>
<td>Amoosed Syrup</td>
<td>12 - 600 ml bottles</td>
<td>13</td>
</tr>
<tr>
<td>Boston Crab Meat</td>
<td>24 - 4 oz tins</td>
<td>123</td>
</tr>
<tr>
<td>Camembert Pierrot</td>
<td>15 - 300 g rounds</td>
<td>19</td>
</tr>
<tr>
<td>Carnivore Tigers</td>
<td>15 kg pkg</td>
<td>42</td>
</tr>
<tr>
<td>Chai</td>
<td>10 boxes x 20 bags</td>
<td>59</td>
</tr>
<tr>
<td>Charg</td>
<td>24 - 12 oz bottles</td>
<td>17</td>
</tr>
<tr>
<td>Chanreuse verte</td>
<td>750 cc per bottle</td>
<td>89</td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>48 - 8 oz jars</td>
<td>53</td>
</tr>
<tr>
<td>Chef Anton's Gumbo Mix</td>
<td>36 boxes</td>
<td>0</td>
</tr>
<tr>
<td>Chocolade</td>
<td>10 pkgs</td>
<td>15</td>
</tr>
<tr>
<td>Coque de Blaye</td>
<td>12 - 75 cl bottles</td>
<td>17</td>
</tr>
<tr>
<td>Escargots de Bourgogne</td>
<td>24 pieces</td>
<td>92</td>
</tr>
<tr>
<td>Fino Mix</td>
<td>15 - 2 kg boxes</td>
<td>30</td>
</tr>
<tr>
<td>Filled Yeast</td>
<td>10 - 500 g pkgs</td>
<td>26</td>
</tr>
<tr>
<td>Gefilte</td>
<td>500 g</td>
<td>112</td>
</tr>
<tr>
<td>Genin Shouyu</td>
<td>24 - 250 ml bottles</td>
<td>39</td>
</tr>
<tr>
<td>Genovese di nona Alice</td>
<td>24 - 250 g pkgs</td>
<td>21</td>
</tr>
<tr>
<td>Gorgonzola Talno</td>
<td>12 - 100 g pkgs</td>
<td>0</td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>12 - 8 oz jars</td>
<td>120</td>
</tr>
<tr>
<td>Grassi Truffle</td>
<td>12 - 500 g pkgs</td>
<td>11</td>
</tr>
<tr>
<td>Guarana Fantastica</td>
<td>12 - 355 ml cans</td>
<td>20</td>
</tr>
<tr>
<td>Grandpa's Gouda</td>
<td>10 kg pkg</td>
<td>26</td>
</tr>
<tr>
<td>Guia Malacca</td>
<td>20 - 2 kg bags</td>
<td>27</td>
</tr>
<tr>
<td>Gumbir Gummibärchen</td>
<td>100 - 250 g bags</td>
<td>15</td>
</tr>
<tr>
<td>Guineda's Kondensbrei</td>
<td>12 - 500 g pkgs</td>
<td>104</td>
</tr>
<tr>
<td>Helmi</td>
<td>12 - 200 ml jars</td>
<td>31</td>
</tr>
<tr>
<td>Ichac Stil</td>
<td>24 - 250 g jars</td>
<td>112</td>
</tr>
<tr>
<td>Ion Coffee</td>
<td>15 - 500 g tins</td>
<td>17</td>
</tr>
<tr>
<td>Jack's New England Clam Chowder</td>
<td>12 - 12 oz cans</td>
<td>85</td>
</tr>
<tr>
<td>Kontu</td>
<td>2 kg box</td>
<td>24</td>
</tr>
<tr>
<td>Lakkiskorol</td>
<td>500 ml</td>
<td>97</td>
</tr>
<tr>
<td>Laughing Limericjack Lager</td>
<td>24 - 12 oz bottles</td>
<td>52</td>
</tr>
</tbody>
</table>
Report With Dynamic Collapsing in Preview

The report with dynamic collapsing is an interactive report in which items can collapse/expand its contents by clicking the title of the block. 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`:

   ![Collapsing Enabled](image)

8. Change the value of the `Interaction.Collapsed`. In this case, set this property to `{GroupLine!=1}`, all the groups except the first one will be collapsed:

   ![Collapsed](image)

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:
To expand or collapse the group, select the **GroupHeaderBand** in the rendered report. If you want to collapse the group together with 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:
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.
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:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Customers</th>
<th>...</th>
</tr>
</thead>
</table>

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.

<table>
<thead>
<tr>
<th>CompanyName</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahrea Futterkiste</td>
<td>Berlin</td>
<td>Germany</td>
</tr>
<tr>
<td></td>
<td>Contact:</td>
<td>Maria Andes</td>
</tr>
<tr>
<td></td>
<td>Phone:</td>
<td>000-007421</td>
</tr>
<tr>
<td></td>
<td>Fax:</td>
<td>000-0076845</td>
</tr>
<tr>
<td>Ana Trujillo Empanadados y Harinas</td>
<td>México D.F.</td>
<td>México</td>
</tr>
<tr>
<td></td>
<td>Contact:</td>
<td>Ana Trujillo</td>
</tr>
<tr>
<td></td>
<td>Phone:</td>
<td>555-4729</td>
</tr>
<tr>
<td></td>
<td>Fax:</td>
<td>555-0745</td>
</tr>
<tr>
<td>Antonio Moreno Taqueria</td>
<td>México D.F.</td>
<td>México</td>
</tr>
<tr>
<td></td>
<td>Contact:</td>
<td>Antonio Moreno</td>
</tr>
<tr>
<td></td>
<td>Phone:</td>
<td>555-0992</td>
</tr>
<tr>
<td></td>
<td>Fax:</td>
<td></td>
</tr>
<tr>
<td>Around the Horn</td>
<td>London</td>
<td>UK</td>
</tr>
<tr>
<td></td>
<td>Contact:</td>
<td>Thomas Hardy</td>
</tr>
<tr>
<td></td>
<td>Phone:</td>
<td>0711-555-7753</td>
</tr>
<tr>
<td></td>
<td>Fax:</td>
<td>0711-555-6750</td>
</tr>
<tr>
<td>Berglunds snabklip</td>
<td>Luleå</td>
<td>Sweden</td>
</tr>
<tr>
<td></td>
<td>Contact:</td>
<td>Christian Berglund</td>
</tr>
<tr>
<td></td>
<td>Phone:</td>
<td>0212-12 24 68</td>
</tr>
<tr>
<td></td>
<td>Fax:</td>
<td>0212-12 24 67</td>
</tr>
<tr>
<td>Blauer See Delikatessen</td>
<td>Mannheim</td>
<td>Germany</td>
</tr>
<tr>
<td></td>
<td>Contact:</td>
<td>Henne Moos</td>
</tr>
<tr>
<td></td>
<td>Phone:</td>
<td>0241-0440</td>
</tr>
<tr>
<td></td>
<td>Fax:</td>
<td>0241-09924</td>
</tr>
<tr>
<td>Biondeodal père et fils</td>
<td>Strasbourg</td>
<td>France</td>
</tr>
<tr>
<td></td>
<td>Contact:</td>
<td>Patricio Citeaux</td>
</tr>
<tr>
<td></td>
<td>Phone:</td>
<td>0560-15 31</td>
</tr>
<tr>
<td></td>
<td>Fax:</td>
<td>0560-15 32</td>
</tr>
<tr>
<td>Bolido Comidas preparadas</td>
<td>Madrid</td>
<td>Spain</td>
</tr>
<tr>
<td></td>
<td>Contact:</td>
<td>Martin Sommer</td>
</tr>
<tr>
<td></td>
<td>Phone:</td>
<td>915 555 22 82</td>
</tr>
<tr>
<td></td>
<td>Fax:</td>
<td>915 555 91 98</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Data Relation</th>
<th>Categories</th>
</tr>
</thead>
</table>

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 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 Data Bands on a page of the report template

6. Edit Data Band 1 and Data Band 2:
   6.1. Align them by height;
   6.2. Change values of required properties. For example, if to set the PrintIfDetailEmpty property of the Data Band 1 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 Data Band 2 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 Data Bands;
   6.4. Enable Borders of the band, if required;

7. Specify the data sources for Data Bands, as well as assign the Master component. In this case, the Master component is the upper Data Band 1, and hence in the Data Setup window the lower Data Band 2 on the Wizard tab in the Master Component should indicate Data Band 1 as a Master component. Indicate the data sources for Data Bands using the Data Source property:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Categories</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>Products</td>
<td>...</td>
</tr>
</tbody>
</table>

8. Fill the Data Relation property of the Data Band 2, which is the Detail component:

| Data Source | Categories | ... |

9. Put text components with expressions on Data Bands. For example: on the Data Band 1, which is the Master component, we put the text component with the following expression: \{Categories.CategoryName\}, and on the Data Band 2, 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 **Data Bands** 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:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Categories</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>Products</td>
<td>...</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Beverages</th>
<th>Dairy Products</th>
<th>Grains/Cereals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>Queso Cabrillas</td>
<td>Gusta's Chocolate</td>
</tr>
<tr>
<td>Charo</td>
<td>Queso Manchego La Pastora</td>
<td>Turmeric</td>
</tr>
<tr>
<td>Guarana Fantatica</td>
<td>Gorgonzola Talloso</td>
<td>Singapore Hokkien</td>
</tr>
<tr>
<td>Sambuca Ale</td>
<td>Mascarpone Faggio</td>
<td>Fried Fish</td>
</tr>
<tr>
<td>Steeave Stout</td>
<td>Sabot</td>
<td>Wimmers Jüt Semmelknödel</td>
</tr>
<tr>
<td>Chai Tea</td>
<td>Raclette Courdavault</td>
<td></td>
</tr>
<tr>
<td>Chai Tea</td>
<td>Semmelhart Pfanni</td>
<td></td>
</tr>
<tr>
<td>Chai Tea</td>
<td>Subbrandtskål</td>
<td></td>
</tr>
<tr>
<td>Chai Tea</td>
<td>Potermedel</td>
<td></td>
</tr>
<tr>
<td>Chai Tea</td>
<td>Mazzarela di casavardini</td>
<td></td>
</tr>
<tr>
<td>Condiments</td>
<td>Grains/Cereals</td>
<td></td>
</tr>
<tr>
<td>Aired Syrup</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChefAnton's Cajun Seasoning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChefAnton's Gumbo Mix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Northcote Cranberry Sauce</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gran Nan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guia Macaack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strog o'Gable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strog o'Gable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Louisiana Piery Hot Pepper Sau</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Louisiana Hot Spiced Cider</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original Frankfurter Pilsner</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confections</td>
<td>Meat/Poultry</td>
<td></td>
</tr>
<tr>
<td>Devinsa</td>
<td>Monk's Lobor Nows</td>
<td></td>
</tr>
<tr>
<td>Traditional Chocolate Biscuits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sir Rodney's Marmalade</td>
<td>Alice Mutton</td>
<td></td>
</tr>
<tr>
<td>Sir Rodney's Scones</td>
<td>Thörlinger Raostbratwurst</td>
<td></td>
</tr>
<tr>
<td>NuNuCa Nuß-Douglas-Creme</td>
<td>Herr Passat</td>
<td></td>
</tr>
<tr>
<td>Gummär Gummibärchen</td>
<td>Touriste</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pita chinos</td>
<td></td>
</tr>
</tbody>
</table>
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, **HeaderView**;
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
- Chappi
- Guarana Fantastica
- Sanquater Ale
- Steeave Stout
- Caffeine Rye
- Château verre
- Java Coffee
- Laughing Lumberjack Lager
- Outback Lager
- Robin's Kölster
- Lakeland

### Dairy Products
- Queso Carlota
- Queso Manchego La Pastora
- Sorong Co Talco
- Mascarpone Frico
- Sahno
- Raclette Courdavault
- Semolina Blanquet
- Sylva-detae
- Newtons
- Macaronel de camiones

### Condiments
- Arrested Serum
- Chef Anton's Cajun Seasoning
- Chef Anton's Gumbo Mix
- Grandma's Blackberry Spread
- Northwoods Cranberry Sauce
- Canh Shmoo
- Guilt-free
- Soup of Paradise
- Vege-spread
- Louisiana Fiery Hot Pepper Sauce
- Louisiana Hot Spiced Okra
- Original Franks Hot Dog Sauce

### Confections
- Devils
- French Chocolate Biscuits
- Sir Rodney's Marmalade
- Sir Rodney's Scones
- Muller's Nut-Coffee-Creme
- Gumbier Gummibärchen
- Schooqi Schokolade
- Zupaia koken
- Chocolate
- Maitake
- Välkommen suhka
- Tarte au sucre
- Scottish Longbreads

### Grains/Cereals
- Sultana's Chocolates
- Tofu
- Singapore Hokiin Fried Nuts
- Too Mix
- Grünkorn de norna Alisa
- Rapido Artigo
- Wimmers Jute Sammekarolle

### Meat/Poultry
- Wutsi Lober Nuts
- Alice Hutton
- Tylkiner Rostbratwurst
- Himl Hausas
- Tourke
- Pita chions
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.
<table>
<thead>
<tr>
<th>Beverages</th>
<th>Dairy Products</th>
<th>Condiments</th>
<th>Grains/Cereals</th>
<th>Confections</th>
<th>Meat/Poultry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>Queso Cabreras</td>
<td>Arised Syrup</td>
<td>Sustaf's Chokolade</td>
<td>Devils</td>
<td>Aprikos</td>
</tr>
<tr>
<td>Ch}.{</td>
<td>Queso Manchego La Pastor</td>
<td>ChefAnton's Cajun Season</td>
<td>Turmböd</td>
<td>Festive Chocolate Biscuits</td>
<td>Aprikos</td>
</tr>
<tr>
<td>Guarana Fantastica</td>
<td>Qorgonzola Talino</td>
<td>ChefAnton's Gumbo Mix</td>
<td>Singepor Hockian Fried Mei</td>
<td>Sir Rodney's Marmalade</td>
<td>Alice Mutton</td>
</tr>
<tr>
<td>Sagra Ale</td>
<td>Mascarpone Faggio</td>
<td>Grandma's Boysenberry Spread</td>
<td>Too Mix</td>
<td>Sir Rodney's Scones</td>
<td>Tämminger Rostbratwurst</td>
</tr>
<tr>
<td>Stiegl Stout</td>
<td>Salbei</td>
<td>Northwoods Cranberry Sauce</td>
<td>Gnoochi di nomma Alice</td>
<td>NuNuCa Nuß-Krapfen-Creme</td>
<td>Herr Passas</td>
</tr>
<tr>
<td>Caffeine Risvea</td>
<td>Raclette Courdavault</td>
<td>Ranchyards Kustersherr</td>
<td>Rivoli Agnello</td>
<td>Gumnär Gummärchen</td>
<td>Tournolage</td>
</tr>
<tr>
<td>Chaihausa Vera</td>
<td>Semmelhart Platter</td>
<td>Laktiokodri</td>
<td>Wimmers yuta Sammekrodel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Irish Coffee</td>
<td>Sudbranddinkel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laughing Lumberjack Lager</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outback Lager</td>
<td>Rottnord</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rhodnau Kosterlabor</td>
<td>Mazzareul di cesuvar</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condiments</td>
<td>Grains/Cereals</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beef</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

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:

![Report Template with Bands](image)

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:

```
Data Source   Products   ...
```

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:
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:

![Report template with FooterBand, rectangle and primitives](image)

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:

![Report Template Image]

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, 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:
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:
2.31 Invoice Report With Parameters

Do the following steps 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, 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.
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:

![Invoice Template with 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, 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 crucial 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 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
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 variables 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 the calculation. In text expressions, the result of the calculation is automatically converted into a string. For example, if you enter the following expression:

\[ \text{Value} = \{1 + 2\} \]

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

\[ \text{Value} = 3 \]

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:

\[ \text{ValueA} = \{1 + 2\}, \text{ValueB} = \{2 + 3\} \]
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, 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: 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 the name of a data source;
✓ **ParentCategories** is the name of the 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 the name of a data source;

✓ **ParentProducts** is the name of the relations between **OrdersDetails** and **Products** data sources;

✓ **ParentCategories** is the name of the 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, but the 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 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 enter the 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 a
hyperlink 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 behavior 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 language, it is also possible to use the ternary operator. The syntax for the ternary operator is as follows:

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

In 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 the name, you would create an expression like this:
{MyVariable}

To use the variable by the 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 expression can 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
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 identifiers 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 the 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 of 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 a 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 that are used to set up and draw text, the **Text**, and the **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 of fonts are shown below:
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 the 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:
using properties such as `Font.Bold`, `Font.Italic`, `Font.Underline`, and `Font.Strikeout`, and the styles may be combined to produce different effects like bold and underlined or bold and italic. Examples of font styles are shown below:

```
AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
```

**Font Brushes**

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

```
AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
AaBbCcDd
```

**Custom Fonts**

When designing reports, it is possible to use custom fonts that 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 in 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 borders and the differences between them.

3.2.3.1 Simple Borders

**Border Sides**
Each border consists 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, you can set up on which sides a border will be visible.

**Border Style**
Seven styles of borders 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:
The style of borders 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.
You should 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 a 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:

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:

![Advanced Border Examples]

**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.

<table>
<thead>
<tr>
<th>4. Appearance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Brush</strong></td>
<td>Solid</td>
</tr>
<tr>
<td><strong>Border</strong></td>
<td>All</td>
</tr>
</tbody>
</table>
The simple **Border** dialog will be displayed. To access the advanced border features click the Advanced tab at the top to bring it to the front.

![Border dialog]

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 or 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.

<table>
<thead>
<tr>
<th></th>
<th>Left</th>
<th>Center</th>
<th>Right</th>
<th>Justify</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>An example of aligned text</td>
<td>An example of aligned text</td>
<td>An example of aligned text</td>
<td>An example of aligned text</td>
</tr>
</tbody>
</table>

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 a 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**

Also, 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 set to 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 or 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.
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.

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.

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:

![Component Style Selection](image)

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:

<table>
<thead>
<tr>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
</tr>
<tr>
<td>Ana Trujillo Emparedados y helados</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
</tr>
<tr>
<td>Around the Horn</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
</tr>
<tr>
<td>Blauer See Delikatessen</td>
</tr>
<tr>
<td>Blondesdds! père et fils</td>
</tr>
</tbody>
</table>

### 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:

![Style Designer dialog](image)
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:
   - **Open...** command allows you to open a previously saved file (*.sts) with a list of styles and style collections.
   - **Save As...** command saves styles and style collections to the *.sts file.
   - **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:
   - **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.
   - **Chart** style is used for any type of charts.
   - **Map** style is used only for maps in reports.
   - **Gauge** style is used for both linear and radial gauges.
   - **Cross-Tab** style is used for the cross-tab components.
   - **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:
   - Select the types of styles that you want to display in the list of report styles.
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.

* 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.

* 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:

- **Add Style**
- **Remove**
- **Duplicate Style**
- **Cut**
- **Copy**
- **Paste**
- **Create Style Collection**

As you can see from the picture, the context menu contains:

* The commands are used to create styles of various types;
* The command is used to delete the selected item (collection or style);
* 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.
* Commands are used to work with the clipboard (Cut, Copy, Paste);
* 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.

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

**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. 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:
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**

**Step 1.** 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**

**Step 1.** Run the Style Designer.

**Step 2.** Select the Create Style Collection command from the **Actions** menu or the
context menu of the style designer;

**Step 3.** Set the parameters of the future collection of styles.

1. **Collection Name** field is used to specify the name of the collection.
2. **Color** field is used to select the main color for the style collection.
3. **Nested Level** field is used to select the value corresponding to the level of nesting in the report.
4. **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. **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. **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. **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, properties of the component will meet two requirements - placement and naming. In addition, you can use the multi-conditions, 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, to the component of the same type. The first level of nesting is organized when the component is added to the report template, 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, 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, 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 **ReportTitleBand**s and **ReportSummaryBand**s:

![Diagram showing nesting levels]

For the **PageHeaderBand**s and **PageFooterBand**s, 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.
Parameters of a condition
Specify parameters of condition on this panel.

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:

```
Value 1  Operation  Value 2
1       ≥           2
```

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:
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 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:
<table>
<thead>
<tr>
<th>Operator</th>
<th>Types of data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Numerical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Logic</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expression</td>
<td></td>
</tr>
<tr>
<td>equal to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>not equal to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>between</td>
<td></td>
<td></td>
</tr>
<tr>
<td>not between</td>
<td></td>
<td></td>
</tr>
<tr>
<td>greater than</td>
<td></td>
<td></td>
</tr>
<tr>
<td>greater than or equal to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>less than</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td>Truth Value 1</td>
<td>Truth Value 2</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>less than or equal to</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>containing</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>not containing</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>beginning with</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>ending with</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the first value is less than the second value or equal to the second value, then the condition is true.

If the first value contains the second value, then the condition is true. This operator is used only for strings.

If the first value does not contain the second value, then the condition is true. This operator is used only for strings.

If the first value starts with the second value, then the condition is true. This operator is used only for strings.

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:

![Configuration Panel](image)

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:
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.

### 3.3.5 Defining Formatting

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

\[(\text{Categories.CategoryID}) = 1 \text{ or } (\text{Categories.CategoryID} = 2)\]

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:

- [ ] And
- [ ] Or
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 significant 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:
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 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: \texttt{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:

<table>
<thead>
<tr>
<th>CompanyName</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
<td>Berlin</td>
<td>Germany</td>
</tr>
<tr>
<td>Ana Trujillo Empaños</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>London</td>
<td>UK</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
<td>Luleå</td>
<td>Sweden</td>
</tr>
<tr>
<td>Bauer See Delikatessen</td>
<td>Mannheim</td>
<td>Germany</td>
</tr>
<tr>
<td>Blondesdøl père et fils</td>
<td>Strasbourg</td>
<td>France</td>
</tr>
<tr>
<td>Bólido Comidas preparadas</td>
<td>Madrid</td>
<td>Spain</td>
</tr>
<tr>
<td>Bon app’</td>
<td>Marseille</td>
<td>France</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>CompanyName</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
<td>Berlin</td>
<td>Germany</td>
</tr>
<tr>
<td>Ana Trujillo Empaños</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>London</td>
<td>UK</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
<td>Luleå</td>
<td>Sweden</td>
</tr>
<tr>
<td>Bauer See Delikatessen</td>
<td>Mannheim</td>
<td>Germany</td>
</tr>
<tr>
<td>Blondesdøl père et fils</td>
<td>Strasbourg</td>
<td>France</td>
</tr>
<tr>
<td>Bólido Comidas preparadas</td>
<td>Madrid</td>
<td>Spain</td>
</tr>
<tr>
<td>Bon app’</td>
<td>Marseille</td>
<td>France</td>
</tr>
</tbody>
</table>
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:
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:

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:

<table>
<thead>
<tr>
<th>CompanyName</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
<td>Berlin</td>
<td>Germany</td>
</tr>
<tr>
<td>Ana Trujillo Emparedados</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>London</td>
<td>UK</td>
</tr>
<tr>
<td>Berglunds snabblop</td>
<td>Luleå</td>
<td>Sweden</td>
</tr>
<tr>
<td>Bauer See Delikatesen</td>
<td>Mannheim</td>
<td>Germany</td>
</tr>
<tr>
<td>Blondesddsl père et fils</td>
<td>Strasbourg</td>
<td>France</td>
</tr>
<tr>
<td>Bólido Comidas preaparadas</td>
<td>Madrid</td>
<td>Spain</td>
</tr>
<tr>
<td>Bon app'</td>
<td>Marseille</td>
<td>France</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>EmployeeID</th>
<th>LastName</th>
<th>FirstName</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Davolio</td>
<td>Nancy</td>
<td>USA</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>USA</td>
</tr>
<tr>
<td>3</td>
<td>Leverling</td>
<td>Janet</td>
<td>USA</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>USA</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>UK</td>
</tr>
<tr>
<td>6</td>
<td>Suyama</td>
<td>Michael</td>
<td>UK</td>
</tr>
<tr>
<td>7</td>
<td>King</td>
<td>Robert</td>
<td>UK</td>
</tr>
<tr>
<td>8</td>
<td>Callahan</td>
<td>Laura</td>
<td>USA</td>
</tr>
<tr>
<td>9</td>
<td>Dodsworth</td>
<td>Anne</td>
<td>UK</td>
</tr>
</tbody>
</table>

For example, you can make a text underlined for components that contain a Nancy word in the FirstName column. Select a text component with the (Employees.LastName) expression, in the DataBand and call the Conditions editor. Then, you should set a condition: select the Employees.FirstName data column, as the first value, and indicate the Nancy letter, as a second value. Also set the Operation comparison to the not equal to value. Change the formatting parameters, in this case, set the font style to underlined. The picture below shows the Conditions editor dialog box:
After making changes in the report template, the report engine will perform conditional formatting of text components, according to the specified parameters. In this case, the underlined font will be applied for the content of text components that match the specified condition. The picture below shows a page of the rendered report with conditional formatting:
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:

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:

<table>
<thead>
<tr>
<th>CompanyName</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
<td>Berlin</td>
<td>Germany</td>
</tr>
<tr>
<td>Ana Trujillo Emparedados</td>
<td>México D.F.</td>
<td>México</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>London</td>
<td>UK</td>
</tr>
<tr>
<td>Berglunds snabblop</td>
<td>Luleå</td>
<td>Sweden</td>
</tr>
<tr>
<td>Bauer See Delikatessen</td>
<td>Mannheim</td>
<td>Germany</td>
</tr>
<tr>
<td>Blondesddsl père et fils</td>
<td>Strasbourg</td>
<td>France</td>
</tr>
<tr>
<td>Bólido Comidas preparadas</td>
<td>Madrid</td>
<td>Spain</td>
</tr>
<tr>
<td>Bon app'</td>
<td>Marseille</td>
<td>France</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>CompanyName</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
<td>Berlin</td>
<td>Germany</td>
</tr>
<tr>
<td>Ana Trujillo Emparedados</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>México D.F.</td>
<td>Mexico</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>London</td>
<td>UK</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
<td>Luleå</td>
<td>Sweden</td>
</tr>
<tr>
<td>Bauer See Delikatessen</td>
<td>Mannheim</td>
<td>Germany</td>
</tr>
<tr>
<td>Bónodesdssl père et fils</td>
<td>Strasbourg</td>
<td>France</td>
</tr>
<tr>
<td>Bóldo Comidas preparadas</td>
<td>Madrid</td>
<td>Spain</td>
</tr>
<tr>
<td>Bon app’</td>
<td>Marseille</td>
<td>France</td>
</tr>
</tbody>
</table>

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:

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, replace its textual expression on a text expression, specified in the condition. The picture below shows a report page:

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:

<table>
<thead>
<tr>
<th>EmployeeID</th>
<th>LastName</th>
<th>FirstName</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Devolio</td>
<td>Nancy</td>
<td>USA</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>USA</td>
</tr>
<tr>
<td>3</td>
<td>Levering</td>
<td>Janet</td>
<td>USA</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>USA</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>UK</td>
</tr>
<tr>
<td>6</td>
<td>Suyama</td>
<td>Michael</td>
<td>UK</td>
</tr>
<tr>
<td>7</td>
<td>King</td>
<td>Robert</td>
<td>UK</td>
</tr>
<tr>
<td>8</td>
<td>Callahan</td>
<td>Laura</td>
<td>USA</td>
</tr>
<tr>
<td>9</td>
<td>Dodsworth</td>
<td>Anne</td>
<td>UK</td>
</tr>
</tbody>
</table>

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.
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.

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. 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. 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:

<table>
<thead>
<tr>
<th>EmployeeID</th>
<th>LastName</th>
<th>FirstName</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Davolio</td>
<td>Nancy</td>
<td>USA</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>USA</td>
</tr>
<tr>
<td>3</td>
<td>Levetling</td>
<td>Janet</td>
<td>USA</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>USA</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>UK</td>
</tr>
<tr>
<td>6</td>
<td>Suyama</td>
<td>Michael</td>
<td>UK</td>
</tr>
<tr>
<td>7</td>
<td>King</td>
<td>Robert</td>
<td>UK</td>
</tr>
<tr>
<td>8</td>
<td>Callahan</td>
<td>Laura</td>
<td>USA</td>
</tr>
<tr>
<td>9</td>
<td>Dodsworth</td>
<td>Anne</td>
<td>UK</td>
</tr>
</tbody>
</table>

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, i.e. beginning from which a histogram of negative and positive values. For example, the minimum value is -1, while the maximum is three, i.e. 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
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:

<table>
<thead>
<tr>
<th>ProductName</th>
<th>QuantityPerUnit</th>
<th>UnitPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getost</td>
<td>2.5</td>
<td>62</td>
</tr>
<tr>
<td>Guaraná Fantástia</td>
<td>4.5</td>
<td>-30</td>
</tr>
<tr>
<td>Konbu</td>
<td>5</td>
<td>-26</td>
</tr>
<tr>
<td>Flo Mix</td>
<td>7</td>
<td>-12</td>
</tr>
<tr>
<td>Tourtière</td>
<td>7.45</td>
<td>-29</td>
</tr>
<tr>
<td>Rhönbräu Klostermär</td>
<td>7.75</td>
<td>-75</td>
</tr>
<tr>
<td>Tunnbröd</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>Testime Chocolate Biscuits</td>
<td>9.2</td>
<td>-25</td>
</tr>
<tr>
<td>Zaanse koeken</td>
<td>3.5</td>
<td>-14</td>
</tr>
<tr>
<td>Rogeda sild</td>
<td>9.6</td>
<td>-45</td>
</tr>
<tr>
<td>Jack's New England Clam Chowder</td>
<td>9.65</td>
<td>35</td>
</tr>
</tbody>
</table>

The picture below shows an example of a rendered report with negative and positive values:

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
The **Color Scale Condition** is working according to 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:
The **Column** field. This field indicates the data column from which the value for the condition will be taken.

The **Color Scale Type** fields provide 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 of color scale:

![Color Scale Type Menu](image)

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:

![Type Menu](image)

The **Value** field. Used for a minimum color scale;

The **Color** field. Used for a minimum color scale;

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;

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

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:
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:

<table>
<thead>
<tr>
<th>EmployeeID</th>
<th>LastName</th>
<th>FirstName</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Davolio</td>
<td>Nancy</td>
<td>USA</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>USA</td>
</tr>
<tr>
<td>3</td>
<td>Leverling</td>
<td>Janet</td>
<td>USA</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>USA</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>UK</td>
</tr>
<tr>
<td>6</td>
<td>Suyama</td>
<td>Michael</td>
<td>UK</td>
</tr>
<tr>
<td>7</td>
<td>King</td>
<td>Robert</td>
<td>UK</td>
</tr>
<tr>
<td>8</td>
<td>Callahan</td>
<td>Laura</td>
<td>USA</td>
</tr>
<tr>
<td>9</td>
<td>Dodsworth</td>
<td>Anne</td>
<td>UK</td>
</tr>
</tbody>
</table>
Add an **Icon Set** condition. To do this, select a text component, for example a component with the **{Employees.Country}** expression. Add the **Icon Set** condition. Change the parameters of the condition. The picture below shows the **Conditions** dialog:

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

4. The **Alignment** field is used to align icons in text components. The picture below shows the Alignment menu options:

   - Middle Left
   - Top Left
   - Top Center
   - Top Right
   - Middle Left
   - Middle Center
   - Middle Right
   - Bottom Left
   - Bottom Center
   - Bottom Right

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:

<table>
<thead>
<tr>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;=</td>
</tr>
<tr>
<td>&gt;</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>1. Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Percentage</td>
</tr>
<tr>
<td>3. Value</td>
</tr>
</tbody>
</table>

In the Value field 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:
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**

![System Variable Panel](image)

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:
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.

<table>
<thead>
<tr>
<th>0 Degrees</th>
<th>45 Degrees</th>
<th>90 Degrees</th>
<th>180 Degrees</th>
<th>270 Degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>This example of the text under specified angle</td>
<td>This example of the text under specified angle</td>
<td>This example of the text under specified angle</td>
<td>This example of the text under specified angle</td>
<td>This example of the text under specified angle</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th><strong>Beverages</strong></th>
<th><strong>Condiments</strong></th>
<th><strong>Confections</strong></th>
<th><strong>Dairy Products</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chai</strong></td>
<td><strong>Ai-seed Syrup</strong></td>
<td><strong>Chocolate</strong></td>
<td><strong>Camembert Pienot</strong></td>
</tr>
<tr>
<td>10 boxes x 20 bags</td>
<td>12 - 500 ml bottles</td>
<td>10 pkgs.</td>
<td>15 - 300 g rounds</td>
</tr>
<tr>
<td>19,00p</td>
<td>19,00p</td>
<td>12,75p</td>
<td>34,00p</td>
</tr>
<tr>
<td><strong>Chang</strong></td>
<td><strong>Chef Antoine’s Cajun</strong></td>
<td></td>
<td><strong>Fleuretto</strong></td>
</tr>
<tr>
<td>24 - 12 oz bottles</td>
<td>48 - 6 oz jars</td>
<td>100 - 250 g bags</td>
<td>100 - 500 g pkgs.</td>
</tr>
<tr>
<td>19,00p</td>
<td>22,00p</td>
<td>31,23p</td>
<td>21,50p</td>
</tr>
<tr>
<td><strong>Chartreuse verte</strong></td>
<td><strong>Chef Antoine’s Oombo</strong></td>
<td></td>
<td><strong>Flotemysopt</strong></td>
</tr>
<tr>
<td>750 oz per bottle</td>
<td>Mix</td>
<td>24 - 50 g pkgs.</td>
<td>10 pkgs.</td>
</tr>
<tr>
<td>16,00p</td>
<td>21,35p</td>
<td>28,00p</td>
<td>21,50p</td>
</tr>
<tr>
<td><strong>Côte de Blaye</strong></td>
<td><strong>Sen Ma Shouyu</strong></td>
<td><strong>NuNuCa Null Neuget-Creme</strong></td>
<td>20 - 400 g glasses.</td>
</tr>
<tr>
<td>12 - 75 cl bottles</td>
<td>24 - 250 ml bottles</td>
<td>29,00p</td>
<td>14,00p</td>
</tr>
<tr>
<td>209,50p</td>
<td>19,50p</td>
<td>14,00p</td>
<td>7,45p</td>
</tr>
<tr>
<td><strong>Guarana Fantástica</strong></td>
<td><strong>Gramma’s Boysenberry</strong></td>
<td></td>
<td><strong>Pavlava</strong></td>
</tr>
<tr>
<td>12 - 355 ml cans</td>
<td>12 - 8 oz jars</td>
<td>25,00p</td>
<td>17,45p</td>
</tr>
<tr>
<td>4,50p</td>
<td>25,00p</td>
<td>17,45p</td>
<td>40,00p</td>
</tr>
<tr>
<td><strong>Boh Coffee</strong></td>
<td><strong>Gula Maloca</strong></td>
<td><strong>Northwoods Cranberry</strong></td>
<td><strong>Schoaggi Schokolade</strong></td>
</tr>
<tr>
<td>16 - 500 g ths</td>
<td>20 - 2 kg bags</td>
<td>40,00p</td>
<td>100 - 100 g pieces</td>
</tr>
<tr>
<td>46,00p</td>
<td>19,45p</td>
<td>40,00p</td>
<td>40,00p</td>
</tr>
<tr>
<td><strong>Lakkaikööri</strong></td>
<td><strong>Louisiana Fiery Hot</strong></td>
<td><strong>Original Frankfurter</strong></td>
<td><strong>Scottish Longbreads</strong></td>
</tr>
<tr>
<td>500 ml</td>
<td>Pepper Sauce</td>
<td>grüne Soße</td>
<td>10 boxes x 9 pieces</td>
</tr>
<tr>
<td>16,00p</td>
<td>21,05p</td>
<td>12 boxes</td>
<td>12,50p</td>
</tr>
<tr>
<td><strong>Laughing Lumberjack</strong></td>
<td><strong>Louisiana Hot Spiced</strong></td>
<td></td>
<td><strong>Sir Rodney’s Marmalade</strong></td>
</tr>
<tr>
<td>Lager</td>
<td>Olla</td>
<td>100 - 8 oz jars</td>
<td>30 gift boxes</td>
</tr>
<tr>
<td>24 - 12 oz bottles</td>
<td>24 - 8 oz jars</td>
<td>17,00p</td>
<td>81,00p</td>
</tr>
<tr>
<td>14,00p</td>
<td>17,00p</td>
<td>49,30p</td>
<td>40,00p</td>
</tr>
<tr>
<td><strong>Outback Lager</strong></td>
<td><strong>Northwoods Cranberry</strong></td>
<td></td>
<td><strong>Tarte au sucre</strong></td>
</tr>
<tr>
<td>24 - 355 ml bottles</td>
<td>Sauce</td>
<td>12 - 12 oz jars</td>
<td>48 pies</td>
</tr>
<tr>
<td>16,00p</td>
<td>40,00p</td>
<td>46,00p</td>
<td>49,30p</td>
</tr>
<tr>
<td><strong>Röhrbräu Klosterbier</strong></td>
<td><strong>Original Frankfurter</strong></td>
<td></td>
<td><strong>Tarte au sucre</strong></td>
</tr>
<tr>
<td>24 - 0.5 l bottles</td>
<td>grüne Soße</td>
<td>12 boxes</td>
<td>48 pies</td>
</tr>
<tr>
<td>7,75p</td>
<td>120,00p</td>
<td><strong>Végé-spread</strong></td>
<td>49,30p</td>
</tr>
<tr>
<td><strong>Sasquatch Ale</strong></td>
<td><strong>Original Frankfurter</strong></td>
<td></td>
<td>15 - 625 g jars</td>
</tr>
<tr>
<td>24 - 12 oz bottles</td>
<td>grüne Soße</td>
<td>42,00p</td>
<td>42,00p</td>
</tr>
<tr>
<td>18,00p</td>
<td>15 - 625 g jars</td>
<td>24,00p</td>
<td>24,00p</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Assistant Sales Agent</th>
<th>Assistant Sales Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assistant Sales Agent</td>
<td></td>
</tr>
<tr>
<td>Assistant Sales Represent</td>
<td>Assistant Sales Represent</td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td></td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td></td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td></td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td></td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Assistant Sales Agent</th>
<th>Assistant Sales Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assistant Sales Agent</td>
<td></td>
</tr>
<tr>
<td>Assistant Sales Agent</td>
<td></td>
</tr>
<tr>
<td>Assistant Sales Represent</td>
<td>Assistant Sales Represent</td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td></td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td></td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td></td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td></td>
</tr>
</tbody>
</table>

**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).

<table>
<thead>
<tr>
<th>HideZeros = false</th>
<th>HideZeros = true</th>
</tr>
</thead>
<tbody>
<tr>
<td>9,00$  61,00</td>
<td>9,00$  61,00</td>
</tr>
<tr>
<td>33,25$ 22,00</td>
<td>33,25$ 22,00</td>
</tr>
<tr>
<td>39,00$ 0,00</td>
<td>39,00$</td>
</tr>
<tr>
<td>97,00$ 29,00</td>
<td>97,00$ 29,00</td>
</tr>
<tr>
<td>24,00$ 115,00</td>
<td>24,00$ 115,00</td>
</tr>
<tr>
<td>32,80$ 0,00</td>
<td>32,80$</td>
</tr>
<tr>
<td>123,79$ 0,00</td>
<td>123,79$</td>
</tr>
</tbody>
</table>
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.

![Image of report components output]

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.
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

```
ShrinkFont
```

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

```
ShrinkFontToFit ShrinkFontToFit
```
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.
Images below show how this property works

The **Shrink Font to Fit Minimum Size** property is set to 1. The font **Arial**, size **8pt**

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

> **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, 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":

![Zip Code Example](image)

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:

![Increased Width Zip Code](image)

### 3.5 Text Formatting

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

The Format window is divided into three parts.

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

2. Shows how the formatted text will look like;
3. Shows the format settings.
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.

![Format dialog]

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.
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.

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.

1. **Group separator**
   When the Group Separator is used then number will be separated into number positions.

2. **Local setting**
   When using the Local settings, numerical values are formatted according to the current OS installations.

3. **Decimal digits**
   Number of decimal digits, which are used to format numerical values.

4. **Decimal separator**
Used as a decimal separator to separate numerical values in formatting.

5 **Group separator**
Used as a group separator when numerical values formatting.

6 **Group size**
The number of digits in each group in currency values formatting.

7 **Negative pattern**

This pattern is used to format negative values.

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

<table>
<thead>
<tr>
<th>ProductName</th>
<th>ProductID</th>
<th>CategoryID</th>
<th>UnitPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>1.0</td>
<td>1.0</td>
<td>18.00</td>
</tr>
<tr>
<td>Chang</td>
<td>2.0</td>
<td>1.0</td>
<td>19.00</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>3.0</td>
<td>2.0</td>
<td>10.00</td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>4.0</td>
<td>2.0</td>
<td>22.00</td>
</tr>
<tr>
<td>Chef Anton's Sumbo Mk</td>
<td>5.0</td>
<td>2.0</td>
<td>21.35</td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>6.0</td>
<td>2.0</td>
<td>25.00</td>
</tr>
<tr>
<td>Uncle Bob's Organic Dried Pears</td>
<td>7.0</td>
<td>7.0</td>
<td>30.00</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>8.0</td>
<td>2.0</td>
<td>40.00</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>9.0</td>
<td>6.0</td>
<td>57.00</td>
</tr>
<tr>
<td>Ikura</td>
<td>10.0</td>
<td>8.0</td>
<td>31.00</td>
</tr>
<tr>
<td>Queso Cabrales</td>
<td>11.0</td>
<td>4.0</td>
<td>21.00</td>
</tr>
<tr>
<td>Queso Manchego La Pastora</td>
<td>12.0</td>
<td>4.0</td>
<td>38.00</td>
</tr>
</tbody>
</table>

⚠️ **Notice:** 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

To display numeric values as a currency you should use the Currency format. This format is designed specifically to output monetary values.
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.
It should be noted that previously there were two ways 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.
When the Group Separator is used then currency values will be separated into number positions.

When using the Local settings, currency values are formatted according to the current OS installations.

Number of decimal digits, which are used to format currency values.

Used as a decimal separator to separate currency values in formatting.

Used as a group separator when currency values formatting.

The number of digits in each group in currency values formatting.

This pattern is used to format positive values.

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.

<table>
<thead>
<tr>
<th>ProductName</th>
<th>ProductID</th>
<th>CategoryID</th>
<th>UnitPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>1.0</td>
<td>1.0</td>
<td>€10.00</td>
</tr>
<tr>
<td>Chang</td>
<td>2.0</td>
<td>1.0</td>
<td>€19.00</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>3.0</td>
<td>2.0</td>
<td>€10.00</td>
</tr>
<tr>
<td>Chef Anton’s Cajun Seasoning</td>
<td>4.0</td>
<td>2.0</td>
<td>€22.00</td>
</tr>
<tr>
<td>Chef Anton’s Gumbo Mix</td>
<td>5.0</td>
<td>2.0</td>
<td>€21.35</td>
</tr>
<tr>
<td>Grandma’s Boysenberry Spread</td>
<td>6.0</td>
<td>2.0</td>
<td>€25.00</td>
</tr>
<tr>
<td>UncleBob’s Organic Dried Pears</td>
<td>7.0</td>
<td>7.0</td>
<td>€30.00</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>8.0</td>
<td>2.0</td>
<td>€40.00</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>9.0</td>
<td>6.0</td>
<td>€97.00</td>
</tr>
<tr>
<td>Ikura</td>
<td>10.0</td>
<td>8.0</td>
<td>€31.00</td>
</tr>
<tr>
<td>Queso Cabrales</td>
<td>11.0</td>
<td>4.0</td>
<td>€21.00</td>
</tr>
<tr>
<td>Queso Manchego La Pastorra</td>
<td>12.0</td>
<td>4.0</td>
<td>€30.00</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>ProductName</th>
<th>OrderDate</th>
<th>RequiredDate</th>
<th>ShippedDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queso Cabrileo</td>
<td>8/3/2008 11:00:00 PM</td>
<td>8/31/2008 11:00:00 PM</td>
<td>8/15/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Singapore Hokkien Fried Mee</td>
<td>8/3/2008 11:00:00 PM</td>
<td>8/31/2008 11:00:00 PM</td>
<td>8/15/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Mozzarella di Giovanni</td>
<td>8/3/2008 11:00:00 PM</td>
<td>8/31/2008 11:00:00 PM</td>
<td>8/15/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Tofu</td>
<td>8/4/2008 11:00:00 PM</td>
<td>9/15/2008 11:00:00 PM</td>
<td>8/9/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Manjinup Dried Apples</td>
<td>8/4/2008 11:00:00 PM</td>
<td>9/15/2008 11:00:00 PM</td>
<td>8/9/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Jack’s New England Clam Chowder</td>
<td>8/7/2008 11:00:00 PM</td>
<td>9/4/2008 11:00:00 PM</td>
<td>8/11/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Manjinup Dried Apples</td>
<td>8/7/2008 11:00:00 PM</td>
<td>9/4/2008 11:00:00 PM</td>
<td>8/11/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Louisiana Fiery Hot Pepper Sauce</td>
<td>8/7/2008 11:00:00 PM</td>
<td>9/4/2008 11:00:00 PM</td>
<td>8/11/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Gustaf’s Knäckebröd</td>
<td>8/7/2008 11:00:00 PM</td>
<td>9/4/2008 11:00:00 PM</td>
<td>8/14/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Ravioli Angelo</td>
<td>8/7/2008 11:00:00 PM</td>
<td>9/4/2008 11:00:00 PM</td>
<td>8/14/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Louisiana Fiery Hot Pepper Sauce</td>
<td>8/7/2008 11:00:00 PM</td>
<td>9/4/2008 11:00:00 PM</td>
<td>8/14/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Sir Rodney’s Marmalade</td>
<td>8/8/2008 11:00:00 PM</td>
<td>9/5/2008 11:00:00 PM</td>
<td>8/10/2008 11:00:00 PM</td>
</tr>
<tr>
<td>Gelbst</td>
<td>8/8/2008 11:00:00 PM</td>
<td>9/5/2008 11:00:00 PM</td>
<td>8/10/2008 11:00:00 PM</td>
</tr>
</tbody>
</table>

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.
The list of formatting types.

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

1 Date format
The list of formatting types.

And then, the dates in the report will be displayed with certain formats.
<table>
<thead>
<tr>
<th>Product Name</th>
<th>Order Date</th>
<th>Required Date</th>
<th>Shipped Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queso Cabrales</td>
<td>8/3/2008</td>
<td>Sunday, August 31, 2008</td>
<td>15/08/2008</td>
</tr>
<tr>
<td>Gustaf's Knäckebröd</td>
<td>8/7/2008</td>
<td>Thursday, September 4, 2008</td>
<td>14/08/2008</td>
</tr>
<tr>
<td>Ravioli Angelo</td>
<td>8/7/2008</td>
<td>Thursday, September 4, 2008</td>
<td>14/08/2008</td>
</tr>
<tr>
<td>Louisiana Fiery Hot Pepper Sauce</td>
<td>8/7/2008</td>
<td>Thursday, September 4, 2008</td>
<td>14/08/2008</td>
</tr>
<tr>
<td>Geitost</td>
<td>8/8/2008</td>
<td>Friday, September 5, 2008</td>
<td>10/08/2008</td>
</tr>
</tbody>
</table>

⚠️ **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

The list of formatting types

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

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Unit Price</th>
<th>Profitability Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>$18.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Chang</td>
<td>$19.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>$10.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>$22.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Chef Anton's Gumbo Mix</td>
<td>$21.35</td>
<td>0.01</td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>$25.00</td>
<td>0.02</td>
</tr>
<tr>
<td>Uncle Bob's Organic Dried Pears</td>
<td>$30.00</td>
<td>0.02</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>$40.00</td>
<td>0.03</td>
</tr>
<tr>
<td>MishiKobe Niku</td>
<td>$97.00</td>
<td>0.09</td>
</tr>
<tr>
<td>Ikura</td>
<td>$31.00</td>
<td>0.02</td>
</tr>
<tr>
<td>Queso Cabrales</td>
<td>$21.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Queso Manchego La Pastora</td>
<td>$38.00</td>
<td>0.03</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>ProductName</th>
<th>UnitPrice</th>
<th>Profitability Index</th>
<th>Profitability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>$ 15.00</td>
<td>0.01</td>
<td>0.84 %</td>
</tr>
<tr>
<td>Chang</td>
<td>$ 19.00</td>
<td>0.01</td>
<td>0.95 %</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>$ 10.00</td>
<td>0.00</td>
<td>0.00 %</td>
</tr>
<tr>
<td>Chef Anton’s Cajun Seasoning</td>
<td>$ 22.00</td>
<td>0.01</td>
<td>1.26 %</td>
</tr>
<tr>
<td>Chef Anton’s Gumbo Mix</td>
<td>$ 21.35</td>
<td>0.01</td>
<td>1.19 %</td>
</tr>
<tr>
<td>Grandma’s Boysenberry Spread</td>
<td>$ 25.00</td>
<td>0.02</td>
<td>1.58 %</td>
</tr>
<tr>
<td>Uncle Bob’s Organic Dried Pears</td>
<td>$ 30.00</td>
<td>0.02</td>
<td>2.11 %</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>$ 40.00</td>
<td>0.03</td>
<td>3.16 %</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>$ 97.00</td>
<td>0.09</td>
<td>9.16 %</td>
</tr>
<tr>
<td>Ikura</td>
<td>$ 31.00</td>
<td>0.02</td>
<td>2.21 %</td>
</tr>
<tr>
<td>Queso Obrero</td>
<td>$ 21.00</td>
<td>0.01</td>
<td>1.16 %</td>
</tr>
<tr>
<td>Queso Manchego La Pastor</td>
<td>$ 38.00</td>
<td>0.03</td>
<td>2.95 %</td>
</tr>
</tbody>
</table>

It should be noted that previously there were two ways 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.
1 **Group separator**
When the Group Separator is used then currency values will be separated into number positions.

2 **Use local setting**
When using the Local settings, numerical values are formatted according to the current OS installations.

3 **Decimal digits**
Number of decimal digits, which are used to format numerical values.

4 **Decimal separator**
Used as a decimal separator to separate numerical values in formatting.

5 **Group separator**
Used as a group separator when numerical values formatting.

6 **Group size**
The number of digits in each group in currency values formatting.

7 **Positive pattern**
This pattern is used to format positive values.

8 **Negative pattern**
This pattern is used to format negative values.
9 Percentage symbol
The symbol will be used as a percent sign.

3.5.7 Boolean Values Formatting

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

<table>
<thead>
<tr>
<th>Formats</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>False</td>
</tr>
<tr>
<td>Number</td>
<td>False</td>
</tr>
<tr>
<td>Currency</td>
<td>False</td>
</tr>
<tr>
<td>Date</td>
<td>False</td>
</tr>
<tr>
<td>Time</td>
<td>False</td>
</tr>
<tr>
<td>Percentage</td>
<td>True</td>
</tr>
<tr>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>Custom</td>
<td>True</td>
</tr>
</tbody>
</table>

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.

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

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.
**Mask**
A string or an expression that sets formatting mask.

**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.
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 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` specifier is used:

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Order Date</th>
<th>Shipped Date</th>
<th>Unit Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queso Cabrales</td>
<td>2008-08-03</td>
<td>08-15-2008</td>
<td>14.00 dollars of USA</td>
</tr>
<tr>
<td>Singaporean Hokkien Fried Mee</td>
<td>2008-08-03</td>
<td>08-15-2008</td>
<td>9.80 dollars of USA</td>
</tr>
<tr>
<td>Mozzarella di Giovanni</td>
<td>2008-08-03</td>
<td>08-15-2008</td>
<td>34.60 dollars of USA</td>
</tr>
<tr>
<td>Tofu</td>
<td>2008-08-04</td>
<td>08-09-2008</td>
<td>18.60 dollars of USA</td>
</tr>
<tr>
<td>Manjimup Dried Apples</td>
<td>2008-08-04</td>
<td>08-09-2008</td>
<td>42.40 dollars of USA</td>
</tr>
<tr>
<td>Jack's New England Clam Chowder</td>
<td>2008-08-07</td>
<td>08-11-2008</td>
<td>7.70 dollars of USA</td>
</tr>
<tr>
<td>Manjimup Dried Apples</td>
<td>2008-08-07</td>
<td>08-11-2008</td>
<td>42.40 dollars of USA</td>
</tr>
<tr>
<td>Louisiana Fiery Hot Pepper Sauce</td>
<td>2008-08-07</td>
<td>08-11-2008</td>
<td>16.80 dollars of USA</td>
</tr>
<tr>
<td>Gustaf's Knäckebrodd</td>
<td>2008-08-07</td>
<td>08-14-2008</td>
<td>16.80 dollars of USA</td>
</tr>
<tr>
<td>RavioliAngelo</td>
<td>2008-08-07</td>
<td>08-14-2008</td>
<td>15.60 dollars of USA</td>
</tr>
<tr>
<td>Louisiana Fiery Hot Pepper Sauce</td>
<td>2008-08-07</td>
<td>08-14-2008</td>
<td>16.80 dollars of USA</td>
</tr>
<tr>
<td>Sir Rodney's Marmalade</td>
<td>2008-08-08</td>
<td>08-10-2008</td>
<td>64.60 dollars of USA</td>
</tr>
</tbody>
</table>

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)}`
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:

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

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

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

```html
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:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1+2}</td>
<td>Incorrect</td>
</tr>
<tr>
<td>{1}+2</td>
<td>Incorrect</td>
</tr>
<tr>
<td>{1+2}</td>
<td>Incorrect</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;b&gt;simple&lt;/b&gt; &lt;i&gt;expression {1+2}&lt;/i&gt;</td>
<td>Correct</td>
</tr>
</tbody>
</table>

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 `<font>` Tag

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

**Syntax:**

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

**Parameters:**

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

Not all of these 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:

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

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

```
Test "Test" Test
```
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:

```html
<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:

```html
<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 the tag 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.

Alternative 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 be 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: <b>, <i>, <u>, <s>, <sup>, <sub>, <strong>, <p>, <br>. 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 <b> tag works in a text expression. If you enter the following expression:

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

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

This **text** is bold.

Note that the word ‘text’ is enclosed within the opening and closing <b> and </b> 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 is bold italic.

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

This text is bold italic.

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

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

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

3.6.2.1 HTML `<b>` Tag

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

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

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

Test Test Test

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

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 <em> Tag

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

Emphasis <em>Emphasis</em> Emphasis

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

Emphasis Emphasis Emphasis

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

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 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.

3.6.2.5 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 <sup>Test</sup> Test

then after calculation the result appearing in the report will be:
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 `<sub>Test</sub>` Test

The result of output:

Test Test Test

### 3.6.2.8 HTML `<strong>` Tag

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

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

The result of output:

Text **Text** Text
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 <br> Tag

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

How it<br> works.

The result of output:

How it works.

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 the tag, 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<br>
<text-align="right">Test</text-align><br>
Test<br>

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

<table>
<thead>
<tr>
<th>Test</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>Test</td>
</tr>
</tbody>
</table>
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<br> <letter-spacing="0.5">Test</letter-spacing>

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

Test  Test

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:
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, to display the "greater-than" sign or the opening "<" brace, the "&lt;" character is used. Each character has its &-ASCII code, which has a specific &### format, where **** is a numeric character. Pointing a &-ASCII code, the appropriate symbol will be output on the page. Also, some characters have &-Name codes, which have the &**** formats where **** is an alphabetic names of characters. Below are the tables with the most frequently used characters:

<table>
<thead>
<tr>
<th>Common Name</th>
<th>ISO Latin-1 Numeric Entity</th>
<th>&amp;-ASCII</th>
<th>&amp;-Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quotation mark</td>
<td>&quot;</td>
<td>&amp;034;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Ampersand</td>
<td>&amp;</td>
<td>&amp;038;</td>
<td>&amp;</td>
</tr>
<tr>
<td>Non-breaking space</td>
<td></td>
<td>&amp;160;</td>
<td> </td>
</tr>
<tr>
<td>Inverted exclamation point</td>
<td>i</td>
<td>&amp;161;</td>
<td>¡</td>
</tr>
<tr>
<td>Cent</td>
<td>¢</td>
<td>&amp;162;</td>
<td>¢</td>
</tr>
<tr>
<td>Pound sterling</td>
<td>£</td>
<td>&amp;163;</td>
<td>£</td>
</tr>
<tr>
<td>General currency</td>
<td>₤</td>
<td>&amp;164;</td>
<td>¤</td>
</tr>
<tr>
<td>Symbol</td>
<td>HTML Entity</td>
<td>ASCII</td>
<td>Unicode</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>Yen sign</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>Broken vertical bar</td>
<td></td>
<td></td>
<td>\brvbar;</td>
</tr>
<tr>
<td>Section sign</td>
<td>§</td>
<td>§</td>
<td>§</td>
</tr>
<tr>
<td>Dieresis</td>
<td>¨</td>
<td>¨</td>
<td>¨</td>
</tr>
<tr>
<td>Copyright</td>
<td>©</td>
<td>©</td>
<td>©</td>
</tr>
<tr>
<td>Feminine ordinal</td>
<td>a</td>
<td>a</td>
<td>ª</td>
</tr>
<tr>
<td>Left guillemot</td>
<td>«</td>
<td>«</td>
<td>&amp;larrqu;</td>
</tr>
<tr>
<td>Not sig</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>Soft hyphen</td>
<td>-</td>
<td>-</td>
<td>­</td>
</tr>
<tr>
<td>Registered trademark</td>
<td>®</td>
<td>®</td>
<td>®</td>
</tr>
<tr>
<td>Macron</td>
<td>¯</td>
<td>¯</td>
<td>¯</td>
</tr>
<tr>
<td>Degree sign</td>
<td>°</td>
<td>°</td>
<td>°</td>
</tr>
<tr>
<td>Plus or minus</td>
<td>±</td>
<td>±</td>
<td>±</td>
</tr>
<tr>
<td>Superscript 2</td>
<td>²</td>
<td>²</td>
<td>²</td>
</tr>
<tr>
<td>Superscript 3</td>
<td>³</td>
<td>³</td>
<td>³</td>
</tr>
<tr>
<td>Acute accent</td>
<td>’</td>
<td>‘</td>
<td>´</td>
</tr>
<tr>
<td>Mu</td>
<td>µ</td>
<td>µ</td>
<td>µ</td>
</tr>
<tr>
<td>Pilcrow</td>
<td>¶</td>
<td>¶</td>
<td>¶</td>
</tr>
<tr>
<td>Middle dot</td>
<td>·</td>
<td>·</td>
<td>·</td>
</tr>
<tr>
<td>Cedilla</td>
<td>¸</td>
<td>¸</td>
<td>¸</td>
</tr>
<tr>
<td>Superscript 1</td>
<td>¹</td>
<td>¹</td>
<td>¹</td>
</tr>
<tr>
<td>Masculine ordinal</td>
<td>o</td>
<td>o</td>
<td>º</td>
</tr>
<tr>
<td>Right guillemot</td>
<td>»</td>
<td>»</td>
<td>»</td>
</tr>
<tr>
<td>Fraction one-fourth</td>
<td>¼</td>
<td>¼</td>
<td>¼</td>
</tr>
<tr>
<td>Fraction one-half</td>
<td>½</td>
<td>½</td>
<td>½</td>
</tr>
<tr>
<td>Fraction three-fourths</td>
<td>¾</td>
<td>¾</td>
<td>¾</td>
</tr>
<tr>
<td>Inverted question</td>
<td>¥</td>
<td>¥</td>
<td>?</td>
</tr>
</tbody>
</table>
## UPPERCASE LATIN-1 CHARACTERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Character</th>
<th>&amp;-ASCII</th>
<th>&amp;-Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capital A, grave accent</td>
<td>À</td>
<td>À</td>
<td>À</td>
</tr>
<tr>
<td>Capital A, acute accent</td>
<td>Á</td>
<td>Á</td>
<td>Á</td>
</tr>
<tr>
<td>Capital A, circumflex accent</td>
<td>Â</td>
<td>Â</td>
<td>Â</td>
</tr>
<tr>
<td>Capital A, tilde</td>
<td>Ä</td>
<td>Ã</td>
<td>Ã</td>
</tr>
<tr>
<td>Capital A, dieresis</td>
<td>Ä</td>
<td>Ä</td>
<td>Ä</td>
</tr>
<tr>
<td>Capital A, ring</td>
<td>Å</td>
<td>Å</td>
<td>Å</td>
</tr>
<tr>
<td>Capital AE diphthong</td>
<td>Æ</td>
<td>Æ</td>
<td>Æ</td>
</tr>
<tr>
<td>Capital C, cedilla</td>
<td>Ç</td>
<td>Ç</td>
<td>Ç</td>
</tr>
<tr>
<td>Capital E, grave accent</td>
<td>È</td>
<td>È</td>
<td>È</td>
</tr>
<tr>
<td>Capital E, acute accent</td>
<td>É</td>
<td>É</td>
<td>É</td>
</tr>
<tr>
<td>Capital E, circumflex accent</td>
<td>Ê</td>
<td>Ê</td>
<td>Ê</td>
</tr>
<tr>
<td>Capital E, dieresis</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
</tr>
<tr>
<td>Capital I, grave accent</td>
<td>Ì</td>
<td>Ì</td>
<td>Ì</td>
</tr>
<tr>
<td>Capital I, acute accent</td>
<td>Í</td>
<td>Í</td>
<td>Í</td>
</tr>
<tr>
<td>Capital I, circumflex accent</td>
<td>Í</td>
<td>Î</td>
<td>Î</td>
</tr>
<tr>
<td>Capital I, dieresis</td>
<td>Ë</td>
<td>Ï</td>
<td>Ï</td>
</tr>
<tr>
<td>Capital Eth</td>
<td>Ð</td>
<td>Ð</td>
<td>Ð</td>
</tr>
<tr>
<td>Capital N, tilde</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
</tr>
<tr>
<td>Capital O, grave accent</td>
<td>Ò</td>
<td>Ò</td>
<td>Ò</td>
</tr>
<tr>
<td>Capital O, acute accent</td>
<td>Ó</td>
<td>Ó</td>
<td>Ó</td>
</tr>
<tr>
<td>Capital O, circumflex accent</td>
<td>Ô</td>
<td>Ô</td>
<td>Ô</td>
</tr>
<tr>
<td>Capital O, tilde</td>
<td>Ö</td>
<td>Õ</td>
<td>Õ</td>
</tr>
<tr>
<td>Capital O, dieresis</td>
<td>Ö</td>
<td>Ö</td>
<td>Ö</td>
</tr>
<tr>
<td>Multiply sign</td>
<td>x</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Name</td>
<td>Character</td>
<td>&amp;-ASCII</td>
<td>&amp;-Name</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Capital O, slash</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td>Capital U, grave accent</td>
<td>Ù</td>
<td>Ù</td>
<td>Ù</td>
</tr>
<tr>
<td>Capital U, acute accent</td>
<td>Ú</td>
<td>Ú</td>
<td>Ú</td>
</tr>
<tr>
<td>Capital U, circumflex accent</td>
<td>Ù</td>
<td>Û</td>
<td>Û</td>
</tr>
<tr>
<td>Capital U, dieresis</td>
<td>Ü</td>
<td>Ü</td>
<td>Ü</td>
</tr>
<tr>
<td>Capital Y, acute accent</td>
<td>Ý</td>
<td>Ý</td>
<td>Ý</td>
</tr>
<tr>
<td>Capital Thorn</td>
<td>Þ</td>
<td>Þ</td>
<td>Þ</td>
</tr>
<tr>
<td>German sz ligature</td>
<td>ß</td>
<td>ß</td>
<td>ß</td>
</tr>
</tbody>
</table>

**LOWERCASE LATIN-1 CHARACTERS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Character</th>
<th>&amp;-ASCII</th>
<th>&amp;-Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase a, grave accent</td>
<td>à</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>Lowercase a, acute accent</td>
<td>á</td>
<td>á</td>
<td>á</td>
</tr>
<tr>
<td>Lowercase a, circumflex accent</td>
<td>à</td>
<td>â</td>
<td>â</td>
</tr>
<tr>
<td>Lowercase a, tilde</td>
<td>ä</td>
<td>ã</td>
<td>ã</td>
</tr>
<tr>
<td>Lowercase a, dieresis</td>
<td>å</td>
<td>ä</td>
<td>&amp;auuml;</td>
</tr>
<tr>
<td>Lowercase a, ring</td>
<td>â</td>
<td>å</td>
<td>å</td>
</tr>
<tr>
<td>Lowercase ae ligature</td>
<td>æ</td>
<td>æ</td>
<td>æ</td>
</tr>
<tr>
<td>Lowercase c, cedilla</td>
<td>ç</td>
<td>ç</td>
<td>ç</td>
</tr>
<tr>
<td>Lowercase e, grave accent</td>
<td>è</td>
<td>è</td>
<td>è</td>
</tr>
<tr>
<td>Lowercase e, acute accent</td>
<td>é</td>
<td>é</td>
<td>é</td>
</tr>
<tr>
<td>Lowercase e, circumflex accent</td>
<td>ê</td>
<td>ê</td>
<td>ê</td>
</tr>
<tr>
<td>Lowercase e, dieresis</td>
<td>è</td>
<td>ë</td>
<td>ë</td>
</tr>
<tr>
<td>Lowercase i, grave accent</td>
<td>ì</td>
<td>ì</td>
<td>ì</td>
</tr>
<tr>
<td>Lowercase i, acute accent</td>
<td>í</td>
<td>í</td>
<td>í</td>
</tr>
<tr>
<td>Lowercase i, circumflex accent</td>
<td>î</td>
<td>î</td>
<td>î</td>
</tr>
<tr>
<td>Lowercase i, dieresis</td>
<td>ï</td>
<td>ï</td>
<td>ü</td>
</tr>
<tr>
<td>----------------------</td>
<td>---</td>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>Lowercase eth</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
</tr>
<tr>
<td>Lowercase n, tilde</td>
<td>ñ</td>
<td>ñ</td>
<td>ñ</td>
</tr>
<tr>
<td>Lowercase o, grave accent</td>
<td>ô</td>
<td>ò</td>
<td>ò</td>
</tr>
<tr>
<td>Lowercase o, acute accent</td>
<td>ó</td>
<td>ó</td>
<td>ó</td>
</tr>
<tr>
<td>Lowercase o, circumflex accent</td>
<td>ô</td>
<td>ô</td>
<td>ô</td>
</tr>
<tr>
<td>Lowercase o, tilde</td>
<td>ô</td>
<td>õ</td>
<td>õ</td>
</tr>
<tr>
<td>Lowercase o, dieresis</td>
<td>ô</td>
<td>ö</td>
<td>ö</td>
</tr>
<tr>
<td>Division sign</td>
<td>÷</td>
<td>÷</td>
<td>÷</td>
</tr>
<tr>
<td>Lowercase o, slash</td>
<td>ø</td>
<td>ø</td>
<td>ø</td>
</tr>
<tr>
<td>Lowercase u, grave accent</td>
<td>ù</td>
<td>ù</td>
<td>ù</td>
</tr>
<tr>
<td>Lowercase u, acute accent</td>
<td>ú</td>
<td>ú</td>
<td>ú</td>
</tr>
<tr>
<td>Lowercase u, circumflex accent</td>
<td>ú</td>
<td>û</td>
<td>û</td>
</tr>
<tr>
<td>Lowercase u, dieresis</td>
<td>ü</td>
<td>ü</td>
<td>ü</td>
</tr>
<tr>
<td>Lowercase y, acute accent</td>
<td>ý</td>
<td>ý</td>
<td>ý</td>
</tr>
<tr>
<td>Lowercase thorn</td>
<td>þ</td>
<td>þ</td>
<td>þ</td>
</tr>
<tr>
<td>Lowercase y, dieresis</td>
<td>ý</td>
<td>ÿ</td>
<td>ÿ</td>
</tr>
</tbody>
</table>

### 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.
**RichText**

**Category:** Beverages  
**Description:** Soft drinks, coffees, teas, beers, and ales

**Category:** Condiments  
**Description:** Sweet and savory sauces, relishes, spreads, and seasonings

⚠️ **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:
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:

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

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

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

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

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

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

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

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

You should know that in the expressions of the RichText component only plain text can be inserted this way (without formatting commands). So it is not possible to insert the RTF text. You can only assign all of its properties with 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.
<table>
<thead>
<tr>
<th>Country name</th>
<th>Flag</th>
<th>Country name</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td><img src="afghanistan_flag.png" alt="Flag" /></td>
<td>Albania</td>
<td><img src="albania_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Algeria</td>
<td><img src="algeria_flag.png" alt="Flag" /></td>
<td>American Samoa</td>
<td><img src="american_samoa_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Andorra</td>
<td><img src="andorra_flag.png" alt="Flag" /></td>
<td>Angola</td>
<td><img src="angola_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Anguilla</td>
<td><img src="anguilla_flag.png" alt="Flag" /></td>
<td>Antigua and Barbuda</td>
<td><img src="antigua_and_barbuda_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Argentina</td>
<td><img src="argentina_flag.png" alt="Flag" /></td>
<td>Armenia</td>
<td><img src="armenia_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Aruba</td>
<td><img src="aruba_flag.png" alt="Flag" /></td>
<td>Ashmore and Carter Islands</td>
<td><img src="ashmore_and_carter_islands_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Australia</td>
<td><img src="australia_flag.png" alt="Flag" /></td>
<td>Austria</td>
<td><img src="austria_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Azerbaijan</td>
<td><img src="azerbaijan_flag.png" alt="Flag" /></td>
<td>Bahamas</td>
<td><img src="bahamas_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Bahrain</td>
<td><img src="bahrain_flag.png" alt="Flag" /></td>
<td>Baker Island</td>
<td><img src="baker_island_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Bangladesh</td>
<td><img src="bangladesh_flag.png" alt="Flag" /></td>
<td>Barbados</td>
<td><img src="barbados_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Bassas da India</td>
<td><img src="bassas_da_india_flag.png" alt="Flag" /></td>
<td>Belarus</td>
<td><img src="belarus_flag.png" alt="Flag" /></td>
</tr>
<tr>
<td>Belgium</td>
<td><img src="belgium_flag.png" alt="Flag" /></td>
<td>Belize</td>
<td><img src="belize_flag.png" alt="Flag" /></td>
</tr>
</tbody>
</table>
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.

### 1. Image
- **Image**
  - [Not Assigned]
- **Data Column**
  - [Not Assigned]
- **Image Data**
  - [Not Assigned]
- **Image URL**
  - [Not Assigned]

### 2. Image Additional

### 3. Position

### 4. Appearance

### 5. Behavior
- Can Grow
  - [ ]
- Can Shrink
  - [ ]
- Grow to Height
  - [ ]
- Can Break
  - [ ]
- Dock Style
  - ![Dock Style Icons]
- Enabled
  - [ ]

### Interaction
- **Printable**
  - [ ]
- **Print on**
  - All Pages

### Shift Mode

### 6. Design

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:
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.
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:

![CanGrow=false](image1) ![CanGrow=true](image2)

**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:

![CanShrink=false](image3) ![CanShrink=true](image4)

**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.

<table>
<thead>
<tr>
<th>Soft drinks, coffees,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft drinks, coffees, teas, beers, and ales</td>
</tr>
</tbody>
</table>

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:

**CanGrow** 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:

**CanShrink** property
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:

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:

<table>
<thead>
<tr>
<th></th>
<th>Beverages</th>
<th>Condiments</th>
<th>Confections</th>
<th>Dairy Products</th>
<th>Grains,Cereals</th>
<th>Meat/Poultry</th>
<th>Produce</th>
<th>Seafood</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Soft drinks, coffees, teas, beers,</td>
<td>Sweet and savory sauces,</td>
<td>Desserts, candies, and sweet</td>
<td>Cheese</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>and ales</td>
<td>relishes, spreads, and</td>
<td>breads</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>seasonings</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 Examples](image)

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:

![Only In Width Examples](image)

Or:

![Only In Width Examples](image)

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 represents 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 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;
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 it 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, you should 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, in 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.

![Barcode Properties Panel](image)

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 you need to double-click the component to call the editor.

The Barcode editor consists of two tabs:

- The **Bar Code** tab. Select the bar code you need to use in the report. For example, the **QR Code**:
The **Settings** tab. Set up the barcode. The tabs have 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 entirely 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 bar code will have. For example, you can determine a custom value. For the 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. You should know that the larger is the amount of information in a bar code, the more graphical elements are in it.
So, if the bar code contains a large volume of data and, at the same time, you minimize the component size, the barcode reader could misread it. 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 the 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 misread. 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 barcode type, the number of
properties and their names 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:

<table>
<thead>
<tr>
<th>Graphics density</th>
<th>Printing resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very high density</td>
<td>&lt; 4 mils</td>
</tr>
<tr>
<td>High density</td>
<td>4 mils .. 6 mils</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Density</th>
<th>Width Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium density</td>
<td>7 mils .. 13 mils</td>
</tr>
<tr>
<td>Low density</td>
<td>14 mils .. 20 mils</td>
</tr>
<tr>
<td>Very low density</td>
<td>&gt; 20 mils</td>
</tr>
</tbody>
</table>
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.


### 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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>fixed, 12 characters</td>
</tr>
<tr>
<td>Check digit:</td>
<td>one, modulo-10 algorithm</td>
</tr>
</tbody>
</table>
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](image)

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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>fixed, 8 characters</td>
</tr>
<tr>
<td>Check digit:</td>
<td>one, modulo-10 algorithm</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>fixed, 13 characters</td>
</tr>
<tr>
<td>Check digit:</td>
<td>one, modulo-10 algorithm</td>
</tr>
</tbody>
</table>

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.
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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>fixed, 8 characters</td>
</tr>
<tr>
<td>Check digit:</td>
<td>one, modulo-10 algorithm</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>fixed, 2 or 5 characters</td>
</tr>
<tr>
<td>Check digit:</td>
<td>no</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th><strong>Valid symbols:</strong></th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Length:</strong></td>
<td>fixed, 14 characters</td>
</tr>
<tr>
<td><strong>Check digit:</strong></td>
<td>one, modulo-10 algorithm</td>
</tr>
</tbody>
</table>

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.
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.

3.10.3.11 ISBN-10

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

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Not variable, 10 symbols</td>
</tr>
<tr>
<td>Check digit:</td>
<td>One</td>
</tr>
</tbody>
</table>

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.


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


<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>fixed, 13 symbols</td>
</tr>
<tr>
<td>Check digit:</td>
<td>one, algorithm modulo-10</td>
</tr>
</tbody>
</table>

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.


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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>A whole number from 3 to 131070</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Variable, 1..6 characters of a digit</td>
</tr>
<tr>
<td>Check digit:</td>
<td>No</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789ABCDEF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Variable</td>
</tr>
<tr>
<td>Check digit:</td>
<td>No, one or two;</td>
</tr>
</tbody>
</table>
**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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Variable</td>
</tr>
<tr>
<td>Check digit:</td>
<td>none, one or two; algorithm modulo-10 or modulo-11</td>
</tr>
</tbody>
</table>

**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 has 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.

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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Variable</td>
</tr>
<tr>
<td>Check digit:</td>
<td>no</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Variable, even</td>
</tr>
<tr>
<td>Check digit:</td>
<td>No</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>ABCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Fixed, 1 symbol</td>
</tr>
<tr>
<td>Check digit:</td>
<td>No</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789 - $ : / . + ABCD (only as start/stop symbols)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Variable</td>
</tr>
<tr>
<td>Check digit:</td>
<td>no</td>
</tr>
</tbody>
</table>

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.
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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Fixed, 5, 9 or 11 characters</td>
</tr>
<tr>
<td>Check digit:</td>
<td>One, algorithm modulo-10</td>
</tr>
</tbody>
</table>

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.](image)

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

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>FCC - fixed, 2 characters, DPID - fixed, 8 characters, CustomerInfo variable</td>
</tr>
<tr>
<td>Check digit:</td>
<td>Four, ReedSolomon algorithm</td>
</tr>
</tbody>
</table>

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 is 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 TP 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.

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ABCDEFGHIJKLMNOPQRSTUVWXYZ</td>
</tr>
<tr>
<td>Length:</td>
<td>Variable</td>
</tr>
<tr>
<td>Check digit:</td>
<td>none</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>numeric characters 0-9; alpha characters A-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Length:</strong></td>
<td>Variable</td>
</tr>
<tr>
<td><strong>Check digit:</strong></td>
<td>none</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Valid symbols:</th>
<th>0123456789 -</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Variable</td>
</tr>
<tr>
<td>Check digit:</td>
<td>None, one or two; modulo-10 algorithm</td>
</tr>
</tbody>
</table>

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  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ABCDEFGHIJKLMNOPQRSTUVWXYZ</td>
</tr>
<tr>
<td></td>
<td>-./+% space</td>
</tr>
<tr>
<td><strong>Length:</strong></td>
<td>Variable</td>
</tr>
<tr>
<td><strong>Check digit:</strong></td>
<td>Two, algorithm modulo-47</td>
</tr>
</tbody>
</table>

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.

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 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Length:</td>
<td>Variable</td>
</tr>
<tr>
<td>Check digit:</td>
<td>One, algorithm modulo-103</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Check symbols</th>
<th>Checksum algorithm</th>
<th>0-9</th>
<th>A-Z</th>
<th>a-z</th>
<th>other symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPC-A</td>
<td>12</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPC-E</td>
<td>8</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAN-13</td>
<td>13</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAN-8</td>
<td>8</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAN-128a</td>
<td>var</td>
<td>1</td>
<td>modulo-103</td>
<td>+</td>
<td>+</td>
<td></td>
<td>ASCII 0 to 95</td>
</tr>
<tr>
<td>EAN-128b</td>
<td>var</td>
<td>1</td>
<td>modulo-103</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>ASCII 32 to 127</td>
</tr>
<tr>
<td>EAN-128c</td>
<td>var</td>
<td>1</td>
<td>modulo-103</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>ITF-14</td>
<td>14</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAN-13</td>
<td>13</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAN-8</td>
<td>8</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISBN-10</td>
<td>10</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Length</td>
<td>Digits</td>
<td>Modulo</td>
<td>Checksum</td>
<td>Characters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>--------</td>
<td>--------------</td>
<td>----------</td>
<td>------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISBN-13</td>
<td>13</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pharmaco de</td>
<td>1..6</td>
<td>-</td>
<td>-</td>
<td>int 3..13, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plessey</td>
<td>var</td>
<td>0-2</td>
<td>modulo-10/11</td>
<td>+ A B C D E F</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Msi</td>
<td>var</td>
<td>0-2</td>
<td>modulo-10/11</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2of5 Standard</td>
<td>var</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2of5 Interleaved</td>
<td>var</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIM</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>A B C D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Codabar</td>
<td>var</td>
<td>-</td>
<td>-</td>
<td>+ $ / . +</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Postnet</td>
<td>5, 9, 11</td>
<td>1</td>
<td>modulo-10</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Australia Post</td>
<td>10[+var]</td>
<td>4</td>
<td>ReedSolomon</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 11</td>
<td>var</td>
<td>0-2, A</td>
<td>modulo-11</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 39</td>
<td>var</td>
<td>0-1</td>
<td>modulo-43</td>
<td>+</td>
<td>-$ / + % space</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 39 ext</td>
<td>var</td>
<td>0-1</td>
<td>modulo-43</td>
<td>+</td>
<td>+ full ASCII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 93</td>
<td>var</td>
<td>2</td>
<td>modulo-47</td>
<td>+</td>
<td>-.$/+ % space</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 93 ext</td>
<td>var</td>
<td>2</td>
<td>modulo-47</td>
<td>+</td>
<td>+ full ASCII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code128a</td>
<td>var</td>
<td>1</td>
<td>modulo-103</td>
<td>+</td>
<td>ASCII 0 to 95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code128b</td>
<td>var</td>
<td>1</td>
<td>modulo-</td>
<td>+</td>
<td>ASCII 32 to</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reports and Dashboards

<table>
<thead>
<tr>
<th>Code128c</th>
<th>var</th>
<th>103</th>
<th>modulo-103</th>
<th>+</th>
</tr>
</thead>
</table>

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.

- Plessey
- MSI
- 2of5 Standard
- 2of5 Interleaved
- Codabar
- Code11
- Code39
- Code93
- Code128a
- Code128c

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.

<table>
<thead>
<tr>
<th>Encoding mode</th>
<th>Valid symbols</th>
<th>Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte</td>
<td>ASCII 0 to 255</td>
<td>1,2 bytes per word</td>
</tr>
<tr>
<td>Text</td>
<td>ASCII 9,10,13 &amp; 32-127</td>
<td>2 characters per word</td>
</tr>
<tr>
<td>Numeric</td>
<td>0123456789</td>
<td>2,9 digits per word</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Level of Error Correction</th>
<th>Number of Codewords</th>
</tr>
</thead>
</table>
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 $8 \times 8$ to $144 \times 144$. 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:
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 $8 \times 8$ to $144 \times 144$. All available combinations of sizes is shown on the table below:

<table>
<thead>
<tr>
<th>Barcode size</th>
<th>Length, bites</th>
<th>Barcode size</th>
<th>Length, bites</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 × 10</td>
<td>3</td>
<td>32 × 32</td>
<td>62</td>
</tr>
<tr>
<td>12 × 12</td>
<td>5</td>
<td>36 × 36</td>
<td>86</td>
</tr>
<tr>
<td>8 × 18</td>
<td>5</td>
<td>40 × 40</td>
<td>114</td>
</tr>
<tr>
<td>14 × 14</td>
<td>8</td>
<td>44 × 44</td>
<td>144</td>
</tr>
<tr>
<td>8 × 32</td>
<td>10</td>
<td>48 × 48</td>
<td>174</td>
</tr>
<tr>
<td>16 × 16</td>
<td>12</td>
<td>52 × 52</td>
<td>204</td>
</tr>
<tr>
<td>12 × 26</td>
<td>16</td>
<td>64 × 64</td>
<td>280</td>
</tr>
<tr>
<td>18 × 18</td>
<td>18</td>
<td>72 × 72</td>
<td>368</td>
</tr>
<tr>
<td>20 × 20</td>
<td>22</td>
<td>80 × 80</td>
<td>456</td>
</tr>
<tr>
<td>12 × 36</td>
<td>22</td>
<td>88 × 88</td>
<td>576</td>
</tr>
<tr>
<td>22 × 22</td>
<td>30</td>
<td>96 × 96</td>
<td>696</td>
</tr>
<tr>
<td>16 × 36</td>
<td>32</td>
<td>104 × 104</td>
<td>816</td>
</tr>
<tr>
<td>24 × 24</td>
<td>36</td>
<td>120 × 120</td>
<td>1050</td>
</tr>
<tr>
<td>26 × 26</td>
<td>44</td>
<td>132 × 132</td>
<td>1304</td>
</tr>
<tr>
<td>16 × 48</td>
<td>49</td>
<td>144 × 144</td>
<td>1558</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Encoding mode</th>
<th>Valid symbols</th>
<th>Bits per symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>ASCII character 0 to 127</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>ASCII character 128 to 255</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>ASCII numeric</td>
<td>4</td>
</tr>
<tr>
<td>C40</td>
<td>Upper-case alphanumeric</td>
<td>5,33</td>
</tr>
<tr>
<td></td>
<td>Lower-case letters and punctuation</td>
<td>10,66</td>
</tr>
<tr>
<td>TEXT</td>
<td>Lower-case alphanumeric</td>
<td>5,33</td>
</tr>
<tr>
<td></td>
<td>Upper-case letters and punctuation</td>
<td>10,66</td>
</tr>
<tr>
<td>X12</td>
<td>ANSI X12</td>
<td>5,33</td>
</tr>
<tr>
<td>EDIFACT</td>
<td>ASCII character 32 to 94</td>
<td>6</td>
</tr>
<tr>
<td>BASE 256</td>
<td>ASCII character 0 to 255</td>
<td>8</td>
</tr>
</tbody>
</table>

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):

<table>
<thead>
<tr>
<th>Numeric mode:</th>
<th>0123456789</th>
<th>Maximum 7089 characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphanumeric mode:</td>
<td>ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 $ % * + -. / : space</td>
<td>Maximum 4296 characters</td>
</tr>
<tr>
<td>Binary mode (8 bits byte data):</td>
<td>JIS 8-bit (Latin and Kana)</td>
<td>Maximum 2953 bytes</td>
</tr>
<tr>
<td>Kanji mode:</td>
<td>Shift JIS (8140H-9FFCH and E040H-EBBFH)</td>
<td>Maximum 1817 characters</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Correction Level</th>
<th>Percentage of the recovered information</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>7%</td>
</tr>
<tr>
<td>M</td>
<td>15%</td>
</tr>
<tr>
<td>Q</td>
<td>25%</td>
</tr>
<tr>
<td>H</td>
<td>30%</td>
</tr>
</tbody>
</table>

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.

![Diagram of Report Rendering]

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.
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**.

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.

![Margins Diagram]

**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.

| Margins | 1;2;3;0 |

**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.

| Margins |  
|---------|-------------------------------------------------|
| Left    | 1                                               |
| Right   | 2                                               |
| Top     | 3                                               |
| Bottom  | 0                                               |

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

Stimulsoft Reports builds its reports using bands (sometimes bands are called sections in other products). A band consists 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. To become familiar with how they are processed when rendering the report, please read the article [Rendering Order of Bands](#).

<table>
<thead>
<tr>
<th>Icon</th>
<th>Band Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Report Title</td>
<td>This band is printed in the beginning of a report</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Report Summary</td>
<td>This band is printed in the end of a report</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Page Header</td>
<td>This band is printed on the top of each page</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Page Footer</td>
<td>This band is printed on the bottom of each page</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Group Header</td>
<td>This band is printed in the beginning of a group</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Group Footer</td>
<td>This band is printed in the end of a group</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Header</td>
<td>This band is printed before data</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Footer</td>
<td>This band is printed after data</td>
</tr>
<tr>
<td>Band Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Column Header</td>
<td>This band is printed before a column is output</td>
<td></td>
</tr>
<tr>
<td>Column Footer</td>
<td>This band is printed after a column is output</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>This band is printed as many times as there are rows in the data source</td>
<td></td>
</tr>
<tr>
<td>Hierarchical Data</td>
<td>This band is printed as many times as there are rows in the data source. Data items are output as a tree</td>
<td></td>
</tr>
<tr>
<td>Child</td>
<td>This band is printed only once, after the band beneath which it is placed</td>
<td></td>
</tr>
<tr>
<td>Empty Data</td>
<td>Fills the free space at the bottom of a page</td>
<td></td>
</tr>
<tr>
<td>Overlay</td>
<td>This band is printed on the background of a page. It does not effect on other bands.</td>
<td></td>
</tr>
</tbody>
</table>

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:
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:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>Cross-Group Header</td>
<td>This band is printed in the beginning of a group</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cross-Group Footer</td>
<td>This band is printed in the end of a group</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cross-Header</td>
<td>This band is printed before data</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cross-Footer</td>
<td>This band is printed after data</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cross-Data</td>
<td>This band is printed as many times as there are rows in the data source</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Order</th>
<th>Band name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Page Header</td>
<td>On each page. Output on the first page is optional.</td>
</tr>
<tr>
<td>2</td>
<td>Report Title</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>Header, Column Header</td>
<td>Once before data output (for the Column Header - once for every column. Output on each new</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>4</td>
<td><strong>Group Header</strong></td>
<td>At the beginning of each group. Output on each new page is optional.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Data</strong></td>
<td>Once for every row of data.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Empty Band</strong></td>
<td>For each empty row on every page of the report.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Group Footer</strong></td>
<td>At the end of each group.</td>
</tr>
<tr>
<td>8</td>
<td><strong>Footer, Column Footer</strong></td>
<td>After all data has been output (for the Column Footer - once for every column). Output on each new page is optional.</td>
</tr>
<tr>
<td>9</td>
<td><strong>Report Summary</strong></td>
<td>Once at the end of a report.</td>
</tr>
<tr>
<td>10</td>
<td><strong>Overlay</strong></td>
<td>Once on every page of the report.</td>
</tr>
<tr>
<td>11</td>
<td><strong>Page Footer</strong></td>
<td>On every page. Output on the first page is optional.</td>
</tr>
</tbody>
</table>

**Information:** Components placed directly on the page (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.

  **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.

<table>
<thead>
<tr>
<th>DataBand2: Data Source: 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

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

⚠️ **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.

<table>
<thead>
<tr>
<th>Order</th>
<th>Band name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HeaderBand3</td>
</tr>
<tr>
<td>2</td>
<td>HeaderBand2</td>
</tr>
<tr>
<td>3</td>
<td>DataBand2</td>
</tr>
<tr>
<td>4</td>
<td>FooterBand3</td>
</tr>
<tr>
<td>5</td>
<td>FooterBand2</td>
</tr>
</tbody>
</table>

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.

| DataBand1: Data Source: Categories |
| {Categories.CategoryName} | {Categories.Description} |

| DataBand2: Data Source: Products |
| {Products.ProductName} | {Products.UnitPrice} | {Products.UnitsInStock} |

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).
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.
As mentioned above, footers in the report template refer 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.

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

<table>
<thead>
<tr>
<th>ProductName</th>
<th>UnitPrice</th>
<th>UnitsInStock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>18</td>
<td>39</td>
</tr>
<tr>
<td>Chang</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>22</td>
<td>53</td>
</tr>
<tr>
<td>Chef Anton's Gumbo Mix</td>
<td>21.35</td>
<td>0</td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>25</td>
<td>120</td>
</tr>
<tr>
<td>Uncle Bob's Organic Dried Pears</td>
<td>30</td>
<td>16</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>97</td>
<td>29</td>
</tr>
<tr>
<td>Ikura</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>Queso Cabrales</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>Queso Manchego La Pastora</td>
<td>38</td>
<td>86</td>
</tr>
<tr>
<td>Konbu</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>Tofu</td>
<td>23.25</td>
<td>35</td>
</tr>
<tr>
<td>Genen Shouyu</td>
<td>15.5</td>
<td>39</td>
</tr>
<tr>
<td>Pavlova</td>
<td>17.45</td>
<td>29</td>
</tr>
<tr>
<td>Alice Mutton</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>Carnarvon Tigers</td>
<td>62.5</td>
<td>42</td>
</tr>
<tr>
<td>Teatime Chocolate Biscuits</td>
<td>9.2</td>
<td>25</td>
</tr>
<tr>
<td>Sir Rodney's Marmalade</td>
<td>81</td>
<td>40</td>
</tr>
<tr>
<td>Sir Rodney's Scones</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Gustaf's Knäckebröd</td>
<td>21</td>
<td>104</td>
</tr>
<tr>
<td>Tunnbröd</td>
<td>9</td>
<td>61</td>
</tr>
</tbody>
</table>
The report page will look the following way.
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.

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

### DataBand1
**List of Categories**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>18</td>
</tr>
<tr>
<td>Chang</td>
<td>19</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>10</td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>22</td>
</tr>
<tr>
<td>Chef Anton's Gumbo Mix</td>
<td>21.35</td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>25</td>
</tr>
<tr>
<td>Uncle Bob's Organic Dried Pears</td>
<td>30</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>40</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>87</td>
</tr>
<tr>
<td>Ikura</td>
<td>31</td>
</tr>
<tr>
<td>Queso Cabrales</td>
<td>21</td>
</tr>
<tr>
<td>Queso Manchego La Pastora</td>
<td>38</td>
</tr>
<tr>
<td>Konbu</td>
<td>6</td>
</tr>
<tr>
<td>Tofu</td>
<td>23.25</td>
</tr>
<tr>
<td>Genen Shouyu</td>
<td>16.5</td>
</tr>
<tr>
<td>Pavlova</td>
<td>17.45</td>
</tr>
<tr>
<td>Alice Mutton</td>
<td>39</td>
</tr>
<tr>
<td>Carnarvon Tigers</td>
<td>62.5</td>
</tr>
<tr>
<td>Teatime Chocolate Biscuits</td>
<td>9.2</td>
</tr>
<tr>
<td>Sir Rodney's Marmalade</td>
<td>81</td>
</tr>
<tr>
<td>Sir Rodney's Scones</td>
<td>10</td>
</tr>
<tr>
<td>Gustaf's Knäckebröd</td>
<td>21</td>
</tr>
</tbody>
</table>

### DataBand2
**List of Products**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drink</td>
<td>39</td>
</tr>
<tr>
<td>Meat</td>
<td>17</td>
</tr>
<tr>
<td>Confection</td>
<td>13</td>
</tr>
<tr>
<td>Dairy Product</td>
<td>53</td>
</tr>
<tr>
<td>Grains/Cereal</td>
<td>0</td>
</tr>
<tr>
<td>Meat/Poultry</td>
<td>120</td>
</tr>
<tr>
<td>Produce</td>
<td>6</td>
</tr>
<tr>
<td>Seafood</td>
<td>3</td>
</tr>
<tr>
<td>Totals</td>
<td>104</td>
</tr>
</tbody>
</table>
And then the first page of the report will look the following way.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

And then the first page of the report will look the following way.
<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beverages</td>
<td>Soft drinks, coffees, teas, beers, and ales</td>
</tr>
<tr>
<td>Condiments</td>
<td>Sweet and savory sauces, relishes, spreads, seasonings</td>
</tr>
<tr>
<td>Confections</td>
<td>Desserts, candies, and sweet breads</td>
</tr>
<tr>
<td>Dairy Products</td>
<td>Cheeses</td>
</tr>
<tr>
<td>Grains/Cereals</td>
<td>Breads, crackers, pasta, and cereal</td>
</tr>
<tr>
<td>Meat/Poultry</td>
<td>Prepared meats</td>
</tr>
<tr>
<td>Produce</td>
<td>Dried fruit and bean curd</td>
</tr>
<tr>
<td>Seafood</td>
<td>Seaweed and fish</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ProductName</th>
<th>UnitPrice</th>
<th>UnitsInStock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>18</td>
<td>39</td>
</tr>
<tr>
<td>Chang</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>22</td>
<td>15</td>
</tr>
<tr>
<td>Chef Anton's Gumbo Mix</td>
<td>21.35</td>
<td>0</td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>25</td>
<td>120</td>
</tr>
<tr>
<td>Uncle Bob's Organic Dried Pears</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>97</td>
<td>29</td>
</tr>
<tr>
<td>Ikura</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>Queso Cabrales</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>Queso Manchego La Pastora</td>
<td>38</td>
<td>86</td>
</tr>
<tr>
<td>Konbu</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>Tofu</td>
<td>23.25</td>
<td>35</td>
</tr>
<tr>
<td>Genen Shouyu</td>
<td>16.5</td>
<td>39</td>
</tr>
<tr>
<td>Pavlova</td>
<td>17.45</td>
<td>29</td>
</tr>
<tr>
<td>Alice Mutton</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>Carnarvon Tigers</td>
<td>62.5</td>
<td>42</td>
</tr>
<tr>
<td>Teatime Chocolate Biscuits</td>
<td>9.2</td>
<td>25</td>
</tr>
<tr>
<td>Sir Rodney's Marmalade</td>
<td>81</td>
<td>40</td>
</tr>
<tr>
<td>Sir Rodney's Scones</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

Sum Total: 22

**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, all headers located above some footer band or another data band. Footers that relate to this data band are defined next, 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.
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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beverages</td>
<td>Soft drinks, coffees, teas, beers, and alcohols</td>
</tr>
<tr>
<td>Condiments</td>
<td>Sweet and savory sauces, relishes, spices, seasonings</td>
</tr>
<tr>
<td>Confections</td>
<td>Desserts, candies, and sweet breads</td>
</tr>
<tr>
<td>Dairy Products</td>
<td>Cheeses</td>
</tr>
<tr>
<td>Grains/Cereals</td>
<td>Breads, crackers, pastas, and cereals</td>
</tr>
<tr>
<td>Meat/Poultry</td>
<td>Prepared meats</td>
</tr>
<tr>
<td>Produce</td>
<td>Dried fruit and bean curd</td>
</tr>
<tr>
<td>Seafood</td>
<td>Seaweed and fish</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>18</td>
<td>39</td>
</tr>
<tr>
<td>Chang</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>22</td>
<td>53</td>
</tr>
<tr>
<td>Chef Anton's Gumbo Mix</td>
<td>21.25</td>
<td>0</td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>25</td>
<td>120</td>
</tr>
<tr>
<td>Uncle Bob's Organic Dried Pears</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>Mishi Kobe Niku</td>
<td>97</td>
<td>29</td>
</tr>
<tr>
<td>Ikura</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>Queso Cabrales</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>Queso Manchego La Pastora</td>
<td>38</td>
<td>86</td>
</tr>
<tr>
<td>Konbu</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>Tofu</td>
<td>23.25</td>
<td>35</td>
</tr>
<tr>
<td>Genen Shoyyu</td>
<td>15.5</td>
<td>35</td>
</tr>
<tr>
<td>Pavlova</td>
<td>17.45</td>
<td>29</td>
</tr>
<tr>
<td>Alice Mutton</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>Carnarvon Tigers</td>
<td>62.5</td>
<td>42</td>
</tr>
<tr>
<td>Teatime Chocolate Biscuits</td>
<td>9.2</td>
<td>25</td>
</tr>
<tr>
<td>Sir Rodney's Marmalade</td>
<td>81</td>
<td>40</td>
</tr>
<tr>
<td>Sir Rodney's Scones</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Gustaf's Knäckebröd</td>
<td>21</td>
<td>104</td>
</tr>
<tr>
<td>Tunnbröd</td>
<td>9</td>
<td>61</td>
</tr>
</tbody>
</table>
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) 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 band 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 band 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.

**Virtual Data Band**

Sometimes it is necessary to print a Data band several times without specifying a data source. The CountData property is used for this purpose.
It is possible to specify number of elements in the **Data** band editor. On the picture below the **Data** editor is shown.

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

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

### 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.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>[Not Assigned]</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Object</td>
<td>[Data.Categories.Products]</td>
<td>...</td>
</tr>
</tbody>
</table>

**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.
<table>
<thead>
<tr>
<th>Data Source</th>
<th>Count Data: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation</td>
<td></td>
</tr>
<tr>
<td>Master Component</td>
<td></td>
</tr>
<tr>
<td>Sort</td>
<td></td>
</tr>
<tr>
<td>Filters</td>
<td></td>
</tr>
</tbody>
</table>

[Not Assigned]
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.
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.

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Address</th>
<th>Phone</th>
<th>Contact Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
<td>Obere Str. 67</td>
<td>030-6074321</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Ana Trujillo Emparedados y helados</td>
<td>Apdo. de la Constitución 2222</td>
<td>(6) 666-4720</td>
<td>Owner</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>Mataderos 2312</td>
<td>(5) 666-3832</td>
<td>Owner</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>120 Hanover Sq.</td>
<td>(171) 666-7718</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Berglunds snabbklop</td>
<td>Berguvsvägen 8</td>
<td>0921-12 34 65</td>
<td>Order Administrator</td>
</tr>
<tr>
<td>Bauer See Delikatessen</td>
<td>Forsterstr. 57</td>
<td>0821-08460</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Bondesøssl père et fils</td>
<td>24, place Klåber</td>
<td>88.66.15.31</td>
<td>Marketing Manager</td>
</tr>
<tr>
<td>Bildo Comidas preparadas</td>
<td>C/ Araquil, 67</td>
<td>(91) 665 22 82</td>
<td>Owner</td>
</tr>
<tr>
<td>Bin app’</td>
<td>12, rue des Bouchers</td>
<td>91.24.46.40</td>
<td>Owner</td>
</tr>
<tr>
<td>Bottom-Dollar Markets</td>
<td>23 Tsawassen Blvd.</td>
<td>(604) 665-4729</td>
<td>Accounting Manager</td>
</tr>
<tr>
<td>B’s Beverages</td>
<td>Fauntleroy Circus</td>
<td>(171) 665-1212</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Cactus Comidas para llevar</td>
<td>Centro 333</td>
<td>(1) 125-5555</td>
<td>Sales Agent</td>
</tr>
<tr>
<td>Centro comercial Moctezuma</td>
<td>Sierras de Granada 9993</td>
<td>(5) 665-3392</td>
<td>Marketing Manager</td>
</tr>
<tr>
<td>Chop-suey Chinese</td>
<td>Hauptstr. 28</td>
<td>0462-076356</td>
<td>Owner</td>
</tr>
<tr>
<td>Comério Mineiro</td>
<td>Av. dos Luídias, 23</td>
<td>(11) 665-7647</td>
<td>Sales Associate</td>
</tr>
<tr>
<td>Consolidated Holdings</td>
<td>Berkeley Gardens 12 Brewery</td>
<td>(171) 665-2262</td>
<td>Sales Associate</td>
</tr>
<tr>
<td>Drachenblut Delikatessen</td>
<td>Walsenweg 21</td>
<td>0241-039123</td>
<td>Order Administrator</td>
</tr>
<tr>
<td>Du monde entier</td>
<td>87, rue des Cinquante Etapes</td>
<td>40.67.88.88</td>
<td>Owner</td>
</tr>
</tbody>
</table>

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.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chai</td>
<td>10 boxes x 20 bags</td>
<td>39,00</td>
</tr>
<tr>
<td>2</td>
<td>Chang</td>
<td>24 - 12 oz bottles</td>
<td>17,00</td>
</tr>
<tr>
<td>3</td>
<td>Chartreuse verte</td>
<td>750 cc per bottle</td>
<td>69,00</td>
</tr>
<tr>
<td>4</td>
<td>Côte de Blaye</td>
<td>12 - 75 cl bottles</td>
<td>17,00</td>
</tr>
<tr>
<td>5</td>
<td>Guaraná Fantástica</td>
<td>12 - 355 ml cans</td>
<td>20,00</td>
</tr>
<tr>
<td>6</td>
<td>Ipoh Coffee</td>
<td>16 - 500 g tins</td>
<td>17,00</td>
</tr>
<tr>
<td>7</td>
<td>Lakkaliküööri</td>
<td>500 ml</td>
<td>57,00</td>
</tr>
<tr>
<td>8</td>
<td>Laughing Lumberjack Lager</td>
<td>24 - 12 oz bottles</td>
<td>52,00</td>
</tr>
<tr>
<td>9</td>
<td>Outback Lager</td>
<td>24 - 355 ml bottles</td>
<td>15,00</td>
</tr>
<tr>
<td>10</td>
<td>Rhönbräu Klosterbier</td>
<td>24 - 0.5 l bottles</td>
<td>125,00</td>
</tr>
<tr>
<td>11</td>
<td>Sasquatch Ale</td>
<td>24 - 12 oz bottles</td>
<td>111,00</td>
</tr>
<tr>
<td>12</td>
<td>Steelyeye Stout</td>
<td>24 - 12 oz bottles</td>
<td>20,00</td>
</tr>
</tbody>
</table>

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:

\{	extbf{Line}\}.{\textbf{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:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Chai</td>
<td>10 boxes x 20 bags</td>
<td>39,00</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Chang</td>
<td>24 - 12 oz bottles</td>
<td>17,00</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Chartreuse verte</td>
<td>750 cc per bottle</td>
<td>69,00</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Côte de Blaye</td>
<td>12 - 75 cl bottles</td>
<td>17,00</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Guaraná Fantástica</td>
<td>12 - 355 ml cans</td>
<td>20,00</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Ipoh Coffee</td>
<td>16 - 500 g tins</td>
<td>17,00</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Lakkalikööri</td>
<td>500 ml</td>
<td>57,00</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Laughing Lumberjack Lager</td>
<td>24 - 12 oz bottles</td>
<td>52,00</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Outback Lager</td>
<td>24 - 355 ml bottles</td>
<td>15,00</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>Rhönbräu Klosterbier</td>
<td>24 - 0.5 l bottles</td>
<td>125,00</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>Sasquatch Ale</td>
<td>24 - 12 oz bottles</td>
<td>111,00</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>Steeleye Stout</td>
<td>24 - 12 oz bottles</td>
<td>20,00</td>
<td></td>
</tr>
</tbody>
</table>

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.

**Simple List**

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfred's Fish</td>
<td>Erie St. 57</td>
<td>030-407-4221</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Alex's Taqueria</td>
<td>Avenida de la Constitución 2222</td>
<td>055-4739</td>
<td>Owner</td>
</tr>
<tr>
<td>André's Taquero</td>
<td>Montefrío 2312</td>
<td>055-3602</td>
<td>Owner</td>
</tr>
<tr>
<td>Arduino's Heros</td>
<td>120 Handers Sq.</td>
<td>(717) 555-7188</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Beer's Warehouse</td>
<td>Seaport 98</td>
<td>0521-123-456</td>
<td>Owner/Manager</td>
</tr>
<tr>
<td>Baker's Deli</td>
<td>Freedom St. 57</td>
<td>082-168460</td>
<td>Sales Representative</td>
</tr>
</tbody>
</table>

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**:

```javascript
myvariable = 0;
```

Also it is necessary to create the **myvariable** variable in the data dictionary. Write the following code in the **RenderingEvent**:

```javascript
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:
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 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.

![Data Sorting Editor](image-url)

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.

![Bookmark Structure](image-url)
The Sort bookmark;
The button to add a new level of sorting;
The button to remove the selected level of sorting;
Move the selected level of sorting upwards;
Move the selected level of sorting downwards;
Level of sorting;
The column or expression which are used for sorting;
The button to add or edit expressions of the sorting level;
The button the select a column for sorting;
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.

<table>
<thead>
<tr>
<th>ProductName</th>
</tr>
</thead>
<tbody>
<tr>
<td>✕ [No Sorting]</td>
</tr>
<tr>
<td>☑ Products</td>
</tr>
<tr>
<td>☑ ProductID</td>
</tr>
<tr>
<td>☑ ProductName</td>
</tr>
<tr>
<td>☑ SupplierID</td>
</tr>
<tr>
<td>☑ CategoryID</td>
</tr>
<tr>
<td>☑ QuantityPerUnit</td>
</tr>
<tr>
<td>☑ UnitPrice</td>
</tr>
<tr>
<td>☑ UnitsInStock</td>
</tr>
<tr>
<td>☑ Discontinued</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>FilterOn</th>
<th>Filters</th>
</tr>
</thead>
</table>
|          | [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.

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Data Type</th>
<th>Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>String</td>
<td>[No]</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Data Type</th>
<th>Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>String</td>
<td>[No]</td>
</tr>
</tbody>
</table>

1 The way of selecting a condition;
2 This field specifies the type of data with what the condition will work. There are five types of data: String, Numeric, DateTime, Boolean, Expression. Data type has affect
on how the reporting tool processes a condition. For example, if the data type is a string, then the method of work with strings is used. In addition, depending on the data type the list of available operations of conditions is changed. For example, only for the **String** data type is **Containing** operation is available;

3. The column of the data source is specified in the field. The value from this column will be used as the first value of a condition;

4. The type of operation, using what the calculation of the value of a condition is done. All available types of operation are grouped in the table and shown on the picture below;

5. The second value of a condition of a filter. It is required to specify two values for some operations. For example, for the **between** operation it is required to specify two values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>DateTime</td>
<td>[No]</td>
</tr>
<tr>
<td>between</td>
<td>04.02.2012</td>
<td>And 04.02.2013</td>
</tr>
</tbody>
</table>

The table below shows operations and their description for each data type.

<table>
<thead>
<tr>
<th>Name of operation</th>
<th>Types of data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equal to</td>
<td>✔️ ✔️ ✔️ ✔️ ✔️</td>
<td>If the first value is equal to the second value, then the condition is true.</td>
</tr>
<tr>
<td>not equal</td>
<td>✔️ ✔️ ✔️ ✔️ ✔️</td>
<td>If the first value is not</td>
</tr>
<tr>
<td>Operation</td>
<td>Condition 1</td>
<td>Condition 2</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>between</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>not between</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>greater than</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>greater than or equal to</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>less than</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>less then or equal to</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>containing</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>not containing</td>
<td>If the first value does not contain the second value, then the condition is true. This operation can be applied only to strings.</td>
</tr>
<tr>
<td>beginning with</td>
<td>If the first value begins with the second value, then the condition is true. This operation can be applied only to strings.</td>
</tr>
<tr>
<td>ending with</td>
<td>If the first value ends with the second value, then the condition is true. This operation can be applied only to strings.</td>
</tr>
</tbody>
</table>

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:

![Field Js](image)

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#:

```csharp
expression
```
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.

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Crocker Box</td>
<td>55 Grizzly Peak Rd.</td>
<td>(406) 665-5834</td>
<td>Marketing Assistant</td>
</tr>
<tr>
<td>Toms Spezialitäten</td>
<td>Luisenstr. 40</td>
<td>0251-001259</td>
<td>Marketing Manager</td>
</tr>
<tr>
<td>Tortuga Restaurante</td>
<td>Avda. Ateca 123</td>
<td>(5) 665-2033</td>
<td>Owner</td>
</tr>
<tr>
<td>Tradição Hipermarcados</td>
<td>Av. Inês de Castro, 414</td>
<td>(11) 665-2167</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Tail’s Head Gourmet Provisioners</td>
<td>722 DaMnsi Blvd.</td>
<td>(206) 665-8257</td>
<td>Sales Associate</td>
</tr>
<tr>
<td>Valiesgut</td>
<td>Stragloges 45</td>
<td>03 21 22 31</td>
<td>Sales Manager</td>
</tr>
<tr>
<td>Victuailles en stock</td>
<td>2, rue du Commerce</td>
<td>78 32 64 88</td>
<td>Sales Agent</td>
</tr>
<tr>
<td>Mls et alcools Chevalier</td>
<td>59 rue de l’Abbaye</td>
<td>20 47 15 11</td>
<td>Accounting Manager</td>
</tr>
<tr>
<td>Wartian Heriku</td>
<td>Torikatu 38</td>
<td>981-449355</td>
<td>Accounting Manager</td>
</tr>
<tr>
<td>Wellington Importadores</td>
<td>Rua do Mercado, 12</td>
<td>(14) 666-0122</td>
<td>Sales Manager</td>
</tr>
<tr>
<td>White Clover Markets</td>
<td>305 - 14th Ave. S. Suite 38</td>
<td>(206) 665-4112</td>
<td>Owner</td>
</tr>
<tr>
<td>Wllman Valla</td>
<td>Keskuskatu 46</td>
<td>90 224 0060</td>
<td>Owner/Marketing Assistant</td>
</tr>
<tr>
<td>Wolski Zajajd</td>
<td>ul. Filtowa 66</td>
<td>(28) 642-7012</td>
<td>Owner</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Category</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Mutton</td>
<td>Meat/Poultry</td>
<td>39</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>Condiments</td>
<td>10</td>
</tr>
<tr>
<td>Boston Crab Meat</td>
<td>Seafood</td>
<td>18.4</td>
</tr>
<tr>
<td>Camembert Pirot</td>
<td>Dairy Products</td>
<td>34</td>
</tr>
<tr>
<td>Camaron Von Tigres</td>
<td>Seafood</td>
<td>62.5</td>
</tr>
<tr>
<td>Chai</td>
<td>Beverages</td>
<td>18</td>
</tr>
</tbody>
</table>

© 2003-2020 Stimulsoft
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:
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.

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.

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

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

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

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

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 row of the Master band.

**Data Relation**: Categories

Selection of relation occurs using the **Data** band editor, as well as in case with the **MasterComponent** property.

Selection is done between relations which were created between Master and Detail data.
sources, and in which the Detail data source is subordinate data source. There can be more than one relation (for example, as seen on the picture above). Therefore, it is important to select the correct relation.

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. 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

<table>
<thead>
<tr>
<th>CategoryID</th>
<th>CategoryName</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beverages</td>
<td>Soft drinks, coffees, teas, beer</td>
</tr>
<tr>
<td>2</td>
<td>Condiments</td>
<td>Sweet and savory sauces</td>
</tr>
<tr>
<td>3</td>
<td>Confections</td>
<td>Desserts, candies, and sweet</td>
</tr>
<tr>
<td>4</td>
<td>Dairy Products</td>
<td>Cheeses</td>
</tr>
<tr>
<td>5</td>
<td>Grains/Cereals</td>
<td>Breads, crackers, pasta, and</td>
</tr>
<tr>
<td>6</td>
<td>Meat/Poultry</td>
<td>Prepared meats</td>
</tr>
<tr>
<td>7</td>
<td>Produce</td>
<td>Dried fruit and bean curd</td>
</tr>
<tr>
<td>8</td>
<td>Seafood</td>
<td>Seaweed and fish</td>
</tr>
</tbody>
</table>

### Products

<table>
<thead>
<tr>
<th>ProductID</th>
<th>ProductName</th>
<th>SupplierID</th>
<th>CategoryID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chai</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Chang</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>Guarana Font</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>34</td>
<td>Sasquatch Al</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>35</td>
<td>Steeleye Sto</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>38</td>
<td>Côte de Blay</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>Chartreuse v</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>43</td>
<td>Ipoh Coffee</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>87</td>
<td>Laughing Lu</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>70</td>
<td>Outback Lag</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>75</td>
<td>Rhônbräu Klo</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>76</td>
<td>Lakkalikööri</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Aniseed Syru</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Chef Anton's</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
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), 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, 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, 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, if the master column-key of the **String** type, then the child column-key should be of the **String** type;
3 And so on;

Control panel of data columns in the **New Relation** dialog box is represented by 4 buttons.

1 The button to move all data columns from the field 6 or 7 in the field 8 or 9, respectively;
2 The button to move the selected data column from the field 6 or 7 in the field 8 or 9, respectively;
3 The button to move the selected data column from the field 8 or 9 in the field 6 or 7, respectively;
4 The button to move all the data columns from the field 8 or 9 in the field 6 or 7, respectively.

### 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:

<table>
<thead>
<tr>
<th>1. Beverages</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Chai</td>
<td>10 boxes x 20 bags</td>
</tr>
<tr>
<td>1.2 Chang</td>
<td>24 - 12 oz bottles</td>
</tr>
<tr>
<td>1.3 Chartreuse verte</td>
<td>750 cc per bottle</td>
</tr>
<tr>
<td>1.4 Côte de Blaye</td>
<td>12 - 75 cl bottles</td>
</tr>
<tr>
<td>1.5 Guaraná Fantástica</td>
<td>12 - 355 ml cans</td>
</tr>
<tr>
<td>1.6 Ipoh Coffee</td>
<td>16 - 500 g tins</td>
</tr>
<tr>
<td>1.7 Lakkalikööri</td>
<td>500 ml</td>
</tr>
<tr>
<td>1.8 Laughing Lumberjack Lager</td>
<td>24 - 12 oz bottles</td>
</tr>
<tr>
<td>1.9 Outback Lager</td>
<td>24 - 355 ml bottles</td>
</tr>
<tr>
<td>1.10 Rhönbräu Klosterbier</td>
<td>24 - 0.5 l bottles</td>
</tr>
<tr>
<td>1.11 Sasquatch Ale</td>
<td>24 - 12 oz bottles</td>
</tr>
<tr>
<td>1.12 Steeleye Stout</td>
<td>24 - 12 oz bottles</td>
</tr>
<tr>
<td>2. Condiments</td>
<td></td>
</tr>
<tr>
<td>2.1 Aniseed Syrup</td>
<td>12 - 550 ml bottles</td>
</tr>
<tr>
<td>2.2 Chef Anton's Cajun Seasoning</td>
<td>48 - 6 oz jars</td>
</tr>
<tr>
<td>2.3 Chef Anton's Gumbo Mix</td>
<td>36 boxes</td>
</tr>
<tr>
<td>2.4 Genma Shouyu</td>
<td>24 - 250 ml bottles</td>
</tr>
<tr>
<td>2.5 Grandma's Boysenberry Spread</td>
<td>12 - 6 oz jars</td>
</tr>
<tr>
<td>2.6 Gula Malacca</td>
<td>20 - 2 kg bags</td>
</tr>
<tr>
<td>2.7 Louisiana Fiery Hot Pepper Sauce</td>
<td>32 - 8 oz bottles</td>
</tr>
<tr>
<td>2.8 Louisiana Hot Spiced Okra</td>
<td>24 - 8 oz jars</td>
</tr>
<tr>
<td>2.9 Northwoods Cranberry Sauce</td>
<td>12 - 12 oz jars</td>
</tr>
<tr>
<td>2.10 Original Frankfurter grüne Soße</td>
<td>12 boxes</td>
</tr>
<tr>
<td>2.11 Sirop d'érable</td>
<td>24 - 500 ml bottles</td>
</tr>
<tr>
<td>2.12 Vegie-spread</td>
<td>15 - 625 g jars</td>
</tr>
</tbody>
</table>

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 \texttt{PrintIfDetailEmpty} property of the Master band to \texttt{true}. An example of a report for this case is shown below:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beverages</td>
<td>Soft drinks, coffees, teas, beers, and ales</td>
</tr>
<tr>
<td>2</td>
<td>Condiments</td>
<td>Sweet and savory sauces, relishes, spreads, and seasonings</td>
</tr>
<tr>
<td>3</td>
<td>Akneed Syrup</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Confections</td>
<td>Desserts, candies, and sweet breads</td>
</tr>
<tr>
<td>5</td>
<td>Dairy Products</td>
<td>Cheeses</td>
</tr>
<tr>
<td>6</td>
<td>Grains/Cereals</td>
<td>Breads, crackers, pasta, and cereal</td>
</tr>
<tr>
<td>7</td>
<td>Meat/Poultry</td>
<td>Prepared meats</td>
</tr>
<tr>
<td>17</td>
<td>Allo Mutton</td>
<td>39</td>
</tr>
<tr>
<td>7</td>
<td>Produce</td>
<td>Dried fruit and bean curd</td>
</tr>
<tr>
<td>8</td>
<td>Seafood</td>
<td>Seaweed and fish</td>
</tr>
</tbody>
</table>

As seen on the picture Master records were output (see numbers 1,3,4,5,7,8) all Master records. Moreover, they are output without Detail records. By default, the property is set to \texttt{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 \texttt{GroupHeader} band and the \texttt{GroupFooter} band.

The \texttt{GroupHeader} band is output in the beginning of each group. The \texttt{GroupFooter} band is output in the end of each group. The picture below shows how a report with grouping may look:
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, 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. 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:
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, 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.

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Company</th>
<th>Address</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Familia Arquitecto</td>
<td>Rua Orla, 82</td>
<td>Marketing Assistant</td>
</tr>
<tr>
<td>23</td>
<td>RBBA Fabric Inter Baciches S.A.</td>
<td>Div Morizart, 96</td>
<td>Accounting Manager</td>
</tr>
<tr>
<td>24</td>
<td>Foiles gourmandes</td>
<td>164, chaussée de Tournai</td>
<td>Assistant Sales Agent</td>
</tr>
<tr>
<td>25</td>
<td>Folk och Heklo</td>
<td>Åkergetan 24</td>
<td>Owner</td>
</tr>
<tr>
<td>26</td>
<td>Fierce restauration</td>
<td>54, rue Royale</td>
<td>Marketing Manager</td>
</tr>
<tr>
<td>27</td>
<td>Fancchi B.C.A.</td>
<td>Via Monte Bianco 34</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>28</td>
<td>Fanciello-Riande</td>
<td>Berliner Platz 13</td>
<td>Marketing Manager</td>
</tr>
<tr>
<td>29</td>
<td>Funch Badehau e Fitos do Mar</td>
<td>Jardim das rosas n. 32</td>
<td>Sales Manager</td>
</tr>
</tbody>
</table>

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:

\{\text{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:

![Simple Group](image)

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

![The Main Page](image)

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.

![Simple List](image)

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:

![Header](image)

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
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 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.

<table>
<thead>
<tr>
<th>3. Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
</tr>
<tr>
<td>Column Width</td>
</tr>
<tr>
<td>Column Gaps</td>
</tr>
<tr>
<td>Right to Left</td>
</tr>
</tbody>
</table>

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.

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.

![Diagram Showing Column Width and Gaps](image)

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

⚠️ Note: The number of columns on a Data band is unlimited.

There are two output modes for columns on the Data band: `AcrossThenDown` and `DownThenAcross`.

### 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.
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.
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 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 AcrossThenDown 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 DownThenAcross 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**:
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.
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.

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**.

© 2003-2020 Stimulsoft
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, 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 \textbf{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 \textbf{PageNumber} system variable was used on the \textbf{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

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**, 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 \texttt{ResetPageNumber} property of the page is set to \texttt{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. 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. 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, 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, 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, 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 than the page of a report, the 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. 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:

![Hierarchical report](image)

#### 3.23.1 Data Output

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:

![DataSource](image)

Set the **KeyDataColumn**, select the data column by what an identification number of data rows will be assigned. For example, a **EmployeeID** data column;

Set the **MasterKeyDataColumn**, 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**, 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**, 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:

![Hierarchical Report Example](image)

### 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 **EmployeeID** data column, because the entry of this column is the key and contains unique codes of employees.
3.23.3 MasterKeyDataColumn Property

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:

![Hierarchy Scheme](image1.png)

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:

![Example Scheme](image2.png)
As can be seen in the ReportsTo data column the following entries are: (null), 2 and 5, 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, 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:
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:
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:

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.
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:

![Simple List](image1)

Even when using the **GrowToHeight** property, then visually it cannot be convenient:

![Simple List](image2)

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.

![Report Title and Child Bands](image)

### 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.

![Data and Child Bands](image)

Run a report for execution. As you can see, the **Child** band was printed as many times as the **Data** band. 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.

![Child Band Example](image)

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.

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compaso Minero</td>
<td>Av. 035 Lisboa 23</td>
<td>(11) 555-7647</td>
<td>Sales Associate</td>
</tr>
<tr>
<td>Consolidated Holdings</td>
<td>Berkeley Gardens 12 Brewery</td>
<td>(17) 555-0283</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Dechenoluk Deinnesen</td>
<td>Walsenweg 21</td>
<td>0241-039123</td>
<td>Order Administrator</td>
</tr>
<tr>
<td>Du monde entier</td>
<td>67, rue des Cinquante Dages</td>
<td>40 67 60 80</td>
<td>Owner</td>
</tr>
<tr>
<td>Eastern Connection</td>
<td>36 King George</td>
<td>(11) 555-0297</td>
<td>Sales Agent</td>
</tr>
</tbody>
</table>

Add data to the Child band, for example Country. The picture below shows that instead of empty space, the country name will be output.

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centro comercial Moctezuma</td>
<td>Bieras de Braulio 9992</td>
<td>(5) 555-0392</td>
<td>Marketing Manager</td>
</tr>
<tr>
<td>Mexico</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>China</td>
<td>Hauoustr 29</td>
<td>0452-076845</td>
<td>Owner</td>
</tr>
<tr>
<td>Switzerland</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compania Minera</td>
<td>Av. dos Lisboa 23</td>
<td>(11) 555-7647</td>
<td>Sales Associate</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Contact Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[count] [{Suppliers.CompanyName}]</td>
<td>[{Suppliers.ContactName}]</td>
</tr>
</tbody>
</table>

If you then run the report you will see that the empty space is replaced with formatted empty data rows:

| 23. Arco-Chek | Fred Frecheville | Customer Service |
| 24. B & B | John Smith | Marketing Manager |
| 25. C & D | Jane Doe | Sales Representative |
| 26. E & F | Michael Jackson | Sales Representative |
| 27. G & H | Sarah Ross | Sales Representative |

⚠️ **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.

![IncreaseLastRow](image1)

*DecreaseLastRow*. The last row of the *Empty Band* will be decreased by height. The picture below shows this.

![DecreaseLastRow](image2)

*AlignFooterToBottom*. If there is no free space for the *Empty* band then this band is not output. The picture below shows this.

![AlignFooterToBottom](image3)
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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>A text that is used to output a watermark</td>
</tr>
<tr>
<td>Text Brush</td>
<td>A brush to output a watermark</td>
</tr>
<tr>
<td>Font</td>
<td>A font that is used to output a watermark</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image</strong></td>
<td>An image to output</td>
</tr>
<tr>
<td><strong>ImageAlignment</strong></td>
<td>This property is used to align an image on a page</td>
</tr>
<tr>
<td><strong>ImageMultipleFactor</strong></td>
<td>A multiplier that is used to change image size</td>
</tr>
<tr>
<td><strong>AspectRatio</strong></td>
<td>Saves proportions of an image</td>
</tr>
<tr>
<td><strong>ImageTiling</strong></td>
<td>If to set this property to <strong>true</strong>, then it will be tiled throughout a page</td>
</tr>
<tr>
<td><strong>ImageTransparency</strong></td>
<td>This property is used to set image transparency</td>
</tr>
<tr>
<td><strong>ImageStretch</strong></td>
<td>Stretches an image on a page</td>
</tr>
<tr>
<td><strong>ShowImageBehind</strong></td>
<td>Shows an image of a watermark on the background or foreground</td>
</tr>
</tbody>
</table>

Also there is another **Enabled** property. This property enables or disables watermark
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 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 placed on a panel. But not all bands can be placed on a panel. A table below shows which bands can placed.

<table>
<thead>
<tr>
<th>Band name</th>
<th>It is possible to place a band on a panel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Title</td>
<td>No</td>
</tr>
<tr>
<td>Report Summary</td>
<td>No</td>
</tr>
<tr>
<td>Page Header</td>
<td>No</td>
</tr>
<tr>
<td>Page Footer</td>
<td>No</td>
</tr>
<tr>
<td>Group Header</td>
<td>Yes</td>
</tr>
<tr>
<td>Group Footer</td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>Yes</td>
</tr>
<tr>
<td>Hierarchical Data</td>
<td>Yes</td>
</tr>
<tr>
<td>Child</td>
<td>Yes</td>
</tr>
<tr>
<td>Header</td>
<td>Yes</td>
</tr>
<tr>
<td>Footer</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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.

![Data Source](image)

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.
It is allowed to specify one or several columns at once. For example, in cross table only one column is specified:

As a result we get grouping by values of this column:

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:

<table>
<thead>
<tr>
<th>CategoryName, ProductName</th>
<th>Chai</th>
<th>Chai long</th>
<th>Chai hojicha</th>
<th>Cola de Baja</th>
<th>Iced Coffee</th>
<th>Iced Tea</th>
<th>Latté</th>
<th>Oatmeal Latte</th>
<th>Robusta Instant</th>
<th>Seattle Ale</th>
<th>Shakti Stout</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eeevages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condiments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.28.2.2 Rows

On a picture below you may see how the rows are positioned on a table.
Grouping is done only by its values for one row:

Get the result shown on a picture below. All values of the specified row are represented in one level.
Specify two rows:

A cross table is grouped in two levels vertically:
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:

<table>
<thead>
<tr>
<th>CategoryName</th>
<th>CompanyName</th>
<th>Units In Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beverages</td>
<td>Aux joyeux ecclésiastiques</td>
<td>281.5</td>
</tr>
<tr>
<td></td>
<td>Bigfoot Breweries</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Exotic Liquids</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>Kaikki Oy</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Leka Trading</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Pavlova, Ltd.</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Plutzer Lebensmittelgroßmärkte AG</td>
<td>7.75</td>
</tr>
<tr>
<td></td>
<td>Refrescos Americanas LTDA</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td><strong>455.75</strong></td>
</tr>
<tr>
<td>Condiments</td>
<td>Exotic Liquids</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Forêts d’érables</td>
<td>28.5</td>
</tr>
<tr>
<td></td>
<td>Grandma Kelly's Homestead</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Leka Trading</td>
<td>19.45</td>
</tr>
<tr>
<td></td>
<td>Marumi's</td>
<td>15.5</td>
</tr>
<tr>
<td></td>
<td>New Orleans Cajun Delights</td>
<td>81.40</td>
</tr>
<tr>
<td></td>
<td>Pavlova, Ltd.</td>
<td>43.9</td>
</tr>
<tr>
<td></td>
<td>Plutzer Lebensmittelgroßmärkte AG</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td><strong>278.75</strong></td>
</tr>
</tbody>
</table>

### 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.
1 The **New Data Source** button. Calls the window to create a new data source.
2 The **New Business Object** button. Calls the window to create a new Business Object.
3 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.
4 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 summary 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. 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.

3. 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:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Do not summarize the cell values</td>
</tr>
<tr>
<td>Sum</td>
<td>Returns the sum of values that are contained in the cell</td>
</tr>
<tr>
<td>Average</td>
<td>Returns the average of values that are contained in the cell</td>
</tr>
<tr>
<td>Min</td>
<td>Returns the minimal of values that are contained in the cell</td>
</tr>
<tr>
<td>Max</td>
<td>Returns the maximal of values that are contained in the cell</td>
</tr>
<tr>
<td>Count</td>
<td>Returns the number of values that are contained in the cell</td>
</tr>
<tr>
<td>CountDistinct</td>
<td>Returns the number of distinct values that are contained in the cell</td>
</tr>
<tr>
<td>Image</td>
<td>A cross table will show the first value as an image</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>SortDirection</th>
<th>Asc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SortType</td>
<td>ByDisplay Value</td>
</tr>
</tbody>
</table>

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.

![Sorted Cross-Table](image)

### 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.

[No Conditions]

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.
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:

<table>
<thead>
<tr>
<th>Country</th>
<th>Beverages</th>
<th>Condiments</th>
<th>Confections</th>
<th>Dairy Products</th>
<th>Grain/Pasta</th>
<th>Meat/Poultry</th>
<th>Produce</th>
<th>Seafood</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>15</td>
<td>24</td>
<td>29</td>
<td>38</td>
<td>20</td>
<td>42</td>
<td></td>
<td></td>
<td>168</td>
</tr>
<tr>
<td>Brazil</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Canada</td>
<td>113</td>
<td>17</td>
<td></td>
<td>135</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>266</td>
</tr>
<tr>
<td>Denmark</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Finland</td>
<td>57</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>132</td>
</tr>
<tr>
<td>France</td>
<td>98</td>
<td>96</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>246</td>
</tr>
<tr>
<td>Germany</td>
<td>125</td>
<td>32</td>
<td>140</td>
<td>22</td>
<td>26</td>
<td>10</td>
<td></td>
<td>355</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td>23</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>Japan</td>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>162</td>
</tr>
<tr>
<td>Netherlands</td>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>51</td>
</tr>
<tr>
<td>Norway</td>
<td>164</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>164</td>
</tr>
<tr>
<td>Singapore</td>
<td>17</td>
<td>27</td>
<td></td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>Spain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>108</td>
<td></td>
<td></td>
<td></td>
<td>108</td>
</tr>
<tr>
<td>Sweden</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>165</td>
<td></td>
<td></td>
<td></td>
<td>224</td>
</tr>
<tr>
<td>UK</td>
<td>99</td>
<td>13</td>
<td>74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>146</td>
</tr>
<tr>
<td>USA</td>
<td>163</td>
<td>259</td>
<td></td>
<td></td>
<td>15</td>
<td>208</td>
<td></td>
<td>686</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>569</td>
<td>507</td>
<td>365</td>
<td>335</td>
<td>165</td>
<td>100</td>
<td>701</td>
<td>3119</td>
<td></td>
</tr>
</tbody>
</table>

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:
By default, the ShowTotal property for rows and columns is set to true, 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:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllValues</td>
<td>All values, contained in a cell</td>
</tr>
<tr>
<td>SkipZerosAndNulls</td>
<td>Skip 0 null values, contained in a cell</td>
</tr>
<tr>
<td>SkipNulls</td>
<td>Skip null values, contained in a cell</td>
</tr>
</tbody>
</table>

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

The **Chart** is a component for visualizing data in a report.

<table>
<thead>
<tr>
<th>Series</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Column:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| ![Clustered Column](image) | **Clustered Column**  
Clustered column charts compare values across categories. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stacked Column</strong></td>
<td>Stacked column charts show the relationship of individual items to the whole, comparing the contribution of each value to a total across categories.</td>
</tr>
<tr>
<td><strong>Full-Stacked Column</strong></td>
<td>Full-Stacked column allows comparing the percentage of each value.</td>
</tr>
<tr>
<td><strong>Pareto</strong></td>
<td>A series that is used to apply the Pareto principle to values.</td>
</tr>
<tr>
<td><strong>Line:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Line</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Stacked Line</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Full-Stacked Line</strong></td>
<td>This is a kind of the Line series by which you can compare the relative proportion of each value of the series among the total aggregate value of specific arguments. Lines without markers are recommended in the approximation of the set of value arguments.</td>
</tr>
<tr>
<td><strong>Spline</strong></td>
<td>This type of series is used to display a smooth line, the points of which are the values of the series. Each point has its coordinates</td>
</tr>
</tbody>
</table>
depending on the value and argument of the chart series. After all points are specified, a spline will be drawn. Points on the chart can be displayed using markers. Spline without markers is recommended in the approximation of the set of value arguments.

**Stacked Spline**
This type of series is used to display a smooth line, the points of which are the values of the series. Each point has its coordinates depending on the value and argument of the chart series. The points of the next row of the chart are located above the smooth line of the previous row of the chart. After all points are specified, a stacked spline will be drawn. Points on the chart can be displayed using markers. Smooth lines without markers are recommended for the approximation of the set of value arguments.

**Full-Stacked Spline**
This is a variety of Spline series, with which you can compare the relative proportion of each value of a series in the total aggregate value of specific arguments. Smooth lines without markers are recommended in the approximation of the set of value arguments.

**Stepped Line**
This is a variation of the Line series, which will be displayed using only vertical and horizontal lines.

**Area:**
<table>
<thead>
<tr>
<th>Area</th>
<th>Area charts display the trend of values over time or categories.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stacked Area</td>
<td>Stacked area charts display the trend of the contribution of each value over time or categories.</td>
</tr>
<tr>
<td>Full-Stacked Area</td>
<td>Full-Stacked Area charts display the trend of the percentage each value contributes over time or categories.</td>
</tr>
<tr>
<td>Spline Area</td>
<td>This is a type of series with smooth lines. In the chart area, points are marked by coordinates - the value and argument of the series. Then, a smooth line passes through these points. The area between the line and the axis of the arguments is filled with color.</td>
</tr>
<tr>
<td>Stacked Spline Area</td>
<td>A series type that is used to display the ratio of smooth areas across multiple series of a chart.</td>
</tr>
<tr>
<td>Full-Stacked Spline Area</td>
<td>A series type is used to display the relative proportion of a spline area in the total aggregate value of specific arguments.</td>
</tr>
<tr>
<td>Stepped Area</td>
<td>This is a type of linear series. In the chart area, points are marked by coordinates - the value and argument of the series. Then, strictly vertical and horizontal lines pass through these points. The area between the line and the axis of the arguments is filled with color.</td>
</tr>
</tbody>
</table>

**Range:**
### 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.

### Spline Range
The series of this type displays the interval of change of values by strictly vertical lines and the time interval by any smooth straight lines.

### Stepped Range
A row of this type displays the interval of changing values by strictly vertical lines and a time interval by strictly horizontal lines.

### Range Bar
This type of series is used to display a range of values as columns for each argument. Also, if the chart has more than one row, it shows the ratio of the values of different rows for the current argument.

### Clustered Bar:

#### 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.
## Scatter:

| Scatter | **Scatter**  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Scatter" /></td>
<td>This type of chart compares pairs of values. Use a scatter chart without lines when you have data in a specific order.</td>
</tr>
</tbody>
</table>
| Scatter Line | **Scatter Line**  
| ![Scatter Line](image2) | 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 | **Scatter Spline**  
| ![Scatter Spline](image3) | 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. |  

## Pie:

| Pie | **Pie**  
| ![Pie](image4) | 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. |  

## Radar:

| Radar Point | **Radar Point**  
| ![Radar Point](image5) | This series is used to display three-dimensional data in two-dimensional space using points on a circular area. |  
| Radar Line | **Radar Line**  
| ![Radar Line](image6) | This series is used to display three-dimensional data in two-dimensional space, using points and lines between them, on a circular area. |
### Radar Area
This series is used to display three-dimensional data in two-dimensional space, using points and lines that form a region, on a circular region.

### Funnel:

<table>
<thead>
<tr>
<th>Funnel</th>
<th>Funnel</th>
</tr>
</thead>
<tbody>
<tr>
<td>This type of series is used to display statistical data, for example, sales stats and attendance of an online store. Depending on the value, the width of the parts of the graphic element will change.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Funnel Weighted Slices</th>
<th>Funnel Weighted Slices</th>
</tr>
</thead>
<tbody>
<tr>
<td>This type of series is used to display statistical data, for example, stats by sales and attendance of an online store. The graphic element will always be in displayed as a funnel, where every part is a separate value. Depending on the value, the height of the parts of the graphic element will change.</td>
<td></td>
</tr>
</tbody>
</table>

### Financial:

<table>
<thead>
<tr>
<th>Candlestick</th>
<th>Candlestick</th>
</tr>
</thead>
<tbody>
<tr>
<td>The financial series, using which you can display stock indicators of stocks, currencies, precious metals, etc.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stock</th>
<th>Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Another series for the financial chart, which displays market trends.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Treemap</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>A series is used to display a hierarchy of values. The chart area is the sum of all major values. This area will be split proportionally for each value of the first row. In turn, this each part will be divided into proportional parts for each value of the second row, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Gantt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Doughnut</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Bubble</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A series type is used to display three-dimensional data in two-dimensional space. In addition to the coordinates for each bubble, the value of its width or weight is indicated.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Pictorial</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A series type that is used to display data as special set of icons.</td>
</tr>
</tbody>
</table>

### 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.
The button **Run Chart Wizard**.

When you press this button, a chart of a certain type with the specified parameters is created.

Pressing this button cancels the creation of a chart but the component remains in 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;
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**.
The button is used to add the constant line.
The button is used to erase the selected line.
The buttons move the selected item in the list on the panel.
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.

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, which means 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.
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.

**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, which means 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.
This panel displays a list of different types of labels.

The preview panel. This panel displays the chart and immediately previews changes made in real time.

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**.
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.

In this panel you can see the list of styles that are available by default.

The panel Preview. This panel displays the chart and immediately previews changes made in real time.

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.
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.
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 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,
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, in these fields arguments of a chart are specified. For example, the arguments A, B, C, D.
4. The list of values fields, in these fields the values of the chart are shown. For example, the values 1, 2, 3, 4.

**Information:** It should be noted that for rendering the chart there must be at least one values, 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.
The list of labels for the chart, with examples of their placing on this type of a chart. Parameters of labels, their angle, the text before the header text after the header, etc.

**Information:** You should know that when you create a chart manually, 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, one type for all series of the chart.

On the next step, it is necessary to define axes settings.
The panel **Preview**.

The most important settings are displayed on the axes. Moreover, this panel has tabs axis X and axis Y.

1. The parameter **Title**. This group of settings specifies the text of the axis title and its alignment.
2. The parameter **Ticks**. It is determined by the number of intermediate ticks and display mode - without labels, only the main, and all labels.
3. The group **Grid Lines**. This group defines the parameters of the grid line.
4. 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:

![Pie chart example](image)

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:

![Doughnut Chart](image)

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
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 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:

![Title Legend](image)

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 **HorizontalAlignment** 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:**

![Legend Left](image)

» **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:**

![Legend Center](image)

» **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:**

![Legend Right](image)
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:
Right 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:
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**:

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**:

![Diagram showing Legend placement](image1)

> **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**:

![Diagram showing Legend placement](image2)

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:

![Image of chart with markers]

**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:

➤ **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**:

![Legend Bottom to Top](image)

➤ **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**:

![Legend Left to Right](image)

➤ **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**:

![Legend Right to Left](image)

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**):

![Legend Columns](image)
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**):

![Image of Legend with markers split into two vertical columns]

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**:

![Image of Legend with markers aligned to the 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**:

![Image of Legend with markers aligned to the right]

By default the **Marker Alignment** property is set to **Left**.
3.29.3.5.4 MarkerVisible Property

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**:

![Legend with markers](image1)

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**:

![Legend without markers](image2)

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:

![Axes Area](image3)
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**.

### 3.29.4.1.2 Labels

Labels are titles of X axis (the axis of the arguments) and Y (the axis values). Labels can take any string value. Any string value is transformed according to the selected format. If the report generator failed to convert a value to the selected format, then a direct string value is output. The picture below shows an example of a chart with arguments of Labels. The Format property is set to N:

![Chart with Labels](chart.png)

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.

### 3.29.4.1.2.1 Angle Property

The **Angle** property is used to change the inclination of Labels. Specifies the angle, in degrees. The **Angle** property is set separately for each axis. The full path to this property is **Area.Axis.Labels.Angle**. By default, the value of the **Angle** property is set...
to 0. So **Labels** are placed as it is shown on the picture below:

![Chart Sample](image)

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 is inclined anticlockwise. The picture below shows the chart sample, which Angle property by the X axis is set to 50:

![Chart Sample](image)

### 3.29.4.1.2.2 Format Property

The **Format** property is used 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**:

![Chart with N format](chart.png)

**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**:

![Chart with C format](chart.png)

**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**:

![Chart with P format](chart.png)
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:

![One Line chart example](image1)

**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:

![Two Lines chart example](image2)

**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:

![None chart example](image3)
By default, the **Placement** property is set to **One Line**.

### 3.29.4.1.2.4 TextAlignm ent 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**:

![Chart with TextAlignment set to Left](chart_left.png)

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**:

![Chart with TextAlignment set to Right](chart_right.png)
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, 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 than 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:

![Chart with hidden axes](image)

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:

![Chart with visible axes](image)

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**. 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:

![Chart with Start from Zero set to true](image)

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:

![Chart with Start from Zero set to false](image)

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:

![Chart with Ticks](image)

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, 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, 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":

![Chart Image]

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**:

![Far Alignment Example](image)

➢ **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**:

![Near Alignment Example](image)
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**, 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:
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:
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**:

![Chart example with Auto Series Key Data Column](image)

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, 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. To select the required settings, data filtering is used. Filtering is set using the **Filters** property in the **Series Editor**. A condition is specified in each filter. If the condition is **true**, 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:

<table>
<thead>
<tr>
<th>Field is</th>
<th>Data Type</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Numeric</td>
<td>equal to</td>
<td>9</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Operation</th>
<th>String</th>
<th>Numeric</th>
<th>DateTime</th>
<th>Boolean</th>
</tr>
</thead>
<tbody>
<tr>
<td>equal to</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not equal to</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>between</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not between</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>greater than</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

If the first value is equal to the second, then the condition is true.

If the first value is not equal to the second, then the condition is true.

If the first value is in the range, then the condition is true.

If the first value is not in the range, then the condition is true.

If the first value is greater then the second value, then the condition is true.
<table>
<thead>
<tr>
<th>Condition</th>
<th>Comparison</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>greater than or equal to</td>
<td>✔</td>
<td>If the first value is greater than or equal to the second value, then the condition is true.</td>
</tr>
<tr>
<td>less than</td>
<td>✔</td>
<td>If the first value is less than the second value, then the condition is true.</td>
</tr>
<tr>
<td>less than or equal to</td>
<td>✔</td>
<td>If the first value is less than or equal to the second value, then the condition is true.</td>
</tr>
<tr>
<td>containing</td>
<td>✔</td>
<td>If the first value contains the second value, then the condition is true. This operation is used only for strings.</td>
</tr>
<tr>
<td>not containing</td>
<td>✔</td>
<td>If the first value does not contain the second value, then the condition is true. This operation is used only for strings.</td>
</tr>
<tr>
<td>beginning with</td>
<td>✔</td>
<td>If the first value starts with the second value, then the condition is true. This operation is used only for strings.</td>
</tr>
<tr>
<td>ending with</td>
<td>✔</td>
<td>If the first value ends with the second value, then the condition is true. This operation is used only for strings.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Data Type</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value 1</td>
<td>Numeric 2</td>
<td>greater than 3</td>
<td>4</td>
</tr>
<tr>
<td>Color</td>
<td></td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

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, 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.

### 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:

![Diagram showing populations comparison](image)

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:

![Series Labels chart](image)

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:**

1. **Empty**
2. **Solid**
3. **Hatch**
4. **Gradient**
5. **Glare**
6. **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**:

![Chart with Brush property set to Glare](image)

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:

```
AaBbCcDd
AaBbCcDd
AaBbCcDd
```

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

**Font Size**

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

Arial, Bold

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**:

![Chart with Label Color set to red](image)

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**:

![Chart with UseSeriesColor set to false](image)
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`:

![Chart with UseSeriesColor 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`:

![Chart with Angle 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**:

![Chart with inclined labels](image)

### 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):

![Chart with borders around labels](image)
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**:

![Chart with Antialiasing set to true](image)

The picture below shows a chart with the **Antialiasing** property set to **false**:

![Chart with Antialiasing set to false](image)

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 \( N \):

![Chart example](image)

**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 \( C \):

![Chart example](image)

**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**:

![Value Type Example](image1)

- **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**:

![Series Title Example](image2)

- **Argument**. The Series Labels are the arguments. The picture below shows an
example of a chart with the **Value Type** property set to **Argument**:

![Chart 1](image1.png)

➤ **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**:

![Chart 2](image2.png)

➤ **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 `ShowOnZeroValues` property is set to `false`:

![Chart with zero value and `ShowOnZeroValues` set to false](image)

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 `ShowOnZeroValues` property is set to `true`:

![Chart with zero value and `ShowOnZeroValues` set to true](image)

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):

![Chart samples with TextBefore and TextAfter properties](image)

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:

![Chart with Visible property](image)
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**:
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:

![Marker Alignment](image)

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. To add the Map component in the report you should go to the Infographics menu on the toolbox or Insert tab:
Setting the map can be done in the Map component. 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:
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

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. 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.
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:

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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Name</th>
<th>Value</th>
<th>Group</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>AL</td>
<td>99</td>
<td>1</td>
<td>#FF000</td>
</tr>
<tr>
<td>Alaska</td>
<td>AK</td>
<td>87</td>
<td>2</td>
<td>#FF7700</td>
</tr>
<tr>
<td>Arizona</td>
<td>AZ</td>
<td>50</td>
<td>3</td>
<td>#CD0074</td>
</tr>
<tr>
<td>Arkansas</td>
<td>AR</td>
<td>134</td>
<td>4</td>
<td>#009999</td>
</tr>
<tr>
<td>California</td>
<td>CA</td>
<td>127</td>
<td>5</td>
<td>#1D7373</td>
</tr>
<tr>
<td>Colorado</td>
<td>CO</td>
<td>74</td>
<td>6</td>
<td>#BF7130</td>
</tr>
<tr>
<td>Connecticut</td>
<td>CT</td>
<td>89</td>
<td>7</td>
<td>#FF7733</td>
</tr>
<tr>
<td>Delaware</td>
<td>DE</td>
<td>90</td>
<td>8</td>
<td>#33CCCC</td>
</tr>
<tr>
<td>Florida</td>
<td>FL</td>
<td>62</td>
<td>9</td>
<td>#FFB273</td>
</tr>
<tr>
<td>Georgia</td>
<td>GA</td>
<td>36</td>
<td>6</td>
<td>#BF7130</td>
</tr>
</tbody>
</table>

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 Wy3iwyg 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 Wy3iwyg 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:

<table>
<thead>
<tr>
<th>Test string</th>
<th>Test string</th>
<th>Test string</th>
<th>Test string</th>
</tr>
</thead>
</table>

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 cannot 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.

<table>
<thead>
<tr>
<th>Table1: Data Source Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

⚠️ 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.
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.

On the picture below the table with 5 rows is shown.

### 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.

```
Master Componen [DataBand1] ....
```

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

<table>
<thead>
<tr>
<th>Beverage</th>
<th>Type Description</th>
<th>Quantity</th>
<th>Unit</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>10 boxes x 20 bags</td>
<td>18</td>
<td></td>
<td>39</td>
</tr>
<tr>
<td>Chang</td>
<td>24 - 12 oz bottles</td>
<td>19</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>Guaraná Fantástica</td>
<td>12 - 355 ml cans</td>
<td>4,5</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Sasquatch Ale</td>
<td>24 - 12 oz bottles</td>
<td>14</td>
<td></td>
<td>111</td>
</tr>
<tr>
<td>Steeleye Stout</td>
<td>24 - 12 oz bottles</td>
<td>18</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Côte de Blaye</td>
<td>12 - 75 cl bottles</td>
<td>263,5</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>Chartreuse verte</td>
<td>750 cc per bottle</td>
<td>18</td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>Ipoh Coffee</td>
<td>16 - 500 g tins</td>
<td>46</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>Laughing Lumberjack</td>
<td>24 - 12 oz bottles</td>
<td>14</td>
<td></td>
<td>52</td>
</tr>
</tbody>
</table>

#### 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.

```plaintext
DataCategories; Data Source: Categories
{Categories.CategoryName}

TableProducts; Data Source: Products Master Component: DataCategories
{Products.ProductName} {Products.QuantityPer} {Products.UnitPrice} {Products.UnitsIn}
```

The number of **Data** bands and **Tables** which interact 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.

```plaintext
GroupHeaderBand1: Condition: {Products.Categories.CategoryName}

Table

<table>
<thead>
<tr>
<th>Products.ProductName</th>
<th>Products.ProductID</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Products.QuantityFe}</td>
<td>{Products.UnitPrice}</td>
</tr>
<tr>
<td>{Products.UnitsInStor}</td>
<td>{Products.SupplierID}</td>
</tr>
</tbody>
</table>
```

See the picture below that demonstrates the report with grouping and a table.

### Beverages

<table>
<thead>
<tr>
<th>Côte de Blaye</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 - 75 cl bottles</td>
<td>263,5</td>
</tr>
<tr>
<td>Chartreuse verte</td>
<td>39</td>
</tr>
<tr>
<td>750 cc per bottle</td>
<td>18</td>
</tr>
<tr>
<td>Steeleye Stout</td>
<td>35</td>
</tr>
<tr>
<td>24 - 12 oz bottles</td>
<td>18</td>
</tr>
<tr>
<td>Guaraná Fantástica</td>
<td>24</td>
</tr>
<tr>
<td>12 - 355 ml cans</td>
<td>4,5</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Address</th>
<th>Phone</th>
<th>Contact Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Franchi S.p.A.</td>
<td>Via Monte Blanco 34</td>
<td>011-4988260</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Furia Bacalhau e Frutos do Mar</td>
<td>Jardim das Rosas n. 22</td>
<td>(1) 354-2534</td>
<td>Sales Manager</td>
</tr>
<tr>
<td>Galería del gastronómetro</td>
<td>Rambla de Cataluña, 23</td>
<td>(93) 203 4560</td>
<td>Marketing Manager</td>
</tr>
<tr>
<td>Godos Cocina Típica</td>
<td>C/ Romero, 33</td>
<td>(95) 555 82 82</td>
<td>Sales Manager</td>
</tr>
</tbody>
</table>

(c) 2003-2009 Stimulsoft

➤ **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.
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**.

**LastColumns**
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.

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
<td>Obere Str. 57</td>
<td>030-0074321</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Ana Trujillo Emparedados y helados</td>
<td>Avda. de la Constitución 2222</td>
<td>(5) 555-4729</td>
<td>Owner</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>Matadero 2312</td>
<td>(5) 555-3932</td>
<td>Owner</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>120 Hanover Sq.</td>
<td>(171) 555-7788</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
<td>Berguvsvägen 3</td>
<td>0921-12 34 65</td>
<td>Order Administrator</td>
</tr>
<tr>
<td>Blauer See Delikatessen</td>
<td>Forsterstr. 57</td>
<td>0621-08460</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Blondesddsl père et fils</td>
<td>24, place Kléber</td>
<td>38.60.15.31</td>
<td>Marketing Manager</td>
</tr>
</tbody>
</table>

On the second picture the **FixedWidth** property of the Phone column is set to **true**.

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Futterkiste</td>
<td>Obere Str. 57</td>
<td>030-0074321</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Ana Trujillo Emparedados y helados</td>
<td>Avda. de la Constitución 2222</td>
<td>(5) 555-4729</td>
<td>Owner</td>
</tr>
<tr>
<td>Antonio Moreno Taquería</td>
<td>Matadero 2312</td>
<td>(5) 555-3932</td>
<td>Owner</td>
</tr>
<tr>
<td>Around the Horn</td>
<td>120 Hanover Sq.</td>
<td>(171) 555-7788</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
<td>Berguvsvägen 3</td>
<td>0921-12 34 65</td>
<td>Order Administrator</td>
</tr>
<tr>
<td>Blauer See Delikatessen</td>
<td>Forsterstr. 57</td>
<td>0621-08460</td>
<td>Sales Representative</td>
</tr>
<tr>
<td>Blondesddsl père et fils</td>
<td>24, place Kléber</td>
<td>38.60.15.31</td>
<td>Marketing Manager</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Cell Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
</tr>
<tr>
<td>Image</td>
</tr>
<tr>
<td>CheckBox</td>
</tr>
<tr>
<td>RichText</td>
</tr>
</tbody>
</table>

Also it is possible to indicate a cell style by clicking the quick access button of a cell.

Or the context menu of a cell.
Primitives 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:
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.
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.
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
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. 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

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Unit Price</th>
<th>Units In Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oros</td>
<td>$13.00</td>
<td>29</td>
</tr>
<tr>
<td>Chang</td>
<td>$19.00</td>
<td>17</td>
</tr>
<tr>
<td>Guanaco Fantástico</td>
<td>$4.50</td>
<td>20</td>
</tr>
<tr>
<td>Bassquatch Ave</td>
<td>$14.00</td>
<td>111</td>
</tr>
<tr>
<td>Steelye Stout</td>
<td>$13.00</td>
<td>20</td>
</tr>
<tr>
<td>Côte de Blaye</td>
<td>$200.50</td>
<td>17</td>
</tr>
<tr>
<td>Chateuse verte</td>
<td>$18.00</td>
<td>69</td>
</tr>
<tr>
<td>Ippon Coffee</td>
<td>$48.00</td>
<td>17</td>
</tr>
<tr>
<td>Laughing Lumberjack Lager</td>
<td>$14.00</td>
<td>52</td>
</tr>
<tr>
<td>Outback Lager</td>
<td>$18.00</td>
<td>15</td>
</tr>
<tr>
<td>Richardo Kisterbier</td>
<td>$7.75</td>
<td>125</td>
</tr>
<tr>
<td>Laxa Vinken</td>
<td>$18.00</td>
<td>37</td>
</tr>
</tbody>
</table>

Count: 12

Condiments
Sweet and savory sauces, relishes, spreads, and seasonings

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Unit Price</th>
<th>Units In Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alkaline syrup</td>
<td>$10.00</td>
<td>13</td>
</tr>
<tr>
<td>Chef Antoine Cajun Seasoning</td>
<td>$22.00</td>
<td>53</td>
</tr>
<tr>
<td>Chef Antoine Gumbo Mix</td>
<td>$21.35</td>
<td>0</td>
</tr>
<tr>
<td>Grandma's Boysenberry Spread</td>
<td>$22.00</td>
<td>100</td>
</tr>
<tr>
<td>Northwoods Cranberry Sauce</td>
<td>$40.00</td>
<td>6</td>
</tr>
<tr>
<td>Geno Bshoyu</td>
<td>$18.50</td>
<td>29</td>
</tr>
<tr>
<td>Guia Malacca</td>
<td>$18.46</td>
<td>27</td>
</tr>
<tr>
<td>Broth d'eleve</td>
<td>$22.50</td>
<td>113</td>
</tr>
<tr>
<td>Veggiespread</td>
<td>$40.90</td>
<td>24</td>
</tr>
<tr>
<td>Louisiana Red Hot Pepper Sauce</td>
<td>$21.05</td>
<td>76</td>
</tr>
<tr>
<td>Louisiana Hot Spiced Ones</td>
<td>$17.00</td>
<td>4</td>
</tr>
<tr>
<td>Original Frankfurter Dose</td>
<td>$19.00</td>
<td>22</td>
</tr>
</tbody>
</table>

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, 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, 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 has the Functions category. This category contains the elements using which you can calculate a specific 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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Date:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{DateDiff,}</code></td>
<td>Calculates the distance between the specified dates</td>
<td><code>{DateDiff(DateSerial(2016,1,30),DateSerial(2016,1,1))}</code> - 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</td>
</tr>
<tr>
<td><code>{DateSerial,}</code></td>
<td>Specifies date. Arguments should be year, month, day</td>
<td><code>{DateSerial(2016,1,30)}</code> - the result is 1/30/2016 12:00:00 AM</td>
</tr>
<tr>
<td><code>{Day,}</code></td>
<td>Shows a day from the specified date Arguments should be of the DateTime type</td>
<td><code>{Day(DateSerial(2016,1,30))}</code> - 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</td>
</tr>
<tr>
<td><code>{DayOfWeek,}</code></td>
<td>Display a day of the week from a specified date in text form.</td>
<td><code>{DayOfWeek(DateSerial(2016,1,30))}</code> - the result is Saturday. {DayOfWeek(DataSource.Column)} - for each value a day of the week will be calculated <code>{DayOfWeek(DateSerial(2016,1,30),&quot;de&quot;))</code> - the result</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| `DayOfWeek()`             | Checks the specified day of the week                                        | `{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                                                  | `{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:                                  | `{DaysInMonth(DateSerial(2016,2,1)))}` - the result will be 29, because 2016 is a
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Date()}</td>
<td>Displays date</td>
<td></td>
</tr>
<tr>
<td>{Year()}</td>
<td>Displays year</td>
<td></td>
</tr>
<tr>
<td>{Month()}</td>
<td>Displays month</td>
<td></td>
</tr>
<tr>
<td>{DayOfWeek()}</td>
<td>Displays day of the week</td>
<td></td>
</tr>
<tr>
<td>{DaysInMonth()}</td>
<td>Displays the number of days in a month</td>
<td></td>
</tr>
<tr>
<td>{DaysInYear()}</td>
<td>Displays the number of days in a year</td>
<td></td>
</tr>
<tr>
<td>{Hour()}</td>
<td>Displays hour</td>
<td></td>
</tr>
<tr>
<td>{Minute()}</td>
<td>Displays minutes</td>
<td></td>
</tr>
<tr>
<td>{Second()}</td>
<td>Displays second</td>
<td></td>
</tr>
<tr>
<td>{DayOfYear()}</td>
<td>Displays day of the year</td>
<td></td>
</tr>
<tr>
<td>{Time()}</td>
<td>Displays time</td>
<td></td>
</tr>
<tr>
<td>{DateSerial()}</td>
<td>Creates a date from year, month, and day</td>
<td></td>
</tr>
<tr>
<td>{DateAdd()}</td>
<td>Adds or subtracts days from a date</td>
<td></td>
</tr>
<tr>
<td>{DatePart()}</td>
<td>Extracts parts of a date</td>
<td></td>
</tr>
<tr>
<td>{DateTimeAdd()}</td>
<td>Adds or subtracts days from a DateTime</td>
<td></td>
</tr>
<tr>
<td>{DateTimePart()}</td>
<td>Extracts parts of a DateTime</td>
<td></td>
</tr>
<tr>
<td>{Day()}{Month()}{Year()}{Hour()}{Minute()}{Second()}</td>
<td>Displays date, month, year, hour, minute, second from a DateTime</td>
<td></td>
</tr>
<tr>
<td>{TimeAdd()}</td>
<td>Adds or subtracts minutes from a time</td>
<td></td>
</tr>
<tr>
<td>{TimePart()}</td>
<td>Extracts parts of a time</td>
<td></td>
</tr>
</tbody>
</table>

**Examples:**

- `{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(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(DataSource.Column)}` - an hour will be displayed from each value. For example, if time is 16:22:36, then the result is 16.
- `{Minute(DataSource.Column)}` - minutes will be displayed from each value. For example, if time is 16:22:36, then the result is 22.
- `{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...
<table>
<thead>
<tr>
<th>{MonthName()}</th>
<th>Displays the month name of the specified date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>➤ Specifies in arguments:</td>
</tr>
<tr>
<td></td>
<td>1. Date (the DateTime type) and culture (the string type)</td>
</tr>
<tr>
<td></td>
<td>2. Culture (the string type),</td>
</tr>
<tr>
<td></td>
<td>3. The true or false value (the bool type), to display the result with a capital letter or with a small letter.</td>
</tr>
<tr>
<td></td>
<td>➤ Returns the string value</td>
</tr>
<tr>
<td>(MonthName(DateSerial(2016,1,1)))</td>
<td>the result is January, because the 1 of January 2016 is set.</td>
</tr>
<tr>
<td>(MonthName(DataSource.Column))</td>
<td>the result is the name of the month for each Column value.</td>
</tr>
<tr>
<td>(MonthName(DateSerial(2016,2,1),&quot;de&quot;))</td>
<td>the result will correspond to the de culture, i.e. Febuar.</td>
</tr>
<tr>
<td>(MonthName(DataSource.Column,&quot;en&quot;))</td>
<td>all the names of months will correspond to the en culture.</td>
</tr>
<tr>
<td>(MonthName(DateSerial(2016,1,1), false))</td>
<td>the name of the month will be in lower case.</td>
</tr>
<tr>
<td>(MonthName(DataSource.Column, true))</td>
<td>the name of the months will start with a capital letter.</td>
</tr>
<tr>
<td>(MonthName(DateSerial(2016,1,1), false))</td>
<td>the name of the month will correspond to the culture used by the system.</td>
</tr>
<tr>
<td>(MonthName(DataSource.Column, true))</td>
<td>the names of months will correspond</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>{Second()}</code></td>
<td>Displays seconds:</td>
</tr>
<tr>
<td><code>{TimeSerial(,,)}</code></td>
<td>Displays time:</td>
</tr>
<tr>
<td><code>{Year()}</code></td>
<td>Displays year:</td>
</tr>
</tbody>
</table>

**Math:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{Abs()}</code></td>
<td>Displays the absolute number.</td>
<td><code>{Abs(-42)} - the result is 42 </code>{Abs(DataSource.Column1)} - the result will be absolute numbers from the values of Column1, i.e. without considering the number.</td>
</tr>
<tr>
<td><code>{Acos()}</code></td>
<td>Displays the angle value in radians.</td>
<td><code>{Acos(-1)) - the angle in radians will be calculated for the value cos = -1, i.e. the angle will be ~ 3.14. </code>{Acos(DataSource.Column)</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td><code>{Asin()}</code></td>
<td>Displays the angle in radians. The sin value in arguments (the double type) Returns the value of the angle in radians of the double type</td>
<td><code>{Asin(0)}</code> - the angle in radians will be calculated for the value sin = 0, i.e. the angle is 0 <code>{Asin(DataSource.Column1)}</code> - for all sin values, the angle will be calculated in radians.</td>
</tr>
<tr>
<td><code>{Atan()}</code></td>
<td>Displays the angle in radians. The tan value in arguments (the double type) Returns the value of the angle in radians of the double type</td>
<td><code>{Atan(-1)}</code> - the angle in radians will be calculated for the value tan = -1, i.e. the angle will be ~ -0.79 <code>{Atan(DataSource.Column1)}</code> - for all tan values the angle will be calculated.</td>
</tr>
<tr>
<td><code>{Ceiling()}</code></td>
<td>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</td>
<td><code>{Ceiling(25.124))</code> - It is worth noting that when this function is used, the number is not rounded. <code>{Ceiling(25.9))</code> - the result is 26 <code>{Ceiling(DataSource.Column1)}</code> - for all Column values, the nearest maximal integers will be found and displayed.</td>
</tr>
<tr>
<td><code>{Cos()}</code></td>
<td>Calculates and displays the cos value: The value of the angle in radians is specified in arguments (the double type) Returns double, decimal values</td>
<td><code>{Cos(0)}</code> - the result is 1. <code>{Cos(DataSource.Column1)}</code> - for all values, the cos of the angle will be calculated.</td>
</tr>
</tbody>
</table>
| **{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,DataSource.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 the division, in this line, the values from Column3 will be displayed. |
| **{Exp()}** | Displays the result of rising to the specified degree the number e:  
- The arguments indicate the degree to which the number e must rise (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 |
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Log()}</td>
<td>Calculates the natural logarithm: The value is specified in arguments (the double type) Returns the value of the double type</td>
<td>{Log(x)}, where x is a number or an expression, the result is a calculation of the natural logarithm.</td>
</tr>
<tr>
<td>{Maximum(,)}</td>
<td>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</td>
<td>{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.</td>
</tr>
<tr>
<td>{Minimum(,)}</td>
<td>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</td>
<td>{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.</td>
</tr>
<tr>
<td>{Round()}</td>
<td>Rounds up the value to an integer or up to the certain number of decimal: In arguments, the following is specified: 1. The value (the decimal, double types), 2. Number of characters to</td>
<td>{Round(7.56)} - the result is 8 {Round(DataSource.Column1)} - all Column1 values will be rounded according to the mathematical rounding rules.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td><code>Round()</code></td>
<td>Returns the value of the decimal, double types</td>
<td><code>{Round(5.7896541897, 3)}</code></td>
</tr>
<tr>
<td></td>
<td>which the fractional part should be rounded (the int type)</td>
<td>- the result is 5.789</td>
</tr>
<tr>
<td></td>
<td>Returns the value of the decimal, double types</td>
<td><code>{Round(DataSource.Column1, 2)}</code></td>
</tr>
<tr>
<td></td>
<td>- all values from the data column will be rounded up to two decimal places in the fractional part,</td>
<td>according to the mathematical</td>
</tr>
<tr>
<td></td>
<td>- according to the mathematical rounding rules.</td>
<td>rounding rules.</td>
</tr>
<tr>
<td><code>{Sign()}</code></td>
<td>Displays an indicator. For positive numbers 1, 0 - for all zero values, -1 - for negative values:</td>
<td><code>{Sign(256)}</code> - the result is 1.</td>
</tr>
<tr>
<td></td>
<td>The value is specified in arguments (the long, decimal, double types).</td>
<td><code>{Sign(0)}</code> - the result is 0.</td>
</tr>
<tr>
<td></td>
<td>Returns the value of the long type</td>
<td><code>{Sign(-157)}</code> - the result is -1.</td>
</tr>
<tr>
<td><code>{Sin(0)}</code></td>
<td>Calculates sin of an angle:</td>
<td><code>{Sin(0)}</code> - the result is 0.</td>
</tr>
<tr>
<td></td>
<td>The value of an angle in radians is specified in arguments (the double type).</td>
<td><code>{Sin(DataSource.Column1)}</code></td>
</tr>
<tr>
<td></td>
<td>Returns the value of the long type</td>
<td>- for all values, the sin angle</td>
</tr>
<tr>
<td></td>
<td>is calculated.</td>
<td>will be assigned.</td>
</tr>
<tr>
<td><code>{Sqrt()}</code></td>
<td>Calculates the square root of the number:</td>
<td><code>{Sqrt(4)}</code> - the result will be</td>
</tr>
<tr>
<td></td>
<td>The number is specified in arguments (the double type).</td>
<td>2 because the square root of 4 is 2.</td>
</tr>
<tr>
<td></td>
<td>Returns the value of the double type</td>
<td><code>{Sqrt(DataSource.Column1)}</code></td>
</tr>
<tr>
<td></td>
<td>- for all Column1 values, the square root will be calculated.</td>
<td>- for all Column1 values, the</td>
</tr>
<tr>
<td><code>{Tan()}</code></td>
<td>Calculates tg of an angle:</td>
<td><code>{Tan(90)}</code> - the result is ~ -</td>
</tr>
<tr>
<td></td>
<td>The value of an angle in radians is specified in arguments (the double type).</td>
<td>1.995</td>
</tr>
<tr>
<td></td>
<td>Returns the value of the double type</td>
<td><code>{Tan(DataSource.Column1)}</code></td>
</tr>
<tr>
<td></td>
<td>- for all values, the tan of the angle will be</td>
<td>- for all values, the tan of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>angle will be</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
</tbody>
</table>
| **{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. The whole part in this number is 2.  
{Truncate(DataSource.Column1)} - only the integer part of all Column1 values will be displayed. |
| **{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:*  
1. The data source (the object type)  
2. The column name (the string type).  
*Returns the value of the bool type* | {IsNull(,)} - 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. |
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{NextIsNull(,)}</code></td>
<td>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.</td>
<td>For example, the Column data column contains the values 2, 0, 9. Then, using the function <code>{NextIsNull (DataSource, &quot;Column&quot;)}</code>, the first value is true; the second is false; the third is true.</td>
</tr>
<tr>
<td><code>{Previous(,)}</code></td>
<td>Displays the value from the previous line. If the value of the next line is null, the result is 0.</td>
<td>For example, the Column column contains values 2, 5, 9. Then, using the function <code>{Previous (DataSource, &quot;Column&quot;)}</code>, the first value will be empty, the second value will be 2, the third value will be 5.</td>
</tr>
<tr>
<td><code>{PreviousIsNull(,)}</code></td>
<td>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.</td>
<td>For example, the Column data column contains the values 2, 9, 0. Then, using the function <code>{PreviousIsNull (DataSource, &quot;Column&quot;)}</code>, the first value is true; the second is false; the third is false.</td>
</tr>
<tr>
<td>String type</td>
<td>Programming Shortcut:</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------</td>
<td></td>
</tr>
<tr>
<td>{Choose()}</td>
<td>Displays the value by index. The arguments specify the index and values. Returns values by index.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{Choose(DataSource.Column1, &quot;expensive&quot;, &quot;average&quot;, &quot;cheap&quot;)} - instead of index 1, the value expensive will be displayed, instead of index 2 - average, instead of index 3 - cheap.</td>
<td></td>
</tr>
<tr>
<td>{IIF(,,)}</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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. To highlight critical positions in the report visually, you</td>
<td></td>
</tr>
</tbody>
</table>
| 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:
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Arabic()}</td>
<td>Converts these numbers to Arabic numerals:</td>
<td>{Arabic()} - the number 2 will have an Arabic spelling. {Arabic(DataSource.Column1)} - all the numbers from Column1 will have an Arabic spelling.</td>
</tr>
<tr>
<td></td>
<td>➤ The value is specified in arguments (the string or int types)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>➤ Returns the value of the string type</td>
<td></td>
</tr>
<tr>
<td>{DateToStr()}</td>
<td>Converts date to text value:</td>
<td>{DateToStr()} - all dates from Column1 will be displayed in text form.</td>
</tr>
<tr>
<td></td>
<td>➤ In arguments, the following is specified:</td>
<td>{DateToStrPl(DataSource.Column1,true)} - dates will be displayed in text form, in Polish and the first character is a capital letter.</td>
</tr>
<tr>
<td></td>
<td>1 Date (the DateTime type)</td>
<td>{DateToStrPl(DataSource.Column1,false)} - dates will be displayed in text form, in Polish and the first character is a lowercase letter.</td>
</tr>
<tr>
<td></td>
<td>2 Boolean values (true or false) for displaying the header that starts with a capital letter or with a lowercase letter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>➤ Returns the value of the string type</td>
<td></td>
</tr>
<tr>
<td>{Insert(,,)}</td>
<td>Inserts a value after a certain character into another value:</td>
<td>{Insert(&quot;25&quot;,2,&quot; dollars&quot;)} - in the value 25, after the second symbol, the value dollars will be inserted, i.e. the result will be 25 dollars.</td>
</tr>
<tr>
<td></td>
<td>➤ In arguments, the following is specified:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 The value in which to insert text (the string type),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 The number of a character, after which the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>{Left()}</td>
<td>Displays the specified number of characters from the left side of the value.</td>
<td>{Left(&quot;Beverages&quot;, 4)) - only four characters from the Beverages value will be displayed, the result will be Beve. {Left(DataSource.Column1, 2)) - only the first two characters for each Column1 value will be displayed.</td>
</tr>
<tr>
<td>{Length()}</td>
<td>Displays the number of characters for the specified value.</td>
<td>{Length(&quot;Beverages&quot;)}) - the result will be 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.</td>
</tr>
<tr>
<td>{Mid()}</td>
<td>Displays characters from a value. In this case, you can set the reference position.</td>
<td>{Mid(&quot;Beverages&quot;,2,3)) - three symbols will be displayed after the first two, the result will be ver.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>Mid()</code></td>
<td>Returns the value of the int type</td>
<td><code>Mid(DataSource.Column1, 3,2)</code> - 2 characters will be displayed after the first three for all values.</td>
</tr>
<tr>
<td><code>{Persian()}</code></td>
<td>Converts specified numbers to numbers in Persian:</td>
<td><code>{Persian(5)}</code> - number 2 will have Persian spelling. <code>{Persian(DataSource.Column1)}</code> - all the numbers from Column1 will have Persian spelling.</td>
</tr>
<tr>
<td><code>{Remove()}</code></td>
<td>Deletes the specified number of characters from the index of a specific position:</td>
<td><code>{Remove(&quot;Beverages&quot;,2,3)}</code> - after the second character, three characters will be deleted, the result is Beages. <code>{Remove(DataSource.Column1,3,2)}</code> - for all values from Column1, two characters will be deleted after the first three.</td>
</tr>
<tr>
<td><code>{Replace(, ,)}</code></td>
<td>Replaces certain characters or their combination with other characters:</td>
<td><code>{Replace(&quot;Beverages&quot;,&quot;ver&quot;,&quot;NEW&quot;)}</code> - in the value Beverages, the ver characters will be replaced by the characters NEW, the result is BeNEWages. <code>{Replace(DataSource.Column1,&quot;be&quot;,&quot;BE&quot;)}</code> - for all values from Column1, the letter b will be replaced with BE.</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Right()}</td>
<td>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) (The value is specified in arguments of the string type and the number of characters which should be displayed (the int type) (The value is specified in arguments of the string type and the number of characters which should be displayed (the int type)</td>
<td>{Right(&quot;Beverages&quot;,3)) - three characters from the right side of the value will be displayed, ges.</td>
</tr>
<tr>
<td>{Roman()}</td>
<td>Converts Arabic numerals to Roman numerals: (The value is specified in arguments of the string type and the number of characters which should be displayed (the int type) (The value is specified in arguments of the string type and the number of characters which should be displayed (the int type) (The value is specified in arguments of the string type and the number of characters which should be displayed (the int type)</td>
<td>{Roman(4)) - the number 4 will have a Roman spelling. {Roman(DataSource.Column1,1)) - all the numbers from Column1 will have a Roman spelling.</td>
</tr>
<tr>
<td>{Substring()}</td>
<td>Displays a certain number of characters from the specified position: (The value (the string type) from which the characters will be displayed (The value (the string type) from which the characters will be displayed (The value (the string type) from which the characters will be displayed (The value (the string type) from which the characters will be displayed</td>
<td>{Substring(&quot;Beverages&quot;,6,3)) - the first six characters are skipped and three characters will be displayed, the result is ges.</td>
</tr>
</tbody>
</table>

2 Characters to be replaced (the string type)
3 Characters to be inserted (the string type)
4 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("Beverages",3)) - three characters from the right side of the value will be displayed, ges.
{Right(DataSource.Column1,4)) - for each Column1 value, four characters will be displayed from the right side.
{Roman(4)) - the number 4 will have a Roman spelling.
{Roman(DataSource.Column1,1)) - all the numbers from Column1 will have a Roman spelling.
{Substring("Beverages",6,3)) - the first six characters are skipped and three characters will be displayed, the result is ges.
{Substring("Beverages",0,2)) - two characters will be displayed starting with zero, the result will be Be.
{Substring(DataSource.Column1,1,4)) - the first
<table>
<thead>
<tr>
<th>Returns the value of the string type</th>
<th>Displays the currency value as the text:</th>
<th>{ToCurrencyWords()}</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ToCurrencyWords()}</td>
<td>In arguments, the following is specified:</td>
<td></td>
</tr>
<tr>
<td>➤ Returns the value of the string type</td>
<td>1. A numeric value (the double, decimal, long types)</td>
<td>{ToCurrencyWords(100)} - the used currency is dollars of the USA, so that the result will be One hundred dollars and zero cents.</td>
</tr>
<tr>
<td>➤ In arguments, the following is specified:</td>
<td>2. You can also specify values (true or false) to display them with a capital letter and display cents (type bool)</td>
<td>{ToCurrencyWords(Products.UnitPrice)} - all product prices will be displayed as text.</td>
</tr>
<tr>
<td>➤ Single and plural formats for currency and cents (the string type)</td>
<td>3. Single and plural formats for currency and cents (the string type)</td>
<td>{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), the result will be One hundred dollars.</td>
</tr>
<tr>
<td>➤ It is possible to specify the ISO code (the string type)</td>
<td>4. It is possible to specify the ISO code (the string type)</td>
<td>{ToCurrencyWordsEnGb(1.25,&quot;EUR&quot;,2)} - the ISO code EUR will be applied, and the result will be one euro and twenty-five cents.</td>
</tr>
<tr>
<td>➤ You can also specify a base unit for the integer part and a fractional</td>
<td>5. You can also specify a base unit for the integer part and a fractional</td>
<td>{ToCurrencyWordsEnIn(&quot;dollars&quot;,&quot;cents&quot;,1.25M,0,true)} - the base unit for the integer part as dollars will be specified, the fractional</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>{ToLowerCase()}</code></td>
<td>Displays the value in lowercase:</td>
<td><code>{ToLowerCase(&quot;EURO&quot;))</code> - the result is euro.</td>
</tr>
<tr>
<td></td>
<td>- The value is specified in arguments (the string type)</td>
<td><code>{ToLowerCase(DataSource.Column1))</code> - all values of this column will be</td>
</tr>
<tr>
<td></td>
<td>- Returns the value of the string type</td>
<td>displayed in lowercase.</td>
</tr>
<tr>
<td><code>{ToOrdinal()}</code></td>
<td>Converts numerals to ordinal:</td>
<td><code>{ToOrdinal(25))</code> - the result is 25th.</td>
</tr>
<tr>
<td></td>
<td>- The value is specified in arguments (the long type)</td>
<td><code>{ToOrdinal(DataSource.Column1))</code> - all the values of this column will</td>
</tr>
<tr>
<td></td>
<td>- Returns the value of the string type</td>
<td>be converted to ordinal numerals.</td>
</tr>
<tr>
<td><code>{ToProperCase()}</code></td>
<td>Converts the text to the format - the first character is capital, the rest</td>
<td>`{ToProperCase(&quot;dOllars&quot;)}) - the result is Dollars.</td>
</tr>
<tr>
<td></td>
<td>characters are in lowercase:</td>
<td>`{ToProperCase(&quot;dollars&quot;)}) - the result is Dollars.</td>
</tr>
<tr>
<td></td>
<td>- The value is specified in arguments (the string type)</td>
<td>`{ToProperCase(&quot;dOLLARS&quot;)}) - the result is Dollars.</td>
</tr>
<tr>
<td></td>
<td>- Returns the value of the string type</td>
<td><code>{ToProperCase(DataSource.Column1))</code> - all values from this column will</td>
</tr>
<tr>
<td></td>
<td></td>
<td>be with the first capital letter and the rest ones in lowercase.</td>
</tr>
<tr>
<td><code>{ToUpperCase()}</code></td>
<td>Displays the value in uppercase:</td>
<td>`{ToUpperCase(&quot;dollars&quot;)}) - the result is DOLLARS.</td>
</tr>
<tr>
<td></td>
<td>- The value is specified in arguments (the string type)</td>
<td>`{ToUpperCase(&quot;dOllars&quot;)}) - the result is DOLLARS.</td>
</tr>
<tr>
<td></td>
<td>- Returns the value of the string type</td>
<td>`{ToUpperCase(&quot;dOLLARS&quot;)}) - the result is DOLLARS.</td>
</tr>
<tr>
<td>{ToWords()}</td>
<td>Displays the numerals as text:</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In arguments, the following is specified:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 A numeric value that will be converted to text (decimal, double, long)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 True or false values whether to display the first character with a capital letter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 True or false values to return null and empty values</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 It is also possible to specify true or false values to provide the feminine form for the result</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns the value of the string type</td>
<td></td>
</tr>
<tr>
<td>{ToWords(100)}</td>
<td>- the result is one hundred.</td>
<td></td>
</tr>
<tr>
<td>{ToWords(100, true)}</td>
<td>- the result is One hundred.</td>
<td></td>
</tr>
<tr>
<td>{ToWordsEnIn(0,false)}</td>
<td>- the result is Zero.</td>
<td></td>
</tr>
<tr>
<td>{ToWordsEnIn(0,true)}</td>
<td>- there will be no results.</td>
<td></td>
</tr>
<tr>
<td>{ToWordsEs(100,true,true)}</td>
<td>- the result starts with a capital letter and in feminine form, Cien</td>
<td></td>
</tr>
<tr>
<td>{ToProperCase(DataSource .Column1)}</td>
<td>- all values will be displayed in text.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>{Trim()}</th>
<th>Trims the spaces at the beginning or end of the line:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The value is specified in arguments (the string type)</td>
</tr>
<tr>
<td></td>
<td>Returns the value of the string type</td>
</tr>
<tr>
<td>{Trim(&quot; &lt;1 dollars&gt; &quot;)}</td>
<td>- the result in this case is &lt;1 dollars&gt;&quot;.</td>
</tr>
<tr>
<td>{Trim(DataSource.Column 1)}</td>
<td>- the spaces before each value and after each value will be truncated.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>{TryParseDecimal()}{TryParseDouble()}{TryParseLong()}</th>
<th>Checks the value for conversion to decimal, double, long:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The value is specified in arguments (the string type)</td>
</tr>
<tr>
<td></td>
<td>Returns a value of the string type</td>
</tr>
<tr>
<td>{TryParseLong(&quot;100&quot;)}</td>
<td>- The value can be converted to long.</td>
</tr>
</tbody>
</table>
| {TryParseLong("{ 100"})} | - the result is false. The value cannot be converted
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 band which associates directly with any band that is associates with the Data band.

- **Not associated with bands.** In this case, the 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

To calculate and display the total, you should place a text component in the report, call the editor and go to the Summary tab.

1. **The Expression field.** This field specifies an expression of calculating totals. The expression can be specified manually, or it will be generated automatically, depending on the type of other parameters.
2. **The Summary Function field.** In this field, the function of calculating totals is selected.
3. 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.
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.

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).

The Conditions parameter. If the flag is checked, then the condition will be considered when calculating the totals. If unchecked, then the total will be calculated without considering conditions.

The field specifies an expression of a condition.

### The type of the total function result

By default, the function for calculating totals returns 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 avoiding losses in the calculation of totals.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type of returned value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum()</td>
<td>Decimal</td>
</tr>
<tr>
<td>SumD()</td>
<td>Double</td>
</tr>
<tr>
<td>SumI()</td>
<td>Int64</td>
</tr>
</tbody>
</table>

⚠️ 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 the 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.

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 conditions.

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 conditions.
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.

The 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 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**

To calculate the total by the page or panel, you should add the Latin letter "c" in lowercase 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.

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 software does not have any restrictions on 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**

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 is 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 on this. It is also allowed to combine the footers by the column with the condition:

\[ \{\text{ColCountIf(DataBand1, Products.UnitsInStock = 0)}\} \] - 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 software, you can calculate functions in the 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 influence this process. To make this calculation, you should create a variable in the data dictionary, which will store the value of the function.

Notice: Do not use variables declared in the code to store the result of the 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 a 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:

\[ \text{Variable} + = \text{Products.ItemsInStock}; \]

To display the result of calculation, you should place the text component with the expression:

\[ \{\text{Variable}\} \]

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 the 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 is not correct in relation to the result of the 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 (a master record), totals will be calculated:

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, 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:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Sample</th>
</tr>
</thead>
</table>

© 2003-2020 Stimulsoft
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{Avg()}</code></td>
<td>Calculates the arithmetic mean:</td>
<td><code>{Avg(DataSource.Column1)}</code> - the arithmetic mean of the Column1 column will be calculated. <code>{AvgDate(DataSource.ColumnDate)}</code> - the average of the date on the ColumnDate column will be calculated. <code>{AvgTime(DataSource.ColumnTime)}</code> - the average time by the ColumnTime column will be calculated.</td>
</tr>
<tr>
<td><code>{Count()}</code></td>
<td>Calculates the number of values or the number of unique values:</td>
<td><code>{Count()}</code> - the result is the number of entries in the data source. <code>{Count(DataSource1.Column1)}</code> - the result is the number of entries in Column1 column for DataSource1. <code>{CountDistinct(DataSource.Column1)}</code> - the result is the number of unique entries in DataSource.Column1. <code>{CountDistinct(DataBand2, DataSource.Column2)}</code> - the result is the number of unique entries in Column2 column for DataBand2.</td>
</tr>
<tr>
<td><code>{First()}</code></td>
<td>Displays the first value from the specified object:</td>
<td><code>{First(DataSource1.Column1)}</code> - the result is the first value of Column1 from the DataSource1. <code>{First(DataBand2, DataSource.Column2)}</code> - the result is the first value of Column2 of the</td>
</tr>
</tbody>
</table>
## Reports and Dashboards

<table>
<thead>
<tr>
<th>{Last()}</th>
<th>Displays the last value from the specified object: In arguments, specify an object or two objects Returns the values of various object types</th>
<th>{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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Max()}</td>
<td>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.</td>
<td>{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.</td>
</tr>
<tr>
<td>{Median()}</td>
<td>Displays the mean (non-arithmetic) value from the list: 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.</td>
<td>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.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>{Min()}</td>
<td>Displays the maximum value from the specified object.</td>
<td>{Min(DataSource1.Column1)} - the result is the minimum value from Column1 of DataSource1.</td>
</tr>
<tr>
<td></td>
<td>In arguments, specify an object or two objects</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns the values of various double, decimal, long, DateTime, TimeSpan, string types depending on the function selected.</td>
<td>{MinDate(DataSource1.ColumnDate)} - the result is the minimum date from the ColumnDate of the DataSource1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{MinTime(DataSource1.ColumnTime)} - the result is the minimum time from the ColumnTime of the DataSource1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{MinStr(DataSource1.Column1)} - all values will be sorted alphabetically. The result is the first value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{Min(DataBand2, DataSource.Column2)} - the result is the minimum value of Column2 of the DataBand2 band.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Mode()}</td>
<td>Displays the value that is most common in the list of values.</td>
<td>{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.</td>
</tr>
<tr>
<td></td>
<td>In arguments, specify an object or two objects</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns the values of various double, decimal, long types depending on the function selected.</td>
<td>{Mode(DataBand2, DataSource.Column2)} - the result will be the value from Column2 of the DataBand2, which is the most common.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Rank()}</td>
<td>Display the rank of the value. The prefix Totals is mandatory.</td>
<td>{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; 44 - rank 3; 55 - rank 4; 71 - rank 5.</td>
</tr>
<tr>
<td>processing and assigning rank (the object type)</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>2 The value (true or false) for assigning a tight or not tight rank</td>
<td>(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).</td>
<td></td>
</tr>
<tr>
<td>3 Sorting direction of values.</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

| {Sum()} | Displays the result of the sum of the values: |
| ➢ Specify in arguments: | {Sum(DataSource1.Column1)} - the result is the sum of all Column1 values in the DataSource1. |
| 1 Objects for processing and assigning rank (type object) | {SumDistinct(DataSource1.Column1)} - the result is the sum of all the unique Column1 values in the DataSource1. |
| 2 Condition | SumTime(DataSource1.Column1) - the result is the sum of the time from Column1 in the DataSource1. |
| 3 Summation expression | {Sum(DataBand2, DataSource2.Column2)} - the result is the sum of the values from Column2 of the DataBand2. |
| ➢ Returns the values of various long types | |
### TimeSpan types

\[
\{\text{SumDistinct(DataBand1, DataSource.Column1, DataSource.Column2)}\} - \text{the result is the sum of the Column2 values that correspond to the unique values from Column1 of the DataBand2.}
\]

### Samples to calculate totals not associated with bands.

For example, there is a Master-Detail Report, which is a list of products by categories:

![Master Detail Report]

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 \(\{\text{Sum(DataBand2, Products.UnitPrice)}\}\). 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 - 

\[
\frac{\text{Sum(DataBand2,Products.UnitsInStock)}}{\text{Totals.Sum(Products, Products.UnitsInStock)}}
\]

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 the 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 refer to 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 is 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 Bookmarks 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 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:

- Report
  - My Bookmark [My Bookmark]

To add a bookmark of the second level to the tree, it is necessary to write the following code:

`Bookmark["My Bookmark"].Add("Bookmark Level2");`

- Report
  - My Bookmark [My Bookmark]
    - Level 1

...and for the third level:

`Bookmark["My Bookmark"]["Level2"].Add("Bookmark Level3");`

- Report
  - My Bookmark [My Bookmark]
    - Level 1
    - Level 2

To create all three bookmarks, the code sample shown above can be used. Stimulsoft Reports automatically checks the 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:

```csharp
Bookmark.Add(Customers.CompanyName);
```

So the values of the `Interaction.Bookmark` property should be as follow:

```csharp
{Customers.CompanyName}
```

As a result, all components will be marked with a 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 a 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 the name of the highest level bookmark;

`nameN` is the name of the lowest level bookmark.

The picture below shows the expression hierarchy of a common type:

```
name1
   \name2
      ...\nameN
```

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, you should to specify an expression (see an example below):

```
%\{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 that correspond to the name of the country.
The middle level will be represented as bookmarks that correspond to the name of the company.
The lowest level will be represented as bookmarks that 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**
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:

![Report template pages](image)

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:

![Select Drill-Down Page](image)
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:

1. Speedy Express
2. United Package
3. Federal Shipping

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: \texttt{Products.ProductID}, \texttt{Products.ProductName} and \texttt{Products.UnitPrice}. For this band, you should select the data source \texttt{Products}. Also add the Header Band. The picture below shows a page template with detailed information:

![Page template with detailed information](image)

Add a filter with the expression \texttt{(int)this["CategoryID"] == Products.CategoryID} in the Data Band. After that, you must save the report template. For example save the report to: \texttt{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 \texttt{Categories.CategoryName}. For this band, you should select the data source \texttt{Categories}. The picture below shows a page of the report template with the main data:

![Page template with main data](image)

Then, select the text component and change the values of some properties. The Interaction.Drill-Down Enabled property must be set to \texttt{true}. Then, set the value of the Interaction.Drill-Down Report property to the full path to the report with detailed data.

```
Drill-Down Report | D:\Products.mrt
```

Also, specify the Drill-Down Parameters. In each parameter you must change the following properties: \texttt{Name} and \texttt{Expression}. In this case, define a detailed parameter with the name \texttt{CategoryId} and the expression \texttt{Categories.CategoryID}. Then render a
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** 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. 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 has names of categories, respectively.


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**

<table>
<thead>
<tr>
<th>Country</th>
<th>Year2005</th>
<th>Year2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argentina</td>
<td>319755</td>
<td>513224</td>
</tr>
<tr>
<td>Australia</td>
<td>384713</td>
<td>227238</td>
</tr>
<tr>
<td>Austria</td>
<td>263279</td>
<td>71714</td>
</tr>
<tr>
<td>Belarus</td>
<td>28996</td>
<td>11520</td>
</tr>
<tr>
<td>Belgium</td>
<td>920965</td>
<td>522810</td>
</tr>
<tr>
<td>Brazil</td>
<td>330340</td>
<td>2182617</td>
</tr>
<tr>
<td>Canada</td>
<td>300363</td>
<td>1409651</td>
</tr>
</tbody>
</table>

**Click this component**

<table>
<thead>
<tr>
<th>Country</th>
<th>Year2005</th>
<th>Year2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zimbabwe</td>
<td>950</td>
<td>830</td>
</tr>
<tr>
<td>Vietnam</td>
<td>21600</td>
<td>25480</td>
</tr>
<tr>
<td>Venezuela</td>
<td>135425</td>
<td>116297</td>
</tr>
<tr>
<td>Uzbekistan</td>
<td>84437</td>
<td>117900</td>
</tr>
<tr>
<td>United States</td>
<td>11946883</td>
<td>5711823</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>1803109</td>
<td>1090139</td>
</tr>
<tr>
<td>Ukraine</td>
<td>215759</td>
<td>85295</td>
</tr>
</tbody>
</table>

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:

When rendering the report without sorting, data are taken from the data source sequentially. 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. 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:

<table>
<thead>
<tr>
<th>CategoryName</th>
<th>ProductName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beverages</td>
<td>SteelEye Stout</td>
</tr>
<tr>
<td>Beverages</td>
<td>Sasquatch Ale</td>
</tr>
<tr>
<td>Beverages</td>
<td>Rhônbushlau Klosterbier</td>
</tr>
<tr>
<td>Beverages</td>
<td>Outback Lager</td>
</tr>
<tr>
<td>Beverages</td>
<td>Laughing Lumberjack Lager</td>
</tr>
<tr>
<td>Beverages</td>
<td>Lakkolikööri</td>
</tr>
<tr>
<td>Beverages</td>
<td>Ipoh Coffee</td>
</tr>
<tr>
<td>Beverages</td>
<td>Guarana Fantástica</td>
</tr>
<tr>
<td>Beverages</td>
<td>Côte de Blaye</td>
</tr>
<tr>
<td>Beverages</td>
<td>Chartreuse verte</td>
</tr>
<tr>
<td>Beverages</td>
<td>Chang</td>
</tr>
<tr>
<td>Beverages</td>
<td>Chai</td>
</tr>
<tr>
<td>Condiments</td>
<td>Veggie-spread</td>
</tr>
<tr>
<td>Condiments</td>
<td>Sirop d'étable</td>
</tr>
<tr>
<td>Condiments</td>
<td>Original Frankfurter grüne Soße</td>
</tr>
<tr>
<td>Condiments</td>
<td>Northwoods Cranberry Sauce</td>
</tr>
</tbody>
</table>

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 \[\text{icon} \] is displayed and the icon \[\text{icon} \] 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.

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 viewer.
This chapter covers the following:

- Creating dashboards;
- The size of a dashboard;
- Elements of data analysis;
- Elements of data filtering;
- Other dashboard elements;
- Actions with the dashboard;
- Adding items to the dashboard;
- Changing the item type.
Creating dashboards

Do the following steps 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 stretch by width and height, in proportion to the original dimensions in the report designer.

Information

The data filtering elements of the drop-down type (**combo box**, **date picker**, **tree view box**) do not stretch by height.
When creating the dashboard in the designer, its size has a working area that looks like a white sheet with a grid. 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 increase, while the size of the elements will remain unchanged.

All elements of the dashboard are grouped into the following categories according to their functionality:

**Elements of data analysis**

- Table;
- Chart;
- Pivot;
- Indicator;
- Progress;
- Maps;

**Elements of data filtering**

- List box;
- Combo box;
❯ Tree View;
❯ Tree View Box;
❯ Date Picker.

Other dashboard elements
❯ Panel;
❯ Text;
❯ Image;
❯ Shapes.

Actions with the dashboard
❯ View the dashboard;
❯ Export the dashboard and its items;
❯ Share the dashboard and embed it into your website;
❯ Publish the dashboard.

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 to which you want to convert the current one.
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:

- **Styles**;
- **Fore Color**;
- **Back Color**;
- **Margins and padding**;
- **Titles elements**;
- **Text Format**.

**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.

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.
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. 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.

![Theme Colors](image)

**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. 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`.

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.

For charts, you can specify the formatting of the chart axis values. To do this:

- Select a chart on a dashboard panel;
- Click the Browse button of the Argument Format or Value Format property and respectively set up the formatting of the arguments or chart values.
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.

This chapter will cover the following:

- Data fields;
- Data filed expression;
- Adding data to an element;
- Putting values manually;
- Removing data from elements;
- Table of Functions.

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 one 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:
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. 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 do 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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions that are available from the menu of the Expression field</td>
<td></td>
</tr>
<tr>
<td>Count()</td>
<td>Calculates the number of values in the current data field.</td>
</tr>
<tr>
<td>DistinctCount()</td>
<td>Calculates the number of unique values in the current data field.</td>
</tr>
<tr>
<td>First()</td>
<td>Shows the first value of the current data field.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Last()</td>
<td>Shows the last value of the current data field.</td>
</tr>
<tr>
<td>Sum()</td>
<td>Shows the result of the sum of values in the current data field.</td>
</tr>
<tr>
<td>Avg()</td>
<td>Calculates the arithmetic average for the values of the current data field.</td>
</tr>
<tr>
<td>Min()</td>
<td>Shows the minimum value from the current data field.</td>
</tr>
<tr>
<td>Max()</td>
<td>Shows the maximum value from the current data field.</td>
</tr>
<tr>
<td>Median()</td>
<td>Shows the average (non arithmetic) value from the current data field.</td>
</tr>
<tr>
<td>Year()</td>
<td>Shows the year from date encoding.</td>
</tr>
<tr>
<td>Quarter()</td>
<td>Shows the quarter from the date encoding.</td>
</tr>
<tr>
<td>Month()</td>
<td>Shows the month from date encoding.</td>
</tr>
<tr>
<td>Day()</td>
<td>Shows the day from date encoding.</td>
</tr>
</tbody>
</table>

**Functions that can be added from the data dictionary or entered manually**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode()</td>
<td>Shows the most frequently repeating values in the current data field.</td>
</tr>
<tr>
<td>List()</td>
<td>Enters a list of values for the current data field of an item.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Array()</td>
<td>Enters an array of values for the current data field of an item.</td>
</tr>
<tr>
<td>ToUpperCase()</td>
<td>Converts all data field values to uppercase.</td>
</tr>
<tr>
<td>ToLowerCase()</td>
<td>Converts all data field values to lowercase.</td>
</tr>
<tr>
<td>ToProperCase()</td>
<td>Sets the first character value to uppercase, and the remaining characters to lowercase.</td>
</tr>
<tr>
<td>Insert(,,)</td>
<td>Inserts text into data field values, after a specific character. Three arguments are specified through the &quot;,&quot; delimiter: 1 Data field; 2 The ordinal number of the character after which another value will be inserted. 3 The value to be inserted.</td>
</tr>
<tr>
<td>Replace(,,)</td>
<td>Replaces certain characters in values. Three arguments are specified through the &quot;,&quot; delimiter: 1 Data field; 2 A character or combination of characters that needs to be replaced. 3 The value to be replaced.</td>
</tr>
<tr>
<td>Remove(,,)</td>
<td>Removes the specified number of characters in the values. Three arguments are specified</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>DayOfWeek()</td>
<td>Shows day of week from date encoding.</td>
</tr>
<tr>
<td>DaysInMonth()</td>
<td>Shows the number of days in a month.</td>
</tr>
<tr>
<td>DaysInYear()</td>
<td>Shows the number of days per year.</td>
</tr>
<tr>
<td>Month()</td>
<td>Shows the number of the month.</td>
</tr>
<tr>
<td>ISO2()</td>
<td>Shows the two-letter code of the geographical object.</td>
</tr>
<tr>
<td>ISO3()</td>
<td>Shows the three-letter code of the geographical object.</td>
</tr>
<tr>
<td>NormalizeName()</td>
<td>Shows the names of the geographical objects by default.</td>
</tr>
<tr>
<td>Left(,)</td>
<td>Shows the specified number of characters from the left side of the value. Two arguments are specified through the &quot;,&quot; delimiter: 1 Data field; 2 The number of characters to show.</td>
</tr>
<tr>
<td>Mid(,)</td>
<td>Shows characters from a value. Three arguments</td>
</tr>
</tbody>
</table>
are specified through the "," delimiter:
1. Data field;
2. The ordinal number of the character with which to start the display.
3. 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:
1. Data field;
2. The number of characters to show. |
|----------|--------------------------------------------------------------------------------|

| Substring(,,) | Shows characters from a value. Three arguments are specified through the "," delimiter:
1. Data field;
2. The ordinal number of the character with which to start the display.
3. 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.
This chapter will cover the following:

- Creating groups;
- Removing elements from the group.

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

- 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.
This chapter will cover the following:

- **Table Editor**;
- **The Order of Elements**;
- **Size Mode**;
- **Grouping data**;
- **Images in Table**;
- **Header Menu**;
- **Table of Properties**.

The options for displaying the values of the Table element are made in its editor. 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.
Information

Text formatting and Interaction can be applied to the values of the current element.

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. 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**

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.
Images in Table
In the table, you can display images obtained from data sources, as well as images obtained by URL. To display images in a table from a data source, you should add the data field to the list of table fields.

If the data field contains image URLs, then by default, these URLs will be displayed as text in the table. To get images by URL and display them in a table, you should:

- Select the data field with the image URL in the Table editor;
- Apply the `Image()` function to the expression of this field. For example, `Image(DataSource DataColumn1)`.
- Specify the height and width of the image in the function arguments, if the URL redirects to an SVG image - `Image(DataSource DataColumn, height, width)`.

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. 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.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Size Mode</td>
<td>Sets the size mode of the columns of the element:</td>
</tr>
<tr>
<td></td>
<td>➤ <strong>AutoSize</strong> - optimal column widths will be calculated;</td>
</tr>
<tr>
<td></td>
<td>➤ <strong>Fit</strong> - the columns will be proportionally stretched across the entire width of the element.</td>
</tr>
<tr>
<td>Back Color</td>
<td>Changes the background color of the element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the element will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Footer Font</td>
<td>A group of properties defines the font family, its style, and size for the footer values of the element.</td>
</tr>
<tr>
<td>Footer Fore Color</td>
<td>Specifies the color of the footer values of the element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the footer values will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>Specifies the color of the values of the element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the values will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Header Font</td>
<td>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.</td>
</tr>
<tr>
<td>Header Fore Color</td>
<td>Determines the color of the headers of the values of the element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the value headers will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Style</td>
<td>Selects a style for the current element. The default it is set to <strong>Auto</strong>, i.e. the style of this element is inherited from the style of the dashboard.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to <strong>True</strong>, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
</tr>
<tr>
<td>Interaction</td>
<td>Sets <strong>interaction</strong> of the current element.</td>
</tr>
<tr>
<td>Margin</td>
<td>A group of properties that allows you to define margin (left, top, right, bottom) of the value area</td>
</tr>
</tbody>
</table>
### Padding

A group of properties that allows you to define padding (left, top, right, bottom) of the columns from the range of values.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>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.</td>
</tr>
<tr>
<td>Font</td>
<td>The group property that allows you to define the font family, its style and size for the title of the current element.</td>
</tr>
<tr>
<td>Horizontal Alignment</td>
<td>The property provides the ability to change the title alignment relative to the element - Left, Center, Right.</td>
</tr>
<tr>
<td>Text</td>
<td>The property is used to set the title text of the current element.</td>
</tr>
<tr>
<td>Visible</td>
<td>The property is used to enable or disable displaying of the title of the current item. If the property is set to True, then the title will be displayed.</td>
</tr>
</tbody>
</table>

---

**© 2003-2020 Stimulsoft**
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Changes the name of the current element.</td>
</tr>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Change</strong> option enables or disables changes of the element.</td>
</tr>
</tbody>
</table>
|           |   - 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 |
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.
Text formatting and Interaction can be applied to the values of the current element.

This chapter will cover the following:
- Chart Editor;
- Chart Values;
- Chart Types;
- Chart Arguments;
- Chart Series;
- The Color Each Property;
- Chart Legend;
- Constant Lines;
- Constant Lines Editor;
- Constant Line Parameters;
- Table of Properties.

**Chart Editor**
You can configure the Chart element in the special editor. 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**

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.

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.
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.
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.
Reports and Dashboards

Information

Setting the legend is done using properties from the Legend group on the property panel. You can adjust the alignment of the legend horizontally and vertically, the title of the legend, the text of the legend, and etc.

Constant lines

Constant lines are used to display value lines in the chart area.
You should do the following to add constant lines in the chart:

- Select a chart in the dashboard;
- On the property panel, click the **Browse** button of the **Constant Lines** property.

After that, the editor will open. Configure constant lines for the current chart in the editor.

**Constant Lines editor**

In the current editor, you can add, setup and remove constant lines for the current chart.
1 The **Add** button. It is used to add a new constant line in the constant lines list.
2 The **Remove** button. It is used to delete the selected constant line.
3 The arrow buttons. They are used to move the selected constant line in the list.
4 The First Constant Line.
5 The Second Constant Line.

**Constant Line parameters**
Each constant line has parameters. You can setup a constant line with using these parameters.
1 The **Title** parameter. It allows you to add a constant line title. If This Field is Empty, Constant Line will not have a title.

2 The **Value** parameter. It allows you to specify a constant line value. Constant Line will be displayed in chart area by this value. Also, you can specify expression in the field of this parameter. The result of the expression calculation will be the constant line value.

3 The **Color** parameter. It allows you to select a constant line color.

4 The **Style** parameter. It allows you to change a constant line style.

5 The **Width** parameter. It allows you to select a width of a constant line.

**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>A property group that is used to customize the chart area: ▶ The <strong>Color Each</strong> property is used to set a unique shade for every graphic element of the chart. If this property is set to <strong>true</strong>, 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,</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Constant Lines</th>
<th>Customizes the constant lines of the chart element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
</tbody>
</table>

**Labels**

A group of properties that is used to customize the chart labels:

- The **Auto Rotate** property is used to enable or disable the auto rotate mode of chart labels.
- The **Font** group property allows you to change the text color of the title of the current item. By default, this property is set to From Style, the text color of the title will be obtained from the settings of the current element style.
- The **Fore Color** property allows you to change the text color of the labels of the current item. By default, this property is set to From Style, the text color of the labels will be obtained from the settings of the current element style.
- The **Style** property allows you to change the label style.
- The **Text After** property is used to specify text after a label value.
- The **Text Before** property is
| Legend | A property group that is used to customize the chart legend:

- **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.
- **Labels** group of properties is used to change the color and font of the legend label.
- **Title** group of properties is used to customize the title of the legend - specify the text of a title, change its color and font.
- **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. |

| Marker | A group of properties that is used to customize the chart markers:

- **Angle** property allows you to change the inclination angle of markers. The value of the property can be negative and positive. If a value of the property is negative then the marker is inclined anticlockwise. If the value of the property is positive then the marker is inclined clockwise.
- **Size** property is used to set marker size in pixels.
- **Type** property allows you to set the marker type. |
The **Visible** property is used to define the display mode of markers:

1. The **From Style** value - displaying of markers will depend on the visibility property in the chart style.
2. The **True** value - markers will always be displayed.
3. The **False** value - markers will not be displayed always.

<table>
<thead>
<tr>
<th>X axis</th>
<th>A group of properties that is used to customize the value axis:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>❯ The <strong>Labels</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>❯ The <strong>Title</strong> 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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Y axis</th>
<th>A group of properties that is used to customize the value axis:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>❯ The <strong>Labels</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>❯ The <strong>Title</strong> 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.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Back Color</td>
<td>Changes the background color of the Chart element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the element will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of a table - color, sides, size, and style.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Customizes the <a href="#">conditions element of the chart.</a></td>
</tr>
<tr>
<td>Negative Series Color</td>
<td>Customizes the list of colors for negative values of the rows of the Chart element.</td>
</tr>
<tr>
<td>Series Color</td>
<td>Customizes the list of colors for the values of the rows of the Chart element.</td>
</tr>
<tr>
<td>Style</td>
<td>Selects a style for the current element. The default it is set to <strong>Auto</strong>, i.e. the style of this element is inherited from the style of the dashboard.</td>
</tr>
<tr>
<td>Argument Format</td>
<td>Customizes the formatting of the arguments the Chart element.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to <strong>True</strong>, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
</tr>
<tr>
<td>Interaction</td>
<td>Customizes the <a href="#">interaction element of the chart.</a></td>
</tr>
<tr>
<td>Margin</td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Padding</td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
<tr>
<td>Title</td>
<td>A group of properties that allows you to customize the title of the Table element:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Back Color</strong> property provides the ability to change the background color of the title of the current item. By default, this property is set to <strong>From Style</strong>, i.e. the background color will be obtained from the style settings of the current element.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Fore Color</strong> allows you to change the text color of the title of the current item. By default, this property is set to <strong>From Style</strong>, i.e. the text color of the title will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td></td>
<td>- The group property <strong>Font</strong> allows you to define the font family, its style and size for the title of the current element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Horizontal Alignment</strong> property provides the ability to change the title alignment relative to the element - Left, Center, Right.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Text</strong> property is used to set the title text of the current element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Visible</strong> property is used to</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Value Format</td>
<td>Customizes the formatting of the values of the Chart element.</td>
</tr>
<tr>
<td>Name</td>
<td>Changes the name of the current element.</td>
</tr>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>- The Allow Change option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>- The Allow Delete option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>- The Allow Move option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>- The Allow Resize option enables or disables resizing of an element.</td>
</tr>
<tr>
<td></td>
<td>- The Allow Select option enables or disables the element selection.</td>
</tr>
<tr>
<td>Locked</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If this property is set to False, then this</td>
</tr>
<tr>
<td>Linked</td>
<td>Binds the current location to the dashboard or another element. If the property is set to <strong>True</strong>, then the current item is bound to the current location. If this property is set to <strong>False</strong>, then this element is not tied to the current location.</td>
</tr>
</tbody>
</table>

### 4.6 Gauge

**Gauge** is an element of the dashboard panel using which you can display the processed value from the data field.

This chapter will cover the following:

- **Gauge Editor**;
- **Gauge Values**;
- **Gauge Series**;
- **Gauge Types**;
- **Range of Values**;
- **Colored Ranges**;
- **Table of Properties**.
You may adjust the **Gauge** element in its editor. 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.

**Gauge Editor**

In this editor adjusts the gauge.
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**
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.

![Full Circular Gauge](image)

**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.

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.
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.

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.
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 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific <a href="#">group of items</a>.</td>
</tr>
<tr>
<td>Short Value</td>
<td>Applies an abbreviation for the values of the gauge. Has two values - true and false.</td>
</tr>
<tr>
<td>Back Color</td>
<td>Changes the background color of the element. By default, this property is set to <a href="#">From Style</a>, i.e. the color of the element will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>Specifies the color of the values of the element. By default, this property is set to <a href="#">From Style</a>, i.e. the color of the values will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Style</td>
<td>Selects a style for the current element. The default it is set to <a href="#">Auto</a>, i.e. the style of this element is inherited from the style of the dashboard.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>Sets interaction of the current element.</td>
</tr>
<tr>
<td><strong>Margin</strong></td>
<td>A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td><strong>Padding</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>
| **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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Move</strong> option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Resize</strong> option</td>
</tr>
</tbody>
</table>
enables or disables resizing of an element.

The **Allow Select** option enables or disables the element selection.

<table>
<thead>
<tr>
<th>Locked</th>
<th>Locks or unlocks resizing and movement of the current element. If the property is set to <strong>True</strong>, the current element cannot be moved or resized. If this property is set to <strong>False</strong>, then this element can be moved and resized.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linked</td>
<td>Binds the current location to the dashboard or another element. If the property is set to <strong>True</strong>, then the current item is bound to the current location. If this property is set to <strong>False</strong>, then this element is not tied to the current location.</td>
</tr>
</tbody>
</table>

### 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.
This chapter will cover the following:

- **Pivot Editor**;
- **Totals**;
- **Rows**;
- **Columns**;
- **Table of Properties**.

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. 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.
**Information**

Text formatting can be applied to the values of the current element.

**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. 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.

<table>
<thead>
<tr>
<th>Pivot Table</th>
<th>CategoryName</th>
<th>Beverages</th>
<th>Condiments</th>
<th>Grains/Cereals</th>
<th>Meat/Poultry</th>
<th>Seafood</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ProductName</strong></td>
<td><strong>Country</strong></td>
<td><strong>Age</strong></td>
<td><strong>Gender</strong></td>
<td><strong>Income</strong></td>
<td><strong>Education</strong></td>
<td><strong>Occupation</strong></td>
<td><strong>Total Sales</strong></td>
</tr>
<tr>
<td>Chai</td>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chang</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chartreuse</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>verte</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chef Anton’s Gumbo Mix</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Côte de Biaye</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guaraná Fantástica</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gula Malacca</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rød Kaviar</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Røgede sild</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sasquatch Ale</td>
<td>111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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.
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.

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current element.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Style</td>
<td>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.</td>
</tr>
<tr>
<td>Enabled</td>
<td>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.</td>
</tr>
<tr>
<td>Interaction</td>
<td>Sets interaction of the current element.</td>
</tr>
<tr>
<td>Margin</td>
<td>A group of properties that allows</td>
</tr>
<tr>
<td><strong>Padding</strong></td>
<td>A group of properties that allows you to define padding (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
</tbody>
</table>
| **Title** | A group of properties that allows you to customize the title of the element:  
- **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.  
- **Font** that allows you to define the font family, its style and size for the title of the current element.  
- **Horizontal Alignment** property provides the ability to change the title alignment relative to the element - Left, Center, Right.  
- **Text** property is used to set the title text of the current element.  
- **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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard: The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed. The <strong>Allow Delete</strong> option enables or disables the deletion of an element. The <strong>Allow Move</strong> option allows or prohibits moving an element. The <strong>Allow Resize</strong> option enables or disables resizing of an element. The <strong>Allow Select</strong> option enables or disables the element selection.</td>
</tr>
<tr>
<td>Locked</td>
<td>Locks or unlocks resizing and movement of the current element. If the property is set to <strong>True</strong>, the current element cannot be moved or resized. If this property is set to <strong>False</strong>, then this element can be moved and resized.</td>
</tr>
<tr>
<td>Linked</td>
<td>Binds the current location to the</td>
</tr>
</tbody>
</table>
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.
This chapter will cover the following:

- Indicator Editor;
- Indicator Value;
- Indicator Target Value;
- Indicator Series;
- Indicator Icon;
- Table of Properties.

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 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.

**Information**

Text formatting can be applied to the values of the current element.

**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.
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
Flø temysost
Geitost
Genen Shouyu
Gnocchi di nonna Alice
Gorgonzola Telino
Grandma’s Boysenberry Spread
Gravad lax
Guaraná Fantástica
Gudbrandsdalsost
Gula Malacca
Gumbär Gummibärchen
Gustaf’s Knäckebrot

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.
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.
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.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current element.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Icon Alignment</td>
<td>Changes alignment of the element icon.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Customizes the conditions element of the indicator.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>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.</td>
</tr>
<tr>
<td>Glyph Color</td>
<td>Changes the color of the glyph.</td>
</tr>
<tr>
<td>Style</td>
<td>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</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Enables or disables the current item on the dashboard. If the property is set to <strong>True</strong>, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>Sets interaction of the current element.</td>
</tr>
<tr>
<td><strong>Margin</strong></td>
<td>A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td><strong>Padding</strong></td>
<td>A group of properties that allows you to define padding (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
<tr>
<td><strong>Text Format</strong></td>
<td>Sets the formatting values of the element.</td>
</tr>
</tbody>
</table>
| **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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locked</td>
<td>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.</td>
</tr>
<tr>
<td>Linked</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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.
This chapter will cover the following:

- **Progress Editor**;
- **Progress Value**;
- **Progress Target Value**;
- **Progress Series**;
- **Progress Types**;
- **Table of Properties**.

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. 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.

**Information**

Text formatting can be applied to the values of the current element.

**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
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.
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.
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.
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.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current element.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Color Each</td>
<td>Sets a unique shade for every graphic element of the progress.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>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.</td>
</tr>
<tr>
<td>Series Color</td>
<td>Customizes the list of colors for the series of the element.</td>
</tr>
<tr>
<td>Style</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Enables or disables the current item on the dashboard. If the property is set to <strong>True</strong>, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>Sets interaction of the current element.</td>
</tr>
<tr>
<td><strong>Margin</strong></td>
<td>A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td><strong>Padding</strong></td>
<td>A group of properties that allows you to define padding (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
<tr>
<td><strong>Text Format</strong></td>
<td>Sets the formatting values of the element.</td>
</tr>
<tr>
<td><strong>Title</strong></td>
<td>A group of properties that allows you to customize the title of the element: The <strong>Back Color</strong> property provides the ability to change the background color of the title of the current item. By default, this property is set to <strong>From Style</strong>, 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 <strong>From</strong></td>
</tr>
</tbody>
</table>
**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>The <strong>Allow Move</strong> option allows</td>
</tr>
<tr>
<td><strong>Locked</strong></td>
<td>The <strong>Allow Resize</strong> option enables or disables resizing of an element. The <strong>Allow Select</strong> option enables or disables the element selection.</td>
</tr>
<tr>
<td><strong>Linked</strong></td>
<td>Locks or unlocks resizing and movement of the current element. If the property is set to <strong>True</strong>, the current element cannot be moved or resized. If this property is set to <strong>False</strong>, then this element can be moved and resized.</td>
</tr>
</tbody>
</table>

### 4.10 Maps

Map is an element of the dashboard, which provides the ability to display data with reference to geographic location.
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.
This chapter will cover the following:

- Region Map Editor;
- Region Map Editor (Data Columns Mode);
- Adding a custom map;
- Creating a map file;
- Editing a custom map;
- An example of adding a map;
- Table of Properties.
Interaction can be applied to the values of the current element.

The Region Map element can be placed anywhere on the dashboard. This item is configured in its editor. 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.
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.

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.

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.

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.
The **Key** field indicates a data field with a list of keys of map elements of a certain type.

The **Name** field indicates a data field with names for map elements of a certain type.

The **Value** field indicates a data field with values for every map element of a certain type.

The **Color** field indicates a data field that contains the color as `#FFFFFF` 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.

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.

The **Expression** field displays the expression of the selected data item.

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.
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.

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.

**Adding a custom map**

When designing dashboards, you can add a custom map. This map will be displayed in the common list of maps and in the user category.

To use a custom map in the design you should:

- Add a map file to resources of a report;
- In the editor of the **Regional map**, select this type of the map or drag and drop the resource from the dictionary to the dashboard.
Information

If you an invalid map file to the report resources, this type of the map will be marked in the list with the icon as on the picture below.

Creating a map file
A map file has the *.map extension, with the JSON markup of geographic data. The map file must contain the following fields:

- **Name.** This is the name of the map;
- **Width** and **Height.** Sets the width and height of the map.
- The **Paths** array. Contains data of geographic objects of the map.

Each geographic object in the **Paths** array must contain the following fields:

- **Key.** This is the identifier of the geographic object. It may only contain English characters "a-z". It cannot contain spaces, special characters, dashes, etc.
- **EnglishName.** This is the name of the geographic object.
- **Data.** This is a patch of a geographic object.
- **ISOCODE.** This is the ISO code of a geographic object.
Editing a custom map

You may edit each map that is added to report resources. To do this:

- Call the map resource editing form;
- Click the **Edit** button in the resource editing form.
After that, the map editor will be called. In this editor, you can enable or disable geographic objects, customize the titles of geographic objects, and assign an icon to the map.

**Information**

Titles of geographic objects will be obtained from the `EnglishName` fields in the `*.map` file. Each title has an area in which the title text is placed. This area can be moved using the cursor keys (left, right, top, bottom). To resize an area, hold down the `Shift` key and use the cursor keys (top, right, left, bottom) to increase or decrease
the size of the area in the corresponding directions.

Keys (Left, Top, Right, Bottom) change the position of the item.
Shift + keys (Left, Top, Right, Bottom) change the size of the item.
A panel displays a list of geographic objects of the map. If the check box is selected, then the geographic object will be displayed on the preview panel of the current editor. If the box is unchecked, then the geographic object will not be displayed.

Map preview. This panel displays only enabled geographic objects.

The commands are used to align a title of a geographic object vertically.

The commands are used to align a title of a geographic object horizontally.

The option is used to wrap the title text. If the **Word Wrap** option is enabled, the title will be wrapped to the next line. Otherwise the text wrapping will be cut off along the border of the title area.

The **Hide Text** option. It is used to hide the title of the selected geographic object.

The **Icon** option. It is used to load a map icon. This icon will appear as a thumbnail in the map selection window.

---

**An example of adding a map**

Consider the example of adding a **Haiti** map to the list of regional maps.

**Step 1**: You should find the source of the map, which you will integrate into the list of maps, for example, **Haiti.svg**.

**Step 2**: Open this file using the editor, in the example case, using VSCode.

**Step 3**: Create a text file named **Haiti.txt** and open it in VSCode;

**Information**

Since the map file is created in the **JSON format**, you should first check the rules how to format this.

**Step 4**: In the **Haiti.txt** file, add the **Name**, **Width**, **Height** fields with the values. In the current example, the options added are "Name": "Haiti", "Width": 700, "Height": 700.

```
"Name": "Haiti",
"Width": 700,
"Height": 700
```
Step 5: In the file Haiti.txt, add a Paths array and move on to creating geographic objects of the map. To create a geographic object, you should specify the Key, EnglishName, Data, and ISOCODE fields with values. Values for these fields can be taken from the source file Haiti.svg.

Step 6: Create the Key field in Haiti.txt and copy the value from the source file there. In the current Haiti.svg source file, you need to copy the value from the title field.

Information

Keep in mind that the Key field cannot contain spaces, dashes, special characters, etc. Key field can contain only Latin letters. Therefore, if the source file contains invalid characters, then when copying the values, they must be deleted.

All values in the Key field must be unique. It is not allowed to use the same values in several geographic objects. Each geographic object must have its own value in the Key field.

Step 7: Create the EnglishName field in the file Haiti.txt and copy the value from the source file there. This is the name of the geographic objects that will be displayed. Unlike the Key field, the value of this field may contain various characters.

Step 8: Create the Data field in Haiti.txt and copy the value from the source file there. In the current Haiti.svg source file, you need to copy the value from the d field.

Step 9: Create the ISOCODE field in Haiti.txt and copy the value from the source file there. In the current Haiti.svg source file, you need to copy the value from the id field.
Step 10: Add the required number of geographic objects of the map;

Step 11: After adding all the geographic objects, you need to save the changes in the Haiti.txt file;

Step 12: Change the file type of Haiti.txt to Haiti.map;
Step 13: Launch the report designer and drag the Haiti.map file into the data dictionary;
**Step 14:** Add the Regional map element to the dashboard panel;

**Step 15:** In the map editor, select the Custom category, choose Haiti and click OK;

**Step 16:** Set the values of geographic objects and set the parameters of the Regional map element;

**Step 17:** Close the Regional map editor.

Now using this map you can design dashboards.
The added maps can be customized. Do the following to customize them:

**Step 1:** Select the **Haiti** resource in the data dictionary and click the **Edit** button in the data dictionary;
Step 2: Click the **Edit** button in the edit resource dialog;
Step 3: The map editor will be called;

Step 4: Uncheck the geographic objects, if you want to disable displaying them when working in the current editor;
Step 5: Select a geographic object in the list and change the location of the title area using the cursor keys (left, right, top, and bottom);

Step 6: Hold down the Shift key and use the cursor keys (left, right, top, and bottom) to resize the header area;

Step 7: Using the horizontal and vertical alignment, set the location of the title in its area;

Step 8: Enable text wrapping, if it is necessary that the title wraps to the next line when the text reaches the right border of the title area.

Step 9: Select the Hide Text option, if you do not want to display the title text;

Step 10: Click on the icon for the Icon parameter, if it is necessary to load a custom icon for the current map;
**Step 11:** Click the **OK** button in the map editor;

**Step 12:** Click the **OK** button in the resource editor.

Changes will be applied to the map of this type.
**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Short Values</td>
<td>Enable or Disable of using abbreviation mode of map values.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Style</td>
<td>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.</td>
</tr>
<tr>
<td>Enabled</td>
<td>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</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Interaction</td>
<td>Sets <code>interaction</code> of the Region Map element.</td>
</tr>
<tr>
<td>Margin</td>
<td>A group of properties that is used to define margins (left, top, right, bottom) of the map area from the border of this element.</td>
</tr>
<tr>
<td>Padding</td>
<td>A group of properties that is used to define padding (left, top, right, bottom) of the map area from the border of this element.</td>
</tr>
</tbody>
</table>
| 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... |
<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Move</strong> option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Resize</strong> option enables or disables resizing of an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Select</strong> option enables or disables the element selection.</td>
</tr>
<tr>
<td>Locked</td>
<td>Locks or unlocks resizing and movement of the current</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>element</strong></td>
<td>If the property is set to <strong>True</strong>, the current element cannot be moved or resized. If this property is set to <strong>False</strong>, then this element can be moved and resized.</td>
</tr>
<tr>
<td><strong>Linked</strong></td>
<td>Binds the current location to the dashboard or another element. If the property is set to <strong>True</strong>, then the current item is bound to the current location. If this property is set to <strong>False</strong>, then this element is not tied to the current location.</td>
</tr>
</tbody>
</table>

**4.10.2 Online Map**

**Online Map** is used to display any object by geographic coordinates on the online map from Bing.
This chapter will cover the following:

- **Editor by coordinates**;
- **Editor by Location**;
- **Table of Properties**.

The **Online Map** element can be placed anywhere on the dashboard. This item is configured in the element editor. To call the editor, you should:

- Double-click on an item;
- Select **Online Map**, and select the **Design** command in the context menu.

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.

You can display the Objects on Online Map by:
- Geographic coordinates (Latitude and Longitude);
- Location.

![Online Map](image)

**Information**

The Online Map editor will contain various parameters depending on the way objects are displayed.

**Editor by coordinates**

Online map is used to display any object by geographic coordinates and works only with data elements.
The **Latitude** field indicates the data field with the latitude value of the geographical object.

2 The **Longitude** field indicates the data field with the longitude value of the geographical object.

3 The **Expression** field displays the expression of the selected data field.

4 The **Icon** parameter is used to select or load an icon for the value of a geographic object, as well as set the color of this icon.

**Editor by Location**
If you display the objects by location, then the online map editor will contain the following parameters.
1 The **Location** field. It indicates data column with the location of geographic objects. These may be state names, postal codes, etc.

2 The **Value** field. It indicates a data column with values of geographic objects.

3 An expression of the selected data field will be displayed in this field.

4 The **View Mode** parameter. It allows you to determine the option to display the value of a geographical object:
   - **Value.** A numerical value will be displayed for each geographic object.
   - **Bubble.** A separate bubble will represent the value of every geographic object. The larger is the value; the larger is the bubble in diameter.
   - **Icon.** A numerical value with a specific symbol will be displayed for each geographic object. The icon can be selected from the Stimulsoft collection or uploaded from your local storage. Also, for the icons from the collection, you can set its color.
   - **Chart.** If you select this value, an additional field called Arguments will be displayed. In this field, you should specify the data field, the values from which will be the arguments for the chart values of each geographic object. In other words, each value is a pie chart sliced by arguments.

5 The **Icon** parameter is used to select or load an icon for the value of a geographic
object, as well as set the color of this icon.

6 The **Type** parameter. It allows you to set a location. The Bing service does not always correctly determine the data type of a location.

7 The **Culture** parameter. It allows to specify a culture of a map.

8 The **Color** parameter. It allows you to specify color of geographic objects. The following values are available:
   - The **Color Each**. Each geographic objects will have a unique color;
   - The **Value**. The Color Value field will be displayed in the editor. You should specify data field with a color list in this field.
   - The **Fixed Single**. You should select a color, which can be applied for all geographic objects.
   - The **Group**. The Color Group field will be displayed in the editor. You should specify data fields with colors. Geographic objects will be grouped by value and each group of objects will have a specific color assigned.

**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Sets <em>interaction</em> of the Online Map element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Margin</td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td>Padding</td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
</tbody>
</table>
| 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Move</strong> option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Resize</strong> option enables or disables resizing of an element.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Allow Select</td>
<td>Enables or disables the element selection.</td>
</tr>
<tr>
<td>Locked</td>
<td>Locks or unlocks resizing and movement of the current element. If the property is set to <strong>True</strong>, the current element cannot be moved or resized. If this property is set to <strong>False</strong>, then this element can be moved and resized.</td>
</tr>
<tr>
<td>Linked</td>
<td>Binds the current location to the dashboard or another element. If the property is set to <strong>True</strong>, then the current item is bound to the current location. If this property is set to <strong>False</strong>, then this element is not tied to the current location.</td>
</tr>
</tbody>
</table>

### 4.11 Filtering data

One of the main principles of creating and using dashboards is the principle of the interaction of all elements for analysis and displaying data related between them. Thus, all data sources of the dashboard elements form virtual data tables for the current dashboard. This is necessary for the interaction of the dashboard elements with each other.

Data filtering is a selection of values from data sources by a specific condition. As a rule, the condition for selecting data is any value of a certain element in the dashboard panel.
Data filtering using the dashboard panel can be:

- **Prior** - filtering settings are defined in the report designer using the Data Transformation, Filters, Top N and tools.
- **Interactive** - filtering settings are performed in the viewer through the interaction of the dashboard elements, the choice of the value of one element affects the values of other elements. For example, if in the viewer a certain segment is clicked on the map, the data from the virtual data table will be compared with the value of this segment, and filtered for other elements of this dashboard.
- **Using data filtering elements**: List Box, Combo Box, Tree View, Tree View Box, Date Picker.
4.11.1 Relationship of Elements

Interaction means filtering data in the viewer of an analysis element on the dashboard panel, depending on the selected value of another analysis element on this panel. For example, depending on the selected segment on the map, the gauge will display the population size, and the progress will be the population growth rate.

In order for filtering through interaction to occur, the following conditions must be met:
- Data items on the dashboard should be related to each other;
- Items on the dashboard panel should belong to the same group.
All elements of data analysis depend on the values of other elements within their group. However, not all elements can be interactive.

The elements that can affect the values of other elements of the dashboard panel include:

- Table;
- Some types of charts;
- Region Map.

Every element (a chart and regional map) on the dashboard that can filter data have data filtering control buttons. These buttons are displayed when you hover over the element of the dashboard:

1. The button is used to enable and disable the filtering mode by several segments. If this button is enabled, then for filtering data, you should select several segments on one dashboard element. If this button is disabled, then when selecting the next segment, the previous filter will be reset.

For example, when filtering by map, by clicking on each segment in the single mode, other elements of the dashboard will display the associated data only with the current map segment. In the multi-segment filtering mode, other elements of the dashboard will display the associated data with all selected map segments.

2. The button is used to remove all filters. When you click it, all filters of the current
item in the dashboard will be deleted.

4.11.2 Data Transformation

All data that is used in any dashboard is a data column in the virtual table of the dashboard panel. For example, if three data fields are specified in a chart, the chart uses three columns from the virtual data table of the dashboard.

This chapter will cover the following:

➤ Data Transformation Editor;
➤ Sorting Data;
» Skip and Limit Rows;
» Running total;
» Show percentage;
» Replace value;
» Filtering by type of values;
» Custom filter;
» Selecting values.

**Information**

There are always two additional columns of data: Max and Min in the data transformation of the Gauge element.

Filtering using the **Data Transformation** tool is:
» Prior and customizable in the report designer.
» Resetting filter settings are also carried out in the report designer.
» The already filtered data for the current element of the dashboard is displayed in the viewer.

To configure **Data Transformation** you should:
» Select the dashboard element;
» Click the Browse button of the Data Transformation property on the property panel.

**Information**

Data transformation is configured only for a specific element of the dashboard. All data transformation settings are applied only to the current element.

**Data Transformation editor**

Every column in the data transformation consists of:
» Header;
» List of values.
All data transformation settings are located in the command menu. To call this menu, left-click on the header of the data column.

**Information**

Depending on the type of values (numeric, string, Boolean, etc.), the list of commands and actions for the values of the data column may differ.

Consider the commands that can be applied to the values of the data column.

**Data sorting**

By data sorting we mean the order of the element values in a certain direction.
In the Data Transformation element, the values can be:

- Sorted in ascending order. In the case of string values, the sorting is performed From A to Z, and for numeric values From Smallest to Largest;
- Sorted in descending order. In the case of string values, the sorting is From Z to A, and for numeric values From Largest to Smallest;
- Without sorting, values are transferred to the report in the order that they are contained in the data storage.

**Skip and Limit Rows**
One of the ways to filter data when converting data is to skip and set the limit rows in the data table element. For example, defining a range from 3 to 7 lines, or only the first three lines, or only the first four lines, starting from the 3rd line.
To skip lines and (or) set their limit, you should:

- Click on the title of the data column in the **Data Transformation** editor of the element;
- In the menu of the **Actions** item, select the **Skip and Limit Rows** command;
- In the dialog that opens, specify the number of lines to skip. The default value is 0, no rows in the table are skipped.
- Select a predefined number of rows or enter an integer that will be the number of rows in the data table of the element. By default, all rows are selected.

**Running total**

When designing a report, it is often necessary to calculate the cumulative total. The cumulative total is the calculation of a new value, as a result of adding the current string value to the sum of the previous values. You can enable the function to calculate the cumulative total for the item data field in the Data Transformation of the element.
To enable the calculation of the cumulative total for a column, you should:

- Click on the title of the Data Transformation editor;
- Select the **Running total** command in the **Actions** menu.
- Set the initial value. The default value is 0, the cumulative total is calculated only from the data column values. However, if necessary, you can set the initial value. Then the specified value will be added to the first value.

To disable the calculation of the cumulative total you should:

- Click on the title of the element in the Data Transformation editor;
- Select the **Remove Actions** command in the **Actions** menu.
- Delete the value in the cumulative total dialog and click **OK**.

**Displaying percent**

When designing reports, the situations may occur when it is necessary to output a specific weight (percentage) of a value from the list of values of a data column. For example, when analyzing sales, to identify the most profitable regions, it is necessary to calculate the percentage of sales in a particular region in relation to sales in all regions.
To display the percentage of the value from all the values in the data column, you should:

› Click on the title of the element in the Data Transformation editor;
› Select the **Show Percentage** command from the **Actions** menu.

**Replacing values**
You can replace some value with another one or add text to the current value in the Data Transformation.
To replace the value in the Data Transformation editor:

- Click on the title of the data column;
- Select the **Replace Values** command in the **Actions** menu;
- In the editor that opens, you should specify the value to be replaced and the value to be replaced. Also, you can configure the replacement of several values at once.

**Filtering by type of values**

By filtering data we mean the selection of data by any condition. For example, the statistics of visits for the last day, or sales in a certain category, etc.
Data filtering in data conversion can be done the following way:

- Click on the title of the data column;
- Go to the filter (the name depends on the type of the element, i.e. for numeric elements - Number Filter, for string elements - String Filter, etc.);
- In the define logical operation sub-menu.
- Then, an editor will be opened in which you need to specify a value for the logical operation. When this filter is triggered, fulfilling a certain logical condition, the values will be displayed.

**Custom filter**
A custom filter can be applied to any data column of an element.
To add a custom filter, you should:
- Click on the title of the data column;
- Select the **Custom** filter command in the drop-down menu.
- The filter editor will be called. You should add filters, define a logical operation and value. When this filter is triggered, the values will be displayed.

**Selecting values**
Also, you can filter the data simply by selecting values.
Click on the title of the data column in the Data Transformation editor.
In the drop-down menu, check the values that you want to leave, remove flags from values that are not needed.

**Information**

You should know that more than one command can be applied to a single data column. For example, sorting of values, restriction of lines and filtering by the type of values.

Also you should know that if a dashboard element has more than one data column, different commands may be applied to each of them. In this case, the commands of one data column affect the other. For example, if the first data column contains a filter by product category, and the second one has prices for these products, then the first column filter will be applied to the values of these columns, and then the second one will be applied.
4.11.3 Filters

All data that is used in any element of the dashboard is a data column in the virtual table of the dashboard panel. For example, if three data fields are specified in a chart, the chart uses three columns from the virtual data table of the dashboard. Unlike the Data Transformation tool, the Filters tool is used to filter data of an element not only by the used fields, but also by other data fields it is related to.

This chapter will cover the following:
- **Filter editor**;
- **Example of setting the filters of an element**.
- **Table of filter operations**.

**Information**

You may configure filters only for a specific element of the dashboard and apply them only to it. The data of the remaining elements of the current dashboard panel is not
Filtered.

Filtering using the **Filters** tool is:
- Prior and customizable in report designer;
- Reset filter settings are also carried out in the report designer;
- In the viewer, the already filtered data for the current element of the dashboard is displayed.

To set up Filters you should:
- Select the element on the dashboard panel;
- Click the **Filters** button.

And specify the settings for filtering data in the editor.
**Filter editor**
The editor is setting up data filters. Every filter is a data field, a logical operation, and a data filtering value. All added filters work through the logical "AND", the data will be filtered first by the first filter, then by the second, and so on. In other words, only data that matches all filter conditions will be displayed in the element. The order of applying filters is determined by their order in the Filter editor. The higher is the filter in the list, the higher is its order of application.

1. This field indicates any related data fields.
2. This parameter is used to determine the **logical operation** in the data filtering condition.
3. This field indicates the value of the filter condition.
4. This parameter is used to use the expression as the value of the filter condition. If this is enabled, an expression is specified in the Value field. The result of the calculation of this expression will be the value of the filter condition.

**An example of setting the filters of an element**
Suppose there are three elements on the dashboard:
- **Progress** - displays the number of orders in relation to the quantity of goods in stock;
- **Indicator** - displays the total value of goods in stock;
- **Chart** - displays the quantity in stock and the number of orders for each product.
Set up filtering of data in a chart. Display products only from a certain category, the price of which is in the required range.

**Step 1:** Select the **Chart** element in the report designer;

**Step 2:** Click the **Filters** button to open the Filter editor;

**Step 3:** Add a data field with a list of product categories;

**Step 4:** Define a **logical filter operation**. In this case, select the **equal to** operation.
Step 5: Select or enter the value of the filter condition. In this example, the category **Condiments** will be selected.

Products related to the category **Condiments** will be displayed. The relationship of categories and products is defined in the data dictionary. Now add a second filter. We display products from the category **Condiments**, which prices are in a certain range.
Step 6: Add a data field with product prices to the Filter editor;

Step 7: Select this data field and select a logical operation between;

Step 8: Select or set the price range values.
Now, the product list in the chart will first be filtered by the category **Condiments**. After that, products will be filtered by prices and displayed only those which prices are in range within the specified prices.

Note that data filtering using filters:
- It is performed on data fields that are not used in the **Chart** element;
- It is applied only to the element of the dashboard where filters are set, in our example, only to the chart.
# Table of Operations

The list of available operations depends on the data type. Below is a list of operations for each data type and their description. The operation is performed on the value from the data field and the filter value (the value or expression that is specified in the filter).

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type is String</th>
<th>Data Type is Number</th>
<th>Data Type is Date</th>
<th>Data Type is Boolean</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equal to</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>If the data field value is equal to the filter value, then the condition is true.</td>
</tr>
<tr>
<td>not equal to</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>If the data field value is not equal to the filter value, then the condition is true.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>between</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If the data field value is in the specific</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If the data field value is not in the specific range of filter values, then the condition is true.

<table>
<thead>
<tr>
<th>not between</th>
<th>+</th>
<th>+</th>
<th>+</th>
</tr>
</thead>
</table>

range of filter values, then the condition is true.
<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>greater than</td>
<td>If the data field value is greater than the filter value, then the condition is true.</td>
</tr>
<tr>
<td>greater than or equal to</td>
<td>If the data field value is greater than or equal to the filter value, then the condition is true.</td>
</tr>
<tr>
<td>Condition</td>
<td>Symbol</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>less than</td>
<td>+</td>
</tr>
</tbody>
</table>

If the data field value is greater than the filter value of equal to the filter value, then the condition is true.
<table>
<thead>
<tr>
<th>Condition</th>
<th>Action</th>
<th>Action</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than or equal to</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

If the data field value is less than the filter value of equal to
<p>| containing | + | If the data field value contains the filter value, then the condition is true. |</p>
<table>
<thead>
<tr>
<th>Condition</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>not containing</td>
<td>+</td>
<td>If the data field value does not contain the filter value, then the condition is true.</td>
</tr>
<tr>
<td>beginning with</td>
<td>+</td>
<td>If the data field value starts with</td>
</tr>
<tr>
<td>the filter value, then the condition is true.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ending with</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If the data field value ends with the filter value, then the condition is true.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>is blank</td>
<td>+</td>
<td>If the data field value is blank, then the condition is true.</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>is not blank</td>
<td>+</td>
<td>If the data field value is not blank, then the condition is true.</td>
</tr>
<tr>
<td>is null</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>----------------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>in not null</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
4.11.4 Top N

One of the options for filtering data for the elements of the dashboard panel is the ability to display a certain number of maximum or minimum values. This can be applied with the Top N tool. This feature applies to pre-filtering and only to the current element.
This chapter will cover the following:

- **Top N Editor**;
- **An example of Top values**;
- **An example of minimum Top values**.

## Information

The top values can be setup only for a specific element of the dashboard and are applied only to it. The data of the remaining elements of the current dashboard is not...
Filtered using the Top N tool is:
- Prior to filtering and you customize it in the report designer.
- Reset filter settings are also carried out in the report designer.
- In the viewer, the already filtered data for the current element of the dashboard is displayed.

To customize the Top values you should:
- Select the element on the dashboard;
- Click the Browse button for the TopN property on the property panel.

You can specify the top values for the elements of the dashboard panel:

- Chart;
- Indicator;
- Progress;
- Pivot.

**Information**

The Top values for the Pivot element are configured in the editor of this element.

**Top N editor**

In the Top N editor you may define the type of the values (maximum or minimum), the number of the best values, actions with the rest of the element data.
1. The **Mode** parameter allows you to define the type of values that you want to display:
   - **None** - all values of the current item are displayed. This mode is set by default.
   - **Top** - a list of maximum values will be displayed. The first value is the maximum value from the list of values. Depending on the number of values, the values in the direction from the maximum to the minimum will be sequentially displayed.
   - **Bottom** - a list of minimum values will be displayed. The first value is the minimum value from the list of values. Depending on the number of values, the values in the direction from the minimum to the maximum will be sequentially displayed.

2. The **Count** parameter is used to determine the number of maximum or minimum values. For example, if this parameter is set to 10, then 10 maximum or minimum values from the list of values will be displayed.

**Information**

When setting the top values for the **Pivot** element, you should also define the **Measure** parameter. The value for this parameter will be one of the data fields specified in the **Summary** field.

3. The **Show Other** option is used to display a sum of values that were not included in the list of top values:
   - If the **Show other** option is enabled, then all values that were not included in the list of top values will be summed up and displayed as a separate value.
   - If the **Show other** option is disabled, then only values that appear on the list of top values will be displayed on the item.

4. The **Other Text** parameter is used to specify a title for the sum of other values. This parameter is applicable only if the **Show other** option is enabled. If the **Text** parameter of other values is not filled, the default value **Other** is used for the sum of other values.

**An example of Top values**

For example, we have a table and a chart that display the sales volume for every product.
Let's show three products with maximum sales on the chart:

**Step 1**: Select a chart in the dashboard panel in the report designer;

**Step 2**: Click the **Browse** button on the **Top N** property on the property panel;

**Step 3**: Select the **Top** mode in the **Top N** editor;

**Step 4**: Set the number to 3;
Step 5: Uncheck the box next to Show others.

As you can see in the picture, three products with maximum sales will be displayed in the chart. In this case, this filtering does not affect the lists of values of other elements.

Step 6: Go back to the report designer;

Step 7: Click the Browse button on the Top N property on the property panel;

Step 8: Check the box next to Show other.
Step 9: Define text for general value. For example, Another Products.

Now, the chart will display three products with maximum sales. All other values will be summed up and displayed on the chart as a separate graphic element, with the Another Products argument.

An example of minimum Top values
For example, in the dashboard panel, a table and a chart are displayed with the sales
volume for every product.

<table>
<thead>
<tr>
<th>ProductsName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baconator</td>
</tr>
<tr>
<td>Big Mac</td>
</tr>
<tr>
<td>French Fries</td>
</tr>
<tr>
<td>Frosty</td>
</tr>
<tr>
<td>Original Chicken Sandwich</td>
</tr>
<tr>
<td>Snack Wrap</td>
</tr>
<tr>
<td>Spicy Chicken Sandwich</td>
</tr>
<tr>
<td>Ultimate Bacon Cheeseburger</td>
</tr>
<tr>
<td>Whopper</td>
</tr>
</tbody>
</table>

Let's show three products on the chart with minimal sales volumes:

**Step 1**: Select a chart in the dashboard panel in the report designer;

**Step 2**: Click the **Browse** button on the **Top N** property on the property panel;

**Step 3**: Select the **Bottom** mode in the **Top N** editor;
Step 4: Set the number to 3;

Step 5: Uncheck the box next to Show other.

As can be seen in the picture, three products with minimal sales volumes will be displayed on the chart. In this case, this filtering does not affect the lists of values of other elements.

Step 6: Return to the report designer;
**Step 7:** Click the **Browse** button on the **Top N** property on the property panel;

**Step 8:** Check the box next to **Show other**.

**Step 9:** Define text for a general value. For example, **Another Products**.

Now, the chart will display three products with minimal sales. All other values will be summed up and displayed on the chart as a separate graphic element, with the Another Products argument.
4.11.5 List Box

List Box is a filtering element on the dashboard, which is used to filter data for analysis elements in the viewer, depending on the selected value. It can be located anywhere on the dashboard panel. Depending on the size of the dashboard in the viewer, it can grow or shrink in height and width.

This chapter will cover the following:

- List Box editor;
The List Box element can be subordinate to other filtering elements, or can be the main filtering element for them. The List Box can work in two modes:

- **One.** In the viewer you can select only one value of the List Box. Accordingly, data filtering for the elements of the dashboard will be executed only by one value.
- **Multi.** In the viewer, you can select several values of the List Box. Accordingly, data filtering for the elements of the dashboard will be executed by all selected values.

The item is set up in its editor. To call the editor, you should to the following in the report designer:

- Double-click the List Box;
- Select the List Box and select **Design** in the context menu;
- Select the List Box, click the **Browse** button on the **Key** and **Value** properties on the property panel.

**List Box editor**

In the editor of the List Box element, you may add elements with data, set up the mode for selecting values, select the main element of filtering.

![List Box editor](image-url)
1 The **Key** field. The data element is specified there by the values of which the data will be filtered.

2 The **Name** field. Indicates the data item which values will be displayed in the List Box element. If the name is not specified, then the names of keys will be displayed in the list item.

3 The **Field** field. Displays the expression of the selected item data field.

4 The **Selection Mode** parameter. Specifies the number of simultaneously selected values of the **List Box** item — **One** or **Multi**. If one value is selected, the data will be filtered by the current value of the List Box element. If the Multi mode is set, the filtering will be performed for all selected values.

5 The **Show (All) Value** option. Enables the option to select all values in the List Box element. If this option is enabled, then the **Select (All) Value** value will be present in the List Box element.

6 The **Parent Element** parameter. It is used to define the main filtering element for the current List Box element. The data of these filter elements will be interrelated, and depending on the selected value of the main element, the list of values of the current element will be filtered.

**The One mode**

Consider an example where you can select only one value in the List Box element. For example, the dashboard displays data by products and their categories. Using the List Box item, the categories will be switched, data for the remaining elements of the dashboard panel (depending on the selected category) will be filtered.

**Step 1:** Add the elements of the dashboard;

**Step 2:** Add a List Box element;

**Step 3:** Specify the data element with the keys and names in the editor of the List Box element. Also, the selection mode is set to **One**.

**Step 4:** Close the editor and preview the report.

**Step 5:** Left click on the category name to display data on this category.
Since the selection mode of the List Box item is set to One, it is not possible to display data simultaneously across several categories. However, you can display data on all values of the List Box element. To do this, enable it to Multi.

**Step 6:** Go back to the report designer and call the element editor;

**Step 7:** Select Show (All) Values;

**Step 8:** Close the editor and item and go to preview.
Now you can view the data for every category or enable displaying data for all values of the List Box element.

**The Multi mode**

The Multi mode for selecting List Box is used to filter data for the elements of the dashboard by several values of the List Box element.

**Step 1:** Add the elements of the dashboard;
**Step 2**: Add List Box;

**Step 3**: Specify the data element with the keys and names in the editor of the List Box element. Also, the selection mode is set to **Multi**.

**Step 4**: Close the editor and go to preview.

**Step 5**: Check the values by which to filter data for the elements of the dashboard. In this case, check the boxes for several product categories.
The data for the elements of the dashboard will be filtered depending on the selected values of the List Box element. Accordingly, you should uncheck the box for values that do not need to display data. In the Multi mode, you can also disable the All value. In this case, to check (uncheck) the check box for all values of the List Box element, it is enough to check (uncheck) the check box only for the All value. To enable the value All, you should:

**Step 6:** Go back to the report designer and call the element editor;

**Step 7:** Enable **Show (All) values**;

**Step 8:** Close the element editor and go to preview.
Dependent lists
Under dependent lists we mean the dependence of the values of one List Box element on the selected value of another filter element, for example, another List Box element. For example, if the dashboard displays data by categories and products, then using the List Box elements, data can be filtered first by categories, and then by products. In this case, the list of products will depend on the selected category. To create dependent lists you should:

**Step 1**: Add dashboard elements;
Step 2: Add a List Box with the list of categories;

Step 3: Add a List Box with the list of products;

Step 4: From the drop-down menu, select the List Box element with the list of products, in the Parameter Element field, select the List Box with the list of categories;

Step 5: Close the editor and preview the report.

Step 6: Select a category or several categories. The values of the List Box item with the list of products will be filtered depending on the selected categories. By default, the data for all products included in the selected categories will be displayed.

Step 7: Check the products for which you want to display data.
You should know that the dependence of the filter elements can be organized between the filter elements of different types. The number of levels of subordination of elements is unlimited.

### Information

You should know that the dependence of the filter elements can be organized between the filter elements of different types. The number of levels of subordination of elements is unlimited.

### 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>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.</td>
</tr>
<tr>
<td>Style</td>
<td>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.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to True, the current item is enabled and will</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Margin</th>
<th>A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Padding</td>
<td>A group of properties that allows you to define padding (left, top, right, bottom) of values from the range of values.</td>
</tr>
<tr>
<td>Text Format</td>
<td>Sets the formatting of values for the element.</td>
</tr>
<tr>
<td>Title</td>
<td>A group of properties that allows you to customize the title of the element:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Back Color</strong> property provides the ability to change the background color of the title of the current item. By default, this property is set to <strong>From Style</strong>, i.e. the background color will be obtained from the style settings of the current element.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Fore Color</strong> allows you to change the text color of the title of the current item. By default, this property is set to <strong>From Style</strong>, i.e. the text color of the title will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td></td>
<td>- The group property <strong>Font</strong> that allows you to define the font family, its style and size for the title of the current element.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>‣ The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>‣ The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>‣ The <strong>Allow Move</strong> option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>‣ The <strong>Allow Resize</strong> option enables or disables resizing of an element.</td>
</tr>
<tr>
<td></td>
<td>‣ The <strong>Allow Select</strong> option enables or disables the element selection.</td>
</tr>
</tbody>
</table>
Locked

Locks or unlocks resizing and replacement 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.

4.11.6 Combo Box

**Combo Box** is an element of filtering on the dashboard, which is used to filter data for analysis, depending on the selected value. This element is an analogue of the Combo Box element, with the only difference that all values are contained in the drop-down menu. It can be located anywhere on the dashboard. Depending on the size of the dashboard panel in the viewer, it can grow or shrink by width only.
This chapter will cover the following:

- Combo Box editor;
- The One mode;
- The Multi mode;
- Dependent lists;
- Table Of Properties.

The Combo Box element may be subordinate to other filtering elements or be the main filtering element for them. The Combo Box item can work in two selection modes:

- One. In the viewer you can select only one value of the Combo Box. Accordingly, data filtering for the elements of the dashboard will be executed only by one value.
- Multi. In the viewer you can select several values of the Combo Box. Accordingly, data filtering for the elements of the dashboard will be executed by all selected values.

The item is set up in its editor. To call the editor, you should do the following in the report designer:

- Double-click the Combo Box;
- Select the Combo Box and select Design in the context menu;
- Select the Combo Box, click the Browse button on the Key and Value properties on the property panel.
**Combo Box editor**

In the editor of the Combo Box element, you may add elements with data, set up the mode for selecting values, select the main element of filtering.

1. The **Key** field. The data element is specified there by the values of which the data will be filtered.
2. The **Name** field. Indicates the data item which values will be displayed in the Combo Box element. If the name is not specified, then the names of keys will be displayed in the list item.
3. The **Field** field. Displays the expression of the selected item data field.
4. The **Selection Mode** parameter. Specifies the number of simultaneously selected values of a Combo Box item — **One** or **Multi**. If one value is selected, the data will be filtered by the current value of the Combo Box element. If the Multi mode is set, the filtering will be performed for all selected values.
5. The **Show (All) Value** option. Enables the option to select all values in the Combo Box element. If this option is enabled, then the **Select (All) Value** value will be present in the Combo Box element.
6. The **Parent Element** parameter. It is used to define the main filtering element for the current Combo Box element. The data of these filter elements will be interrelated, and depending on the selected value of the main element, the list of values of the current element will be filtered.
The One mode
Consider an example where you can select only one value in the Combo Box element. For example, the dashboard displays data on products and their categories. Using the Combo Box item, the categories will be switched, data for the remaining elements of the dashboard panel (depending on the selected category) will be filtered.

Step 1: Add the elements of the dashboard;

Step 2: Add a Combo Box element;

Step 3: Specify the data element with the keys and names in the editor of the Combo Box element. Also, the selection mode is set to One.

Step 4: Close the editor and preview the report.

Step 5: Left click on the category name to display data on this category.
Since the selection mode of the Combo Box item is set to One, it is not possible to display data simultaneously across several categories. However, you can display data on all values of the Combo Box element. To do this, enable it to **Multi**.

**Step 6:** Go back to the report designer and call the element editor;

**Step 7:** Select **Show (All) Values**;

**Step 8:** Close the editor and item and go to preview.
Now you can view the data for every category or enable displaying data for all values of the Combo Box element.

**The Multi mode**

The **Multi** mode for selecting Combo Box is used to filter data for the elements of the dashboard by several values of the Combo Box element.

**Step 1:** Add the elements of the dashboard;
Step 2: Add Combo Box;

Step 3: Specify the data element with the keys and names in the editor of the Combo Box element. Also, the selection mode is set to Multi.

Step 4: Close the editor and go to preview.

Step 5: Check the values by which to filter data for the elements of the dashboard. In this case, check the boxes for several product categories.
The data for the elements of the dashboard panel will be filtered depending on the selected values of the Combo Box element. Accordingly, you should uncheck the box for values that do not need to display data. In the Multi mode, you can also disable the All value. In this case, to check (uncheck) the check box for all values of the Combo Box element, it is enough to check (uncheck) the check box only for the All value. To enable the value All, you should:

**Step 6**: Go back to the report designer and call the element editor;

**Step 7**: Enable *Show (All) values*;

**Step 8**: Close the element editor and go to preview.
Dependent lists
Under dependent lists we mean the dependence of the values of one Combo Box element on the selected value of another filter element, for example, another Combo Box element. For example, if the dashboard displays data by categories and products, then using the Combo Box elements, data can be filtered first by categories, and then by products. In this case, the list of products will depend on the selected category. To create dependent lists you should:

Step 1: Add dashboard elements;
Step 2: Add a Combo Box with the list of categories;

Step 3: Add a Combo Box with the list of products;

Step 4: From the drop-down menu, select the Combo Box element with the list of products, in the Parameter Element field, select the Combo Box with the list of categories;

Step 5: Close the editor and preview the report.

Step 6: Select a category or several categories. The values of the Combo Box item with the list of products will be filtered depending on the selected categories. By default, the data for all products included in the selected categories will be displayed.

Step 7: Check the products for which you want to display data.
You should know that the dependence of the filter elements can be organized between the filter elements of different types. The number of levels of subordination of elements is unlimited.

**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes <a href="#">the data transformation</a> of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific <a href="#">group of items</a>.</td>
</tr>
<tr>
<td>Back Color</td>
<td>Changes the background color of the element. By default, this property is set to <a href="#">From Style</a>, i.e. the color of the element will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>Specifies the color of the values of the element. By default, this property is set to <a href="#">From Style</a>, i.e. the color of the values will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Style</td>
<td>Selects a style for the current element. The default it is set to <a href="#">Auto</a>, i.e. the style of this element is inherited from the style of the dashboard.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to <a href="#">True</a>, the current item is enabled and will</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Margin</th>
<th>A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Padding</td>
<td>A group of properties that allows you to define padding (left, top, right, bottom) of values from the range of values.</td>
</tr>
<tr>
<td>Text Format</td>
<td>Sets the formatting of values for the element.</td>
</tr>
<tr>
<td>Name</td>
<td>Changes the name of the current element.</td>
</tr>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>– The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>– The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>– The <strong>Allow Move</strong> option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>– The <strong>Allow Resize</strong> option enables or disables resizing of an element.</td>
</tr>
<tr>
<td></td>
<td>– The <strong>Allow Select</strong> option enables or disables the element.</td>
</tr>
</tbody>
</table>
Locked

Locks or unlocks resizing and replacement 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.

### 4.11.7 Tree View

**Tree View** is a filtering element on the dashboard, which is used to create a hierarchy of values and filter data for analysis in the viewer by these values. It can be located anywhere on the dashboard panel. Depending on the size of the dashboard in the viewer, it can grow and shrink by height and width.
This chapter will cover the following:

- **Tree View editor**;
- **The One mode**;
- **The Multi mode**;
- **Dependent lists**;
- **Table Of Properties**.

The **Tree View** can be subordinate to other filtering elements, or be the main filtering element for them. The **Tree View** can work in two selection modes:

- **One**. In the viewer, you can select only one value within one level of the hierarchy of
values. Accordingly, data filtering for the elements of the dashboard will be performed only by one value.

Multi. In the viewer, you can select multiple values within the same level of the hierarchy of values. Accordingly, data filtering for the elements of the dashboard will be performed by all selected values.

You may setup the Tree View element in the editor. To call the editor, you should do the following in the report designer:
- Double-click the Tree View element;
- Select the Tree View element and select the Design command in the context menu;
- Select the Tree View element, and on the property panel, click the Browse button of the Keys property.

The Tree View editor
In the Tree View editor, you can add items with data, set up the value selection mode, select the main filter item.

1. The Key field. The data item is specified, the values of which will form the hierarchy and be displayed in the Tree View item.
2. The Field field. Displays the expression of the selected item data field.
3. The Selection Mode parameter. Specifies the number of simultaneously selected values of the element. A Tree View is one element or many elements.
4. The **Show (All) Value** option. Enables the option to select all values in the **Tree View** element. If this option is enabled, the **Select (All) Value** value will be present in the Tree View element.

5. The **Parent Element** parameter. It is used to define the main filtering element for the current element. The data of these filter elements will be interrelated, and depending on the selected value of the main element, the list of values of the current element will be filtered.

### The One mode
Consider an example, when you can select only one value in the Tree View element. For example, the dashboard displays data by products and their categories. With the help of the Tree View, switching will be carried out by categories and by products included into this category, data will be filtered for the remaining elements of the dashboard, depending on the selected category and product included in this category.

**Step 1:** Add the elements of the dashboard panel;

**Step 2:** Add the Tree View element;

**Step 3:** In the editor of the Tree View element, specify the data item with categories and products.

#### Information
The hierarchy of data values depends on the location of the data elements in the editor. The values of the data field located in the editor above the rest will be initial values in the Tree View.

The values of the lower data field in the editor will be nested with respect to the values of each upper data field.

**Step 4:** Set the Selection Mode to **One**.

**Step 5:** Close the editor and preview the report.

**Step 6:** Left-click on the category name to display data in this category.
**Step 7:** Expand this category by clicking the arrow control and select a product from this category to display data for this product.

Since the selection of Tree View is set to One, it is not possible to display data for several categories and products simultaneously. However, you can display data on all values of the Tree View element. To do this, enable the **Show (All) Value** option.

**Step 8:** Go back to the report designer and call the Tree View editor;
Step 9: Check the box next to Show (All) Values;

Step 10: Close the editor and preview the report.

Now you can view the data for every category and every product, or enable displaying data for all values of the Tree View element by all categories and products included into these categories.

The Multi mode
This mode is used to filter data for dashboard elements by several values of this item.

**Step 1:** Add the elements of the dashboard;

**Step 2:** Add the Tree View item;

**Step 3:** In the editor of the Tree View item, specify the data item with categories and products.

**Information**

The hierarchy of data values depends on the location of the data elements in the editor. The values of the data element located in the editor above the rest will be initial values in the Tree View.

The values of the underlying data element in the editor will be nested with respect to the values of each upper data element.

**Step 4:** Set the Selection Mode to **Multi**.

**Step 5:** Close the editor and go to preview.

**Step 6:** Check the values by which the data for the dashboard items should be filtered. In this case, check the boxes for several categories.

**Step 7:** Expand these categories by clicking on the arrow control and select the products for which you want to display data.
The data for the elements of the dashboard will be filtered depending on the selected values of the Tree View element. Accordingly, you should uncheck the box for values that do not need to display data. In the Multi mode, you can also disable the All value. In this case, to check (uncheck) the check box of all values of the Tree View element, it will be enough to check (uncheck) the check box only for All value. To enable the All value, you should:

Step 6: Go back to the report designer and call the Tree View editor;

Step 7: Enable Show (All) Values;
Step 8: Close the editor and go preview.

Information

To check or uncheck the check boxes for subordinate values, it is enough to set or clear the check box of the main value. For example, to remove the check from the products included into any category, it is enough to remove the check from this category.
**Dependent lists**
Under dependent lists we mean the dependence of the values of one Tree View element on the selected value of another filter element, for example, another Tree View element.

For example, if the dashboard displays data by categories and products, then you can filter data using the Tree View elements:
- The first Tree View will contain, for example, data of suppliers by countries and cities;
- The second one will contain a list of categories and products from these suppliers.

To create a dependent Tree View you should:

**Step 1:** Add dashboard elements;

**Step 2:** Add a Tree View item with a list suppliers' countries and cities;

**Step 3:** Add a Tree View item with a list of categories and products;

**Step 4:** From the drop-down menu, select the first Tree View element with the list of products, in the Parameter Element field, select the Tree View with the list of categories;

**Step 5:** Close the editor and go to preview.

**Step 6:** Select suppliers' countries, expand them and select suppliers' cities. Values of the second Tree View element with a list of categories and products will be filtered depending on the selected countries and cities.

**Step 7:** Check the boxes for categories and products by which you want to display data.
You should know that the dependence of the filter elements can be organized between the filter elements of different types. The number of levels of subordination of elements is unlimited.

**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>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.</td>
</tr>
<tr>
<td>Style</td>
<td>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.</td>
</tr>
</tbody>
</table>
| Enabled               | Enables or disables the current item on the dashboard. If the property is set to True, the current item is enabled and will
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
<td></td>
</tr>
<tr>
<td>Margin</td>
<td>A group of properties that allows you to define margins (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td>Padding</td>
<td>A group of properties that allows you to define padding (left, top, right, bottom) of values from the range of values.</td>
</tr>
<tr>
<td>Text Format</td>
<td>Sets the formatting of values for the element.</td>
</tr>
</tbody>
</table>
| Title | A group of properties that allows you to customize the title of the Table element:  
  - **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. |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal Alignment</td>
<td>The property provides the ability to change the title alignment relative to the element - Left, Center, Right.</td>
</tr>
<tr>
<td>Text</td>
<td>The property is used to set the title text of the current element.</td>
</tr>
<tr>
<td>Visible</td>
<td>The 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.</td>
</tr>
<tr>
<td>Name</td>
<td>Changes the name of the current element.</td>
</tr>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>The Allow Change option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>The Allow Delete option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>The Allow Move option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>The Allow Resize option enables or disables resizing of an element.</td>
</tr>
<tr>
<td></td>
<td>The Allow Select option enables or disables the element selection.</td>
</tr>
<tr>
<td>Locked</td>
<td>Locks or unlocks resizing and replacement of the current element. If the property is set to <strong>True</strong>, the current element cannot be moved or resized. If this property is set to <strong>False</strong>, then this element can be moved and resized.</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Linked</td>
<td>Binds the current location to the dashboard or another element. If the property is set to <strong>True</strong>, then the current item is bound to the current location. If this property is set to <strong>False</strong>, then this element is not tied to the current location.</td>
</tr>
</tbody>
</table>

### 4.11.8 Tree View Box

**Tree View Box** is a filtering element on the dashboard, which is used to create a hierarchy of values and filter data for analysis in the viewer by these values. It can be located anywhere on the dashboard panel. Depending on the size of the dashboard in the viewer, it can grow or shrink by width only.
This chapter will cover the following:

- **Tree View Box editor**;
- **The One mode**;
- **The Multi mode**;
- **Dependent lists**;
- **Table Of Properties**.

The **Tree View Box** can be subordinate to other filtering elements, or be the main filtering element for them. The **Tree View Box** can work in two selection modes:

- **One**. In the viewer, you can select only one value within one level of the hierarchy of
values. Accordingly, data filtering for the elements of the dashboard will be performed only by one value.

- Multi. In the viewer, you can select multiple values within the same level of the hierarchy of values. Accordingly, data filtering for the elements of the dashboard will be performed by all selected values.

You may setup the **Tree View Box** element in the editor. To call the editor, you should do the following in the report designer:
- Double-click the **Tree View Box** element;
- Select the **Tree View Box** element and select the **Design** command in the context menu;
- Select the **Tree View Box** element, and on the property panel, click the **Browse** button of the **Keys** property.

**The Tree View Box editor**
In the **Tree View Box** editor, you can add items with data, set up the value selection mode, select the main filter item.

1. The **Key** field. The data item is specified, the values of which will form the hierarchy and be displayed in the **Tree View Box** item.
2. The **Field** field. Displays the expression of the selected item data field.
3. The **Selection Mode** parameter. Specifies the number of simultaneously selected values of the element. A **Tree View Box** is one element or many elements.
The **Show (All) Value** option. Enables the option to select all values in the *Tree View Box* element. If this option is enabled, the **Select (All) Value** value will be present in the Tree View Box element.

*The Parent Element* parameter. It is used to define the main filtering element for the current element. The data of these filter elements will be interrelated, and depending on the selected value of the main element, the list of values of the current element will be filtered.

**The One mode**

Consider an example, when you can select only one value in the Tree View Box element. For example, the dashboard displays data by products and their categories. With the help of the Tree View Box, switching will be carried out by categories and by products included into this category, data will be filtered for the remaining elements of the dashboard, depending on the selected category and product included in this category.

**Step 1**: Add the elements of the dashboard panel;

**Step 2**: Add the Tree View Box element;

**Step 3**: In the editor of the Tree View Box element, specify the data item with categories and products.

**Information**

The hierarchy of data values depends on the location of the data elements in the editor. The values of the data field located in the editor above the rest will be initial values in the Tree View Box.

The values of the lower data field in the editor will be nested with respect to the values of each upper data field.

**Step 4**: Set the Selection Mode to **One**.

**Step 5**: Close the editor and preview the report.

**Step 6**: Left-click on the category name to display data in this category.
**Step 7:** Expand this category by clicking the arrow control and select a product from this category to display data for this product.

Since the selection of Tree View Box is set to One, it is not possible to display data for several categories and products simultaneously. However, you can display data on all values of the Tree View Box element. To do this, enable the Show (All) Value option.

**Step 8:** Go back to the report designer and call the Tree View Box editor;
**Step 9:** Check the box next to **Show (All) Values**;

**Step 10:** Close the editor and preview the report.

Now you can view the data for every category and every product, or enable displaying data for all values of the Tree View Box element by all categories and products included into these categories.

**The Multi mode**
This mode is used to filter data for dashboard elements by several values of this item.

**Step 1:** Add the elements of the dashboard;

**Step 2:** Add the Tree View Box item;

**Step 3:** In the editor of the Tree View Box, specify the data item with categories and products.

**Information**

The hierarchy of data values depends on the location of the data elements in the editor. The values of the data element located in the editor above the rest will be initial values in the Tree View Box.

The values of the underlying data element in the editor will be nested with respect to the values of each upper data element.

**Step 4:** Set the Selection Mode to Multi.

**Step 5:** Close the editor and go to preview.

**Step 6:** Check the values by which the data for the dashboard items should be filtered. In this case, check the boxes for several categories.

**Step 7:** Expand these categories by clicking on the arrow control and select the products for which you want to display data.
The data for the elements of the dashboard will be filtered depending on the selected values of the Tree View Box element. Accordingly, you should uncheck the box for values that do not need to display data. In the Multi mode, you can also disable the All value. In this case, in order to check (uncheck) the check box of all values of the Tree View Box element, it will be enough to check (uncheck) the check box only for All value. To enable the All value, you should:

**Step 6**: Go back to the report designer and call the Tree View Box editor;

**Step 7**: Enable **Show (All) Values**;
Step 8: Close the editor and go preview.

**Information**

To check or uncheck the check boxes for subordinate values, it is enough to set or clear the check box of the main value. For example, to remove the check from the products included into any category, it is enough to remove the check from this category.
**Dependent lists**

Under dependent lists we mean the dependence of the values of one Tree View Box element on the selected value of another filter element, for example, another Tree View Box element.

For example, if the dashboard displays data by categories and products, then you can filter data using the Tree View Box elements:
- The first Tree View Box will contain, for example, data of suppliers by countries and cities;
- The second one will contain a list of categories and products from these suppliers.

To create a dependent Tree View Box you should:

**Step 1**: Add dashboard elements;

**Step 2**: Add a Tree View Box item with a list suppliers' countries and cities;

**Step 3**: Add a Tree View Box item with a list of categories and products;

**Step 4**: From the drop-down menu, select the first Tree View Box element with the list of products, in the Parameter Element field, select the Tree View Box with the list of categories;

**Step 5**: Close the editor and go to preview.

**Step 6**: Select suppliers' countries, expand them and select suppliers' cities. Values of the second Tree View Box element with a list of categories and products will be filtered depending on the selected countries and cities.

**Step 7**: Check the boxes for categories and products by which you want to display data.
You should know that the dependence of the filter elements can be organized between the filter elements of different types. The number of levels of subordination of elements is unlimited.

**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transformation</td>
<td>Customizes the data transformation of the current item.</td>
</tr>
<tr>
<td>Group</td>
<td>Adds the current item to a specific group of items.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>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.</td>
</tr>
<tr>
<td>Style</td>
<td>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.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to True, the current item is enabled and will</td>
</tr>
<tr>
<td><strong>Reports and Dashboards</strong></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Margin</strong></td>
<td>A group of properties that allows you to define margin (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td><strong>Padding</strong></td>
<td>A group of properties that allows you to define padding (left, top, right, bottom) of values from the range of values.</td>
</tr>
<tr>
<td><strong>Text Format</strong></td>
<td>Sets the formatting of values for the element.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Changes the name of the current element.</td>
</tr>
<tr>
<td><strong>Alias</strong></td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td><strong>Restrictions</strong></td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>➢ The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>➢ The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>➢ The <strong>Allow Move</strong> option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>➢ The <strong>Allow Resize</strong> option enables or disables resizing of an element.</td>
</tr>
<tr>
<td></td>
<td>➢ The <strong>Allow Select</strong> option enables or disables the element</td>
</tr>
<tr>
<td>Locked</td>
<td>Locks or unlocks resizing and replacement 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.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Linked</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 4.11.9 Date Picker

**Date Picker** is a filtering element on the dashboard panel that is used to determine the calendar range and filter the data for the analysis in the viewer, taking into account the specified range. It can be located anywhere on the dashboard panel. Depending on the size of the dashboard panel in the viewer, it can grow or shrink by width only.
This chapter will cover the following:

- **Date Picker editor**;
- **The Single mode**;
- **The Range mode**;
- **The AutoRange mode**;
- **Table Of Properties**.

The **Date Picker** element can only be the main filtering element for other filtering elements and cannot depend on the values of other filtering elements. The **Date Picker** can have the following selection modes:
➤ **Single.** By default, the current date of the operating system and the subsequent range will be determined depending on the value of the **Condition** parameter.

➤ **Range.** By default, the current day range will be set.

➤ **Auto Range.** By default, the range will be set from an earlier date in the data source to the latest date. In other words, the original date range will correspond to the data range of the data source.

You may setup the **Date Picker** element in the editor. To call the editor, you should to the following in the report designer:

➤ Double-click on the **Date Picker** element;

➤ Select the **Date Picker** and choose the Design command in the context menu;

➤ Select the **Date Picker** item, and on the property panel, click the **Browse** button of the **Key** and **Value** properties.

**The Date Picker editor**

In this editor you can add elements with data, set up the mode for selecting values, select the main filtering element.

![Date Picker editor](image)

1. The **Key** field. The data element is specified in it, according to the values of which the data will be filtered.
2. The **Field** field. Displays the expression of the selected item data field.
3. The **Selection Mode** field. It selects the mode of the Date Picker item. The following values can be selected:
   ➤ **Single.** The current date of the operating system and the subsequent range will be determined depending on the value of the **Condition** parameter.
   ➤ **Range.** By default, the current day range will be set.
   ➤ **Auto Range.** By default, the range will be set from an earlier date in the data source to the latest. In other words, the original date range will correspond to the data range of the data source.
range of the data source.

4. The **Condition** field. Depending on the selected item mode, the following parameters may be present:
   - The **Condition** parameter is available only if the **Single** mode is selected. The value of this parameter is a logical operation that determines the continuation of the date range from the current date. For example, if **Greater then** is selected, then the default element range will include all subsequent dates from the current date of the operating system.
   - The **Initial Selection** parameter is available only if **Range** is selected. You can specify the default element range. For example, you can select the previous week. Then when you open the dashboard in the viewer, the range of the **Date Picker** item will be set to the previous week.

**The Single mode example**
The **Single** mode is used to display the time range from the selected date. By default, the initial date is the current date of the operating system. The time range is calculated depending on the logical operation of the condition. For example, you need to filter all the data before a certain date or vice versa after the current date.

**Step 1**: Create or open the dashboard;

**Step 2**: Add a **Date Picker** element;

**Step 3**: Add a **data field** to the Date Picker;

**Step 4**: Set the **Selection Mode** parameter to **Single**;

**Step 5**: Select a logical operation for calculating the time range, for example, **Less than**. In this case, the time range will include the selected date and all previous dates. By default, the current operating system date will be selected.

**Step 6**: Preview the dashboard.
Accordingly, the data that contains the values of dates up to the current date will be filtered. Also, in the viewer or in the preview panel, you can change the date in the Date Picker element, thereby shifting the data filtering time range. The logical calculation operation will remain unchanged.

**The Range sample**

The **Range** mode is used to determine the time range from one date to another. By default, the range is equal to the current date of the operating system, the start and end of the range coincides with the current date of the operating system.
**Step 1**: Create or open the dashboard;

**Step 2**: Add a **Date Picker** element;

**Step 3**: Add a **data field** to the **Date Picker**;

**Step 4**: Set the **Selection Mode** parameter to **Range**;

**Step 5**: Go to the preview.

If there is no data for the current date, the elements of the dashboard will be empty. Therefore, you can manually determine the start and end of the range.
The AutoRange sample
The Auto Range mode is used to set the time range of the Date Selection element based on the values of the specified data field. In this case, the start of the time range will correspond to the earliest date from the data field, and the end - to the latest date from the data field.

Step 1: Create or open the dashboard;

Step 2: Add a Date Picker item;
Step 3: Add a data field to the Date Picker;

Step 4: Set the Selection Mode to Auto Range;

Step 5: Go to preview.

In the viewer, you can change the start and end value of the time range in the Date Picker 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Adds the current item to a specific <a href="#">group of items</a>.</td>
</tr>
<tr>
<td>Back Color</td>
<td>Changes the background color of the element. By default, this property is set to <a href="#">From Style</a>, i.e. the color of the element will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Font</td>
<td>A group of properties defines the font family, its style, and size for the values of the element.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>Specifies the color of the values of the element. By default, this property is set to <a href="#">From Style</a>, i.e. the color of the values will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Style</td>
<td>Selects a style for the current element. The default it is set to <a href="#">Auto</a>, i.e. the style of this element is inherited from the style of the dashboard.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to <a href="#">True</a>, the current item is enabled and will be displayed when previewing</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Margin</th>
<th>A group of properties that allows you to define margins (left, top, right, bottom) of the value area from the border of this element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Padding</td>
<td>A group of properties that allows you to define padding (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
<tr>
<td>Text Format</td>
<td>Sets the formatting of values for the element.</td>
</tr>
<tr>
<td>Name</td>
<td>Changes the name of the current element.</td>
</tr>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Move</strong> option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Resize</strong> option enables or disables resizing of an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Select</strong> option enables or disables the element selection.</td>
</tr>
</tbody>
</table>
4.11.10 Variables

Variables are used to pass a value to the elements of the dashboard panel, while filtering the data of these elements.
Variables can be:
- With user-selectable value, the user selects or enters a value, and the data of the elements of the dashboard panel will be filtered based on this value;
- Without user selection, the user does not select a value, but the elements of the dashboard panel are filtered by the value of the variable.

To create an dashboard with variables, you should:
- Create a variable in the data dictionary;
- Open or create a dashboard;
- Set filters for this item using a variable.
An example of a dashboard with a variable
Suppose there is a dashboard that displays the results of the examination of patients in a clinic.

At the same time, since there are more than one patient, it is necessary, when choosing a patient, to display the results of his/her examination. In this case, it is necessary to create a variable with a list of patients, as well as provide the user to select the value of the variable. To do this:
Step 1: Go to the Data Dictionary;

Step 2: Select the New Variable command from the New Item menu or from the context menu of the dictionary;

Step 3: Specify the name, alias, description of the variable;

Step 4: Specify the data type of the variable. It should match the data type of the column by which the element data will be filtered;

Step 5: Specify the type of variable.

Information

At this moment, the dashboard panel only works with Value variables. If you need to use the selection of several values, then you can use such filter elements of the dashboard panel as the List and the Drop-down list.

Step 6: Select the method of initializing the variable as Value or Expression;

Step 7: Select the Request from User parameter if the user needs to select a value;

Step 8: Select the Allow User Values if you want to allow user input;

Step 9: Create a list of variable elements or select data columns with values;

Step 10: Specify whether the first value is obtained using the Select parameter.
Next, you need to set filters for elements that will be affected by the selected variable value. To do this:

**Step 1**: Select an item;

**Step 2**: Click the Filters button for this item;

**Step 3**: Indicate the data field by which the data will be filtered for the current element;

**Step 4**: Set the operation of the filter condition;
**Step 5:** Set the check box next to **Expression**;

**Step 6:** Provide a reference to the variable by name. For example, `{Variable1}`.

Despite the fact that the Range variable is not supported when filtering on the dashboard panel, if you need to filter a range of values in elements, this can be done using variables. To do this, you should:

- Create two **Value** variables, where the first variable will represent the value of the range, and the second - its end.
- When creating an element filter, the operation of the filter condition is defined as **Between**.
- Indicate the first variable in the initial value field, and the second variable - in the end value field.

So, in the report viewer, changing the values of variables will change the data filtering range.

Now, when viewing the dashboard panel, data can be filtered by the value of the variable. To do this:

**Step 1:** Open the dashboard panel in the preview panel or in the viewer;
**Step 2:** Select or enter a value if the variable provides for the selection or input of values;

**Step 3:** Click the **Submit** button on the options bar.

**Information**

You should know that when filtering data by variable values, you can also use several variables, including dependent variables. In addition, after filtering the data of the
dashboard panel, you can filter the data using the elements List Box, Combo Box, Date Picker, Tree View, Tree View Box.

4.12 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.
Image editor; Element settings; Table of properties.

Information

Interaction can be applied to the current element.

The image can be placed anywhere on the dashboard. Setting up the source for the image element is carried out in its editor. 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.

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 **Icon** field, you can select an icon for the Image element and the color of this icon.
- 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, the image can have only one source. The order in which the object is shown in the image element is as follows:

- An image uploaded from the local storage has the highest priority. This image will overlap the selected icon or image by hyperlink;
- An icon has a medium priority. It will be displayed in the current element if the image from the local storage is not loaded, but regardless of the specified hyperlink.
- An image by a hyperlink has the lowest priority. The hyperlink will upload the image in the current element if the image from the local storage is not loaded, and the icon is not selected.
Thus, if a graphic object is loaded directly in the Image element, the image receiving hyperlink or the selected icon will not work.

**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 **AspectRatio** property. When stretching an image, its proportions may be broken. 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 **AspectRatio** 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 **AspectRatio** 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspect Ratio</td>
<td>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.</td>
</tr>
<tr>
<td>Horizontal alignment</td>
<td>Changes the horizontal</td>
</tr>
</tbody>
</table>

© 2003-2020 Stimulsoft
<table>
<thead>
<tr>
<th><strong>Alignment</strong></th>
<th>alignment of the image in the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vertical alignment</strong></td>
<td>Changes the vertical alignment of the image in the current element.</td>
</tr>
<tr>
<td><strong>Back Color</strong></td>
<td>Changes the background color of the element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the element will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td><strong>Border</strong></td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Enables or disables the current item on the dashboard. If the property is set to <strong>True</strong>, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>Sets interaction of the current element.</td>
</tr>
<tr>
<td><strong>Margin</strong></td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td><strong>Padding</strong></td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
<tr>
<td><strong>Title</strong></td>
<td>A group of properties that allows you to customize the title of the</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current element.</td>
</tr>
</tbody>
</table>
### Restrictions

Configures the permissions to use the current item in the dashboard:

- **Allow Change** option enables or disables changes of the element. If checked, the current item can be changed.
- **Allow Delete** option enables or disables the deletion of an element.
- **Allow Move** option allows or prohibits moving an element.
- **Allow Resize** option enables or disables resizing of an element.
- **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 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back Color</td>
<td>Changes the background color of the element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the element will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of the element - color, sides, size, and style.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to <strong>True</strong>, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
</tr>
<tr>
<td>Margin</td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td>Padding</td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
<tr>
<td>Name</td>
<td>Changes the name of the current element.</td>
</tr>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to</td>
</tr>
</tbody>
</table>
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 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.
This chapter will cover the following:

- **Text editor**
- **Table of properties**

---

**Information**

**Interaction** can be applied to the current element.
**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. 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.

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.
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.
<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Specifies text in a Table element. When you click the <strong>Browse</strong> button, the editor of the element will be opened, in which you can add or delete text, as well as customize its appearance.</td>
</tr>
<tr>
<td>Back Color</td>
<td>Changes the background color of the element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the element will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of a table - color, sides, size, and style.</td>
</tr>
<tr>
<td>Fore Color</td>
<td>Specifies the color of the values of the element. By default, this property is set to <strong>From Style</strong>, i.e. the color of the values will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to <strong>True</strong>, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
</tr>
<tr>
<td>Interaction</td>
<td>Sets <strong>interaction</strong> of the current element.</td>
</tr>
<tr>
<td>Margin</td>
<td>A group of properties that allows</td>
</tr>
</tbody>
</table>
**Reports and Dashboards**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Padding</strong></td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
<tr>
<td><strong>Title</strong></td>
<td>A group of properties that allows you to customize the title of the element:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Back Color</strong> property provides the ability to change the background color of the title of the current item. By default, this property is set to <strong>From Style</strong>, i.e. the background color will be obtained from the style settings of the current element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Fore Color</strong> allows you to change the text color of the title of the current item. By default, this property is set to <strong>From Style</strong>, i.e. the text color of the title will be obtained from the settings of the current element style.</td>
</tr>
<tr>
<td></td>
<td>- The group property <strong>Font</strong> allows you to define the font family, its style and size for the title of the current element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Horizontal Alignment</strong> property provides the ability to change the title alignment relative to the element - Left, Center, Right.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Text</strong> property is used to set the title text of the current element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Visible</strong> property is used to enable or disable displaying of</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Configures the permissions to use the current item in the dashboard:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Change</strong> option enables or disables changes of the element. If checked, the current item can be changed.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Delete</strong> option enables or disables the deletion of an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Move</strong> option allows or prohibits moving an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Resize</strong> option enables or disables resizing of an element.</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Allow Select</strong> option enables or disables the element selection.</td>
</tr>
<tr>
<td>Locked</td>
<td>Locks or unlocks resizing and movement of the current element. If the property is set to <strong>True</strong>, the current element cannot be moved or resized. If this property is set to <strong>False</strong>, then this element can be moved and resized.</td>
</tr>
<tr>
<td>Linked</td>
<td>Binds the current location to the</td>
</tr>
</tbody>
</table>
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 Shape

Shape is an element with the help of which various shapes can be displayed on the dashboard.
This chapter will cover the following:

- **Shape editor**
- **Table of properties**

The **Shape** element can be placed anywhere on the dashboard. The setting of the **Shape** element is carried out in the shape editor. 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;

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill</td>
<td>A property group that is used to change the brush and fill color of a shape in the current element.</td>
</tr>
<tr>
<td>Shape Type</td>
<td>Changes the type of a shape in the current element.</td>
</tr>
<tr>
<td>Size</td>
<td>Changes the stroke width of a shape for the current element.</td>
</tr>
<tr>
<td>Stroke</td>
<td>Changes the stroke color of a shape.</td>
</tr>
<tr>
<td>Back Color</td>
<td>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.</td>
</tr>
<tr>
<td>Border</td>
<td>A group of properties that allows you to customize the borders of a table - color, sides, size, and style.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the current item on the dashboard. If the property is set to <strong>True</strong>, the current item is enabled and will be displayed when previewing the dashboard in the viewer. If this property is set to <strong>False</strong>, this element is disabled and will not be displayed when previewing the dashboard in the viewer.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Margin</td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the value area from the border of this element.</td>
</tr>
<tr>
<td>Padding</td>
<td>A group of properties that allows you to define indents (left, top, right, bottom) of the columns from the range of values.</td>
</tr>
</tbody>
</table>
| 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Changes the name of the current element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Changes the alias of the current item.</td>
</tr>
</tbody>
</table>

**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
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables or disables the element selection.</td>
<td></td>
</tr>
<tr>
<td><strong>Locked</strong></td>
<td>Locks or unlocks resizing and movement of the current element. If the property is set to <strong>True</strong>, the current element cannot be moved or resized. If this property is set to <strong>False</strong>, then this element can be moved and resized.</td>
</tr>
<tr>
<td><strong>Linked</strong></td>
<td>Binds the current location to the dashboard or another element. If the property is set to <strong>True</strong>, then the current item is bound to the current location. If this property is set to <strong>False</strong>, then this element is not tied to the current location.</td>
</tr>
</tbody>
</table>

### 4.16 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.
This chapter will cover the following:

- Interaction editor;
- Table element interaction;
- The Interaction editor of the Table element;
- The Interaction editor for Table data fields;
- Tool tips;
- Showing dashboard;
- Parameters;
- Drill Down;
- Drill down with filtering;
Drill down without filtering;
Drill down order for data fields.

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 user clicks on an element.
Showing dashboard can occur only with a single left-click on the input cursor or when a user touches an element value.
Drill-down of values of a dashboard element.

You can specify interactive element actions in the Interaction editor. To call the editor you should:
Select an element on the dashboard in the report designer;
Click the Interaction button on the Home tab of the Ribbon panel of the report designer.

Currently, interactive actions are available for the following items:
Chart;
Regional map;
Text;
Image.

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.
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.

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.

Show Dashboard. When clicking on the value of a dashboard element, another specified dashboard will open.

Drill Down. When clicking on the value of an element, a transition to the lower level of the data hierarchy will occur. You should enable the data drill down mode using the Allow User Drill Down option.

The On Data Manipulation group of parameters is used to customize the data management of the current item.

- The Allow User Sorting parameter is used to enable/disable interactive sorting when viewing dashboards.
- The Allow User Drill Down parameter is an item of the dashboard that enables data drill down mode.

This group of parameters can be used to enable or disable the Full Screen mode and Save buttons in the viewer or in the preview panel for the current dashboard element.

Information

Drill down of element data can be carried out with the Apply Filter action or without it.

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 tooltip 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.

3. The **Drill-Down Filtered** parameter is used to set the drill-down mode when selecting a value in a table. If the check box of this parameter is enabled, then when the table value is selected, the data of related elements will be filtered, and only the row with the selected value will be displayed in the table. If the check box of this parameter is disabled, then when the table value is selected, only the data filtering of related elements will be carried out. Please note that this parameter is relevant if the **Apply Filter mode** is used in the table.

4. The **Full Row Select** parameter is used to select a row of table values when selecting a value in the current element. If the check box of this parameter is enabled, then when selecting the table value, the entire row of values will be chosen. If the check box of the current parameter is disabled, then only the cell with the value that was clicked will be selected.

5. The **Show 'Full Screen'** option is used to enable or disable the full-screen button for viewing the current item in the viewer or the preview panel. If the check box is enabled, then the Show Full Screen button will be displayed on the element in the viewer. If the check box is disabled, then the button will be disabled.

6. The **Show 'Save'** option is used to enable or disable the Save button for the current item in the viewer or in the preview panel. If the check box is enabled, the Save button will be displayed on the element in the viewer. If the check box is disabled, then the Save button will be disabled.

7. The button is used to call the **Table editor**, in which you can configure the interaction for the **data fields of the current element**.

**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.
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 Tip** - 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.

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.

Tools

A tooltip is a message that appears when you hover over the value of an item. The following types of tooltips are available for dashboard elements:

- Value or text, and combinations of them. To do this, set the Mode property to the Show tooltip value.
The hyperlink that is set for the current values. To do this, you should set the Mode property to the Hyperlink value.

You can setup tooltips (value, text) in the editor. To call the editor, click the Edit button of the Display tooltip action.

The editor will open and you may configure the tooltip.
Options are used to customize text style such as font family, size, style, and color.

Text alignment options - left, center, right, width.

Command is used to delete text of a tooltip.

The function menu contains variables. Each variable is an element data field and contains a list of values for this data field. Adding a variable to the tooltip for the item value, the tooltip will display the corresponding value from a specific data field.

Tooltip template. In the current example, the tooltip uses the text and variables of the Chart element.

Showing dashboard
When designing dashboards when you click on the values of a table, chart, map, you can display another dashboard. In this case, it is also possible to pass parameters. Thus, you can create drill down dashboard. Consider the example of displaying a drill down dashboard.

Suppose a dashboard displays sales statistics by category.
You need to display a drill down dashboard for each category - statistics of products sold and availability for each category. You should do the following:

- Create a new dashboard in the report template;
- Put elements for data analysis and displaying of product statistics;
- Specify data fields for these elements.
Then do the following:
- Go back to the main dashboard;
- Select an element for which, when you click on a value, a drill down dashboard will be displayed. In our example, this is a Chart.
- Call the Interaction editor.

In the Interaction editor, you should:
- Set the Mode parameter to **Show Dashboard**;
- Select the dashboard with product statistics as the value of the **Drill-Down Page** parameter.
- Create and configure drill-down parameters.
Then:

➢ Go to the drill down dashboard.
➢ Specify a filter for the elements of the dashboard panel using the drill down parameters.
Now, when you preview the dashboard, you can click on the chart value - in any category. After that, a drill down dashboard with filtered data will be displayed in another tab in the report viewer.
As you can on the picture, a parameter was passed (in the current example, the category name) to the drill down dashboard and the product data was filtered by this parameter. Only data for products in the selected category is displayed.

**Parameters**
A parameter is a value transmitted from the main dashboard to the drill down dashboard. To create a parameter, you should:

- The Mode parameter should be set to **Show Dashboard** in the interaction editor;
- Press the **New Parameter** button;
Specify the parameter name in the **Name** field;  
Specify the parameter value in the **Value** field.

As a parameter value you can specify:
- Any constant - number, text, etc.;
- Variable, for example, `{Variable1}`;
- Link to the item field. In this case, the parameter value will be the value from the specified field of the element. For example, if a chart refers to the **Arguments** field, the parameter value will be the argument value of the selected graphic element of the chart.

**Information**
Each element of the dashboard has its own fields. To indicate a link to an element field, you should:

➤ Click the **Edit** button in the **Value** field of the Interaction editor;
➤ In the **Hyperlink** editor, click the **Insert Expression** button;
➤ Select what you need from the list of element fields.

After the parameter has been created, you should specify the filtering condition using the drill down parameters in the drill down dashboard:

➤ Select an item in the drill down dashboard;
➤ Press the button to call the **Filters editor**;
➤ Specify the data field by which the data will be filtered;
➤ Set the logical operation of the condition;
➤ Specify the parameter as the second value of the filter value. If the parameter is passed directly without using a variable in the data dictionary, then **this["ParameterName"]** must be specified. If a variable is used, then you must specify a link to this variable - **{Variable1}** in the expression field.
Drill Down
Drill down refers to moving to the lower or upper level in the data hierarchy of an element, without rebuilding the dashboard within the current element.
The picture above shows an example of a data hierarchy - the first chart shows sales statistics by category; the second one shows sales statistics of products from selected categories.

Drill down of element data can be done in the following modes:

- **With filters.** In this case, when choosing element values, data will be filtered for all connected elements of the dashboard panel. To drill down the element data, you will need to click the **Drill Down** button on the current element.

- **Without filters.** In this case, when choosing the element value, you will go to the lower level in the data hierarchy of this element. To drill down to multiple values, you will need to click the **Drill Down** button on the current element.
When drill down element data, the data of other elements of the dashboard does not change. Data drill down applies only to the current item.

**Drill down with filtering**

To view hierarchical data within a single element of the dashboard, you can apply drill down on element data. To do this you should:

- Add the main and subordinate data fields to the element in a **specific order**. In charts, drill down is carried out according to the data fields of the chart’s arguments.
- Select an element in the dashboard;
- Press the call button of the interaction editor;
- Enable the **Allow User Drill Down** parameter.

Now, when you choose chart values, the data of all interconnected elements of the dashboard will be filtered, and to drill down to the data of the current element, you should:

- Click the **Drill Down** button on the dashboard element;
Select the element values for which you want to display detail;
Click the **Drill Down Selected** button;
After that, detailed data of the selected element values will be displayed. Click the **Drill Up** button to return to the previous level in the data hierarchy.
Drill down without filtering
Using this option, it will be impossible to filter data for related elements of the dashboard using the current element, and when you select the element value, it will be drilled down. To do this:

➤ Add the main and subordinate data fields to the element in a **specific order**;
➤ Select a dashboard element;
➤ Press the button to invoke the **Interaction editor**;
➤ Enable the **Allow User Drill Down** parameter.
➤ Set the **Drill Down** mode.
Now, when you select the value of an element, its drill down will be implemented.

To drill down to multiple values, you should:

- Click the **Drill down** button in the dashboard element;
Select the element values for which you want to display detail;
Click the **Drill Down Selected** button;
The drill down data of the selected element values will be displayed. Click the **Drill Up** button to return to the previous level in the data hierarchy.
**Drill down order for data fields**

The order of the data fields in the chart arguments displays the drill-down hierarchy in a top-down direction. In other words, the top field is processed as the top level of the hierarchy, and each subsequent field is treated as the next level in the item hierarchy.

So, by changing the order of the data fields in the arguments, the hierarchy of the item drill down changes, but the data hierarchy does not change. To correctly displaying the data hierarchy in an element, you should follow the order of the data fields in the arguments: Up is the main data field, then the detailed data fields.
The numbers indicate the drill down levels of the dashboard element:

- 1 - sales by category;
- 2 - products sales from selected categories;
- 3 - sales by region for the selected products.

However, if you need to display detailed data first, and then go to the main ones, you can use any order of the data fields in the chart arguments.
The numbers indicate the drill down levels of the dashboard element:

1 – products sales by region;
2 – products sales from selected regions;
3 - sales volume by categories for the products selected on the previous level.

4.17 **Conditions**

Conditional formatting is used to highlight information in a certain color.
This chapter will cover the following:

- **Condition editor**;
- **Chart condition options**;
- **Pivot table condition options**;
- **Condition parameters of Indicator**;
- **Applying conditions**;
- **A sample of a condition parameter of a chart**;
- **An example of using conditional formatting in a pivot table**;
- **An example of conditional formatting for an indicator**;
- **Table of condition operations**.
Conditional formatting can be applied to the following elements of the dashboards:

- **Chart**;
- **Pivot**;
- **Indicator**.

Conditional formatting is configured in the condition editor. To call this editor, you should:

- Select an element on the dashboard panel;
- Click the **Condition** button on the **Home** tab in the report designer.

**Condition editor**

In the editor, conditions can be added, configured, moved and deleted.

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Data Type</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>Numeric</td>
<td>less than</td>
<td>15000</td>
</tr>
<tr>
<td>Color</td>
<td>Color[A=255, R=0, G=]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Data Type</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>Numeric</td>
<td>greater than</td>
<td>15000</td>
</tr>
<tr>
<td>Color</td>
<td>Color[A=255, R=152. ]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Data Type</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>Numeric</td>
<td>greater than</td>
<td>25000</td>
</tr>
<tr>
<td>Color</td>
<td>Color[A=255, R=0, G=]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 The **Add Condition** button is used to add a new condition to the list of conditions.
2. The **Remove Condition** button is used to remove the selected condition from the list.
3. Buttons to move up or down the selected condition in the list of conditions.

**Information**

All conditions are processed sequentially in the direction from top to bottom - the higher is the condition in the list, the earlier it is processed and applied. See the details how to apply the conditions.

For example (see the picture above):

4. The first condition in the list is processed and applied primarily to the dashboard element.
5. The next condition in the list is processed and applied after the first condition.
6. The last condition in the list is processed and applied last.

**Chart condition options**

For each new condition, you should specify the parameters of its applying. The color will be applied to the specific value of the element if the condition is executed.

![Condition options](image)

1. The **Field Is** parameter is used to specify the data field from which the source values will be obtained.
2. The **Data Type** parameter is used to specify the type of condition values.
3. The **Condition** parameter is used to specify the condition operation, which means the operation of logical comparison of the initial value of the series and the value of the condition.
4. The **Value** parameter is used to specify a condition value.
5. The **Color** parameter is used to specify the color that will be applied to the value of the element when the condition is executed.
**Information**

For line charts, two additional options will be displayed:
- **Marker Type** is used to change the type of a marker when the condition is executed;
- **Angle** is used to rotate the marker to the right (positive value) or left (negative value).

### Pivot table condition options

For each new condition, you should specify the parameters of its application and design settings. Design settings will be applied to the cells of the pivot table, if the condition will be executed.

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Data Type</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UnitPrice</code></td>
<td><code>Numeric</code></td>
<td><code>greater than</code></td>
<td><code>3</code></td>
</tr>
</tbody>
</table>

1. **Field Is** parameter. It is used to specify the field from which the initial values will be taken – from the field of rows, columns or totals. You should know that, depending on the selected field, conditional formatting will be applied to its values. If the total field is selected, formatting will be applied to the values of the current field in the pivot table. If a row or column field is selected, formatting will be applied to the row or column headings, respectively.
2. **Data Type** parameter. It is used to specify the type of condition values. This parameter affects how the report engine handles the condition. Also, the list of condition operations depends on this parameter.
3. **Condition** parameter. It is used to specify the condition operation, the operation of a logical comparison of the initial value from the data field, and the value from the condition.
4. **Value** parameter. It is used to specify a condition value.
5. The preview panel for the conditional formatting value, when the condition is executed.
6. Commands with which you can specify the font style in the cell of the pivot table.
when the condition is executed.

- **The Fore Color option.** It is used to specify the text color of the cell in the pivot table to which conditional formatting will be applied.

- **The Back Color parameter.** It is used to specify the background color of the cell in the pivot table to which conditional formatting will be applied.

- **Formatting settings menu.** You should check the conditional formatting parameters that must be applied to the values of the pivot table if the condition is executed.

- **The Font parameter.** It is used to specify the font family for the pivot table cell to which the conditional formatting will be applied.

- **The Font Size parameter.** It is used to set the font size in the cell of the pivot table to which the conditional formatting will be applied.

- **The Icon parameter.** It is used to enable and locate the icon relative to the value in the cell in the pivot table to which the conditional formatting will be applied.

- **The Icon Type parameter.** It is used to select a value icon from the list of Stimulsoft icons. You can also load a custom value icon. To do this, click the **Browse** button and select the icon from the repository.

- **The Icon Color option.** It is used to specify the color of the icon for values.

Also, when using the conditions of the pivot table, you should take into account the procedure for applying conditions. Below is a step-by-step example of using conditional formatting in a pivot table.

**Condition parameters of Indicator**

For each new condition, you should specify the parameters of its application and design settings. Design settings will be applied to the indicator values if the condition is executed.

1. **The Field Is parameter** is used to specify the field from which the initial values will be
taken: from the value field, target, series etc.

1. The **Condition** parameter is used to specify the **condition operation**. This is the operation of logical comparison of the initial value from the data field and the value from the condition.

2. The **Value** parameter is used to specify a condition value.

3. Format settings menu. In this menu, you should check the conditional formatting parameters that must be applied to the indicator, if the condition is executed.

4. The **Font** parameter is used to specify a font family for indicator values.

5. Commands using which you can specify the font style in the indicator.

6. The **Fore Color** option is used to specify the color of the indicator value.

7. The **Back Color** parameter is used to specify the background color of the indicator.

8. The **Font Size** parameter is used to set the font size of indicator values.

9. The **Icon group** of parameters is used to change the appearance of the value icon, its position and color.

10. The **Target Icon** group of parameters is used to change the appearance of the relative value icon, its position and color. You should know that the color of the target icon will also be applied to the deviation value.

Also, when using the indicator conditions, the procedure for **applying the conditions** should be considered. Below is a **step-by-step example of using conditional formatting for an indicator**.

**Terms of using conditions**

All conditions are processed sequentially, in the "from top to down" direction in the list of conditions. When creating multiple conditions for a single element, logical operations should be considered.

Considering the **logical operation** of the condition and the value of the condition, a range of element values is formed to which formatting will be applied. For example, a condition operation is less than, a condition value is 3. Therefore, formatting will be applied to all element values that are less than 3.

For the **example discussed bottom**, change the order of the conditions - move the condition of maximum values (green color) above the average values (yellow color).
And then, there will be no values indicated in green on the chart.
This will happen because:

**Step 1:** The report engine will analyze all the values of the selected data field;

**Step 2:** Apply a red color to all values that are less than 10,000.

**Step 3:** Apply a green color to all values that are greater than 19,000.

**Step 4:** Apply a yellow color to all values that are greater than 10,000. Values greater than 19,000 fall into the range of values of the last condition, and formatting will be applied to them.

Therefore, when using conditions in the elements of the dashboard, it is important to track the logical operations of the conditions and their order in the list of conditions.

**Chart formatting example**

Suppose a bar chart with product sales is put on the dashboard panel. The break-even point of sales is 10,000 and the normal sales level is 19,000.
Information

Learn how to add constant lines in a chart.

All values which are less than 10,000 are specified in red, and all values greater than 10,000 are green, and average values are specified in yellow. To do this:

**Step 1**: Select a chart on the dashboard panel;

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

**Step 3**: Click the **Add condition** button in the Condition editor. In the current example, three conditions are required (for minimum, average, and normal values).

**Step 4**: Specify parameters for each condition.
Step 5: For values less than the break-even point (> 10,000), select the less than condition operation and set 10,000 in the value field. Define the color as red. Also, set the marker as a five-pointed star.

Step 6: For average values, select the greater than condition operation and specify 10,000 in the value field. Define the color as yellow. Also, set the marker as an eight-pointed star.

Step 7: For values greater than normal (< 19,000), select the greater than condition operation and specify 19,000 in the value field. Define the color as green. Also, set the marker as a seven-pointed star.

Step 8: Click OK in the condition editor.
After that, the conditions will be applied to the chart values.
An example of using conditional formatting in a pivot table
Suppose, on the dashboard panel, there is a pivot table with sales by category. Minimum sales are 1,000, and normal sales are above 1,500.
All values that are less than 1,000 are highlighted with a font, replace the background and add a red icon, and all values greater than 1,500 with a green icon. Do the following to achieve this:

**Step 1:** Select the pivot table on the dashboard panel;

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

**Step 3:** Click the **Add condition** button in the condition editor. In the current example, two conditions are required (for the minimum and normal values).
Step 4: Select the data field that is used to calculate the totals of the pivot table and specify the parameters for each condition.

Step 5: For values less than the minimum sales level (>1,000), select the condition operation less than and specify 1000 in the Value field. Also, change the font, using the condition parameters, background, and icon settings.

Step 6: For values greater than normal (>2,500), select the condition operation greater than and specify 2500 in the Value field. Also, change the font, background, and icon settings, using the condition parameters.

Step 7: Click OK in the Condition editor.

After that, the conditions will be applied to the values of the pivot table.
Apply conditional formatting for column and row headers. For this you should do the following:

**Step 1:** Select the Pivot table in the dashboard panel;

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

**Step 3:** Click the Add condition button in the condition editor. In the current example, three conditions are required.

**Step 4:** Select the data field that is used to form the columns of the pivot table.
**Step 5:** Define the condition and define *formatting settings*;

**Step 6:** For the column whose name begins with Cond, change the background, font size, and color, and add a green icon to the name;

**Step 7:** For the column whose name begins with P, change the background, font size, and color, and add a red icon to the name;

**Step 8:** Select the data field that is used to form the rows of the pivot table. Change the background, font size and color for lines whose name ends with t;

Now conditional formatting will also be applied to row and column headings of the
Suppose an indicator with the number of orders and the deviation value is located on the indicator panel. A positive deviation is ensured by the number of orders greater than 164, but the minimum acceptable deviation level is +5 percent.

### Pivot Table

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Category Name</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Mutton</td>
<td>Confections</td>
<td></td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>Beverages</td>
<td>$1.0K</td>
</tr>
<tr>
<td>Boston Crab Meat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camembert Pierrot</td>
<td>Condiments</td>
<td>$3.4K</td>
</tr>
<tr>
<td>Carnavon Tigers</td>
<td></td>
<td>$1.8K</td>
</tr>
<tr>
<td>Chai</td>
<td></td>
<td>$1.8K</td>
</tr>
<tr>
<td>Chang</td>
<td></td>
<td>$1.9K</td>
</tr>
<tr>
<td>Chartreuse verte</td>
<td></td>
<td>$1.8K</td>
</tr>
<tr>
<td>Chef Anton’s Cajun Seasoning</td>
<td></td>
<td>$2.2K</td>
</tr>
<tr>
<td>Chef Anton’s Gumbo Mix</td>
<td></td>
<td>$2.1K</td>
</tr>
<tr>
<td>Chocolade</td>
<td>Dairy Products</td>
<td>$1.3K</td>
</tr>
<tr>
<td>Côte de Blaye</td>
<td></td>
<td>$26.4K</td>
</tr>
<tr>
<td>Escargots de Bourgogne</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filo Mix</td>
<td>Grains/Cereals</td>
<td>$700.0</td>
</tr>
<tr>
<td>Fløtemyost</td>
<td></td>
<td>$2.2K</td>
</tr>
<tr>
<td>Geitost</td>
<td></td>
<td>$250.0</td>
</tr>
<tr>
<td>Genen Shouyu</td>
<td></td>
<td>$1.6K</td>
</tr>
<tr>
<td>Gnocchi di nonna Alice</td>
<td></td>
<td>$3.8K</td>
</tr>
<tr>
<td>Gorgonzola Telino</td>
<td></td>
<td>$1.3K</td>
</tr>
<tr>
<td>Grandma’s Boysenberry Spread</td>
<td></td>
<td>$2.5K</td>
</tr>
<tr>
<td>Gravad lax</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guaraná Fantástica</td>
<td></td>
<td>$450.0</td>
</tr>
<tr>
<td>Gudbrandsdalsost</td>
<td></td>
<td>$3.6K</td>
</tr>
<tr>
<td>Gula Malacca</td>
<td></td>
<td>$1.9K</td>
</tr>
<tr>
<td>Gumbar Gummbärchen</td>
<td></td>
<td>$3.1K</td>
</tr>
<tr>
<td>Gustaf’s Knäckebröd</td>
<td></td>
<td>$2.1K</td>
</tr>
</tbody>
</table>
Step 1: Select an indicator in the dashboard panel;

Step 2: Click the Conditions button on the Home tab in the report designer;

Step 3: Click the Add Condition button in the condition editor. In our example, two conditions are required (for the indicator value and the deviation value).

Step 4: For the first condition, you should select the data field of the indicator values and set the conditional formatting parameters, which will depend on the indicator value. For example, if the indicator value is less than 165, then the color of the value, the
value icon and the color of this icon will change.

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>less than</td>
<td>165</td>
</tr>
</tbody>
</table>

Font: Arial

Information

You should know that if several conditions are used for different types of fields, unused formatting options should be disabled (see section 4).

Step 5: Select the deviation field for the second condition and specify conditional formatting parameters, which will depend on the deviation value. For example, if the deviation value of the indicator is less than 6 percent, the color of the deviation value, the deviation icon, and the color of this icon will change.

<table>
<thead>
<tr>
<th>Field Is</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variation</td>
<td>less than</td>
<td>6</td>
</tr>
</tbody>
</table>

Font: Arial

Information

Please note that the color of the percentage deviation value of the indicator is specified not by the Fore Color parameter, but by the Color parameter of the target icon.
When creating the dashboard panel, the following issues are possible:

- The indicator value is greater than 165 (the first condition is not satisfied and formatting is not applied); the deviation value is more than 6 percent (the second condition is not satisfied and formatting is not applied);
- The indicator value is greater than 165 (the first condition is not satisfied and formatting is not applied); the deviation value is less than 6 percent (the second condition is satisfied and formatting is applied);
- The indicator value is less than 165 (the first condition is satisfied and formatting is applied); the deviation value is less than 6 percent (the second condition is satisfied and formatting is applied);
In the current example, only some situations and several formatting options were considered. Depending on your requirements, you may combine conditions and formatting options.

**Table of Operations**
The list of available operations depends on the data type. Below is a list of operations for each data type and their description. The operation is performed on the value from the data field and the condition value (the value that is specified in the condition).

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type is String</th>
<th>Data Type is Number</th>
<th>Data Type is Data</th>
<th>Data Type is Boolean</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equal to</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>If the data field value is equal to the condition value, then the</td>
</tr>
<tr>
<td>Condition</td>
<td>True</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>not equal to</code></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the data field value is not equal to the condition value, then the condition is true.

<table>
<thead>
<tr>
<th>Condition</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>between</code></td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

If the data field value is between the condition values, then the condition is true.
If the data field value is not in the specific range of conditions, then the condition is true.

<table>
<thead>
<tr>
<th>not between</th>
<th>+</th>
<th>+</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition value, then the condition is true.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>greater than + + + If the data field value is greater then the condition value, then the condition is true.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
greater than or equal to

If the data field value is greater than the condition value of equal to the filter value, then the condition is true.
<table>
<thead>
<tr>
<th>Condition</th>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>less than</td>
<td>+ + +</td>
<td>If the data field value is less than the condition value, then the condition is true.</td>
</tr>
<tr>
<td>less then or equal to</td>
<td>+ + +</td>
<td>If the data field value is less than or equal to the condition value, then the condition is true.</td>
</tr>
</tbody>
</table>
If the data field value contains the string containing the condition value equal to the filter value, then the condition is true.
If the data field value does not contain the condition value, then the condition is true.

<table>
<thead>
<tr>
<th>Condition Value, the condition is true.</th>
</tr>
</thead>
<tbody>
<tr>
<td>not containing</td>
</tr>
<tr>
<td>+</td>
</tr>
</tbody>
</table>
beginning with

If the data field value starts with the condition value, then the condition is true.

ending with

If the data field value ends
<table>
<thead>
<tr>
<th>Condition</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>is blank</td>
<td>+</td>
<td>If the data field value is blank, then the condition is true.</td>
</tr>
<tr>
<td>is not blank</td>
<td>+</td>
<td>If</td>
</tr>
<tr>
<td>is null</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
### 5 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.

#### 5.1 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.
5.1.1 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.
Exit button from the main menu
The New menu item contains submenu where a list of new report components is available for creation is shown.
The Open Report menu item. When calling this menu item, a dialog for opening a report will appear.
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.
The Save Report As menu item. When calling this menu item, a dialog for saving a report will appear.
Call the Report Checker tool.
Call the Report Setup window of report options.
Close a report that is opened in the report designer.
The Recent Documents menu item contains submenu where recently opened reports are shown.
The Options menu item calls a window for designer parameters settings.
The Exit button closes a report designer.
The panel shows a submenu of selected menu item or selected group.

5.1.1.1 Import

YouTube

Watch our video tutorials on using Importing Tool. Subscribe to the Stimulsoft channel 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.
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 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.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:
### 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://www.crystalreports.com/crvs/confirm/)

Please read more about the requirements of those additional installations in the official reply from the Crystal Reports:


### 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 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.

### 5.1.1.2 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

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 to create reports with charts.
6. The **Cross-Tab** wizard. Used to create Cross-Tab reports.

### 5.1.1.3 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**).

### 5.1.1.4 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 1, 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.
As can be seen from the picture above, this tab consists by three groups:

1. The **Grid Options** group includes such parameters as: **Align to Grid** snaps a report component to grid; **Show Grid** enables/disables the grid.
2. The **Grid drawing options** group. The grid style can be applied in this group: **Lines** or **Dots**;
3. The **Grid size** group. This group sets the grid spacing in different units: **Inches**, **Hundredths of Inch**, **Centimeters**, **Millimeters**, **Pixels**.

Parameters of quick info messages are defined in the **Quick Info** tab.
As can be seen from the picture above, this tab contains a single group 1, which defines the following parameters:

- 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.

![Auto save options](image)

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.

![Color Scheme](image)

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.

### 5.1.1.5 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.
5.1.1.6 Share

When designing reports, it often becomes necessary to provide access to it to other users. This can be done in various ways:

- 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.

- 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.

- 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.

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.
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
data 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](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](https://stimulsoft.com/s/55af6/pdf).

### 5.1.2 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.

#### 5.1.2.1 Tab Home

This is a basic tab of the report designer. Main commands for customizing report components are placed on this tab.
5.1.2.1.1 Group Clipboard

This group allows working with the Clipboard of the report designer.

1. Paste components from the Clipboard on the current page of a report.
2. Cut the selected components from the current page to the Clipboard.
3. Copy the selected components on the current page to the Clipboard.
4. Delete selected components on the current page.

5.1.2.1.2 Group Font

This group is used to output text with the specified font type, color etc.

1. Select the font type of the text components on the current page.
2. Select font size of the text components on the current page.
3. Sets the font style as Bold, Italic, Underlined.
4. Set the font color of the text components on the current page.
5. Changes (increases/decreases) the font size.
6. Delete the content of all selected text components.

5.1.2.1.3 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.

5.1.2.1.4 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.

5.1.2.1.5 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.
5.1.2.1.6 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 \texttt{WordWrap} property.

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 \texttt{WordWrap} property of the text component.

5.1.2.2 Tab Page

This tab is used to control page parameters.

5.1.2.2.1 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.

5.1.2.2.2 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
5.1.2.2.3 Group Watermark Image

This group is used to customize the watermark image.

1. Load an image for the watermark
2. Align the watermark image
3. Sets the number of watermarks
4. Set the image transparency
5. Other watermark options

5.1.2.2.4 Group Viewing Options

This group contains settings for different parameters of showing a report and working with the report designer.

1. Control showing grid on a page.
2. Control automatic alignment of components by the grid.
3. Select grid type.
4. Show/hide headers of bands.
5. Show/hide the order of placing components on a page.

© 2003-2020 Stimulsoft
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.

### 5.1.2.3 Tab Layout

This tab is used to control placing different components on a page and also to specify styles of docking components.

#### 5.1.2.3.1 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 Forward.
7. Move selected components Backward.
8. Lock components.
9. Link components.
Move selected components on one level backward.

Choose the size of selected components. It contains submenu and is described in this topic below.

Control the Lock property.

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.
5.1.2.3.2  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.

5.1.2.4  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.
5.1.3 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:0.00; Y:0.00) coordinates corresponds to the top left corner of a page of a report template
5. Page Zooming control.

5.2 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.

5.2.1 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:
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.

5.2.1.1 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 **Data Band**. 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**:

```
| Customers.ContactName | Customers.City | Customers.Country |
```

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:
Filters. On this step, it is possible to set the conditions of filtering. The picture below shows a sample of selection filtering parameters:

Groups. This step defines the condition of grouping. It is necessary to select a data column by what conditions of grouping will be created.

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.

Themes. This step defines the report style.

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:
The New Data Source button is used to create a new data source.

5.2.1.2 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.

5.2.1.3 **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:
1 The Description Panel. Shows description for the current step.
2 The Steps Panel shows step of report creation.
3 The Selection Parameters Panel shows options, actions, settings available on this step.

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.
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.

### 5.2.1.4 Wizard Chart

The **Chart** wizard is used to create reports with charts. The picture below shows a window of the **Chart** wizard.

© 2003-2020 Stimulsoft
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.

### 5.2.1.5 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.

### 5.3 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**.

### 5.3.1 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:

1. The **Actions** menu is a set of commands to work with vocabulary such as: creating,
opening, saving a dictionary, adding, importing and exporting a dictionary to an XML schema, the synchronization of data presented in the dictionary with the data registered in a data store.

2 The **New Item** menu contains commands for creating new elements: new connection, new data source, new connection, new variable, business objects.

3 The **Edit** button is used to edit created elements. Pressing the button runs the editor of the element.

4 The **Delete** button deletes created elements.

5 The **Up** and **Down** buttons move the selected item in the hierarchy of a dictionary within a single level of a tree.

6 The **Sort Items** menu provides the opportunity to choose one of two directions of Sorting: Ascending from A to Z, Descending from Z to A. And also to enable or disable the Auto Sort mode.

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:

![Data Tree Example]

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:
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**
   - The Create Field on Double Click option provides the ability to create a field on the DataBand. Fields are created on the band, that has selected data source by which the fields the double-click in the dictionary is done.

2. **Create Label**
   - The Create Label option attaches the column data header when it is dragged on the report template.

3. **Use Aliases**
   - The Use Aliases option provides an opportunity to show Aliases of components instead of the Name in the report template.

### 5.3.2 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:

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".

### 5.3.3 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 2 3 4 5
```

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):

<table>
<thead>
<tr>
<th>Value Events</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Excel Value</td>
<td></td>
</tr>
<tr>
<td>Get Value</td>
<td></td>
</tr>
<tr>
<td>Get Tool Tip</td>
<td></td>
</tr>
<tr>
<td>Get Tag</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Navigation Events</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Hyperlink</td>
<td></td>
</tr>
<tr>
<td>Get Bookmark</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Print Events</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse Events</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. Behavior</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6. Design</td>
<td></td>
</tr>
<tr>
<td>7. Export</td>
<td></td>
</tr>
<tr>
<td>8. Misc</td>
<td></td>
</tr>
</tbody>
</table>
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.

5.4 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).

<table>
<thead>
<tr>
<th>Buttons</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+B</td>
<td>Makes letters <strong>bold</strong> for the selected text component</td>
</tr>
<tr>
<td>Ctrl+I</td>
<td>Makes letters <em>Italic</em> for the selected text component</td>
</tr>
<tr>
<td>Ctrl+U</td>
<td>Makes letters <strong>Underlined</strong> for the selected text component</td>
</tr>
<tr>
<td>Ctrl+“+”</td>
<td>Increase the font size for the selected component</td>
</tr>
<tr>
<td>Ctrl+“-”</td>
<td>Decrease the font size for the selected component</td>
</tr>
<tr>
<td>Ctrl+L</td>
<td>Align selection or paragraph to the left</td>
</tr>
<tr>
<td>Ctrl+E</td>
<td>Align selection or paragraph to the centre</td>
</tr>
<tr>
<td>Ctrl+R</td>
<td>Align selection or paragraph to the right</td>
</tr>
<tr>
<td>Ctrl+J</td>
<td>Justify selection</td>
</tr>
<tr>
<td>Shortcut</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Ctrl+Shift+D</td>
<td>Activate the “Dictionary” panel</td>
</tr>
<tr>
<td>Ctrl+Shift+M</td>
<td>Activate the “Messages” panel</td>
</tr>
<tr>
<td>Ctrl+Shift+L</td>
<td>Activate the “Report Tree” panel</td>
</tr>
<tr>
<td>F4</td>
<td>Activate the “Properties” panel</td>
</tr>
<tr>
<td>Ctrl+C, Ctrl+Insert</td>
<td>Copy the selected text or object</td>
</tr>
<tr>
<td>Delete, Ctrl+Delete</td>
<td>Delete the selected component</td>
</tr>
<tr>
<td>Ctrl+V, Shift+Insert</td>
<td>Paste the text or object from the Clipboard</td>
</tr>
<tr>
<td>Ctrl+X, Shift+Delete</td>
<td>Cut the selected text or object</td>
</tr>
<tr>
<td>Ctrl+A</td>
<td>Select All</td>
</tr>
<tr>
<td>Ctrl+Z</td>
<td>Undo</td>
</tr>
<tr>
<td>Ctrl+Y</td>
<td>Redo</td>
</tr>
<tr>
<td>Ctrl+F2</td>
<td>Show “Data Store”</td>
</tr>
<tr>
<td>Ctrl+F4</td>
<td>Show “Page Manager”</td>
</tr>
<tr>
<td>Ctrl+F5</td>
<td>Show “Services Configurator”</td>
</tr>
<tr>
<td>Ctrl+N</td>
<td>Create a new report</td>
</tr>
<tr>
<td>Ctrl+Shift+N</td>
<td>Add a page to the report</td>
</tr>
<tr>
<td>Ctrl+Shift+F</td>
<td>Add a form to the report</td>
</tr>
<tr>
<td>Ctrl+O</td>
<td>Load a report from the file</td>
</tr>
<tr>
<td>Ctrl+Shift+O</td>
<td>Load a page from the file</td>
</tr>
<tr>
<td>Ctrl+S</td>
<td>Save a report</td>
</tr>
<tr>
<td>Ctrl+F12</td>
<td>Save a report as...</td>
</tr>
<tr>
<td>Ctrl+Tab</td>
<td>Switching pages in the report template</td>
</tr>
<tr>
<td>F5</td>
<td>Report preview</td>
</tr>
<tr>
<td>Ctrl+Enter</td>
<td>Call the designer be default for the elected component</td>
</tr>
<tr>
<td>Enter</td>
<td>Call the text editor for the selected component</td>
</tr>
</tbody>
</table>

© 2003-2020 Stimulsoft
<table>
<thead>
<tr>
<th>Key Combination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Select the “Select” tool</td>
</tr>
<tr>
<td>F3</td>
<td>Select the “Text Editor” tool</td>
</tr>
<tr>
<td>F6</td>
<td>Select the “Copy Style” tool</td>
</tr>
<tr>
<td>F8</td>
<td>Select the “Hand” tool</td>
</tr>
<tr>
<td><strong>Cursor keys</strong></td>
<td>Move selection</td>
</tr>
<tr>
<td>Shift+Cursor keys</td>
<td>Resize selected component (one step = grid size)</td>
</tr>
<tr>
<td>Shift+Alt+Cursor keys</td>
<td>Resize selected component (one step = half grid size)</td>
</tr>
<tr>
<td>Ctrl+Cursor keys</td>
<td>Move selected component (one step = grid size)</td>
</tr>
<tr>
<td>Ctrl+Alt+Cursor keys</td>
<td>Move selected component (one step = half grid size)</td>
</tr>
<tr>
<td>Ctrl+Drag mouse</td>
<td>Copy selected components</td>
</tr>
<tr>
<td>Alt+Drag mouse</td>
<td>Ignore “Align to Grid” when moving and resizing</td>
</tr>
</tbody>
</table>

### 5.5 Report Checker

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**.

### 5.6 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.

<table>
<thead>
<tr>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make Same Size as DataProductsCategoryID</td>
</tr>
<tr>
<td>Make Same Width as DataProductsCategoryID</td>
</tr>
<tr>
<td>Make Same Height as DataProductsCategoryID</td>
</tr>
<tr>
<td>Make Horizontal Spacing Equal</td>
</tr>
<tr>
<td>Make Vertical Spacing Equal</td>
</tr>
</tbody>
</table>

If you select all the components on the page, i.e. choose the **Select all** command in the context menu, or press the **Ctrl+A** key combination, then the prototype of the size for all components will be the size of the component that is placed on a higher level and higher than other components in the report tree in that level. It is possible to see the report tree on the **Report Tree** panel. The picture below shows an example of a report tree.

As seen on the picture above, the **PageHeaderBand** is located higher, so, in this case, its size is a sample of sizes for the remaining components. If the **PageHeaderBand** is absent, then, as a sample of sizes, the **PageFooterBand** will be taken. If the **PageFooterBand** is
absent, then, as a sample of sizes, the **Text1** will be taken.

- 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:

![Order Submenu](image)

- 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:

![Align Submenu](image)

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.

### 5.7 Previewing Reports

Report Designer allows previewing a report before printing, exporting, sending via Email or any other action, 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.

5.8 Toolbox

The Toolbox panel contains the main tools for creating reports. All items on this panel are divided into the following categories: Select Edit Mode Category, Components Category, Shortcut Category, Shortcut Category Settings. The picture below shows the Toolbox:
The Toolbox is located on the left side in the designer window and looks like a vertical panel. If necessary, the Toolbox bar can be shown or hidden. In Ribbon UI hiding or showing the Toolbox can be done by pressing the Toolbox button, which is located in the View tab. If the button is pressed, the Toolbox is shown. In Standard UI hiding or showing the Toolbox can be done by right-clicking and calling the context menu of the Toolbox or the Main Menu.

⚠️ 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.

### 5.8.1 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:
In addition, the Toolbox has the Setup Toolbox button:

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.

5.9 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.

5.10 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.

5.11 Publish

YouTube

Watch our videos to learn how to publish reports. Subscribe to the Stimulsoft channel and be the first who watches new video tutorials. Leave your questions and suggestions in the comments to the video.
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:

```html
<!DOCTYPE html>
<html>
<head>
  <title>Report.mrt - Viewer</title>
  <link rel="stylesheet" type="text/css"/>
  <script type="text/javascript">
    var str = Stimulsoft.Sys.String localization;
    var report = new Stimulsoft.Report;
    report.load(str);
    var options = new Stimulsoft.Viewer;
    var viewer = new Stimulsoft.Viewer;
    viewer.report = report;
    viewer.renderHtml("viewerContent");
  </script>
</head>
<body>
</body>
</html>
```
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.
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.
The option for selecting a framework type. You can select a JavaScript application without using a framework, or select the Node.js framework.

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.

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.

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:

- Synchronize Report Dictionary. Use this option to synchronize the registered data in the data storage and in the data dictionary of the report.
- 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.

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.

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.

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.

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.

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.

The Hide Options button is used to expand and collapse the options bar in the publish wizard.

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.

5.12 Report Template

A report or dashboard is a way of processing data presented by any structure. The report engine processes the data of the report or dashboard, and its structure is created in the report template. A report template is an area in the report designer in which a report structure or analytical panel is created using components or elements, respectively.

You can create the structure, a position of components or elements:
- On a page or form if you design a report;
- On the dashboard panel, if you create a dashboard.
The report template has its settings that affect both the process of building a report or dashboard and its result. For example, in the properties of a report template, the expression processing mode is determined - compilation or interpretation. Also, using the settings of the report template, you can configure the preview panel, report update time, report culture, and more.

The following ways exit to change the report template settings:

> Click on the report template area (outside the page or dashboard), and set the property values on the **Property** panel in the report designer.
Double-click the left mouse button in the report template area (outside the page or dashboard) to call the **Report Options** window.

The **Report Options** window contains duplicate properties of the report template. A complete list is provided on the **Property** panel.

The table below shows the properties of the report template.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Name</td>
<td>It is used to change the name of the report.</td>
</tr>
<tr>
<td>Report Alias</td>
<td>It is used to change the report alias.</td>
</tr>
<tr>
<td>Report Author</td>
<td>It is used to change the author of the report.</td>
</tr>
<tr>
<td>Report Description</td>
<td>It is used to change the description of the report.</td>
</tr>
<tr>
<td>Report Image</td>
<td>It is used to upload an image that will be a thumbnail for the current report.</td>
</tr>
<tr>
<td>Auto Localize Report on Run</td>
<td>It is used to enable the automatic localization of strings. Learn more about this in this section.</td>
</tr>
<tr>
<td>Cache All Data</td>
<td>It is used to enable or disable the caching mode of all data in one DataSet. If the property is set to True, then all data will be cached in one DataSet. If the property is set to False, then all data will not be cached in one DataSet.</td>
</tr>
<tr>
<td>Cache Totals</td>
<td>It is used to enable or disable caching of totals with the Totals prefix. If the property is set to True, the totals will be cached. If the property is set to False, the totals will not be cached.</td>
</tr>
<tr>
<td>Calculation Mode</td>
<td>It is used to determine the processing mode of report expressions - Compilation or Interpretation. Learn more about this in this section.</td>
</tr>
<tr>
<td>Convert Nulls</td>
<td>It is used to convert null to default values, for numerical</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>values</td>
<td>- to zero. If the property is set to True and the data column type containing null is not Nullable, all null values will be converted to default values. If the property is set to False, null values will not be converted.</td>
</tr>
<tr>
<td>Collate</td>
<td>It is used to shuffle the pages of a rendered report. If the property is set to greater than 1, then all pages of the rendered report will be split into groups, and then one page from each group will be sequentially added to the new page collection. If the property is set to 1, then the report pages will not be shuffled.</td>
</tr>
<tr>
<td>Culture</td>
<td>It is used to change the report culture. You can learn more about the report culture in this section.</td>
</tr>
<tr>
<td>Engine Version</td>
<td>It is used to select the version of the report engine that will be used to build reports.</td>
</tr>
<tr>
<td>Globalization Strings</td>
<td>It is used to customize globalization strings in a report. Click the Browse button in the value field to open the Globalization editor.</td>
</tr>
<tr>
<td>Number of Pass</td>
<td>It is used to select the number of passes when rendering the report - Single Pass, Double Pass.</td>
</tr>
<tr>
<td>Preview Mode</td>
<td>It is used to define the preview mode – Standard, Standard and Dot-Matrix, and Dot-Matrix.</td>
</tr>
<tr>
<td>Preview Settings</td>
<td>It is used to customize the</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Preview Panel of Reports and Dashboards</td>
<td>Click the <strong>Browse</strong> button in the value field to open the <a href="#">preview settings editor</a>.</td>
</tr>
<tr>
<td>Printer Settings</td>
<td>A group of properties that is used to specify print settings - select a printer, set the duplex mode, determine the number of copies, etc.</td>
</tr>
<tr>
<td>Referenced Assemblies</td>
<td>It is used to edit the list of used assemblies. Click the <strong>Browse</strong> button in the value field to open the row collection editor, in which you must add or remove the necessary assemblies.</td>
</tr>
<tr>
<td>Refresh Time</td>
<td>It is used to determine the time of rebuilding a report or dashboard. You can <a href="#">learn more about the refresh time</a> in this chapter.</td>
</tr>
<tr>
<td>Report Cache Mode</td>
<td>It is used to choose the mode of report caching. The next values are available On, Off, and Auto. If the current property is set to <strong>Auto</strong>, the report caching will be activated automatically if the number of report pages is more than 200.</td>
</tr>
<tr>
<td>Report Unit</td>
<td>It is used to select the units in the report- Centimeters, Inches, Hundredths, and Millimeters.</td>
</tr>
<tr>
<td>Retrieve Only Used Data</td>
<td>It is used to request only the necessary data or all dictionary data. You can <a href="#">learn more about requesting only the necessary data</a> in this chapter.</td>
</tr>
<tr>
<td>Parameters Orientation</td>
<td>It is used to select the orientation of the toolbox panel when viewing a report - Vertical or Horizontal.</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>It is used to request input parameters before rendering a report. If the property is set to True, you should enter the parameters before building the report. If the property is set to False, then it is not required to enter parameters before building the report.</td>
</tr>
<tr>
<td>Script Language</td>
<td>It is used to choose a scripting language - CSharp or VB.NET.</td>
</tr>
<tr>
<td>Stop Before Page</td>
<td>It is used to stop render a report when it reaches a specific page. The numerical value is indicated in the value field of this property. This value is the serial number of the page of the rendered report, after which the report rendering will be stopped. By default, the property is set to 0, which means that there are no restrictions on the number of pages of the rendered report. The entire report will be built.</td>
</tr>
<tr>
<td>Styles</td>
<td>It is used to call a style designer. Click the Browse button in the value field to call the Style Designer.</td>
</tr>
</tbody>
</table>

### 5.12.1 Calculation mode

When you design reports and dashboards, expressions can be processed in the **Interpretation** or **Compilation** mode.

In the **Compilation** mode, the CSharp compiler is used to calculate expressions. In this
case, it is allowed to use events, various methods, and functions of the .NET Framework. However, the time taken to build a report or dashboard is slowing down, and it also requires more RAM.

In **Interpretation** mode, the Stimulsoft interpreter is used to calculate expressions. This speeds up the building of a report or dashboard and reduces the required amount of RAM. However, only built-in functions and methods can be used in a report or dashboard. The use of events and third-party scripts is not allowed.

### Information

For some platforms, report generators cannot use the **Compilation** mode. For example, on the .NET Core and JavaScript platforms, all expressions in reports and on dashboards are processed only in the Interpretation mode.

You can change the mode of calculating expressions in a report or on the dashboard in the following way:

➢ In the report designer, select the report template area and choose the expression processing mode as the value of the **Calculation Mode** property on the **Property** panel.
Double-click on the report template area to call the **Report Options** window. Select the expression processing mode as the value of the **Calculation mode** parameter.
You should know that you can only edit the dashboard from the viewer if the expression processing mode is set to Interpretation.

5.12.2 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:

- 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.

**Information**

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.

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.

5.12.3 **Report Culture**

By default, the regional settings of the operating system are used to build reports and dashboards. If you need to show data in the report or dashboard, regardless of the current culture in the operating system, then you should apply a particular culture to this report or dashboard. To apply culture to a report or dashboard, use the **Culture** property of the report template. Set the culture code (format `xx-XX`, for example, `en-GB`) in the field of the property. After that, the report generator, before rendering a report, will set a specific culture and apply regional settings for components and elements. Below, you may find an example of the report with different cultures:

### Russian (Russia) (ru-RU)

<table>
<thead>
<tr>
<th>ProductName</th>
<th>UnitPrice</th>
<th>OrderDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quezo Cabriles</td>
<td>18,00p.</td>
<td>03.03.2005 23:02:00</td>
</tr>
<tr>
<td>Singaporean Hokkien Filied Mee</td>
<td>18,00p.</td>
<td>03.03.2005 23:02:00</td>
</tr>
<tr>
<td>Mozzarella d Giovanni</td>
<td>18,00p.</td>
<td>03.03.2005 23:02:00</td>
</tr>
</tbody>
</table>

### Arabic (Libya) (ar-LY)

<table>
<thead>
<tr>
<th>ProductName</th>
<th>UnitPrice</th>
<th>OrderDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quezo Cabriles</td>
<td>18,000,00،</td>
<td>03/03/2005 11:02:00</td>
</tr>
<tr>
<td>Singaporean HokkienFilied Mee</td>
<td>18,000,00،</td>
<td>03/03/2005 11:02:00</td>
</tr>
<tr>
<td>Mozzarella d Giovanni</td>
<td>18,000,00،</td>
<td>03/03/2005 11:02:00</td>
</tr>
</tbody>
</table>
You should notice that the first columns contain text that is independent of the report culture. The second (currency) and third (date-time) columns are culture-dependent. Therefore, when changing the culture, the type of data record changes.

Information

It is impossible to remember all the codes of cultures. Therefore, for convenience, you can find the list of values of the Culture property in a drop-down menu with a list of cultures and their codes that are available in the operating system on the current computer.

If you need the components to be independent of culture, displayed the same for any culture applied to the report, you should uncheck the Use local settings parameter and define formatting settings in the Text Format editor of the text component. For example, you want to see the price of a product always in the same currency, regardless of regional settings. Below is a report sample with different cultures:

Russian (ru-RU)
As you can see in the picture, the currency depends on the culture applied to the report, which is not entirely true. In order for the price to always be in the same currency, you should select the text component in the report template with reference to the **UnitPrice 2** column and the **Currency format** editor to determine specific parameters – currency, USD. Now, regardless of the report culture, the price in this column will always be in the USD:

**English (United States) (en-US)**

<table>
<thead>
<tr>
<th>ProductName</th>
<th>UnitPrice 1</th>
<th>UnitPrice 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sasquatch Ale</td>
<td>$14.00</td>
<td>$14.00</td>
</tr>
<tr>
<td>Steelye Stout</td>
<td>$18.00</td>
<td>$18.00</td>
</tr>
<tr>
<td>Inugd Sill</td>
<td>$19.00</td>
<td>$19.00</td>
</tr>
</tbody>
</table>

As you can see, when applying the Russian (ru-RU) culture, the currency in the second column has not changed, while in the first one, it depends on the culture used.

**Russian (ru-RU)**

<table>
<thead>
<tr>
<th>ProductName</th>
<th>UnitPrice 1</th>
<th>UnitPrice 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sasquatch Ale</td>
<td>14.00p.</td>
<td>$14.00</td>
</tr>
<tr>
<td>Steelye Stout</td>
<td>18.00p.</td>
<td>$18.00</td>
</tr>
<tr>
<td>Inugd Sill</td>
<td>19.00p.</td>
<td>$19.00</td>
</tr>
</tbody>
</table>

Information

If the culture selected for the report is not supported by the operating system, then the current culture of the operating system will be applied to the report.
5.12.4 Refresh Time

By default, the report is built:
› When you load reports into the viewer;
› When you switch from the report designer to the Preview tab;
› When you click the Refresh button on the preview tab.

In this case, to rebuild the report, you need some action from the user. To rebuild a report automatically at a specified time interval, you should set the Refresh Time report property to:
› One of the predefined values from seconds to hours;
› Enter the value in seconds manually. For example, if you want the report to be rebuilt every hour, enter 3600 (1 hour = 3600 seconds).

After the specified interval, the report will be automatically rebuilt.

Setting the report update time

Step 1: Run the report designer;

Step 2: Create or open a report;

Step 3: Left-click in the area between the page of the report template and the property panel;

Step 4: Select the Refresh Time property on the property panel, select a preset value or manually enter the value in seconds;

Step 5: Save the changes by clicking the Save button in the upper left corner of the report designer;

Step 6: Go to preview or open the report in the viewer by pressing the F5 button.

After the specified interval, the report will be rebuilt.

5.12.5 Preview settings

When designing a report and dashboard, or before exporting, printing, sending by email, you can preview them. Viewing a report or dashboard is done on a separate tab in the report designer. The preview tab for reports and dashboards contains toolbars
with buttons and menus.

In the report designer, you can customize toolbars to preview a report or dashboard. You can do this in the **Preview Settings** editor. To call this editor, you should do the following:

- Left-click in the report template area (outside the page and dashboard).
- Click the **Browse** button of the **Preview Settings** property.

Since reports and dashboards use different viewing modes, their toolbars differ. On the
Report tab of the editor, the toolbar is configured when viewing the current report, and on the Dashboard tab, the current toolbar is configured.

Report preview settings
In the preview editor, on the Report tab, you can find a panel with a preview and a panel of parameters on which you can disable the buttons and controls in the preview. To disable a button or any control, you should uncheck a particular setting. Accordingly, to enable a button or control, a flag must be checked for that parameter. The included buttons and controls in the preview panel are displayed in real-time.
This option is used to enable/disable the display of the **Print** button on the toolbar.

2. This option is used to enable/disable the display of the **Open** button on the toolbar.

3. This option is used to enable/disable the display of the **Save** button on the toolbar.

4. This option is used to enable/disable the display of the **Send Email** button on the toolbar.

5. This option is used to enable/disable the display of the page navigation control in the status bar.
This option is used to enable/disable the display of the **Find** button on the toolbar.

This option is used to enable/disable the display of the **Editor** button on the toolbar.

This option is used to enable/disable the display of the **New Page** button on the toolbar.

This option is used to enable/disable the display of the **Delete Page** button on the toolbar.

This option is used to enable/disable the display of the **Edit Page** button on the toolbar.

This option is used to enable/disable the display of the **Page Size** button on the toolbar.

This option is used to enable/disable the display of the **View Mode** button on the toolbar.

This option is used to enable/disable the display of the status bar on the preview tab.

This option is used to enable/disable the display of the **Bookmarks** button on the toolbar.

This option is used to enable/disable the display of the **Parameters** button on the toolbar.

This option is used to enable/disable the display of the **Resources** button on the toolbar.

This option is used to enable/disable the display of the **Thumbnails** button on the toolbar.

This option is used to enable/disable the display of the zoom control in the status bar.

This option is used to enable/disable the display of the toolbar in the preview tab.

This option is used to enable/disable of the horizontal scroll control on the preview tab.

This option is used to enable/disable the display of the vertical scroll control on the preview tab.

This option is used to enable/disable the display of the context menu on the preview tab.

This option is used to enable/disable the display of the **Close** button on the toolbar.

### Setting the dashboard preview

In the preview editor, on the **Dashboard** tab, a toolbar and a parameter panel are presented, on which you can disable the buttons on the toolbar. To disable the button, you should uncheck the box for a particular parameter. Accordingly, to enable the button, you must check it for any setting. The enabled buttons on the toolbar are displayed in real-time.
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Show 'Tool Bar'</td>
<td>This option is used to enable/disable the display of the toolbar in the preview tab.</td>
</tr>
<tr>
<td>2</td>
<td>Show 'Refresh'</td>
<td>This option is used to enable/disable the display of the Refresh button on the toolbar.</td>
</tr>
<tr>
<td>3</td>
<td>Show 'Open'</td>
<td>This option is used to enable/disable the display of the Open button on the toolbar.</td>
</tr>
<tr>
<td>4</td>
<td>Show 'Edit'</td>
<td>This option is used to enable/disable the display of the Edit button on the toolbar.</td>
</tr>
<tr>
<td>5</td>
<td>Show 'Full Screen'</td>
<td>This option is used to enable/disable the display of the Full Screen button on the toolbar.</td>
</tr>
<tr>
<td>6</td>
<td>Show 'Menu'</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Show 'Report Snapshots'</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Show 'Exports'</td>
<td></td>
</tr>
</tbody>
</table>
This option is used to enable/disable the display of the **Menu** button on the toolbar.

This option is used to enable/disable the display of the **Report Snapshot** command in the **Menu** on the toolbar.

This option is used to enable/disable the display of export commands in the **Menu** on the toolbar.

**Information**

You should know that you can enable or disable the display of buttons on the dashboard elements in the viewer or on the preview tab. Select the element in the report designer, click the **Interaction** button on the **Home** tab of the Ribbon panel, and check/uncheck the box for the parameters if you want to display/hide the buttons of elements.

### 5.12.6 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. 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:
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:
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.

6 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.

Information

When creating report templates, you should consider the value of the Retrieve Only Used Data property.

6.1 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:
The Data Dictionary panel. It contains the necessary 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.

6.1.1 Control Panel

The basic elements to control data dictionary can be found on the control panel. The picture below shows the control panel:

1 The Actions menu. This menu contains the main control commands for the data dictionary;
2 The New Item menu. In this menu the basic commands to create new elements in the data dictionary are placed;
3 The Edit button provides an opportunity to change any element, which can be edited;
4 Using the Delete button one can delete any item in the data dictionary available for deleting;
Pressing the **Up/Down** buttons, the selected item in the data dictionary is moved one position up/down;

6. 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:

![Sorting Items Menu](image)

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.

### 6.1.1.1 Actions Menu

In the **Actions** menu the main commands to control the data dictionary are located. The picture below shows this menu item:

![Actions Menu](image)

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.

**6.1.1.2 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. 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.

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.

### 6.1.2 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.

**Information**

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. To remove the connection...
from the list of favorites, you must click on the "burning" star:

![Star Icon](image)

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.

1. This group contains a list of all connections that support SQL connection strings.
2. This group contains data sources to connect to the data store using REST protocol.
3. The **Other** group contains commands to create connections to data stores such as the XML, Excel, JSON, CSV, DBase.
4. 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.
5. 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.
6. 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.

6.1.2.1 **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.

1 The Name in Source field. In this field, you can enter the name or you can click the
to call a list of names.

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 The **Query Timeout** parameter is used to specify the execution time of a query, which means time during which the request will be executed. If the request timed out and the request failed, the user will see a message about this. The parameter value is indicated in seconds.

8 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.

9 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.

10 Properties panel of the selected data column or a parameter.

**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.

### 6.1.2.1.1 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 \(\text{variable}\).
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 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:

```
SELECT * FROM Categories
WHERE CategoryID = {Parameter1};
```

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:

```
SELECT * FROM Products
WHERE CategoryID = ?
AND ProductID = ?
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:

1. This field displays the parameter Name, which can be edited;
2. This field displays the Type of the parameter, which can be edited;
3. The Expression field displays used expressions in a query parameter, which, if necessary, can be edited;
4. 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:

![New Variable dialog box](image)

Register this variable in the text of the query, using the special "@" symbol before the variable name:
Click **OK**. Now the variable is present in the data source and is used as a parameter in the query.

6.1.2.2 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:

```sql
select * from Products
  where Discontinued = @Variable1
```
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 check box is not checked, the data schema will be retrieved.

Consider a ways of obtaining data in detail.
Retrieving the data schema (the check box 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 check box. 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, 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 check box of the Skip Schema is set)**
After the connection has been created and Skip Schema is checked and you will go to the **New 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.

### 6.1.2.3 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:

![New Column Dialog](image)

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:
6.1.2.4 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:

<table>
<thead>
<tr>
<th>Year2000</th>
<th>Year2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>2069069</td>
<td>13730994</td>
</tr>
<tr>
<td>10140796</td>
<td>7934516</td>
</tr>
<tr>
<td>12798857</td>
<td>5711823</td>
</tr>
<tr>
<td>5526615</td>
<td>5209853</td>
</tr>
<tr>
<td>3114998</td>
<td>3512916</td>
</tr>
<tr>
<td>1681517</td>
<td>3182617</td>
</tr>
<tr>
<td>801360</td>
<td>2632694</td>
</tr>
<tr>
<td>3032874</td>
<td>2170078</td>
</tr>
<tr>
<td>3348361</td>
<td>2049762</td>
</tr>
<tr>
<td>1935527</td>
<td>1557290</td>
</tr>
</tbody>
</table>

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.
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:

1. The **Name** column is used to call this calculated column in the report. Enter in the **Name**.
2. The **Alias** column is used as a prompt. Enter in the **Alias**.
3. The **Type** field provides the ability to choose the data type that will contain the new calculated column.
4. 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.
5. 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:
6.1.2.5 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.

5 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:

<table>
<thead>
<tr>
<th>Column</th>
<th>Aggregate Function</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnitPrice</td>
<td>Sum</td>
<td>UnitPrice.Sum</td>
</tr>
</tbody>
</table>

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:
As can be seen in the picture above, each category corresponds to the total value of all products included in this category.

6.1.2.6 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:

<table>
<thead>
<tr>
<th>Products</th>
<th>CategoryName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>Beverages</td>
</tr>
<tr>
<td>Australia</td>
<td>16</td>
</tr>
<tr>
<td>Brazil</td>
<td>4.5</td>
</tr>
<tr>
<td>Canada</td>
<td>28.6</td>
</tr>
<tr>
<td>Denmark</td>
<td></td>
</tr>
<tr>
<td>Finland</td>
<td>10</td>
</tr>
<tr>
<td>France</td>
<td>211.5</td>
</tr>
<tr>
<td>Germany</td>
<td>7.75</td>
</tr>
<tr>
<td>Italy</td>
<td>79.3</td>
</tr>
<tr>
<td>Japan</td>
<td>15.8</td>
</tr>
<tr>
<td>Netherlands</td>
<td>22.25</td>
</tr>
<tr>
<td>Norway</td>
<td></td>
</tr>
<tr>
<td>Singapore</td>
<td>46</td>
</tr>
<tr>
<td>Spain</td>
<td>59</td>
</tr>
<tr>
<td>Sweden</td>
<td></td>
</tr>
<tr>
<td>Sweden</td>
<td></td>
</tr>
<tr>
<td>UK</td>
<td>37</td>
</tr>
<tr>
<td>USA</td>
<td>46</td>
</tr>
<tr>
<td>Total</td>
<td>451.75</td>
</tr>
</tbody>
</table>

© 2003-2020 Stimulsoft
To create a data source based on cross-table, you should call the **New Data Source** dialog and select the **Data from Cross-tab** item. The picture below shows the **New Data Source** dialog:

![Select Type of Connection](image)

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:
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.

### 6.1.2.7 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 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.
6.1.3 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):

<table>
<thead>
<tr>
<th>CategoryID</th>
<th>CategoryName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beverages</td>
</tr>
<tr>
<td>2</td>
<td>Condiments</td>
</tr>
<tr>
<td>3</td>
<td>Confections</td>
</tr>
<tr>
<td>4</td>
<td>Dairy Products</td>
</tr>
<tr>
<td>5</td>
<td>Grains/Cereals</td>
</tr>
<tr>
<td>6</td>
<td>Meat/Poultry</td>
</tr>
<tr>
<td>7</td>
<td>Produce</td>
</tr>
<tr>
<td>8</td>
<td>Seafood</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CategoryID</th>
<th>ProductName</th>
<th>UnitPrice</th>
<th>UnitsInStock</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chai</td>
<td>18</td>
<td>39</td>
</tr>
<tr>
<td>1</td>
<td>Chang</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>Aniseed Syrup</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Chef Anton’s Cajun Seasoning</td>
<td>22</td>
<td>53</td>
</tr>
<tr>
<td>2</td>
<td>Chef Anton’s Gumbo Mix</td>
<td>21.35</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Grandma’s Boysenberry Spread</td>
<td>25</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>Uncle Bob’s Organic Dried Peas</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Northwoods Cranberry Sauce</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>Mishi Kobe Niku</td>
<td>97</td>
<td>29</td>
</tr>
<tr>
<td>8</td>
<td>Ikura</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>4</td>
<td>Queso Cabrales</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>Queso Manchego La Pastora</td>
<td>38</td>
<td>86</td>
</tr>
</tbody>
</table>

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.

6.1.3.1 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 via 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:

### 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 via 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:
6.1.3.3 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 a page of the report:
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:

As can be seen in the picture above, instead of the product names, the category names to which products are related are output.
6.1.3.4 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:
6.1.3.5 Creating Relation

It is possible to create a relation between data sources in the data dictionary. 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. Field **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 DataSource** 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.

6.1.3.5.1 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.

6.1.4 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.

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 gives an opportunity to use a variable as a parameter in the query when selecting data.

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.

6.1.4.1 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.

### 6.1.4.1.1 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.

6.1.4.1.2 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 check box 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 check box 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:

<table>
<thead>
<tr>
<th>Order</th>
<th>10251</th>
</tr>
</thead>
<tbody>
<tr>
<td>10251</td>
<td></td>
</tr>
<tr>
<td>10435</td>
<td></td>
</tr>
<tr>
<td>10553</td>
<td></td>
</tr>
<tr>
<td>10603</td>
<td></td>
</tr>
<tr>
<td>10619</td>
<td></td>
</tr>
<tr>
<td>10635</td>
<td></td>
</tr>
<tr>
<td>10648</td>
<td></td>
</tr>
<tr>
<td>10651</td>
<td></td>
</tr>
<tr>
<td>10763</td>
<td></td>
</tr>
<tr>
<td>10768</td>
<td></td>
</tr>
</tbody>
</table>

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 check box 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 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 Knäckebröd (product key = 22). In other words, all categories with the key 5 and all the products with the key 22 are displayed.

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 check box is disabled, the filter will occur by the product keys (the report shows the product which key matches). If the check box 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 check box 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.

### 6.1.4.2 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:
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.

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).
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:

<table>
<thead>
<tr>
<th>EmployeeID</th>
<th>LastName</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Tacoma</td>
<td>USA</td>
</tr>
</tbody>
</table>

### 6.1.4.3 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**:

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.

### 6.1.4.4 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:

![New Variable dialog](image)

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:
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:

![Variable Values](image)

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:

![Report Example](image)
6.1.4.5 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:

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 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.

3 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:

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:

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).
6.1.4.6 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:

<table>
<thead>
<tr>
<th>CategoryID</th>
<th>CategoryName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beverages</td>
</tr>
<tr>
<td>2</td>
<td>Condiments</td>
</tr>
<tr>
<td>3</td>
<td>Confections</td>
</tr>
<tr>
<td>4</td>
<td>Dairy Products</td>
</tr>
<tr>
<td>5</td>
<td>Grains/Cereals</td>
</tr>
<tr>
<td>6</td>
<td>Meat/Poultry</td>
</tr>
<tr>
<td>7</td>
<td>Produce</td>
</tr>
<tr>
<td>8</td>
<td>Seafood</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>CategoryID</th>
<th>CategoryName</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Condiments</td>
</tr>
</tbody>
</table>

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:

![Image of a rendered report example]

Add a filter in the **DataStream** 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:

![Image of a variable list]

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:

![Image of a variable panel]

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:

<table>
<thead>
<tr>
<th>CategoryID</th>
<th>CategoryName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beverages</td>
</tr>
<tr>
<td>2</td>
<td>Condiments</td>
</tr>
<tr>
<td>3</td>
<td>Confections</td>
</tr>
<tr>
<td>4</td>
<td>Dairy Products</td>
</tr>
<tr>
<td>5</td>
<td>Grains/Cereals</td>
</tr>
<tr>
<td>6</td>
<td>Meat/Poultry</td>
</tr>
<tr>
<td>7</td>
<td>Produce</td>
</tr>
<tr>
<td>8</td>
<td>Seafood</td>
</tr>
</tbody>
</table>

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:
The New Item button. Creates a new item with the field in which to specify a key;
2 The Remove buttons. Remove the item to which they belong. Each item in the list has such a button.
3 The Remove All button. Removes all items from the list;
4 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:

<table>
<thead>
<tr>
<th>CategoryID</th>
<th>CategoryName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beverages</td>
</tr>
<tr>
<td>3</td>
<td>Confections</td>
</tr>
<tr>
<td>6</td>
<td>Grains/Cereals</td>
</tr>
<tr>
<td>7</td>
<td>Produce</td>
</tr>
</tbody>
</table>

The Reset button, in this case, resets the current list of values to the original one.

6.1.5 Panel Setup

The panel (see the picture below) contains controls that provide an opportunity to change auxiliary parameters of the data dictionary.

1. Create Field on Double Click
2. Create Label
3. Use Aliases

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. 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.

6.1.6 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** property, 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
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;

- **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;

### 6.1.7 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. 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.
The panel **Columns** has three fields. These fields show a list of columns, their properties and description.

### Information

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.

### 6.1.8 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:
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.

<table>
<thead>
<tr>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hyperlink:</strong></td>
</tr>
</tbody>
</table>

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. 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.

7 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.

7.1 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:

![](image1)

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:

![](image2)

In any case a text written in a right-to-left language will be output right to left.

7.2 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:

![](image3)

If the RightToLeft property is set to true, then a text is output from left to right. The
The RightToLeft property of the Text in Cells component works the same way with all languages. So a text characters and symbols 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.

7.3 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:
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**:

<table>
<thead>
<tr>
<th>Products</th>
<th>Category Name</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bev. Cms</td>
<td>Confections</td>
<td>Total</td>
</tr>
<tr>
<td>Brazil</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Canada</td>
<td>113</td>
<td>136</td>
</tr>
<tr>
<td>Denmark</td>
<td>87</td>
<td>100</td>
</tr>
<tr>
<td>Finland</td>
<td>56</td>
<td>52</td>
</tr>
<tr>
<td>France</td>
<td>88</td>
<td>52</td>
</tr>
<tr>
<td>Germany</td>
<td>125</td>
<td>10</td>
</tr>
<tr>
<td>Italy</td>
<td>23</td>
<td>26</td>
</tr>
<tr>
<td>Japan</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>Netherl.</td>
<td>17</td>
<td>164</td>
</tr>
<tr>
<td>Norway</td>
<td>17</td>
<td>164</td>
</tr>
<tr>
<td>Span</td>
<td>108</td>
<td>108</td>
</tr>
<tr>
<td>Sweden</td>
<td>65</td>
<td>224</td>
</tr>
<tr>
<td>UK</td>
<td>74</td>
<td>143</td>
</tr>
<tr>
<td>USA</td>
<td>103</td>
<td>665</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>559</strong></td>
<td><strong>3119</strong></td>
</tr>
</tbody>
</table>

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.
7.4 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.

7.5 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:
7.6 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:
"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:
Report Viewer is a tool for viewing, printing, exporting reports and dashboards.
8.1 Reports

This chapter will cover the following:
- Viewer Structure;
- Viewer Toolbar;
- Viewer Status Bar;
- Displaying Mode;
- Search Panel;
- Resources Panel;
- Sending Report via E-Mail;
- Exporting Report;
- Shortcuts.

Viewer Structure
On the picture below you may find the basic elements of the report viewer.
This panel contains menus which have the basic control commands of the report viewer.
2 The tree of bookmarks of the output report. Using these bookmarks you can navigate through structure elements of a report.
3 The area where the report is shown.
4 The report thumbnails panel. Decreased copies of report pages are shown on this panel. The panel is used to quickly navigate throughout a report.
5 The status bar of viewer.
Toolbar

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.

Information

A report file may contain only a report; only a dashboard; both a report and a dashboard.

- If the report file contains only the report, then this report will be rendered and displayed in the report viewer.
- If the report file contains only the dashboard, then the report viewer will switch to the view mode of the dashboard, with the display of this panel.
- If the report file contains both the report and the dashboard, then the report viewer will switch to the view mode of the dashboard, with the display of this panel. To view the report, go to the tab with the name of the report page.

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 the basic parameters of the rendered report.
7. The Resources button. With this button, you can enable or disable the resource panel in the viewer. If the report does not contain resources that can be displayed in the viewer, this button will be disabled.
8. Show/hide the thumbnails of reports.
9. Enable the search panel.
Run the full-screen mode of report showing. To exit this mode, you can use the Esc or Alt+F4 hot keys.

Change zoom of the report to display only one full page. More than one page by the width can be output.

The Two Pages button is used the zoom the report so that two pages could fit the viewer window by height.

The Multiple Pages button is used to call a menu in which you can select the number of pages by width and height.

Change the zoom of the report to fit the page width to the screen width.

The button is used to open the user manual page.

**Information**

In addition to the above commands, other buttons may be displayed on the toolbar:

- If the report contains editable fields, the **Editor** button will be displayed in the viewer. When you click this button, the edit mode of the report components will be activated.
- Commands are used to create, edit, delete, and customize pages of the rendered report.
- The **Close** button is used to close the preview tab or viewer window.
- The button is used to enable the Dot-matrix mode.

**Status Bar**

The status panel contains navigation controls in the report, report display modes, and its zoom.

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 you click on it, then it is possible to indicate the number of a page that should be the current one.
Set the next page of a report as the current one.
Set the last page of a report as the current page.
The buttons are used to switch display modes for pages.
The zoom control for the report.

Displaying Mode
The viewer for WinForms supports three modes of viewing pages:
- 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.

Search Panel
The search panel is used to search for 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.

- Close the search panel.
- The field to put a text that should be found.
- The button to run the search.
- The button to run the search.
- If the flag is set, then the search will be repeated considering the case.
- If the flag is set, then the search will be done considering the whole word.

Resources Panel
You can display some resources which were added to the report in a separate panel in the viewer. To do this, when adding a resource in the report designer, select Available in the Viewer option. Then, click the Resources button in the viewer to display a panel with these resources.

Each resource in this panel can be viewed or saved.

**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.

**Exporting Report**

Any rendered report can be converted to various formats, for example, to PDF, Excel, Word, etc. Report export has several stages.

- Click the Save button in the viewer;
- Select the type of file you want to convert the report into;
- Set export settings;
- Save the converted file.

See the chapter [Exports](#) to get more information about converting a report to other types of files.

**Shortcuts**

The list of keyboard shortcuts in the report viewer is as follows:

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+P</td>
<td>Print a report</td>
</tr>
</tbody>
</table>
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
Esc | The button is used to exit the Full Screen mode.
Alt+F4 | The buttons are used to close the window, including the full-screen view.

8.1.1 Dot-Matrix

The Dot-Matrix viewer is designed to preview the report before printing it on a 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.

**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.

**Bar Options**

The Options bar is grouped, and each group is located on a separate tab. The picture below shows the options bar:
The **Kill Space Lines** option removes empty rows in the text.

The **Kill Space Graph Lines** option deletes the rows that contain only the "vertical line" pseudographics characters.

The **Put Feed Page Code** option inserts the FormFeed symbol on the bottom of each page.

The **Draw Border** option draws the borders of the objects of the selected type.

The **Cut Long Lines** option cuts long lines of the text that is out of bounds of the text component.

- **Simple** border is drawn with + - | symbols and will be saved and printed in any encoding;
- **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 using the line spacing 1,0.

9 Scale by the X-axis (Zoom X), by page width.
10 Scale by the Y-axis (Zoom Y), by page height.
11 Encoding is the encoding of the displayed text.
12 The **Auto Refresh** parameter automatically updates the rendered report if there are any changes applied to the settings.
13 The **Refresh** button is used to update the rendered report manually.

### 8.1.2 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 edges of the objects may be intersected. Such intersections may lead to the 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 not to overlap objects. Examples of components overlapping are shown in 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 a 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.

**Information**

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 a 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 not to 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.

**Information**

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 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 reports without
preview the report generator prints a report without preview. When choosing the method of printing characteristics of each method should be considered.

**Information**

The **StiWebViewer** component cannot control page parameters (page size, page orientation, page margins) when printing using the 2 and 3 methods. All parameters are controlled with the browser.

### 8.2 Dashboards

The report viewer in the view mode for dashboard panels differs from the report viewer in the report view mode. The dashboard panel and its elements are stretched to the entire area of the viewer. Elements of the dashboard panel - **Combo Box**, **Date Picker**, **Tree View Box** are stretched only in width. Besides, the dashboard and its elements contain various control buttons.
**Control buttons of the dashboard**

Controls of the dashboard are located in the upper right corner above the dashboard panel.
1. The **Refresh** button is used to update the dashboard.
2. The **Open** button is used to open a previously saved report file.

**Information**

A report file may contain the following: only a report, only a dashboard, and both a report and a dashboard.

- If the report file contains only the report, then this report will be rendered and displayed in the report viewer.
If the report file contains only the dashboard, then the report viewer will switch to the view mode of the dashboard, with the display of this panel.

If the report file contains a report and dashboard, then the report viewer will switch to the view mode of the dashboard with this panel displayed. To view a report, in the report viewer window, go to the tab with the name of the report template page.

The **Edit** button is used to change the rendered dashboard in the report designer. You should know that this can only be done if, before rendering the dashboard panel, the **Calculation Mode** property of the template is set to **Interpretation**.

The **Full Screen** button is used to view the dashboard in the full-screen mode. To exit this mode, you can use the Esc or Alt+F4 hot keys.

The **More Options** button invokes a menu with various commands for controlling the dashboard panel. For example, these are commands to convert the dashboard to other files - PDF, Excel, and PNG.

**Element controls**

The control buttons of the dashboard elements are located in the upper right corner of the area of this element and are displayed when you hover over or select this element.
This button is used to enable or disable the filtering mode for several segments.

- If this button is enabled, then to filter data, you can select several segments on one element of the dashboard.
- If the button is disabled, then when selecting the next segment, the previous filter will be reset.

For example, when filtering by map, in the single mode, when you click on each segment, other elements of the dashboard panel will only display related data with the current map segment. In the filtering mode by several segments, other elements of the dashboard will display the associated data with all selected segments of the map.

This button is used to delete all filters. When clicking it, all filters of the current element of the dashboard will be deleted.
The filtering control buttons are present only in the dashboard elements that have active segments for filtering data - Table, Chart, and Region Map.

1. This button is used to switch to the lower level of the drill-down of an element. This button is only displayed if the drill-down mode for the element is enabled.
2. This button is used to go to the upper level of the drill-down of the element. This button is only displayed if the drill-down mode for the element is enabled.
3. The **Full Screen** button is used to display a specific element of the dashboard over the entire area of the viewer.
4. The **Save** button invokes a menu with various control commands for a specific element of the dashboard. For example, commands to convert the current element of the dashboard to other files - PDF, Excel, and PNG.

**Information**

You should know that you can enable or disable displaying of the **Save** and **Full Screen** buttons in the viewer or on the preview tab. Select the element in the report designer, click the **Interaction** button on the **Home** tab of the Ribbon panel, and enable (or disable) the check box for the parameters if you want to display (or not display) the buttons of the element.

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.

<table>
<thead>
<tr>
<th>Export Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDF (Portable Document Format)</td>
</tr>
<tr>
<td>XPS (XML Paper Specification)</td>
</tr>
<tr>
<td>HTML (HyperText Markup Language)</td>
</tr>
<tr>
<td>HTML5 (HyperText Markup Language)</td>
</tr>
<tr>
<td>MHTML (MIME HTML)</td>
</tr>
<tr>
<td>TXT (Text File)</td>
</tr>
<tr>
<td>RTF (Rich Text)</td>
</tr>
<tr>
<td>Microsoft Word 2007/2010</td>
</tr>
<tr>
<td>ODT (Open Document Text)</td>
</tr>
<tr>
<td>Microsoft Excel</td>
</tr>
<tr>
<td>Microsoft Excel Xml</td>
</tr>
<tr>
<td>Microsoft Excel 2007/2010</td>
</tr>
<tr>
<td>Microsoft Power Point 2007/2010</td>
</tr>
<tr>
<td>ODS (Open Document Spreadsheet)</td>
</tr>
<tr>
<td>CSV (Comma Separated Values)</td>
</tr>
<tr>
<td>DBF (DataBase File)</td>
</tr>
<tr>
<td>XML (extensible Markup Language)</td>
</tr>
<tr>
<td>DIF (Data Interchange Format)</td>
</tr>
<tr>
<td>SYLK (Symbolic Link)</td>
</tr>
<tr>
<td>BMP (Bitmap)</td>
</tr>
<tr>
<td>GIF (Graphics Interchange Format)</td>
</tr>
<tr>
<td>PNG (Portable Network Graphics)</td>
</tr>
</tbody>
</table>
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**


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 images, 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
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.

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.

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](#).

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.

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.

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.

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.


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 in 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-primitive. 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
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** 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;
- **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
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:
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.
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.

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.

The checkbox **Put Feed Page Code** provides an opportunity to select the end of the page with a special character.

The checkbox **Draw Border** enables/disables drawing borders of components with graphic symbols.

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.

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.

The option **Zoom** provides the ability to set the report zoom horizontally and vertically.

The option **Encoding** provides the ability to set the text encoding of the report after exporting.

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;
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.

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.

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#PageNumber#</td>
<td>The number of the current page (PAGE)</td>
</tr>
<tr>
<td>#TotalPageCount#</td>
<td>Total number of pages in the document (NUMPAGES)</td>
</tr>
<tr>
<td>#PageRef Bookmark#</td>
<td>The number of pages on what the bookmark is placed (PAGEREF)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtfparagraph</td>
<td>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.</td>
</tr>
<tr>
<td>rtfnewpage</td>
<td>The page break is put before the text box</td>
</tr>
</tbody>
</table>

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
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. 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.

6 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.

7 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.

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:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#PageNumber#</td>
<td>The number of the current page (PAGE)</td>
</tr>
<tr>
<td>#TotalPageCount#</td>
<td>Total number of pages in a document (NUMPAGES)</td>
</tr>
</tbody>
</table>
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
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.
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.

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*

![Microsoft Excel File - Export Settings](image)
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 is 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.

1 The checkbox **All** enables processing of all report pages.
2 The checkbox **Current Page** enables processing only the current (selected) report
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 text box 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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>DBF data type (default size)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>Numeric (15 : 0)</td>
<td>Numeric</td>
</tr>
<tr>
<td>long</td>
<td>Numeric (25 : 0)</td>
<td>Numeric</td>
</tr>
<tr>
<td>float</td>
<td>Numeric (15 : 5)</td>
<td>Decimal</td>
</tr>
<tr>
<td>double</td>
<td>Numeric (20 : 10)</td>
<td>Decimal</td>
</tr>
<tr>
<td>string</td>
<td>Character (auto)</td>
<td>Text</td>
</tr>
<tr>
<td>date</td>
<td>Date (8)</td>
<td>Date</td>
</tr>
</tbody>
</table>

Sample of using elements are shown in the table below.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string : 25</td>
<td>set the column width (25 characters) and cuts all long strings</td>
</tr>
<tr>
<td>float</td>
<td>converts decimal digit with the length 15 characters, 5 characters after comma</td>
</tr>
<tr>
<td>float :10</td>
<td>converts decimal digit with the length 10 characters, 5 characters after comma</td>
</tr>
<tr>
<td>float :10 : 2</td>
<td>converts decimal digit with the length 10 characters, 2 characters after comma</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>int :10 : 2</td>
<td>converts integer digit with the length 10 characters; the second parameter is ignored</td>
</tr>
</tbody>
</table>

⚠️ **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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>Numeric</td>
</tr>
<tr>
<td>long</td>
<td>Numeric</td>
</tr>
<tr>
<td>float</td>
<td>Decimal</td>
</tr>
<tr>
<td>double</td>
<td>Decimal</td>
</tr>
<tr>
<td>string</td>
<td>Text</td>
</tr>
<tr>
<td>date</td>
<td>Date</td>
</tr>
</tbody>
</table>

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*
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 $65535 \times 65535$ 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 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.
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.

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.

### 9.10 Export Dashboard

When viewing the dashboard in the report viewer, you can convert its elements to **PDF**, **Excel files**, as well as **image** files such as BMP, GIF, PNG, TIFF, JPEG, PCX, EMF, SVG, and SVGZ. In addition, you can save the dashboard as a **report snapshot**.
To export the dashboard, click the More Options button and select the appropriate command. If you want to convert only a specific element, click the Save button on that element and select the file type.

**Information**

For the Table element, export formats to **CSV, DBF, XML, JSON, DIF, and SYLK** are also available. To do this, select the Data command in the Save menu. Then, in the export settings, select the file type into which you want to convert the current element.
After selecting the export format, the export options dialog will be called. The parameters may vary depending on the type. Let's consider export settings in more detail.

**Report snapshot**
The **Report Snapshot** command is used to save the dashboard with the current data to the .mrt file. In this case, the created data sources will be embedded into the report as resources. You can open this report both in the report designer and in the report viewer.

**PDF Export Settings**
Export settings for the dashboard or its elements when converting to a PDF file.

- **Paper Size** option allows you to select the page size in the PDF document.
- **Orientation** parameter is used to select the page orientation in the PDF file - Portrait or Landscape.
- **Image Quality** option is used to change the quality of images.
- **Open After Export** parameter allows you to open the exported document after the export process is completed.

**Excel Export Settings**
Export settings for the dashboard or its elements when converting to an Excel file.
1. The **Image Quality** option allows you to change the quality of images.
2. The **Open After Export** parameter allows you to open the output document after the export process is completed.

**Information**

When exporting the Table element to Excel, the **Export Data Only** parameter will also be available in the export settings. This option is used to convert only the values of these elements, without headers and totals.

**Image Export Settings**

Export settings for the dashboard panel or its elements when converting to an image file.

1. The **Image Type** parameter is used to determine the type of image into which the report will be converted - BMP, GIF, PNG, TIFF, JPEG, PCX, EMF, SVG, SVGZ.
2. The **Scale** parameter is used to change the number of pixels per inch.
3. The **Open After Export** parameter allows you to open the exported document after the export process is completed.
Export Settings of Data
Export settings for the Table element when converting it to a data file.

Data File - Export Settings

1. Data Type
   - JSON

2. Open After Export

- **Data Type** parameter is used to specify the type of data file into which the report will be converted - CSV, DBF, XML, JSON, DIF, SYLK.
- **Open After Export** parameter allows you to open the exported document after the export process is completed.

Information

You should know that this type of export is available only for the Table element. However, if you need to convert the values of all the fields of an element into a data file, you may change the type of this element to the Table element, and then, export the Table element to the data file.