REPORT WRITER
USER GUIDE

efi | OPTITEX

Contents

1. Introduction

The Optitex Marker and PDS applications provide you with wide range of reporting features allowing you to generate various reports containing information about your markers or patterns and also generate a tech pack for the current style including essential info of the 2D and 3D. A reporting engine (Report Writer) allows you to generate reports or tech pack based on customizable templates, providing you with the possibility to design your own report / tech pack layout as well as decide on the data appearing in the report / tech pack.

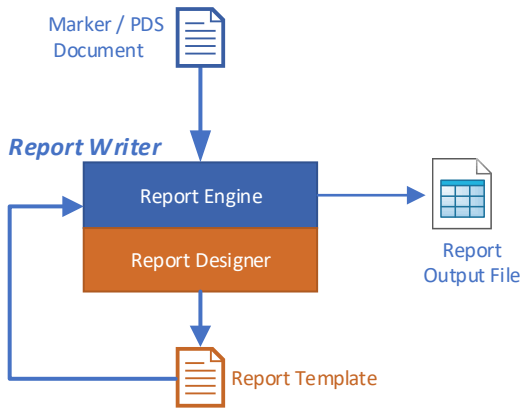
The template creation is a one-time action. Once the template is created it can be used for quick and easy report / tech pack generation.

Once the report or tech pack is generated, the Report Writer provides you with the possibility to save it in following formats:

- PDF – Adobe Portable Document Format
- HTML – Web Page
- MHT – Single File Web Page
- RTF – Rich Text Format
- DOCX – Microsoft Word Document
- XLS, XLSX – Microsoft Excel Workbook
- CSV – Comma-Separated Values Text
- TXT – Plain Text
- BMP, GIF, JPEG, PNG, TIFF, EMF, WMF Image Files

The Report Writer was developed in conjunction with DevExpress XtraReports; therefore, all customizations are according to DevExpress's UI. This includes the Report Designer and Preview Mode. For more information, see https://documentation.devexpress.com/XtraReports/2162/Reporting.

The reporting functionality available in PDS and Marker is represented by two modules: Report Engine and Report Designer. The Report Engine is dedicated for the final report output generation; the Report Designer is a tool dedicated for the report template creation.



This document is intended to provide you with the description of the template creation procedure and does not cover the report generation flow. For a description of the report generation flow, refer to the Optitex Online Help for corresponding information available for PDS and Marker.

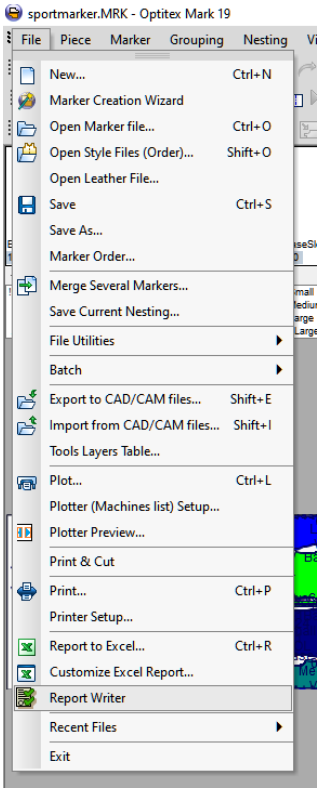
2. Getting Started with Report Designer

This topic provides you with the description of the Report Designer launch.

2.1. Launching Report Designer in Marker

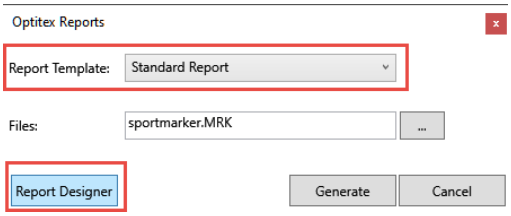
To run the Report Designer tool in the Marker application, perform the following steps:

1. In the Marker application, choose the Report Writer command from the File menu.

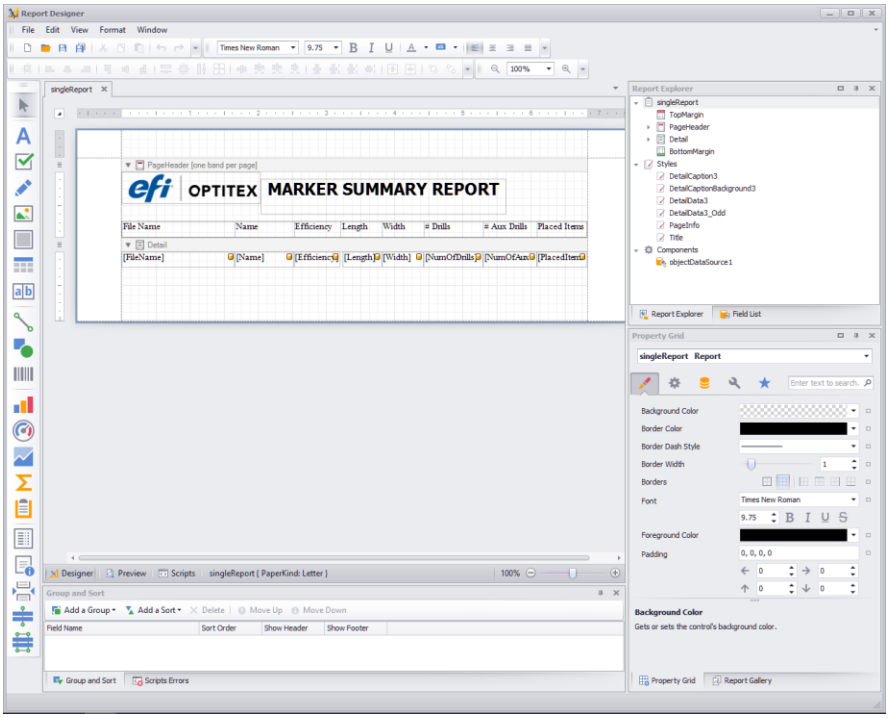


The Optitex Reports dialog box will be displayed.

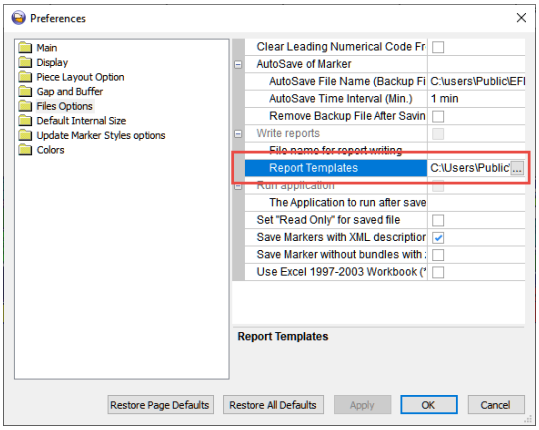
2. To edit an existing template with the Report Designer, choose the report template in the combo box and click the Report Designer button to launch the Report Designer tool.



The Report Designer will be launched; the selected report template will be loaded.



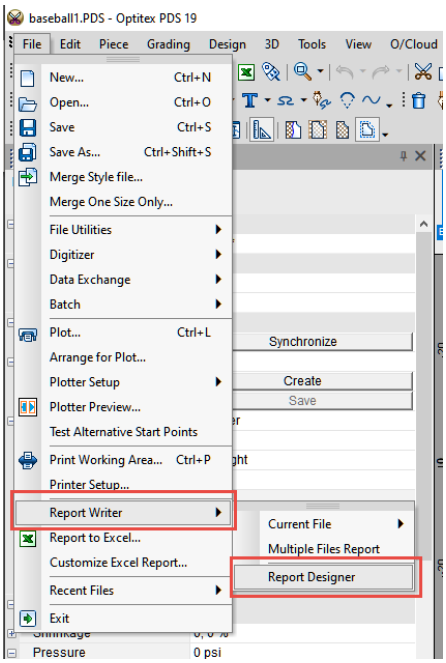
The Report Template combo box mentioned above provides you with the list of report templates (*.repx) available in the Report Templates folder. This is defined in Preferences as shown below.



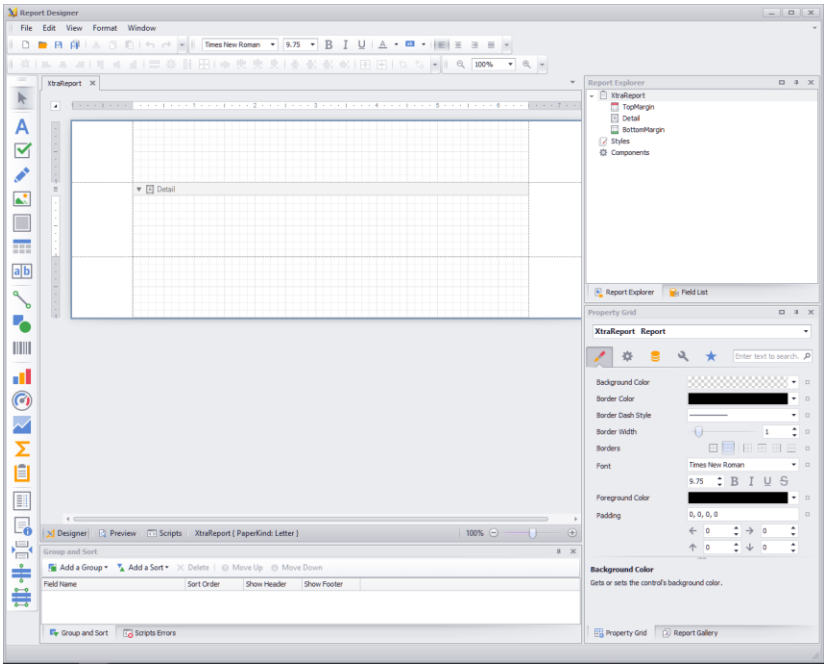
In Marker, the Report Designer session always starts with a template pre-loaded. The Report Designer allows you to edit an existing template or create a new one.

2.2. Launching Report Designer in PDS

1. To run the Report Designer tool in PDS, choose the Report Designer command from the File-Report Writer menu as shown below.

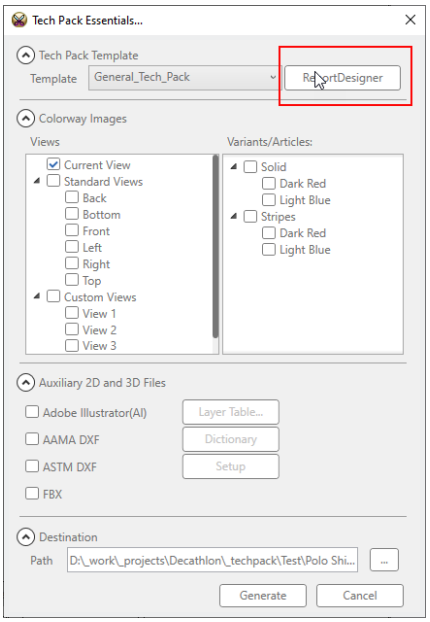


The Report Designer will be launched with a new, empty template.

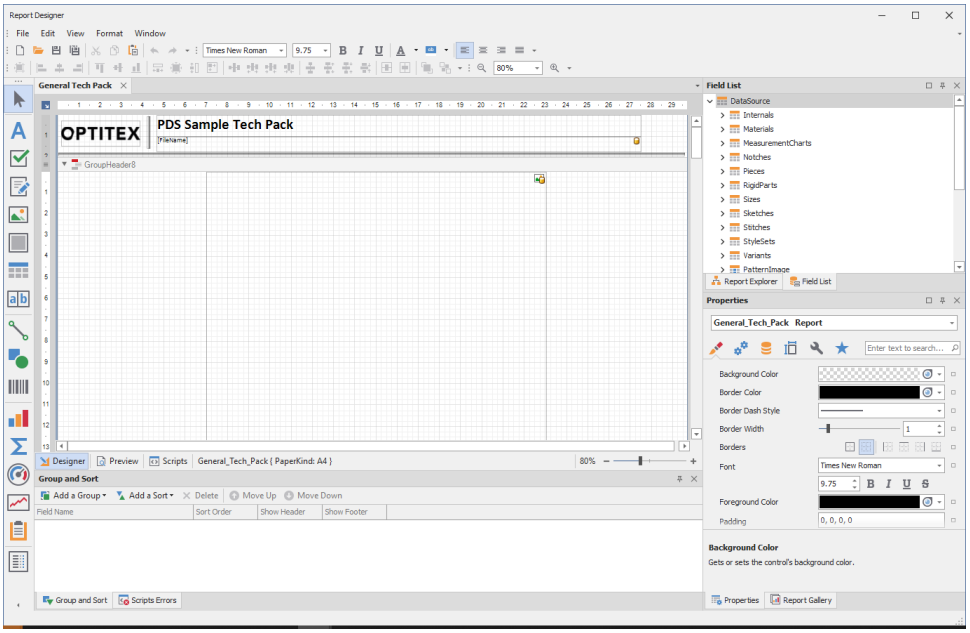


2.3. Launching Report Designer from the Tech Pack Essentials environment

1. To run the Report Designer tool in tech pack environment, choose the Report Designer command from the Tech Pack Essentials dialog box as shown below (the Tech Pack Essentials dialog box is available with the File > Tech Pack Essentials command in PDS).



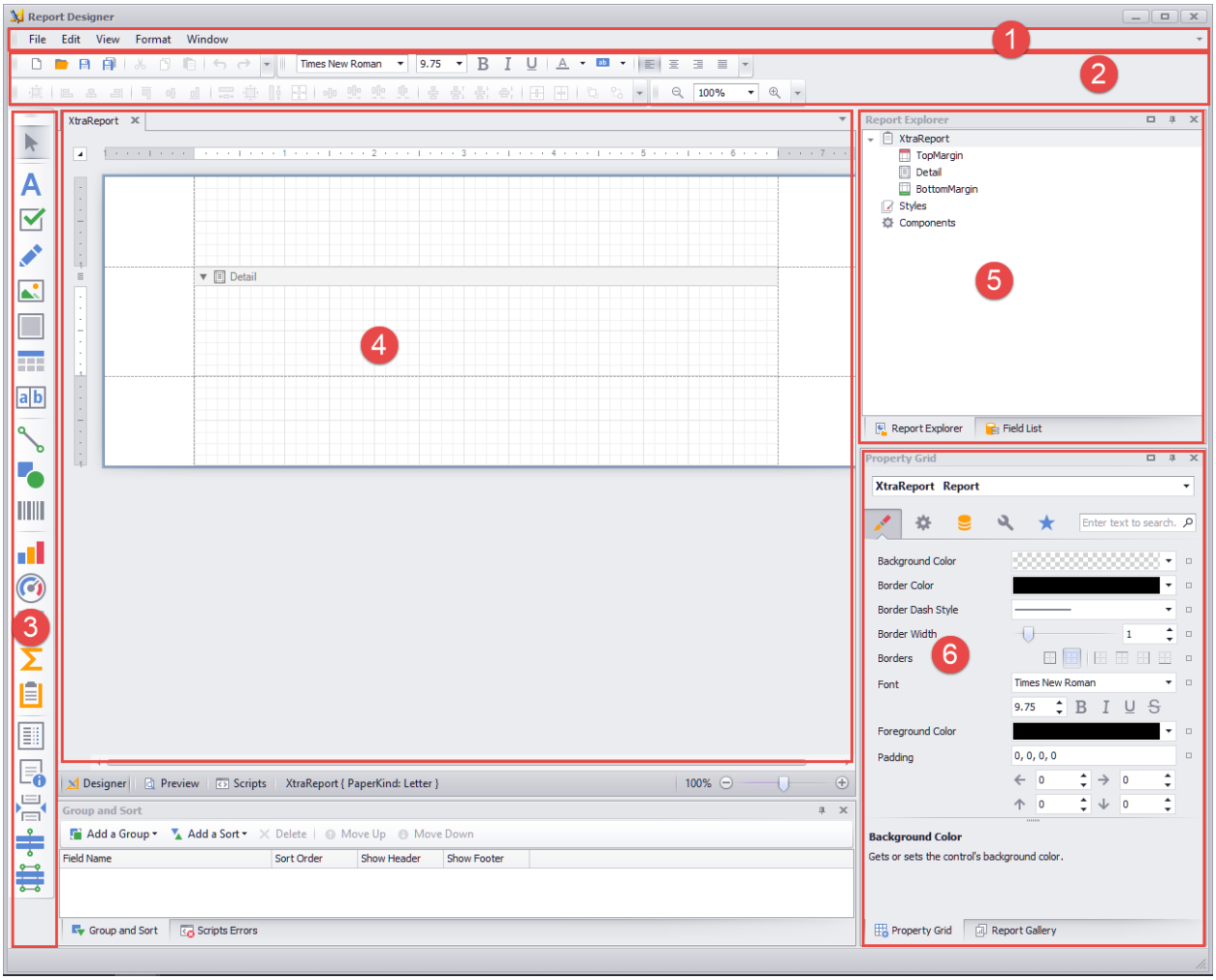
The Report Designer will be launched with a template, currently selected within the Template combo box.



3. Report Designer User Interface

This topic provides you with a description of the Report Designer user interface.

The Report Designer tool provides you with following major user interface elements:

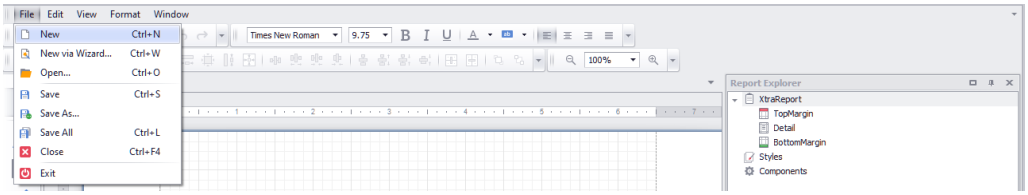


1. **Application menu**. The application menu is composed of several drop-down buttons that display a menu containing commands. These commands expose general functionality related either to a complete template document (such as New, Open, Save) or to the system itself (View, Windows and Exit commands). For more information about the Application menu, refer to topic 3.1.
2. **Toolbars**. The toolbars area provides you with the possibility to display various toolbars containing command buttons allowing you to run most commonly-used operations such as text formatting or file operations. For more information about toolbars, refer to topic 3.2
3. **Standard Controls**. This toolbar provides you with the list of controls available to be used within the report template. For more information about standard controls, refer to topic 3.2.6.
4. **Design panel**. The Design panel provides you with the visual designer environment for the currently opened report template. The Design panel allows you to create a layout of the current report template visually, and fill it in with the required controls.
5. **Report Explorer / Fields list**. The Report Explorer tool provides you with the tree representing the structure of the current report template. The tree provides you with the list of report sections with underlying controls used within these sections. For more information about the report template structure, refer to topic 4.2. The Fields list provides you with the list of variables available

to be placed within the design panel. The list contains variables exposed by the Marker/PDS application, as well as user-defined variables. For more information about variables and their use, refer to topic 4.1.

6. **Property Grid**. The Property grid control provides you with the list of all properties available for a control currently selected within the design panel. The appearance of properties is determined according to the type of the selected control; i.e. the Properties grid provides you with only those properties that are relevant for the currently selected control.

3.1. Application Menu



The application menu provides you with commands that expose general functionality related either to a complete template document (such as New, Open, Save) or to the system itself (View, Windows and Exit commands). The following commands are available through the application menu:

3.1.1 File

- **New**. Provides you with the possibility to create a new template report.
- **New via Wizard**. Provides you with the possibility to create a new template report using a guided wizard-based flow.
- **Open**. Provides you with the possibility to open an existing template to modify it.
- **Save**. Provides you with the possibility to save the current report template.
- **Save As**. Provides you with the possibility to save the current report template in a new location or with the new name.
- **Close**. Closes the currently opened report template
- **Exit**. Closes the Report Designer.

3.1.2 Edit

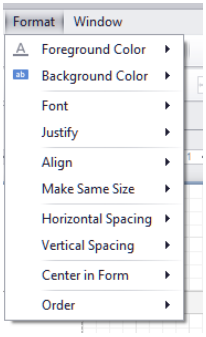
- **Undo**. Provides you with the possibility to cancel the last action performed.
- **Redo**. Once an action is cancelled via the Undo command, the Redo option allows you to redo this action again.
- **Cut** / **Copy** / **Paste**. Provide you with the possibility to perform standard operations using the clipboard.
- **Delete**. Provides you with the possibility to remove the control (or multiple controls) selected within the design panel.
- **Select All**. Provides you with the possibility to automatically select all objects appearing within the design panel.

3.1.3 View

- **Toolbars**. This sub-menu allows you to control the visibility of toolbars: Toolbar, Formatting toolbar, Layout toolbar, Status toolbar, Zoom toolbar and Standard Controls (see topics 3.2.1-3.2.6 for more information about toolbars).
- **Windows**. This sub-menu allows you to control the visibility of windows: Report Explorer (see topic 3.3.1), Field list (see topic 3.3.2) , Property Grid (see topic 3.3.3), Report Gallery (allowing you to store and reuse reports and their elements), Group and Sort and Script Errors.

3.1.4 Format

This sub-menu provides you with the possibility to control layout and formatting of controls placed within the report template in the design panel. The following commands are available in the sub-menu:



- **Foreground color**. This option provides you with the color picker interface to define the foreground color for the selected control. The possibility to customize the foreground color of the control is also available via the Property Grid (see topic 3.3.3).
- **Background color**. This option provides you with the color picker interface to define the background color for the selected control. The possibility to customize the background color of the control is also available via the Property Grid (see topic 3.3.3).
- **Font**. This command allows you to apply a basic effect to the text available within the currently selected control. You can make the text bold, italic and underlined.
- **Justify**. This option allows you to control the justification of the text app-earing within the currently selected control. The control is performed via standard Left, Right, Center and Justify options.
- **Align**. This option allows you to align multiple selected controls in a regular manner. You can align objects in horizontal (left, center, right) or vertical (top, center, bottom) directions. You can also align selected controls with the grid appearing in the design panel.
- **Make same size**. This option provides you with the possibility to adjust the size of selected controls to make it the same. You can adjust controls size in horizontal (width), vertical (height) or both directions. You can also adjust the control size to the grid.
- **Horizontal / Vertical spacing**. These options allow you to control the spacing (distance between two neighbor controls). You can increase or decrease spacing between selected controls or make it equal. You can also remove the spacing between controls to make them adjacent to each other. The spacing can be controlled in either a horizontal or vertical direction.
- **Center in Form**. This option allows you to place the selected control in the center of the form appearing within the design panel. The selected control can be centered in both directions: horizontal and vertical.
- **Order**. This option allows you to control the Z-order of selected controls. You can move the currently selected control to be in front of other objects or move it back to be behind other objects.

3.1.5 Window

This sub-menu provides you with the standard functionality to control over template windows appearing within the design panel. The report designer application provides you with the possibility to open multiple templates simultaneously. Opened templates appear within different templates windows available within the design panel. The **Window** sub-menu allows you to switch between these template windows as well as control over windows layout.

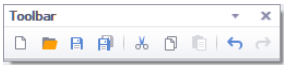
3.2. Toolbars

The Report Designer tool provides you with following toolbars:

- Toolbar
- Formatting toolbar
- Layout toolbar
- Status toolbar
- Zoom toolbar
- Standard Controls

The toolbar appearance is controlled via the View-Toolbars command in the Application menu (see topic 3.1.3)

3.2.1 Toolbar



The Toolbar provides you with the general commonly used commands available in the File and Edit menu as well. Such commands as New, Open, Save, Cut, Copy, Paste etc. are available.

3.2.2 Formatting toolbar

The Formatting toolbar provides you with standard controls that allow you to customize the formatting of the text appearing within the selected control. With the Formatting toolbar, you can specify the font type and font size of the text as well as apply the appearance style like bold or italic. In addition, the toolbar allows you to specify the foreground/background color and justification.

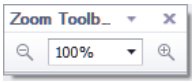
3.2.3 Layout toolbar

The Layout toolbar provides you with the various tools allowing you to control the layout of controls in the template design area. The toolbar provides you with control over the alignment, controls size, spacing and Z-order.

3.2.4 Status toolbar

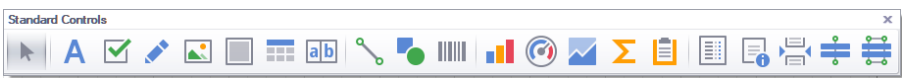
The Status bar appears at the bottom of the Report Designer window, providing you with various notifications.

3.2.5 Zoom toolbar



The Zoom toolbar allows you to control the zoom available within the design panel. The control over the zoom is performed in a regular manner. In addition, the display zoom can be controlled with the Ctrl+Mouse wheel.

3.2.6 Standard Controls

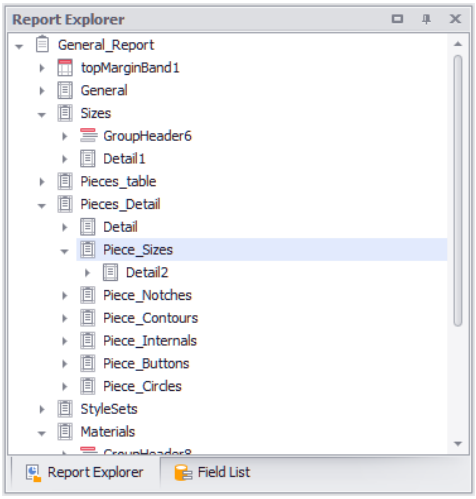


The Standard Controls toolbar provides you with the various tools available for creation of controls in the design panel. For description of commonly used controls and for the procedure of their placement, refer to topic 4.3.

3.3. Windows

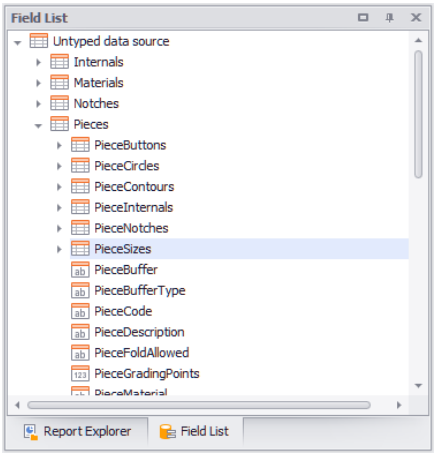
This topic provides you with the description of different windows used for the various data displays within the Report Designer.

3.3.1 Report Explorer



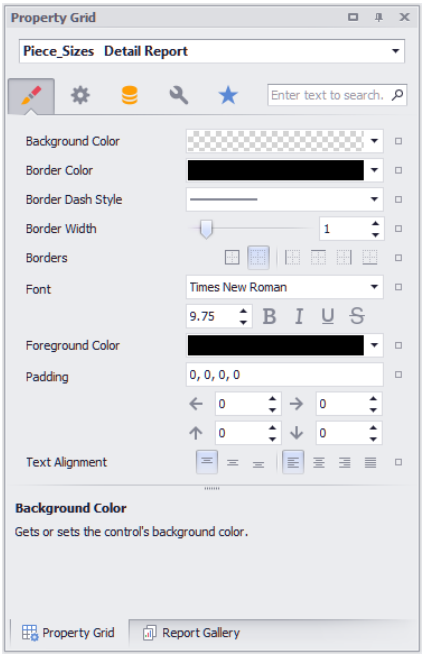
The Report Explorer panel reflects a report structure in a tree-like form and provides access to components assigned to a report (such as its bands or controls). For more information about the report template structure, refer to topic 4.2.

3.3.2 Field list



The Field List panel displays the schema of a report's data sources. It provides you with the structured list of all variables available to be used within the current report. For more information about variables available for the report creations and their structure, refer to topics 4.1 and 4.2.

3.3.3 Property Grid



The Property Grid panel provides you with the possibility to get an access and customize report/report element settings. Each time a certain object is selected within the design panel, the Property Grid is updated to expose the list of properties relevant for the selected object. The appearance, behavior data access and other properties can be accessed and customized via the Property grid.

4. Report Designer Basics

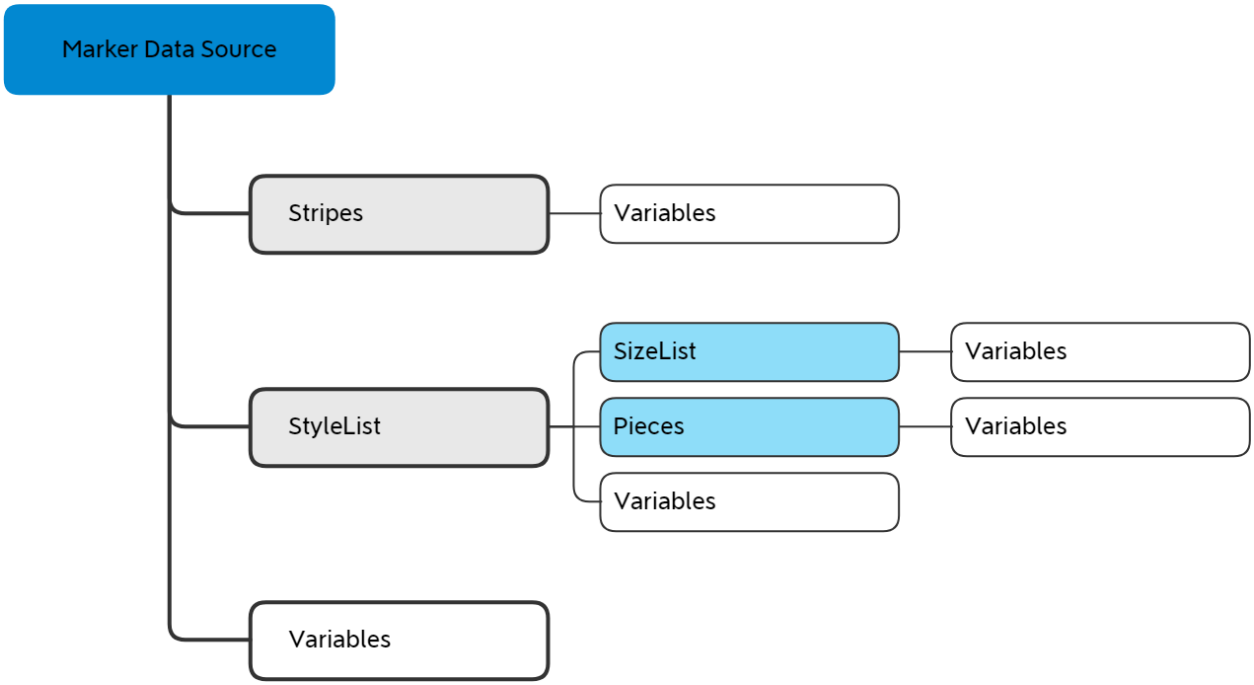
This topic provides you with the description of data sources and report templates layout basics.

4.1. Data Source Fundamentals

The data source is one of the central concepts used throughout this document. The data source for the report generation is a virtual object providing you with the hierarchically-structured data related to the Marker or PDS file for which the report or tech pack is generated.

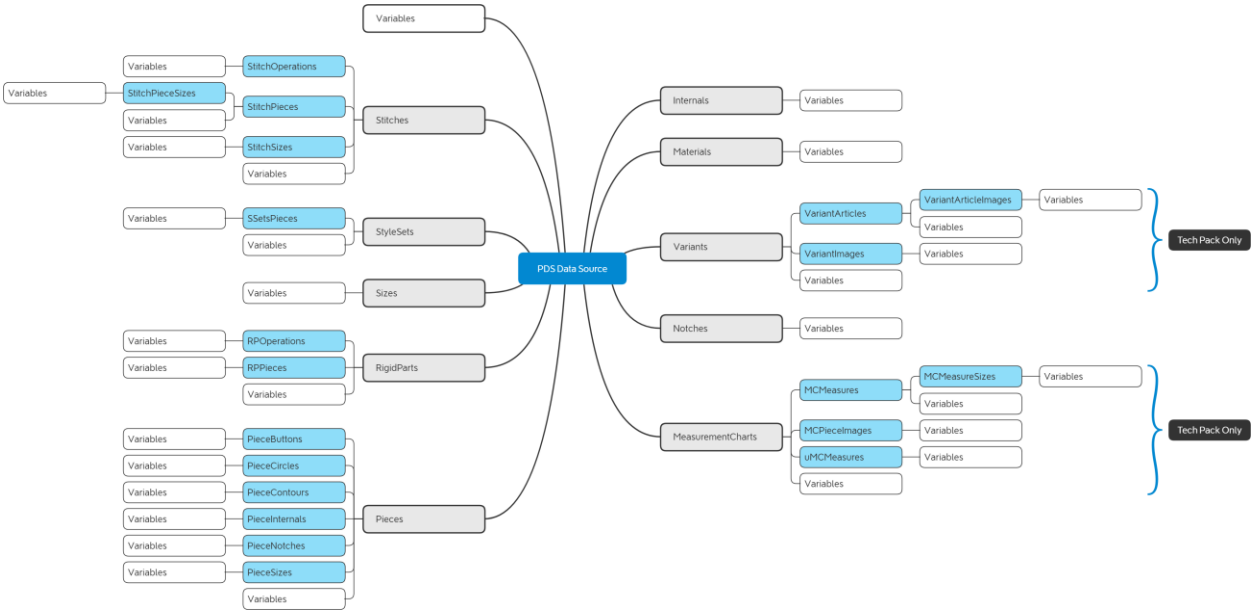
The data source provides you with various properties reflecting various parameters utilized within the PDS and Marker applications. Each time the report (tech pack) is generated for a certain PDS or Marker file, the new data source object is dynamically created and filled in with the actual data.

The scheme below provides you with the data source structure available for Marker reports.



The root space of the Marker data source provides you with the sequence of nested lists intended to represent hierarchical multi-instance data. For example, the StyleList list contains sub-objects representing styles participating within the current Marker; every such style is accompanied with a set of style properties. The StyleList also provides you with the nested SizeList which contains all sizes related to the certain style and size properties and Pieces list containing piece data. In addition, the root space of the Marker data source provides you with multiple general variables describing various properties of the Marker file, like table length, width, efficiency and so on.

The scheme below provides you with the data source structure available for PDS reports / tech pack.



The root space object of the PDS data source provides you with the sequence of nested lists intended to represent hierarchical multi-instance data. There are Internals, Materials, MeasurementCharts, Notches, Pieces, Stitches, RigidParts, Sizes, StyleSets and Variants lists intended to hold corresponding objects. For example, the Pieces list contains pieces available within the current PDS file; every piece is accompanied with corresponding piece properties. The Pieces list also provides you with nested PieceButtons, PieceCircles, PieceContours etc lists containing corresponding properties of various objects belonging to pieces.

In addition to it, the root space of the PDS data source provides you with multiple general variables describing various properties of the PDS file, like style name, number of pieces, number of sizes etc.

4.1.1 Marker Variables

The table below provides you with the list of variables available for the Marker report template creation. The table provides you also with the information of hierarchy of variables as well as their description.

4.1.1.1 General Variables

These variables are available in the root space of the data source. These variables are evaluated for every Marker file for which the report is generated.

Marker Variable	Description
Area	Provides you with the total area used by all placed pieces.
CutPerimeter	Provides you with the cutting perimeter of all placed pieces.
Date	Provides you with the report generation date.
Efficiency	Provides you with the efficiency value.
FileName	Provides you with the current Marker file name.
Folding	Provides you with available folding options.
LayoutMode	Provides you with the Layout mode chosen for the current Marker file.
Length	Provides you with the numeric value of the current Marker length. The length is evaluated using units reported with **UnitsLength** variable.
Material	Provides you with the list of Materials available within the current Marker file.
Name	Provides you with the name of the current Marker.
NLength	Provides you with the string containing the length of the current Marker file accompanied with units, e.g. 17m 50.96cm.
Notches	Provides you with the total number of notches.
NumOfAuxDrills	Provides you with the total number of auxiliary drills.
NumOfBundles	Provides you with the total number of bundles.
NumOfDrills	Provides you with the total number of drills.
NumOfPlies	Provides you with the number of plies defined within the current Marker file.
NWidth	Provides you with the string containing the width of the current Marker file accompanied with units, e.g. 1m 51cm.
NYield	Provides you with the string containing the yield of the current Marker file accompanied with units, e.g. 1m 56.92cm.
OrderedItems	Provides you with the number of ordered pieces.
PlacedItems	Provides you with the number of placed pieces.
Preview	Provides you with the box containing a preview of the Marker file.
UnitsLength	Provides you with units used for linear dimensions
UnitsSquare	Provides you with squared units.
Weight	Provides you with the Weight per Unit Area defined for the currently used material
Width	Provides you with the numeric value of the current Marker width. The width is evaluated using units reported with **UnitsLength** variable.
XMLHeader	Provides you with the XML Header of the current Marker file.
Yield	Provides you with the numeric value of the current Marker yield. The yield is evaluated using units reported with **UnitsLength** variable.

4.1.1.2 StyleList

The StyleList list provides you with sub-lists and variables that are evaluated for every style file used within the current Marker file.

In a case the report is generated for multiple Marker files, the StyleList is evaluated for all of them.

Marker Variable	Description
StyleName	Provides you with the name of the style used through the current Marker.
StyleFileName	Provides you with the file name of actual style.
StyleSetName	Provides you with the name of actual style set used for the style file available in the Marker.

SizeList

The SizeList sub-list is available under the StyleList. This list provides you with size-related variables.

This list is evaluated for every size of a current style file.

Marker Variable	Description
AreaOfSingleBundle	Provides you with the area of a single bundle of to the current size.
NumberOfBundlesPerOrder	Provides you with the number of ordered bundles of the current size.
NumberOfPiecesinPlacedIncompleteBundles	Provides you with the number of pieces appearing within incomplete bundles of the current size.
NumberOfPiecesInSingleBundle	Provides you with the number of pieces within a single bundle of the current size
NumberOfPlacedCompleteBundles	Provides you with the number of completely placed bundles of the current size.
SizeName	Provides you with the name of the current size.
TotalAreaOfOrderedPieces	Provides you with the total area of pieces ordered according to the current size
TotalAreaOfPlacedPieces	Provides you with the total area of pieces of the current size placed on the Marker table.
TotalPerimeterOfPlacedPieces	Provides you with the total perimeter of pieces of the current size placed on the Marker table.
TotalPerimeterPerOrder	Provides you with the total perimeter of pieces ordered according to the current size

Pieces

The Pieces sub-list is available under the StyleList. This list provides you with piece-related variables. This list is evaluated for every combination of piece and size available in the a current style file. For Example, if you have four pieces within style file (Front, Back, Sleeve and Collar) in parallel with three sizes (S, M and L), the list will be evaluated 12 times (Front-S, Front-M, Front-L, Back-S, Back-M and so on).

Marker Variable	Description
NumberOfPiecesInSingleBundle	Provides you with the total number of pieces available within a single bundle where the current piece (of current size) is involved. This number will be the same for all pieces involved into the same bundle.
NumOfBundlesPerOrder	Provides you with the number of ordered bundles where the current piece (of the current size) is participating. This number will be the same for all pieces involved into the same bundle.
NumOfPlacedCompleteBundles	Provides you with the number of placed bundles where the current piece (of the current size) is participating. This number will be the same for all pieces involved into the same bundle.
PieceArea	Provides you with the area of the current piece of the current size.
PieceBlockSizeX	Provides you with the X dimension of the bounding box surrounding the current piece of the current size.
PieceBlockSizeY	Provides you with the Y dimension of the bounding box surrounding the current piece of the current size.
PieceCode	Provides you with the code assigned to the current piece.
PieceDescription	Provides you with the description assigned to the current piece.
PieceMaterial	Provides you with the material assigned to the current piece.
PieceName	Provides you with the current piece name.
PiecePerimeter	Provides you with the perimeter of the current piece of the current size.
PieceSize	Provides you with the actual size name.
Quantity	Provides you with the number of current piece instances within a single bundle.
ScaleShrinkFactorX	Provides you with the Scale/Shrink factor along X-axis specified for the current piece.
ScaleShrinkFactorY	Provides you with the Scale/Shrink factor along Y-axis specified for the current piece.

4.1.2 PDS Variables

The table below provides you with the list of variables available for the PDS report template creation. The table provides you also with the information of hierarchy of variables as well as their description.

4.1.2.1 General Variables

These variables are available in the root space of the data source. These variables are evaluated for every PDS Style file for which the report or tech pack is generated.

PDS Variable	Description
FileName	The style file name including the path.
Name	The style name
NumOfSizes	The total number of sizes in the style file.
NumOfPieces	The total number of pieces in the style file.
TotalNumOfNotches	The total number of notches in the style file.
UnitsLength	The length units used for the style.
UnitsArea	The area units used for the style.
XMLHeader	The XML header of the style file.
PatternImage	The image of the entire pattern as it appears in PDS. **Important:** The pattern image is available only in the scope of the tech pack generation. During the regular report generation, the PatternImage variable is not evaluated.

4.1.2.2 Internals

The list of all internals available in the current style file. The Internals list provides you with variables that are evaluated for every internal used within the current PDS style file.

PDS Variable	Description
InternalCommand	The command (draw or cut) available for the current internal.
InternalCutLength	The cut length of the internal.

4.1.2.3 Materials

The list of all materials used through the current style file. The Materials list provides you with variables that are evaluated for every material used within the current PDS style file.

PDS Variable	Description
MaterialName	The name of the material.
MaterialPerimeter	The perimeter of all pieces of the current material.
MaterialArea	The area of all pieces of the current material.
MaterialPiecesNum	The number of pieces of the current material.

4.1.2.4 MeasurementCharts

The list of all measurement charts available in the current style file.

The MeasurementCharts list provides you with sub-lists and variables evaluated for every measurement chart available in the current style file.

Important:

The MeasurementCharts list is evaluated only in the scope of the tech pack generation. During the regular report generation, the list and underlying variables are not evaluated.

PDS Variable	Description
MCName	The name of the current measurement chart

MCMeasures

The list of all measures defined within the current measurement chart.

The MCMeasures sub-list provides you with sub-lists and variables evaluated for every certain measure defined within the current measurement chart.

PDS Variable	Description
MCMeasureName	The name of actual measure
MCMeasureDescription	The description assigned to the actual measure

MCMeasureSizes

The list of all sizes available within the style.

In a context of the current measure, the MCMeasureSizes sub-list provides you with variables evaluated for every certain size.

PDS Variable	Description
MCMeasureSizeName	The name of actual size
MCMeasureSizeStep	The step value of the current measure calculated between the current size and next one
MCMeasureSizeValue	The value of the current measure calculated for the current size

uMCMeasures

The list of all measures defined within the current measurement chart.

The uMCMeasures sub-list provides you with sub-lists and variables evaluated for every combination of measure and size available within the current measurement chart.

For example if the measurement chart contains 5 measures and the style file contains 4 sizes, the uMCMeasures list will be evaluated 20 times to provide you with the data for all dimension/size combinations.

PDS Variable	Description
uMCMeasureName	The name of actual measure
uMCMeasureDescription	The description assigned to the actual measure
uMCMeasureSizeName	The name of actual size
uMCMeasureSizeStep	The step value of the current measure calculated between the current size and next one

uMCMeasureSizeValue	The value of the current measure calculated for the current size

MCPieceImages

The list of all piece images illustrating measures defined through the current measurement chart.

The sub-list is evaluated for every piece where measures belonging to the current measurement chart are placed.

PDS Variable	Description
MCPieceName	The name of actual piece
MCPieceImage	The image of the actual piece. The image is generated for the certain piece and provides you with all measures placed on it.

4.1.2.5 Notches

The list of all notches defined in the scope of current style file. The underlying variables are evaluated for every notch.

PDS Variable	Description
NotchType	The type of notch.
NotchCommand	The command (draw or cut) available for the current notch.

4.1.2.6 Pieces

The list of all pieces defined in the scope of current style file.

The underlying sub-lists and variables are evaluated for every piece available in the style file.

PDS Variable	Description
PieceBuffer	The value of buffer assigned to the current piece
PieceBufferType	The type of buffer assigned to the current piece
PieceCode	The code assigned to the piece.
PieceDescription	The description assigned to the piece.
PieceFoldAllowed	Determines if folding is allowed for the current piece
PieceGradingPoints	The number of grading points in the current piece.
PieceMaterial	The material that is assigned to the piece.
PieceMaxTilt	The maximum tilting angle allowed.
PieceName	The name of the piece.
PieceNumOfInstances	The quantity of piece instances.
PieceNumOfSizes	The number of sizes where the current piece is available.
PieceOpposite	Determines if the piece has an opposite piece.
PieceOrientation	Determines the piece orientation (right, left or both)
PiecePair	Determines if the current piece is defined as a pair piece
PieceQuality	The quality of the piece
PieceRotation	Rotation allowance assigned for the current piece.
PieceShrinkX	Shrink/Scale factor in X direction
PieceShrinkY	Shrink/Scale factor in Y direction
PieceTotalArea	The area of the piece per all sizes.
PieceTotalPerimeter	The perimeter of the piece per all sizes.
Preview	The SVG image of the piece.

PieceButtons

The list of all buttons defined in the scope of the current piece.

The underlying variables are evaluated for every button defined in the current piece.

PDS Variable	Description
PieceButtonType	The type of button.
PieceButtonRadius	The radius of the button.
PieceButtonCutLength	The cut length of the button.

PieceCircles

The list of all circles defined in the scope of the current piece.

The underlying variables are evaluated for every circle defined in the current piece.

PDS Variable	Description
PieceCircleCommand	The command (draw or cut) available for the current circle.
PieceCircleRadius	The radius of the circle.
PieceCircleCutLength	The cut length of the circle.

PieceContours

The list of all contours defined in the scope of the current piece.

The underlying variables are evaluated for every contour defined in the current piece.

PDS Variable	Description
PieceContourCommand	The command (draw or cut) available for the current contour.
PieceContourCutLength	The cut length of the contour.

PieceInternals

The list of all internals defined in the scope of the current piece.

The underlying variables are evaluated for every internal defined in the current piece.

PDS Variable	Description
PieceInternalCommand	The command (draw or cut) available for the current internal.
PieceInternalCutLength	The cut length of the internal.

PieceNotches

The list of all notches defined in the scope of the current piece.

The underlying variables are evaluated for every notch defined in the current piece.

PDS Variable	Description
PieceNotchType	The type of notch.
PieceNotchCommand	The command (draw or cut) that is available for the current notch.
PieceNotchDepth	The depth of the notch.
PieceNotchCutLength	The cutting length of the notch.

PieceSizes

The list of all sizes defined in the style file providing you with the piece properties that could differ from size to size.

The underlying variables are evaluated for every size defined in the current style.

PDS Variable	Description
PieceSizeName	The name of the size available for the current piece.
PieceSizeArea	The area of the current piece graded to the current size.
PieceSizePerimeter	The perimeter of the current piece graded to the current size.
PieceSizeXDimension	The X dimension of the current piece graded to the current size.
PieceSizeYDimension	The Y dimension of the current piece graded to the current size.
PieceSizeBaseSize	Defines whether the current size is a base size

4.1.2.7 RigidParts

The list of all rigid parts defined in the style file.

The underlying variables are evaluated for every rigid part defined in the current style.

PDS Variable	Description
ButtonShape	The name of button shape file (*.mod) used for the rigid part definition
Code	The code assigned to the current rigid part
Desc	The description assigned to the current rigid part
Name	The name assigned to the current rigid part
Profile	The name of the profile holding physical properties of the stitch associated with the current rigid part
Quantity	The quantity of rigid parts

RPPieces

The list of pieces where the current rigid part is applied.

The underlying variables are evaluated for every piece where the rigid part appears.

PDS Variable	Description
PieceName	The name of the piece

RPOperations

The list of technology operations applicable to the current rigid part.

The underlying variables are evaluated for every technology operation.

PDS Variable	Description
StitchOperationName	The name of technology operation

4.1.2.8 Sizes

The list of all sizes available within the current style file.

The underlying variables are evaluated for every size available in the style.

PDS Variable	Description
SizeName	The name of the current size.
SizeArea	The total area of all the pieces that belong to the current size.
SizePerimeter	The total perimeter of all the pieces that belong to the current size.
SizeBaseSize	Determines whether the current size is a base size

4.1.2.9 Stitches

The list of all stitches appearing within the current style file.

The underlying variables are evaluated for every stitch available in the style.

PDS Variable	Description
StitchCode	The code assigned to the current stitch
StitchDescription	The description assigned to the current stitch
StitchName	The name assigned to the current stitch
StitchProfile	The name of the profile holding physical properties of the stitch
StitchQuantity	The quantity of stitches
StitchImage	The image of the stitch and pieces connected by this stitch.  **Important:** The stitch image is available only in the scope of the tech pack generation. During the regular report generation, the StitchImage variable is not evaluated.

StitchOperations

The list of technology operations applicable to the current stitch.

The underlying variables are evaluated for every technology operation.

PDS Variable	Description
StitchOperationName	The name of technology operation

StitchPieces

The list of pieces where the current stitch is applied.

The underlying variables are evaluated for every piece where the current stitch appears.

PDS Variable	Description
StitchPieceName	The name of the piece

StitchPieceSizes

The list of sizes available within the current style.

The list provides you with those current piece properties relevant in the scope of the current stitch that could differ per size.

The underlying variables are evaluated for every size where the current stitch appears.

PDS Variable	Description
StitchPieceSizeName	The name of the size
StitchPieceSizeLength	The length of current piece segments used for the current stitch definition according to the current size

StitchSizes

The list of sizes available within current style.

The list provides you with those current stitch properties that could differ per size.

The underlying variables are evaluated for every size.

PDS Variable	Description
StitchSize3DLength	The 3D length of current stitch according to the current size
StitchPieceSizeName	The name of the size

4.1.2.10 StyleSets

The list of style sets available within the current style.

The underlying sub-lists and variables are evaluated for every style set.

PDS Variable	Description
SsetName	The name of the style set.
SSetDescription	The description of the style set.

SSetPieces

The list of style pieces defined in scope of the current style set.

The underlying variables are evaluated for every piece belonging to the current style set.

PDS Variable	Description
SSPieceArea	The area of the piece that is assigned to the current style set.
SSPieceBaseline	The baseline of the piece that is assigned to the current style set.
SSPieceBuffer	The buffer values of the piece that is assigned to the current style set.
SSPieceBufferType	The buffer type of the piece that is assigned to the current style set.
SSPieceCode	The code of the piece that is assigned to the current style set.
SSPieceDescription	The description of the piece that is assigned to the current style set.
SSPieceFoldAllowed	If folding is allowed for the piece that is assigned to the current style set.
SSPieceMaterial	The material of the piece that is assigned to the current style set.
SSPieceMaxTilt	The maximum allowed tilt of the piece that is assigned to the current style set.
SSPieceName	The name of the piece that is assigned to the current style set.
SSPieceOpposite	The opposite direction of the piece that is assigned to the current style set.
SSPieceOrientation	The orientation of the piece that is assigned to the current style set.
SSPiecePair	The pair status of the piece that is assigned to the current style set.

SSPiecePerimeter	The perimeter of the piece that is assigned to the current style set.
SSPieceQuality	The quality of the piece that is assigned to the current style set.
SSPieceQuantity	The quantity of the piece that is assigned to the current style set.
SSPieceRotation	If piece rotation is allowed for the piece that is assigned to the current style set.
SSPieceShrinkX	The X shrinkage of the piece that is assigned to the current style set.
SSPieceShrinkY	The Y shrinkage of the piece that is assigned to the current style set.

4.1.2.11 Variants

The list of all variants and articles available in the current style file.

The Variants list provides you with sub-lists and variables evaluated for every variant available in the current style file.

Important: The Variants list is evaluated only in the scope of the tech pack generation. During the regular report generation, the list and underlying variables are not evaluated. The Variants list is also not evaluated in a case when you have no 3D cloth model defined.

PDS Variable	Description
VariantName	The name of the current variant

VariantArticles

The list of articles defined in scope of the current variant.

The underlying variables are evaluated for every article belonging to the current variant.

PDS Variable	Description
VariantArticleName	The name of the current article

VariantArticleImages

The list of images generated for the current article in the scope of the tech pack generation.

The underlying variables are evaluated for every generated image.

PDS Variable	Description
VariantArticleImageViewName	The name of the view used for the image generation
VariantArticleImage	The corresponding image generated during the tech pack generation

VariantImages

The list of images generated for the current variant in the scope of the tech pack generation.

The underlying variables are evaluated for every generated image.

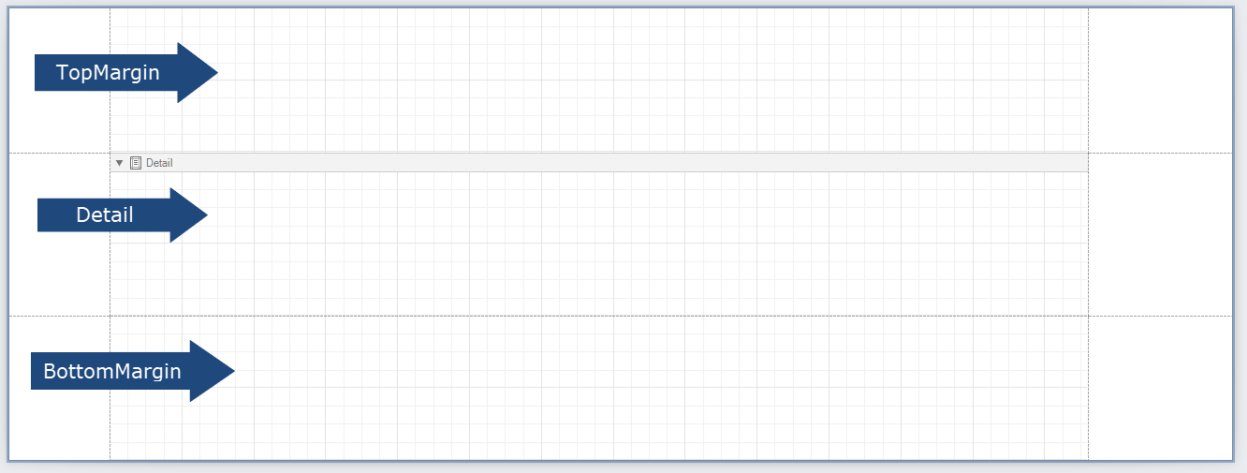
PDS Variable	Description
VariantImageViewName	The name of the view used for the image generation
VariantImage	The corresponding image generated during the tech pack generation

4.2. Understanding the Template Structure and Layout

This topic provides you with the basic description of the report template structure and layout.

4.2.1 Report Layout and Bands

When a new template is created with the **New** command (see topic 3.1.1), it contains three basic sections (bands): TopMargin, Detail and BottomMargin as shown below.



The TopMargin / BottomMargin bands are repeated once at every page of the report. The appearance and content of these bands is the same for all pages. Typically, TopMargin / BottomMargin bands are used for some general report information; e.g. the report generation time & date, user name and so on.

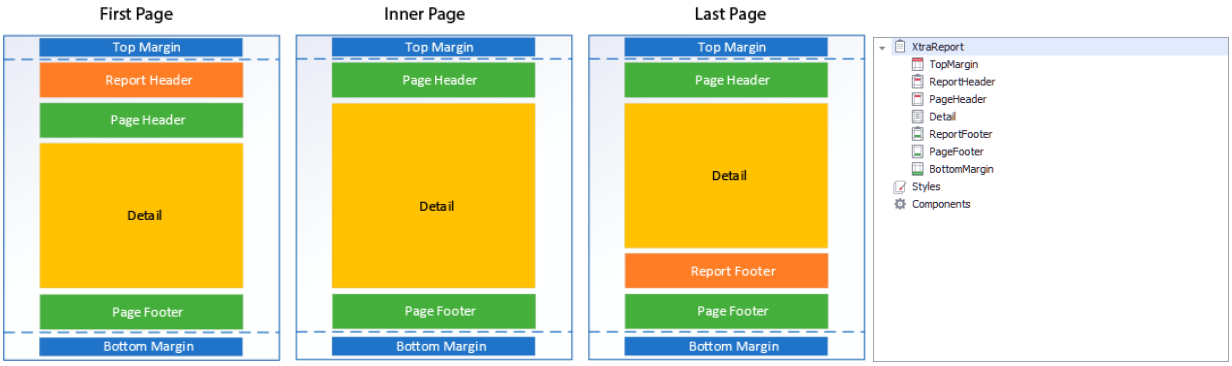
The Detail band is the most important part of the template layout. This band is connected (bound) to the data provided by PDS / Marker. The Detail band and its sub-bands typically contain all variable controls connected to the data; these variables are evaluated during the report generation.

The Detail band is connected to the major data source used for the report generation, which is a Marker (*mrk) or PDS (*pds) file. For more information about the Detail band and its content, refer to topic 4.2.2.

In addition to bands mentioned above, the Report Header and Report Footer bands are available in the report. The Report Header is the report's first band (margins mentioned above are "out-of-page" zones). Use this band to display the report's name, company logo, etc. The Report Header band appears once only on the first page of the report. The Report Footer is placed before the report's last page. You can use the Report Footer band for report summaries or conclusions.

The Page Header and Page Footer appear at the top and bottom of every page in a report. They display information that should be printed on every page.

The following image illustrates a sample report layout and the report representation in Report Explorer (see topic 3.3.1)

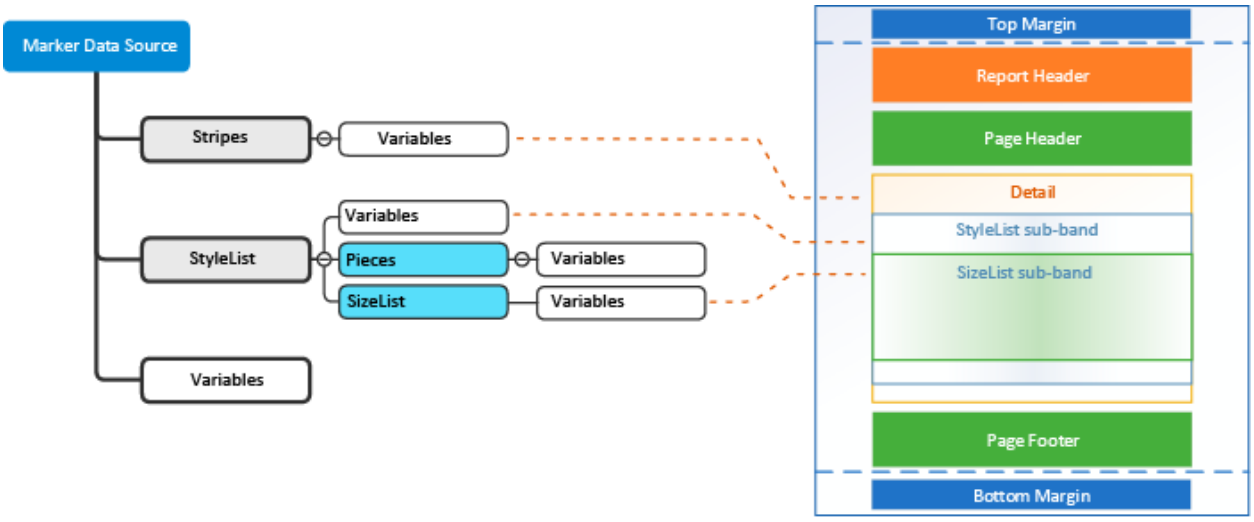


4.2.2 Detail band and sub-reports

As mentioned earlier, the Detail band of the report template is bound to the PDS / Marker file as the data source. The Detail band is dedicated to contain information contained in the root space of the PDS / Marker data source (see topic 4.1); i.e. general properties of the PDS / Marker file, such as file name, units, number of pieces, number of sizes, efficiency etc.

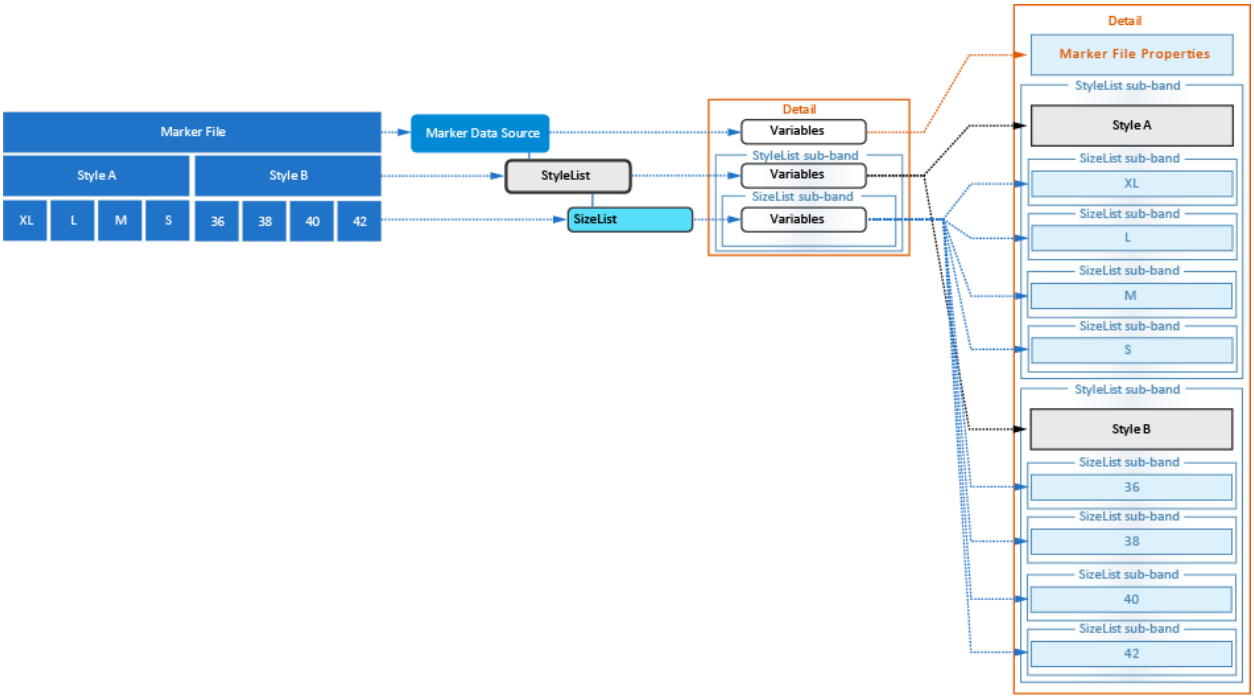
The Detail band provides you with the possibility to create nested sub-reports dedicated to expose hierarchical data available in PDS / Marker files. Such sub-reports can be created in correspondence with the data source structure provided by PDS / Marker for report generation.

To represent such hierarchical structure of data sources described earlier (see topic 4.1), the Report Designer provides you with the capability to create and use nested sub-reports. Each sub-report is bound to the certain list available within the data model and allows you to handle variables belonging to the corresponding list. The image below illustrates sub-bands used for the Marker report generation and their binding to specific lists.



In such a way, the Detail band provides you with access to variables located directly under the top node of the Marker data source (Efficiency, Marker Length, Units, File Name etc.). The underlying StyleList sub-report (also referred to as the Detail Report) provides you with access to styles and their properties (Style Name, Style File Name etc.) The underlying Size List sub-report provides you with the access to sizes (per every style file) and their properties (Size Name, Number of bundles per order, Number of placed bundles, Number of pieces per single bundle etc.).

As mentioned earlier, during the report generation, the Detail band is evaluated for every Marker file added into the scope of report generation. However, the StyleList sub-report is evaluated for every style available within the current Marker file. In a similar manner, within every StyleSet sub-report, the SizeList sub-report is evaluated for every size (see the illustration below).

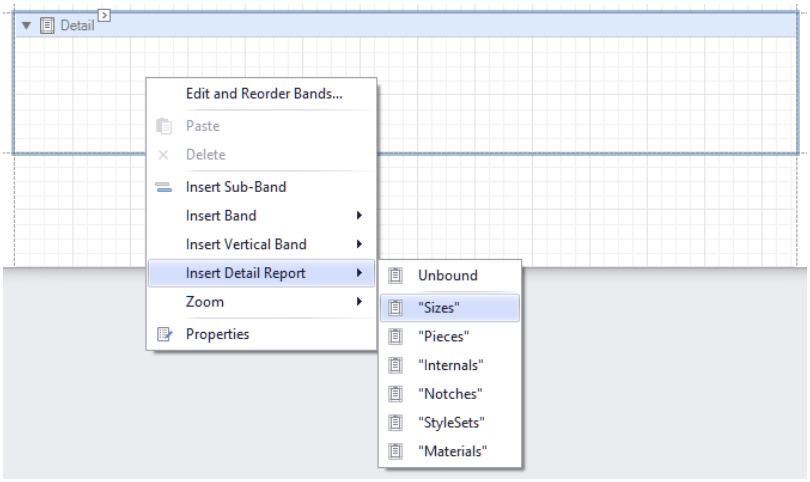


The illustration above provides you with the scheme of data model available for a Marker file containing two styles with 4 sizes each. The illustration also provides you with the data source representation and binding of it to Detail bands and sub-reports. Once defined within the template, binding enables you to maintain the correct data delivery to report bands during the report generation. In a case of listed data, the corresponding band is iterated several times delivering all list entries to corresponding band.

The Detail band is a mandatory part of every report template which is created automatically and cannot be deleted.

To create a sub-band:

1. Select the Detail band as shown below.
2. Choose the Insert Detail Report command from the RMB menu.
3. Choose the corresponding data source list from the sub-menu.

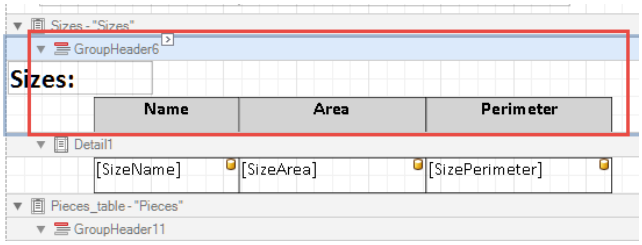


A new Detail Report will be added under the Detail band and bound to the appropriate list exposed by the data source model.

Such a sub-report could contain only controls bound to the corresponding list properties. Binding controls to variables available outside this list will cause incorrect data rendering during the report generation.

Every such sub-report contains its own Detail band containing controls bound to variables. If a data list (like Sizes) is assigned to the sub-report as described above, the Detail band is regenerated for every instance of the list. So, if the Sizes list for the specific style has 4 sizes, the corresponding Detail band will be regenerated four times for every style. In addition to the Detail band, the Detail Report could contain a Group Header and Group Footer bands. These bands are populated once in the beginning and the end of the group of generated Detail bands.

In the exercise below, the sub-report is assigned to the Sizes list. The Group Header band of the sub-report contains the table filled with static labels describing the column names; the Detail band provides you with a similar table which contains cells bound to the list variables.



When the report is generated, the Group Header content appears once and the Detail band content is re-generated for every certain size as shown below.

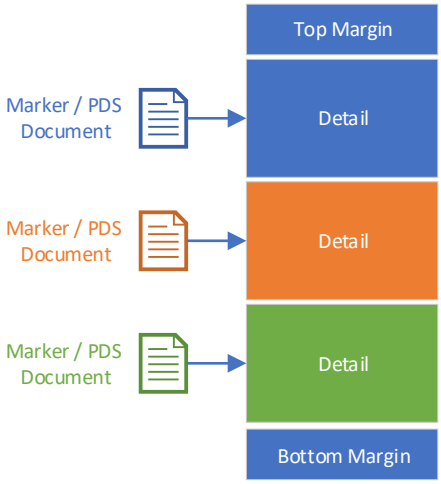
Sizes:

Name	Area	Perimeter
Small	0.7264	795.39
Medium	0.7714	816.54
Large	0.8176	837.77
XLarge	0.8651	859.07

4.2.3 Report generation for multiple PDS / Marker documents

As mentioned earlier, when the report is generated for a single PDS / Marker file, the Detail band is printed once, providing you with the corresponding data.

When the report is generated for a series of PDS / Marker files, the single report is generated with multiple Detail bands. Every such Detail band is generated for a certain PDS / Marker file and exposes the corresponding data.



4.3. Create Basic Controls

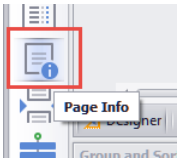
This topic provides you with a description of basic controls provided by Report Designer and used for the PDS / Marker report template creation.

4.3.1 Page Info

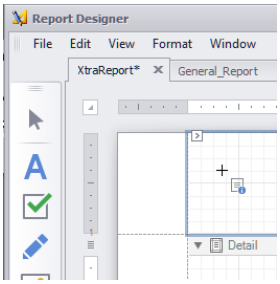
The Page Info control provides you with the possibility to display auxiliary information on report pages, such as date, time, page numbers or user name.

To add the Page Info control:

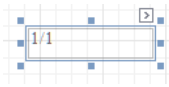
1. Choose the Page Info tool from the Standard Controls toolbar.



2. Click the empty space within the report template to place the control.

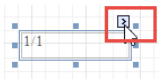


A new control will be added as shown below.

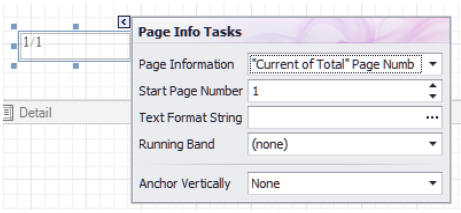


By default, a newly added Page Info control displays the page number in the format of “Current of Total”. To change the content of the Page Info control, perform the following actions:

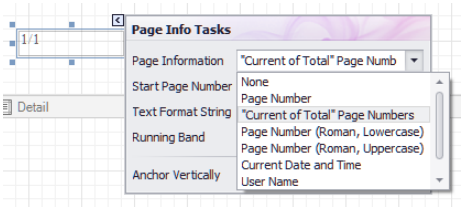
1. Click on the smart tag adjacent to the control’s frame as shown here.



2. The Page Info Tasks panel will be displayed.



3. Open the drop-down list for the Page Information property.



The drop-down list allows you to select one of the following options to define the Page Info control content:

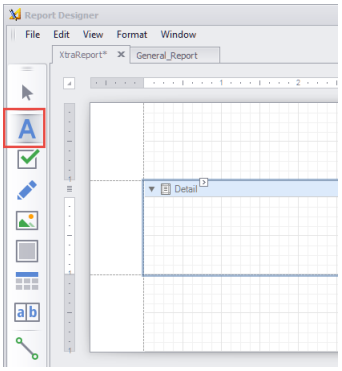
- Page Number
- “Current of Total” Page Number
- Page Number (Roman, lowercase)
- Page Number (Roman, uppercase)
- Current Date and Time
- User Name
- Pages Count

4.3.2 Label

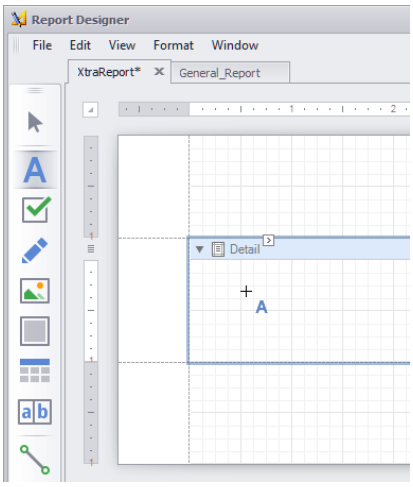
Label control is the most frequently used control for PDS / Marker report template creation. This control provides you with the possibility to display static text within the report as well as dynamic data obtained from the PDS / Marker data source during the report generation.

To add the static text label:

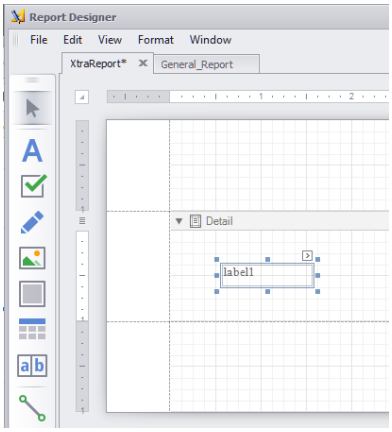
1. Choose the Label command from the Standard Controls toolbar.



2. Click the empty space in the template page displayed in the Design panel.



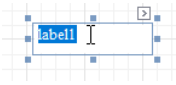
3. The label control will be added and selected.



The automatic selection of the newly created label control provides you with the possibility to see and adjust control properties with the Property Grid.

Make a note that newly created control is static, i.e. it has no relationship to the PDS / Marker data model. This means that during the report generation such a label control will be printed as a static text and will not expose any value from the data source.

To modify the static text of this control, double-click it to enter the text editing mode.

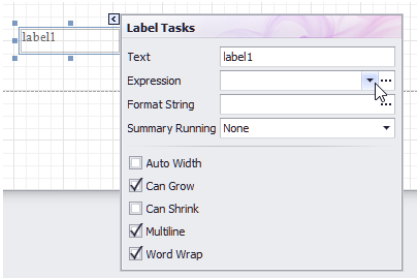


Since the static text label control is not bound to the data source, it could be placed regardless to bands of the report template and their binding. Bands and their binding start to be meaningful when the connection of label to the data source is performed.

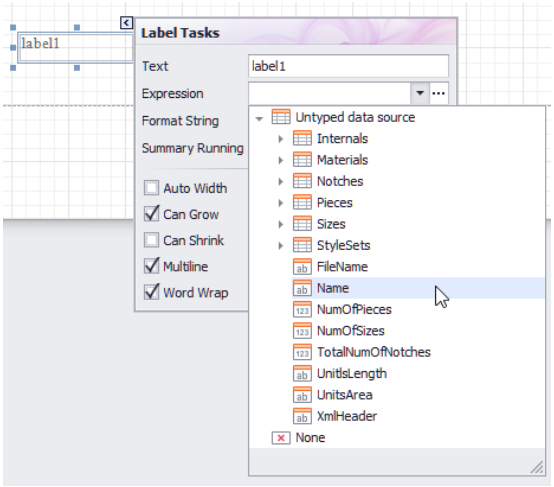
To connect a label to the data source variables:

1. Click on the smart tag adjacent to the label frame as shown here.

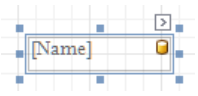
2. The Label Tasks panel will be displayed. Open the drop-down list for the Expression property.



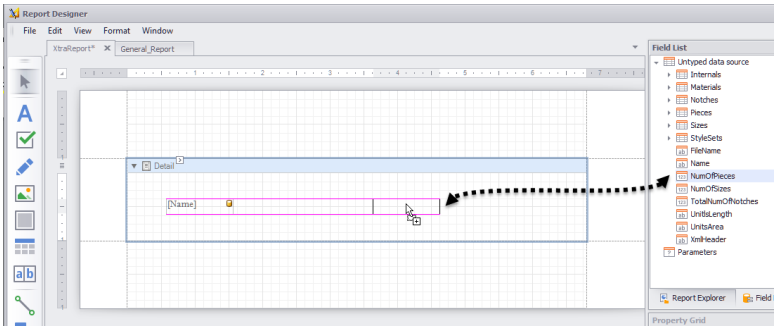
3. Under the Untyped Data Source header, choose the appropriate variable to bind it to the current label.



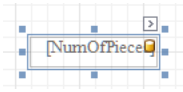
4. Once the variable is selected, its name will appear in square brackets in the Expression field. Click outside the Label Tasks panel to close it. Now you have the label control bound to the variable. The bound variable name is displayed within the label frame in the design panel as shown below. In addition, binding is also emphasized by the database (orange cylinder) icon appearing within the label frame.



As a quick alternative to the flow described above, you could simply drag and drop a required variable from the Untyped Data Source displayed in the Fields list. Make a note that only variables of text and numeric types are displayed as labels in the design panel.

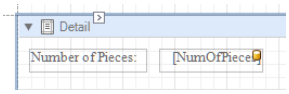


Once the drag and drop is performed, a new label automatically bound to the appropriate variable will be created within the report template.



The static Label with the explicitly types text could be placed within every report band with no limitations. For dynamic labels defined via variables, it is critical to create the Label control within the correct band / detail report connected to the corresponding list of the data source. In such a way, the correct data rendering will be performed during the report generation. The band where the Label is placed should be connected to the same data source list where the variable appears. E.g. variables located in the root space of the Untyped Data Source could be bound to labels located in the Detail band only; variables located under the Sizes list in the data source could be added to the sub-band assigned to the Sizes list and so on. For more information about data source structure available for PDS and Marker, refer to topic 4.1. For more information about band creation, refer to topic 4.2.2.

The combination of a static text label and a dynamic label bound to the data source is used most frequently for the report creation. Such a combination allows you to display the static field name and a value which is dynamically rendered during the report generation.

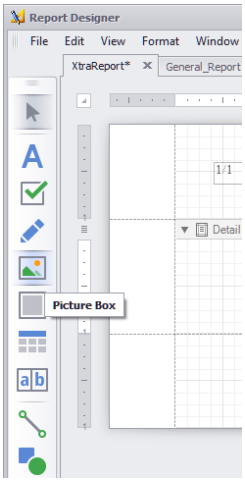


4.3.3 Picture Box

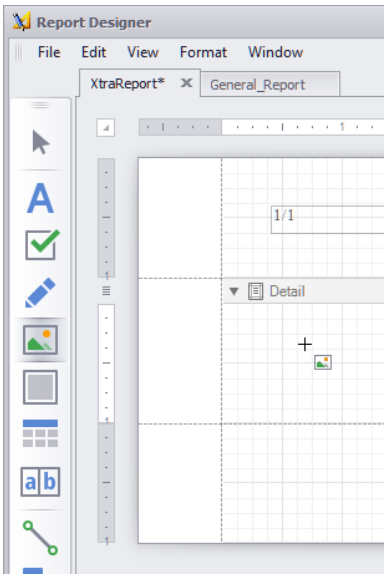
The Picture Box control is used to display images within the PDS / Marker report. This control provides you with the possibility to display a static image within the report (like a company logo) as well as dynamic data obtained from the PDS / Marker data source during the report generation (e.g. piece image).

To add the Picture Box control:

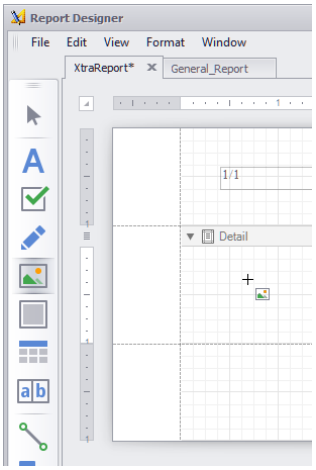
1. Choose the Picture Box command from the Standard Controls toolbar.



2. Click the empty space in the template page displayed in the Design panel.



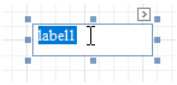
3. The label control will be added and selected.



The automatic selection of the newly created picture box provides you with the possibility to see and adjust control properties with the Property Grid.

A newly created control is created empty; i.e. it does not link to an image. In other words, the control serves as a container for the image which has to be assigned to the control. You can either assign a static image by explicitly choosing the image file, or bind the control to the data source to get the dynamic image during the report generation run time.

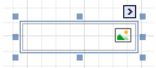
To modify the static text of this control you should double click it to enter the text editing mode.



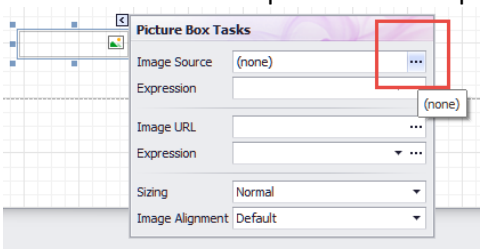
Since the static text label control is not bound to a data source, it could be placed regardless to bands of the report template and their binding. Bands and their binding start to be meaningful when the connection of label to the data source is performed.

To select the static image to appear within the control:

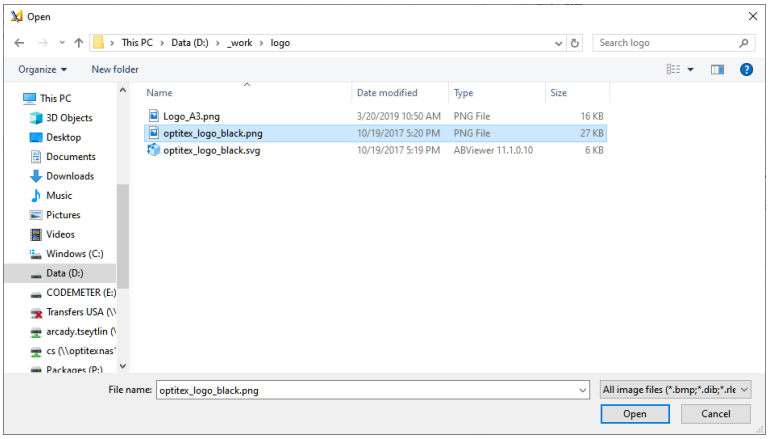
1. Click on the smart tag adjacent to the picture box frame as shown here.



2. The Picture Box Tasks panel will be displayed. Click the (…) icon within the Image Source field.

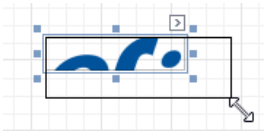


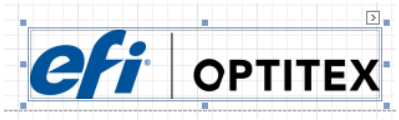
3. The regular Open browser will be displayed providing you with the possibility to browse for the image file. Choose the image file.



4. Once the image is chosen, it is displayed within the control.

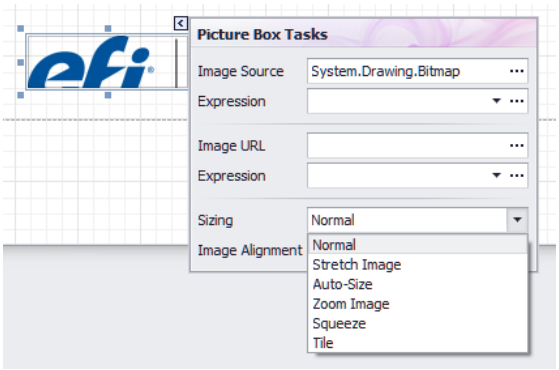
Make a note that typically, the size of the selected image does not fit to the default size of the control frame. To resolve it, you can either manually change the control frame size by dragging (see the image below), or control over the control and image fit via choosing an automatic sizing option.





To specify the sizing option:

1. Click on the smart tag adjacent to the picture box frame.
2. The Picture Box Tasks panel will be displayed. Choose the appropriate option from the Sizing drop down list.



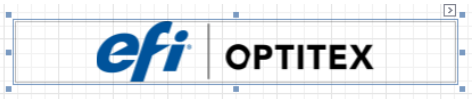
The following options are available to define the sizing:

- **Normal**. The image is displayed with its original dimensions, regardless of the control frame size.

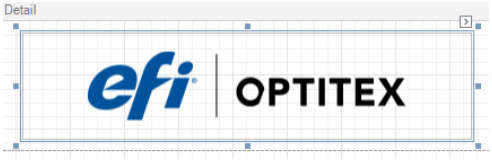
- **Stretch Image**. The image is stretched to completely fill the control frame. The image is stretched independently in both width and height directions; such stretching could cause distortion of the image proportions.

- **Auto-Size**. The image is displayed with the real size; the control's frame dimensions are auto-adjusted to the image size.

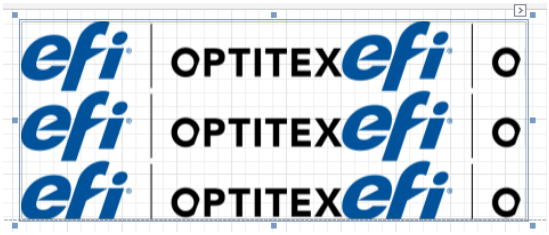
- **Zoom Image**. The image is proportionally resized so that it fits the control.



- **Squeeze**. If the control's dimensions exceed the image size, the image is centered and shown full-size. Otherwise, the image is resized to fit into the control's dimensions.

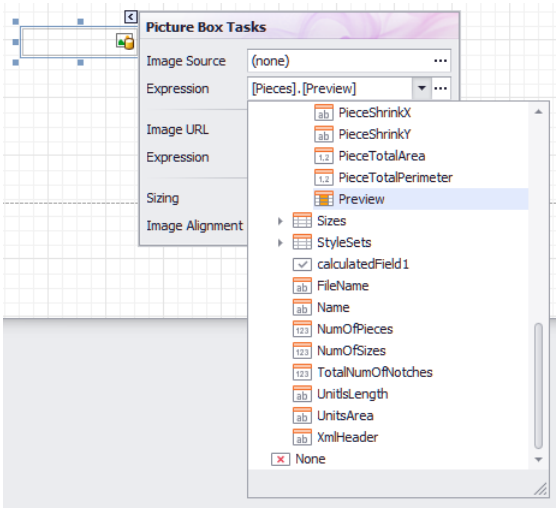


- **Tile**. The original image is replicated within the picture control starting from the upper-left corner. The replicated image is clipped if it does not fit in the picture control that contains it.

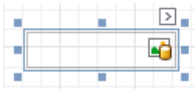


To connect a Picture Box control to the image provided as a variable from the data source:

1. Click on the smart tag adjacent to the picture box frame.
2. The Picture Box Tasks panel will be displayed. Open the drop-down list of the Expression field.
3. Under the Untyped Data Source header, choose the appropriate Image variable to bind it to the current Picture Box control.



4. Once the Image variable is selected, its name will appear in square brackets in the Expression field. Click outside the Label Tasks panel to close it. Now you have the Picture Box bound to the Image variable. The binding is emphasized by a database (orange cylinder) icon appearing within the Picture Box frame.



Once the binding is performed, define the appropriate Sizing option as described earlier to allow dynamic change of the control size during the report generation run time.

As a quick alternative to the flow described above, you could simply drag and drop a required image variable from the Untyped Data Source displayed in the Fields list (in the same manner as it is described for labels). Make a note that only variables of image type are displayed as picture boxes in the design panel. Once the drag and drop is performed, a new picture box which is automatically bound to the appropriate image variable will be created within the report template.

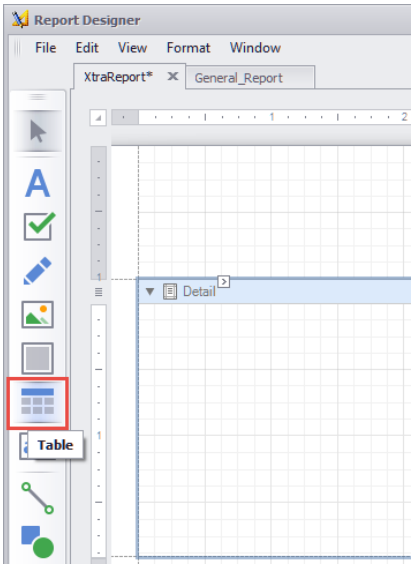
In the same manner as it is described for variables, the static Picture Box bound to the explicitly selected image could be placed within every report band. For dynamic images defined via variables, it is critical to create the Picture Box bound within the correct band / detail report connected to the corresponding list of the data sources. The band where the Picture box is placed should be connected to the same data source list where the image variable appears. E.g. image variables located in the root space of the Untyped Data Source could be bound to Picture Boxes located in the Detail band only; image variables located under the Pieces list in the data source could be added to the sub-band assigned to the Pieces list and so on. For more information about data source structure available for PDS and Marker, refer to topic 4.1. For more information about band creation, refer to topic 4.2.2.

4.4. Create Tables

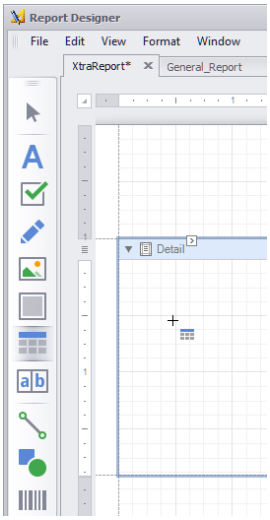
Report Writer provides you with the possibility to create and use tables for better organization and display of report data.

To create a new table:

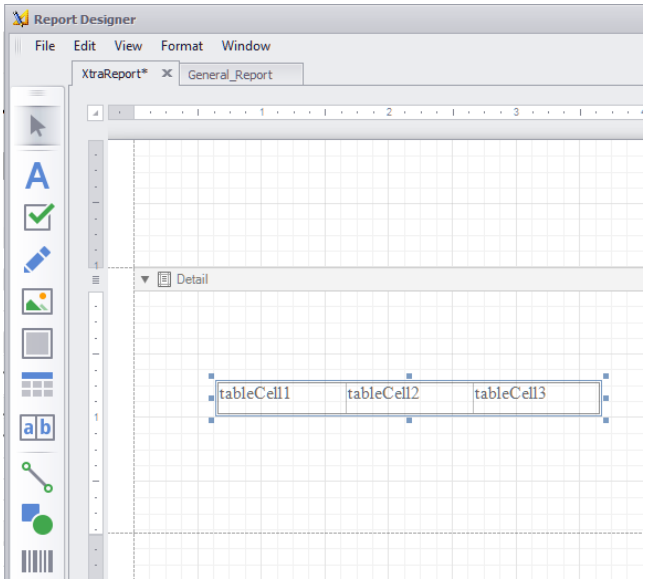
1. Choose the Table command from the Standard Controls toolbar.



2. Click the empty space in the template page displayed in the Design panel.



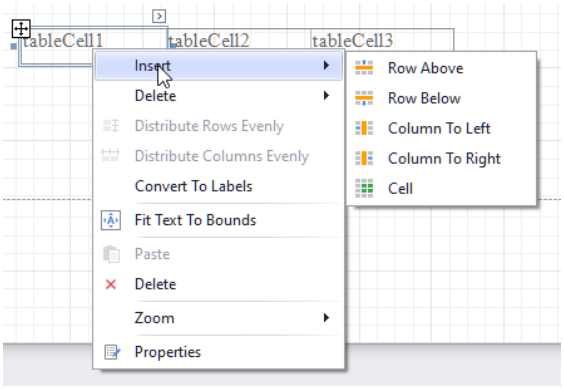
3. A new table control will be added and selected.



By default, every newly table is composed from three columns and a single row, i.e. it has three cells. Such a default table can be adjusted by adding or removing rows/columns.

To add rows / columns:

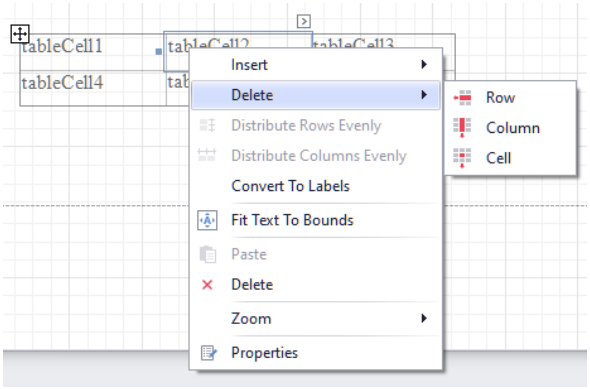
1. Right click on a table cell and choose the Insert sub-menu.



2. Select an appropriate command to add a row or cell. The following commands are available.
   - **Row Above**. This command adds a row above the row where the RMB menu was invoked.
   - **Row Below**. This command adds a row below the row where the RMB menu was invoked.
   - **Column To Left**. This command adds a column at the left of the column where the RMB menu was invoked.
   - **Column To Right**. This command adds a column at the right of the column where the RMB menu was invoked.
   - **Cell**. This command adds a new cell in the same row and next to the cell where the RMB menu was invoked.

To remove rows / columns / cells:

1. Right click on a table cell and choose the Remove sub-menu.

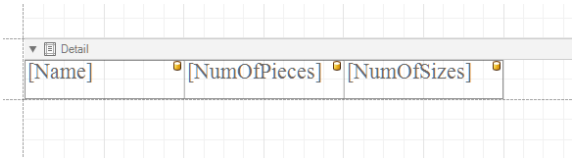


2. Choose an appropriate command from the sub-menu to remove the current cell (where the RMB menu was invoked) or an entire row / column.

The table cell is represented by a regular label which could remain static or be bound to a variable. The binding of cells to variables of the data source is performed in a regular manner as described for labels (see topic 4.3.2 for more information). Like regular labels, it is critical to maintain the correspondence between values used within the table and the report band / sub-report where the table is placed. The band / sub-report should be bound to the same list which owns variables used within the table.

When the report is generated, variables in the table cells are populated with values in the same manner as regular labels. The amount of rows in the table is preserved the same as defined in the template, i.e. rows are not added automatically for the table for every instance of values appearing within the table.

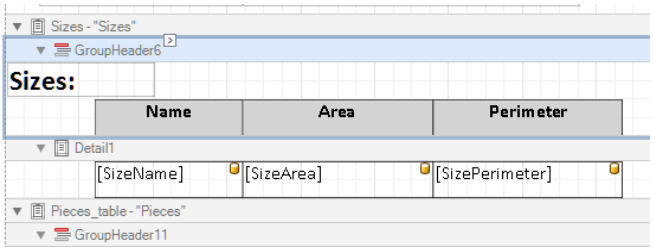
To create a table intended to the dynamic data display, it is recommended to create a single row table within the required band and adjust the band height to be exactly the same as the table height, as shown below.



In such a way, when the report is generated, the multiple re-generation of the certain band (containing the table) for every data set instance will cause an appearance of a new single-row table. In a case when the height of the band matches the height of the table, multiple single row tables look like a regular multi-row table.

The header row for the table should be created as a separate table where all cells are defined as static text. Such a table should be located within the band which is parent relative to the band where the data row of the table is located, or in the group header band (see topic 4.2.2) defined for the current Detail Report.

For example, the image below provides you with two tables: the first one is located in the Group Header band created in the Detail Report bound to the Sizes list. This table contains only static text of table headers. The second table created in the Detail band of the same sub-report (bound to the Sizes list) contains bound variables.



Make a note that the header table is placed adjacent to the bottom of the Group header band and the height of the data table matches the height of the Detail band.

When the report is generated, the header table located in the Group Header band is printed only once, but the data table is printed for every size instance.

Sizes:

Name	Area	Perimeter
Small	0.7264	795.39
Medium	0.7714	816.54
Large	0.8176	837.77
XLarge	0.8651	859.07

The height of the table and the height of the bound could be found and adjusted using the Property Grid.

4.5. Expressions

Report Writer provides you with the possibility to define and manage expressions to perform various calculations during the report generation run time. Expressions are used to specify criteria for retrieving and formatting data, creating calculated fields and calculating summaries, conditionally shaping data and changing a report control's appearance.

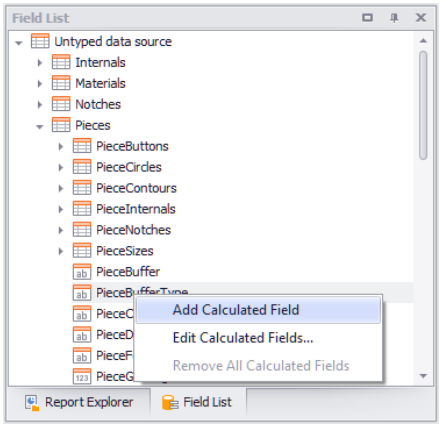
4.5.1 Defining Expressions

Report Writer provides you with two options to use expressions.

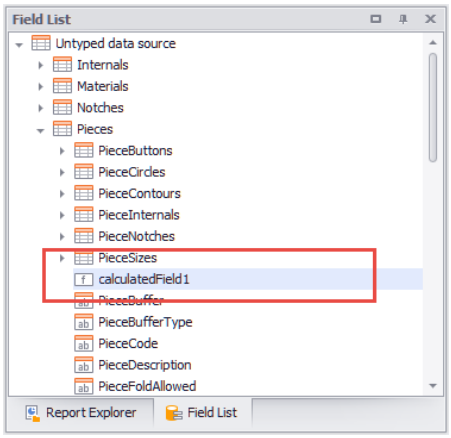
- You can add a calculated field and specify the expression for its calculation. In other words, using calculated fields with assigned expressions is a way to create your own variables in addition to standard ones provided by the PDS / Marker data source. The calculated field is added to the variables tree displayed in the Field List panel in compliance with the available data source structure; once the calculated field appears within the Fields List, it is available for multiple use.
- It is also possible to assign an expression directly to a control displayed in the Design panel, e.g. label. In such a way, the calculated field is not added to the Fields list, which causes a flexibility reduction when using the expression multiple times.

To add a new calculated value:

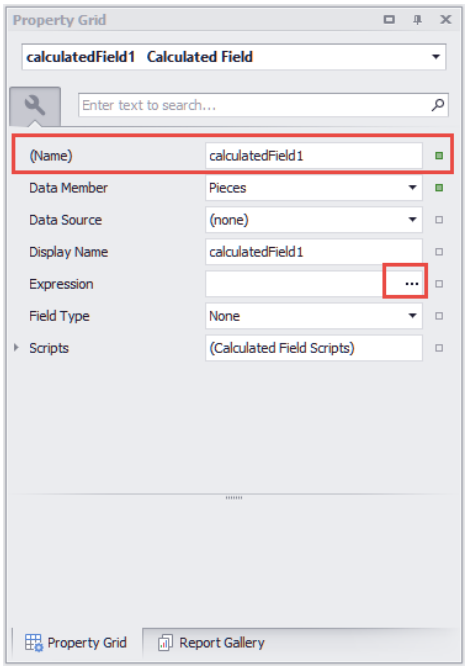
1. Right click on a node in the Field List as shown below. The menu will be displayed.



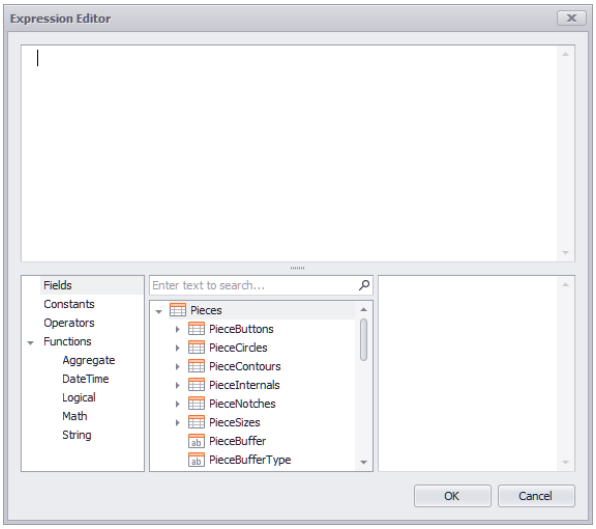
2. Choose the Add Calculated Field command from the menu. A new calculated field with the default name will be added to the list. When the field's list node from which the menu was invoked is a variable, a new calculated field is added to the list which is a parent for the selected variable. When the field's list node from which the menu was invoked is a list, a new calculated field is added to this list.



3. In the Property Grid panel, customize the name of the newly added calculated field.



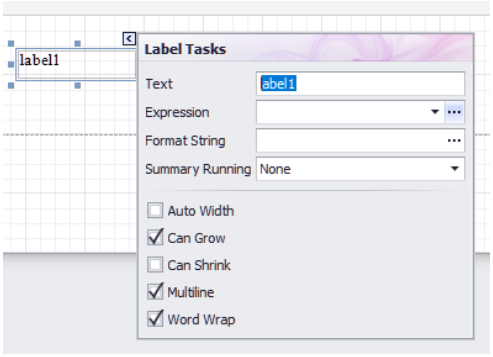
In the Expression field of the Property Grid panel, click the ellipsis icon to launch the Expression Editor to define the expression for the newly added calculated field. The Expression Editor provides you with the possibility to construct expressions using functions, operators, data source fields, report elements, constants and variables.



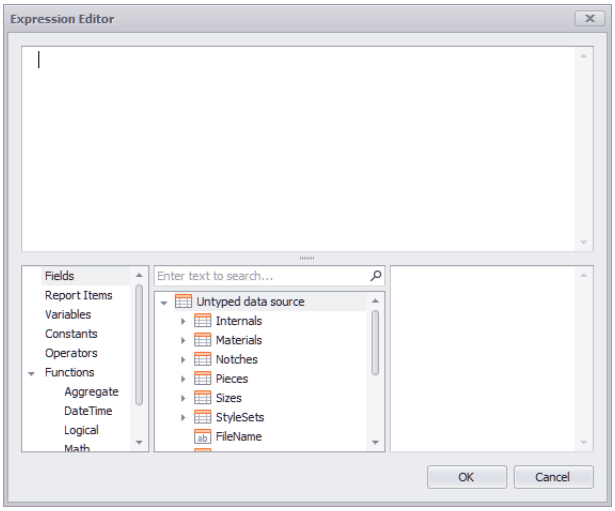
Once the expression is specified, click the OK button in the Expression editor. A newly added calculated field is now fully defined. Once it is available within the Fields list, it could be added to the report template exactly in the same manner as a variable. It could be either assigned to an existing label or dragged from the list in the report template design panel to create a new label.

To specify expression for control (e.g. label):

1. Select an existing label and click the smart tag icon to display the Label Tasks panel.



2. In the Label Tasks Panel, click the ellipsis icon in the Expression field to display the Expression Editor. The Expression Editor provides you with the possibility to construct expressions using functions, operators, data source fields, report elements, constants and variables.



3. Once the expression is specified, click the OK button in the Expression editor. The expression is defined and assigned to the current Label control. However, since the expression is assigned to the specific control and not to a variable, it does not appear in the Field List. Such an approach applies some limitations to the expression reuse.

4.5.2 Create your first expression

This topic provides you with the guided flow of the expression creation.

Generally, an expression is a string that, when parsed and processed, evaluates a value. Expressions consist of field names, constants, operators, and functions. Field names must be wrapped in square brackets. Here are examples of expressions:

*"[Quantity] * [UnitPrice] * (1 - [BonusAmount])"*

"[FirstName] + ' ' + [LastName]"

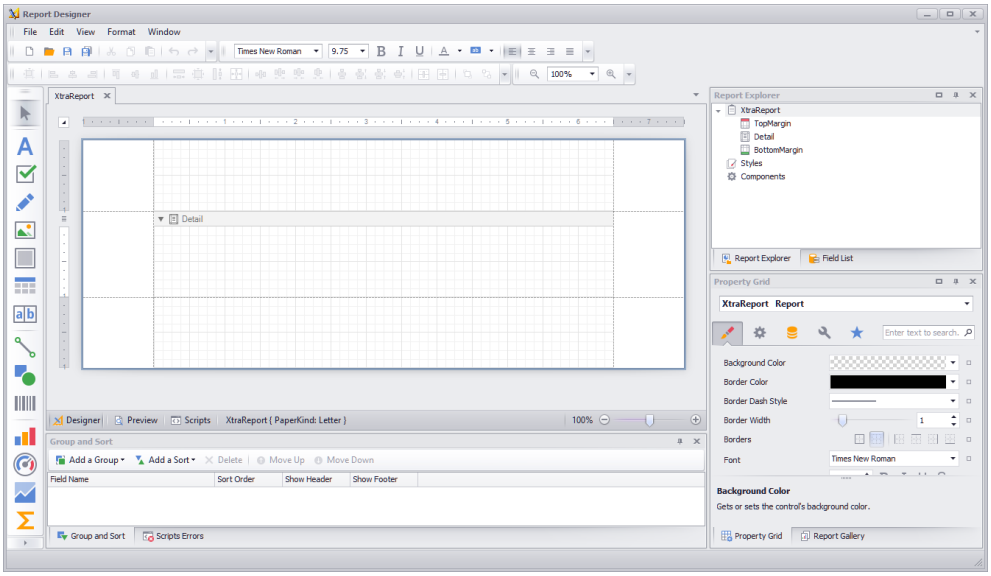
"[Country] == 'USA'"

"[OrderDate] > #8/16/1994# AND [Quantity] > 20"

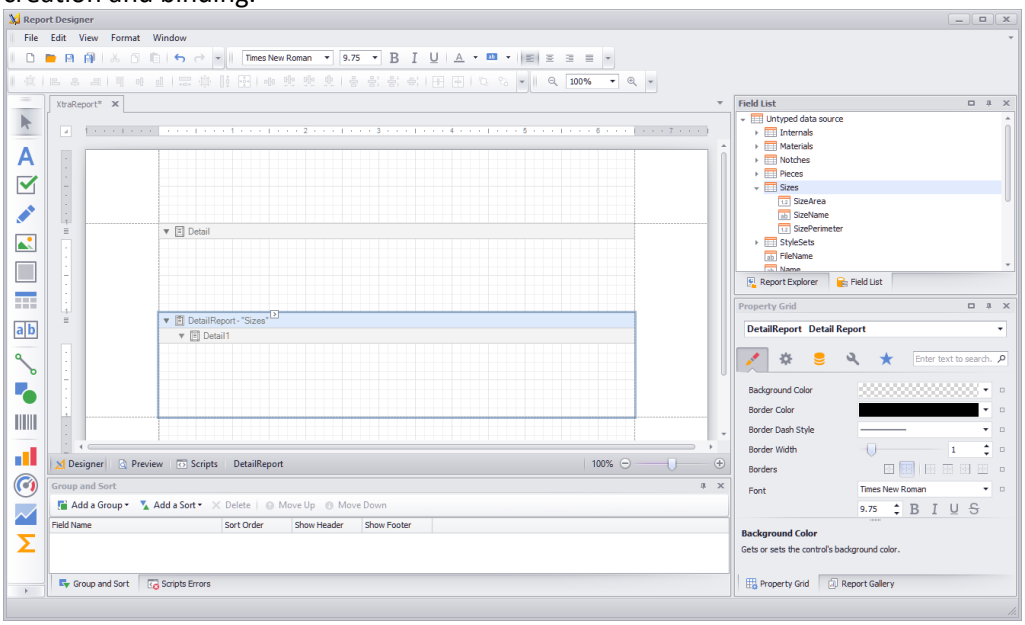
There is a list of operators (refer to topic 4.5.4.2), constants (refer to topic 4.5.4.1) and functions (refer to topic 4.5.4.3) that you can use in expressions.

The expression creation flow described through this exercise is based on PDS reporting functionality. Therefore, perform the following preparation actions prior to the expression definition.

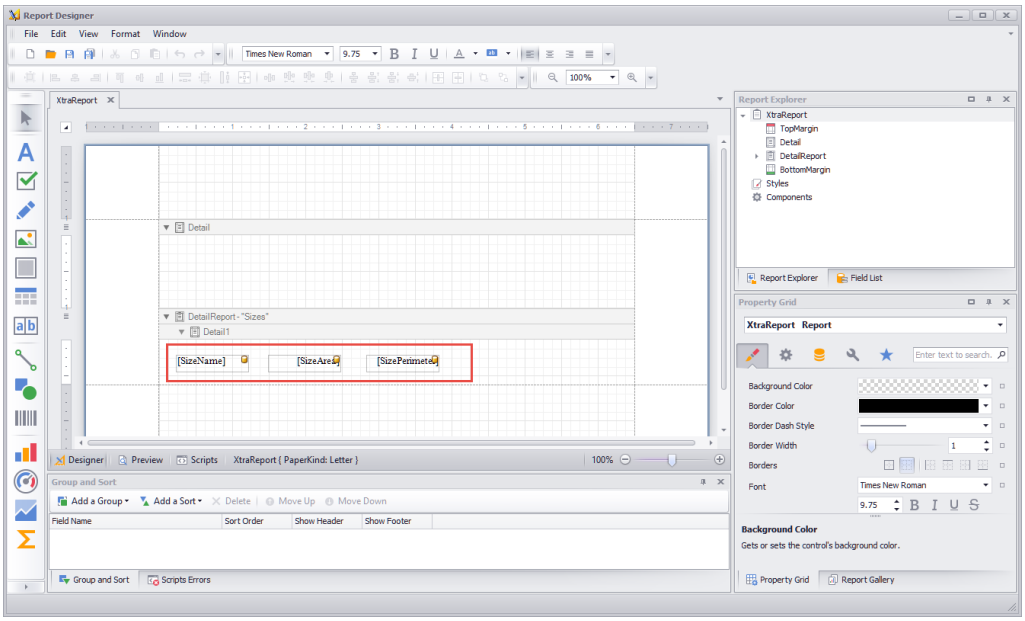
1. Launch the Report Designer from PDS; a new report template will be automatically created and loaded.



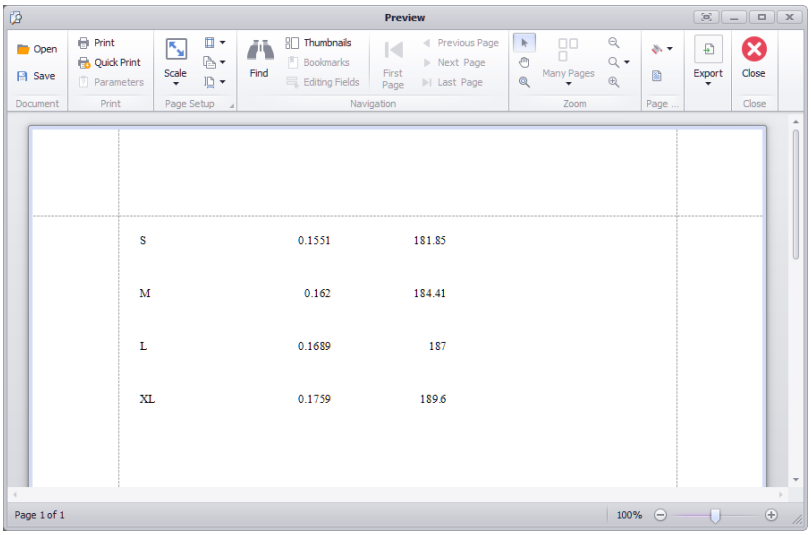
2. Under the Detail band of the report template create a sub-report connected to the Sizes list provided by the PDS data source. Refer to topic 4.2.2 for more information about sub-reports creation and binding.



3. Drag SizeName, SizeArea and SizePerimeter variables from the Sizes list available within the Fields List panel to the newly created sub-report.



4. Save the report template.
5. Close the Report Designer.
6. Run the report generation with the newly created template. Use one of the PDS files provided within sample pack for the report generation.

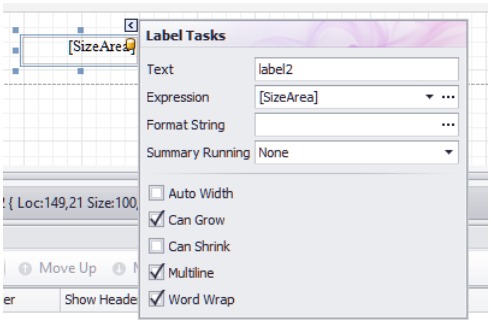


The currently generated report provides you with the list of all sizes used within the PDS file. For every size, the overall area and perimeter are printed. Values are printed using current units: for area it is sq. meters; for perimeter cm units are used. The area value is printed with 4 digits precision, the perimeter is printed with 2-digits precision.

The purpose of expressions described below is to round values of the area to 2-digits and round perimeter values to integer.

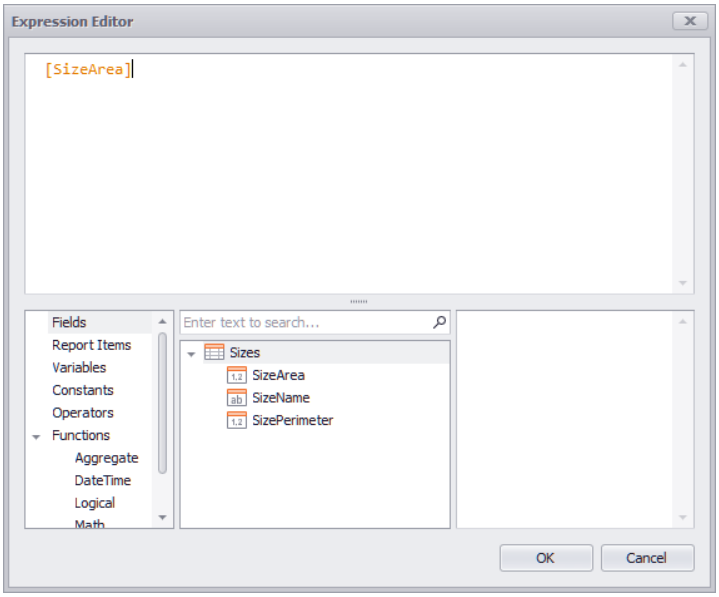
Perform the following actions:

1. Select the SizeArea label in the Design panel and select the smart tag to display the Label Tasks panel.



The Expression edit box provides you with the name of the variable [SizeArea] that is currently assigned to the field.

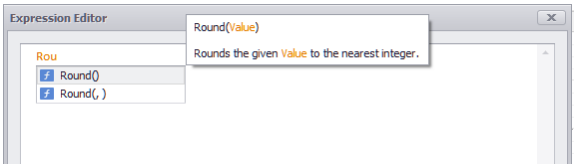
2. Click the ellipsis icon within the Expression field to load the Expression Editor.



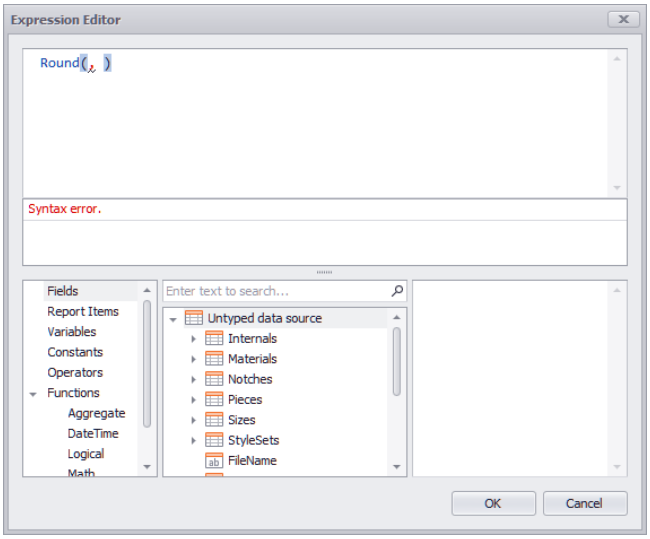
The Expression editor environment allows you to edit the expression.

Generally, the value rounding is performed using the *Round* function. The *Round(Value, Precision)* function rounds the given Value to a specified Precision.

3. Clear the existing variable name from the Expression area and start typing the Round function name. The Expression Editor supports syntax highlighting and intelligent code completion (suggesting functions and available data elements as you type). Once the typed name of the Round function is automatically recognized by intelligent code completion, the menu will be displayed as shown below, providing you with the possibility to choose the complete function name from the list.

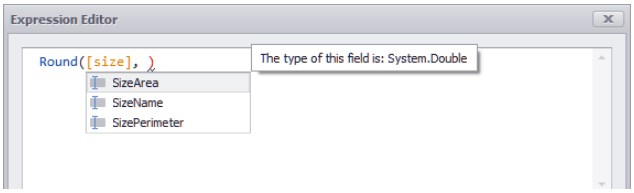


4. Choose the Round(,) function from the menu. Once it is selected, an empty (with no arguments) function appears within the edit area of the Expression Editor as shown below.

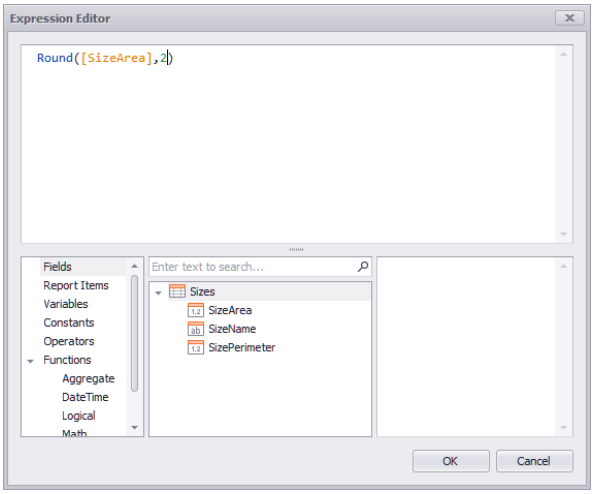


The Expression Editor displays all the errors it finds in the specified expression.

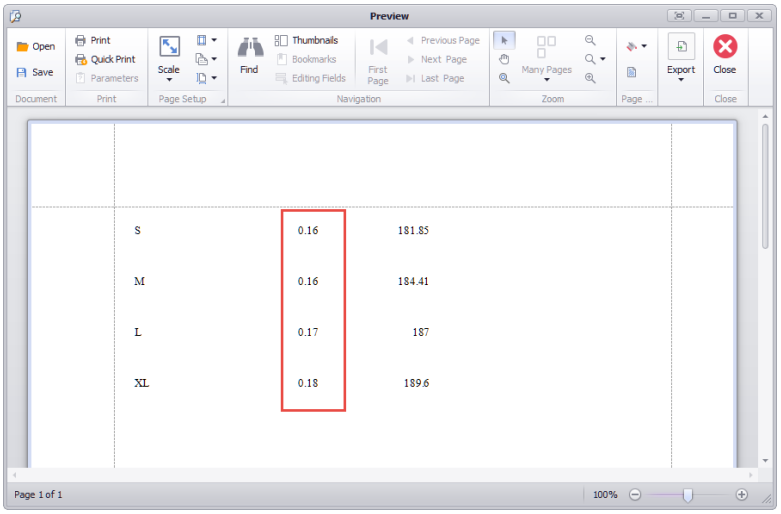
5. Type the name of SizeArea variable as the first argument for the Round function. According to the expression syntax described earlier, the variable name should be enclosed in a square bracket. The intelligent code completion is activated again to allow you to choose the variable name from the list.



6. Once the SizeName variable is chosen for the Round function, specify the rounding precision as the second argument. Type the value of 2 after the comma character within the function parentheses as shown below. Such a value allows you to perform the rounding to two decimal places.



7. Once the expression is completed, confirm the Expression Editor with the OK button. Make a note that the defined expression now is displayed within the label frame in the Design panel.
8. Save the report template and close the Report designer.
9. Generate the report using the updated template.



10. The area value provided for every size is rounded to two decimal places.

4.5.3 Variable Types

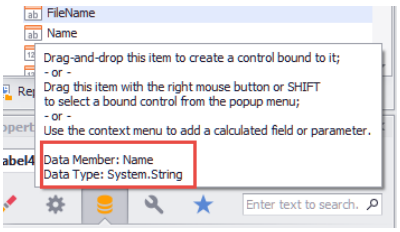
During the expression creation, it is critical to take into consideration the type of variables. The variable type defines the operations applicable for this variable. For example, in the case of rounding described earlier (see topic 4.5.2), the essence of operation imposes the necessity to operate floating-point values. Using integer, Boolean or string data with the Round function will not provide you with the required result or will lead to an error. Therefore, variables should be used in accordance to their type or converted to the required type prior to use (refer to topic 4.5.4.3 for type conversion functions).

The following are they types available for report template creation:

- **String**. A string is a data type used to represent text. It is comprised of a set of characters that can also contain spaces and numbers. E.g. “Fabric Name #1”. This is the type commonly used for various variables provided by Marker / PDS.
- **DateTime**. The DateTime type is used for values that contain both date and time parts. Typically, the DateTime value is displayed in 'YYYY-MM-DD hh:mm:ss' format, however it could be changed.
- **Int16**. The Int16 data type is used to handle integer numbers stored with 16-bit precision. Such a precision allows you to handle the following range: (-32768 to 32767).
- **Int32**. The Int32 data type is used to handle integer numbers stored with 32-bit precision. Such a precision allows you to handle the following range: (-2147483648 to 2147483647). This type is commonly used for PDS / Marker variables holding strongly integer values, e.g. the amount of bundles or quantity of notches.
- **Float**. The Float data type is used to handle double-precision (32-bit) floating-point values. In such a way, the Float type allows you to store values ranging from negative 3.402823e38 to positive 3.402823e38, as well as positive or negative zero.
- **Double**. The Double data type is used to handle double-precision (64-bit) floating-point values. In such a way, the Double type allows you to store values ranging from negative 1.79769313486232e308 to positive 1.79769313486232e308, as well as positive or negative zero. This type is commonly used for PDS / Marker variables holding floating-point values, e.g. area or length.
- **Decimal**. The Decimal type represents a decimal floating-point number using the 128-bit precision ranging from positive 79,228,162,514,264,337,593,543,950,335 to negative 79,228,162,514,264,337,593,543,950,335.
- **Boolean**. The Boolean data type allows you to handle logical variables that has one of two possible values (true or false).

Variables provided within Marker and PDS data sources for the report generation, most frequently use types of String (for all non-numeric data), Int32 (for integer values like piece or size quantity) and Double (for floating-point data like area, efficiency or yield).

The tooltip available for data source variables in the Fields list provides you with the information about the type of certain variable.



4.5.4 Constants, Operators and Functions

During the expression creation, you can use various constants, operators or functions to perform various operations with your data. For example, the rounding operation described earlier (see topic 4.5.2) was performed using the Round function provided within Report Designer.

The topics below provide you with the list of selected constants, operators and functions frequently used for the template creation.

4.5.4.1 Constants

The table below provides you with the list of selected constants frequently used for report template creation. For the full list of constants available within the Report Writer, refer to the DevExpress documentation.

Constant	Description	Example
String constants	Wrap string constants in apostrophes. If a string contains an apostrophe, double the apostrophe.	[Country] == 'France' [Name] == 'O''Neil'
Date-time constants	Wrap date-time constants in '#'.	[OrderDate] >= #2018-03-22 13:18:51.94944#
True	Represents the Boolean True value.	[InStock] == True
False	Represents the Boolean False value.	[InStock] == False
Numeric	Specify different numeric constant types in a string form using suffixes: Int32 (int) - *1* Int16 (short) - *1s* Byte (byte) - *1b* Double (double) - *1.0* Single (float) - *1.0f* Decimal (decimal) - *1.0m*	[Price] == 25.0m

4.5.4.2 Operators

The table below provides you with the list of selected operators frequently used for report template creation. For the full list of operators available within the Report Writer, refer to the DevExpress documentation.

Operator	Description	Example
+	Adds the value of one numeric expression to another or concatenates two strings.	[UnitPrice] + 4 [FirstName] + ' ' + [LastName]
-	Finds the difference between two numbers.	[Price1] - [Price2]
*	Multiplies the value of two expressions.	[Quantity] * [UnitPrice] * (1 - [BonusAmount])
/	Divides the first operand by the second.	[Quantity] / 2

%	Returns the remainder (modulus) obtained by dividing one numeric expression by another.	[Quantity] % 3
== =	Returns true if both operands have the same value; otherwise, it returns false.	[Quantity] == 10
!=	Returns true if the operands do not have the same value; otherwise, it returns false.	[Country] != 'France'
<	Less than operator. Used to compare expressions.	[UnitPrice] < 20
<=	Less than or equal to operator. Used to compare expressions.	[UnitPrice] <= 20
>=	Greater than or equal to operator. Used to compare expressions.	[UnitPrice] >= 30
>	Greater than operator. Used to compare expressions.	[UnitPrice] > 30

4.5.4.3 Functions

This topic provides you with the list of selected functions frequently used for report template creation. For the full list of functions available within the Report Writer, refer to the DevExpress documentation.

Aggregate Functions

Function	Description	Example
Avg(Value)	Evaluates the average of the values in the collection.	[Products].Avg([UnitPrice])
Count()	Returns the number of objects in a collection.	[Products].Count()
Max(Value)	Returns the maximum expression value in a collection.	[Products].Max([UnitPrice])
Min(Value)	Returns the minimum expression value in a collection.	[Products].Min([UnitPrice])
Sum(Value)	Returns the sum of all the expression values in the collection.	[Products].Sum([UnitsInStock])

Date-time Functions

Function	Description	Example
LocalDateTimeNow()	Returns a date-time value corresponding to the current moment in time.	AddDays(LocalDateTimeNow(), 5)
LocalDateTimeToday()	Returns a date-time value corresponding to Today.	AddDays(LocalDateTimeToday(), 5)
Now()	Returns the current system date and time.	AddDays(Now(), 5)
Today()	Returns the current date. Regardless of the actual time, this function returns midnight of the current date.	AddMonths(Today(), 1)

UtcNow()	Returns the current system date and time, expressed as Coordinated Universal Time (UTC).	AddDays(UtcNow(), 7)

Logical Functions

Function	Description	Example
Iif(Expression1, True_Value1, ..., ExpressionN, True_ValueN, False_Value)	Returns one of several specified values depending upon the values of logical expressions. The function can take *2N+1* arguments (*N* - the number of specified logical expressions): Each odd argument specifies a logical expression; Each even argument specifies the value that is returned if the previous expression evaluates to **true**; **...** The last argument specifies the value that is returned if the previously evaluated logical expressions yielded **false**.	Iif(Name = 'Bob', 1, Name = 'Dan', 2, Name = 'Sam', 3, 4)")

Math Functions

Function	Description	Example
Abs(Value)	Returns the given numeric expression's absolute, positive value.	Abs(1 - [Discount])
Ceiling(Value)	Returns the smallest integer that is greater than or equal to the numeric expression.	Ceiling([Value])
Floor(Value)	Returns the largest integer less than or equal to the numeric expression.	Floor([Value])
Max(Value1, Value2)	Returns the maximum value from the specified values.	Max([Value1], [Value2])
Min(Value1, Value2)	Returns the minimum value from the specified values.	Min([Value1], [Value2])
Power(Value, Power)	Returns a specified number raised to a specified power.	Power([Value], 3)
Round(Value)	Rounds the given value to the nearest integer.	Round([Value])
Round(Value, Precision)	Rounds the given value to the nearest integer, or to a specified number of decimal places.	Round([Value], 2)
Sqr(Value)	Returns the square root of a given number.	Sqr([Value])

ToDecimal(Value)	Converts Value to an equivalent decimal number.	ToDecimal([Value])
ToDouble(Value)	Converts Value to an equivalent 64-bit double-precision floating-point number.	ToDouble([Value])
ToFloat(Value)	Converts Value to an equivalent 32-bit single-precision floating-point number.	ToFloat([Value])
ToInt(Value)	Converts Value to an equivalent 32-bit signed integer.	ToInt([Value])
ToLong(Value)	Converts Value to an equivalent 64-bit signed integer.	ToLong([Value])

String Functions

Function	Description	Example
CharIndex(String1, String2)	Returns the starting position of String1 within String2, beginning from the zero character position to the end of a string.	CharIndex('e', 'devexpress')
CharIndex(String1, String2, StartLocation)	Returns the starting position of String1 within String2, beginning from the StartLocation character position to the end of a string.	CharIndex('e', 'devexpress', 2)
Concat(String1, ... , StringN)	Returns a string value containing the concatenation of the current string with any additional strings.	Concat('A', ')', [ProductName])
Contains(String1, SubString1)	Returns True if SubString1 occurs within String1; otherwise, False is returned.	Contains([ProductName], 'dairy')
EndsWith(String1, SubString1)	Returns True if the end of String1 matches SubString1; otherwise, False is returned.	EndsWith([Description], 'The end.')
Insert(String1, StartPosition, String2)	Inserts String2 into String1 at the position specified by StartPositon	Insert([Name], 0, 'ABC-')
Len(Value)	Returns an integer containing either the number of characters in a string or the nominal number of bytes required to store a variable.	Len([Description])
Lower(String)	Returns String in lowercase.	Lower([ProductName])
Remove(String, StartPosition)	Deletes all the characters from this instance, beginning at a specified position.	Remove([Name], 3)
Remove(String, StartPosition, Length)	Deletes a specified number of characters from this instance, beginning at a specified position.	Remove([Name], 0, 3)
Replace(String, SubString2, String3)	Returns a copy of String1, in which SubString2 has been replaced with String3.	Replace([Name], 'The ', '')

Reverse(String)	Reverses the order of elements within String.	Reverse([Name])
StartsWith(String1, SubString1)	Returns True if the beginning of String1 matches SubString1; otherwise, False.	StartsWith([Title], 'The best')
Substring(String, StartPosition, Length)	Retrieves a substring from String. The substring starts at StartPosition and has a specified Length.	Substring([Description], 2, 3)
Substring(String, StartPosition)	Retrieves a substring from String. The substring starts at StartPosition.	Substring([Description], 2)
ToStr(Value)	Returns a string representation of an object.	ToStr([ID])
Trim(String)	Removes all leading and trailing SPACE characters from String.	Trim([ProductName])
Upper(String)	Returns String in uppercase.	Upper([ProductName])

4.5.5 Expression Examples

This topic covers several examples of expression use.

4.5.5.1 Calculate total quantity of pieces within a style

To calculate total quantity of pieces in style, you'll need to calculate a total amount of instances for pieces and then summarize it for all pieces.

The total amount of instances for certain piece should be calculated by adding a new calculated field within the Pieces list and assigning an appropriate expression to this variable.

Within the Pieces list, create a new calculated field *PieceQuantityTotal* and assign the following expression to this field:

ToInt([PieceNumOfInstances])(ToInt([PiecePair])+1)*

Such an expression calculates a total amount of piece instances using the *PieceNumOfInstances* variable (which reflects the value specified for Quantity property available in PDS) and *PiecePair* variable (which reflects the Pair property available in PDS). The ToInt() function is used (see topic 4.5.4.3) to perform the conversion of string variables to integer. Such a conversion is strongly required as an enabler for math operations. The type conversion to integer is required.

At the top level of the PDS data source create a new calculated field *TotalQuantity* and assign the following expression:

[Pieces].Sum([PieceQuantityTotal])

The Sum() function used within this expression enables you to aggregate values of the *PieceQuantityTotal* variable for all pieces and summarize them.

When calculated fields are created, it is critical to maintain the data source hierarchy and define calculated fields within correct lists. In the example above, the *PieceQuantityTotal* expression should be evaluated for every piece; therefore the correct place of this calculated field in the data source hierarchy is the Peices list. The *TotalQuantity* calculated variable is evaluated once per style, therefore it should be defined in the top level of the data source. However, the expression of this variable aggregates data evaluated for every piece.

The place of certain calculated field in the data source hierarchical tree also defines the Detail Report or band where it could be placed within the template. The *TotalQuantity* variable should be placed in the Detail band connected to the root space of the data source; the *PieceQuantityTotal* could be placed within the sub-report connected to the Pieces list.

The calculated field or variable placement is optional; it means that it is not mandatory to place a certain variable within the report template to get it evaluated. During the report calculation, all values are evaluated regardless of their placement status. E.g. you can place the *TotalQuantity* variable without placing the *PieceQuantityTotal.*

4.5.5.2 Calculate area, perimeter and total quantity of pieces within a style set

The series of expressions below provides you with the possibility to calculate the total quantity of pieces in style sets as well as the total pieces area and perimeter per style set.

The total amount of instances for a certain piece within a style set is calculated in a manner similar to the calculation of the total amount of pieces described above. Within the SSetsPieces list, create a new calculated field *SSPieceQuantityTotal* and assign the following expression to this field:

ToInt([SSPieceQuantity])(ToInt([SSPiecePair])+1)*

In such a way, for every piece available in the certain style set, an amount of instances will be calculated. To calculate the area for all these instances, add a new SSPieceAreaTotal variable within the SSetsPieces list and assign the following expression:

*[SSPieceQuantityTotal]*ToFloat([SSPieceArea])*

Such an expression uses a previously evaluated *SSPieceQuantityTotal* variable as well as an area of a single piece instance exposed by the *SSPieceArea* variable provided by the data source. Make a note that the ToFloat function is used here to perform the conversion of the string value provided by the *SSPieceArea* variable to Float type.

In a similar manner, the total perimeter of all instances of the certain piece is calculated:

*[SSPieceQuantityTotal]*ToFloat([SSPiecePerimeter])*

4.5.5.3 Efficiency formatting

In Marker, the efficiency is evaluated as a floating-point number with the precision of 6 decimal places. For a certain report, it is necessary to round it to 3 decimal places and add the % sign after the value.

To do it, it is not necessary to create a calculated field; it's enough to add a new label control to the Design panel of the report template and assign the following expression:

Round([Efficiency], 3) + ' %'

This expression utilizes the Round() function (see topic 4.5.4.3) to perform the rounding to the specified amount of decimal places. The mathematical addition operation between the number and string (containing the % character) results in two atomic operations: implicit conversion of the number to string and concatenation of two strings. In the result of this expression, an initial efficiency value is rounded, and the string of efficiency value accompanied with the % character is printed.

4.5.5.4 Composing size / quantity string

Frequently, it is necessary to provide a size/quantity data within the report using the following format:

S1:S2:S3:S4:SN= Q1:Q2:Q3:Q4:QN

Where S1 is the name of the first size, Q1 is the quantity of bundles for the first size; S2 is the name of the first size, Q2 is the quantity of bundles for the first size and so on. In the example below, such a size/quantity string is generated for sizes 36, 38, 40, 42, 44 and 46 where the quantity for every size is equal to 1.

36: 38: 40: 42: 44: 46 = 1: 1: 1: 1: 1: 1

To compose such a string, it is necessary to perform following actions:

1. Under the SizeList list, create a *StringSize* variable and assign the following expression:

   [SizeName] + ': '

2. Under the SizeList list, create a *StringQuantity* variable and assign the following expression:

   [NumberOfBundlesPerOrder] + ': '

Since both *StringSize* and *StringQuantity* variables are located under the SizeList, they are evaluated per every size. Once evaluated, these variables provide you with the name of the size and number of bundles ordered for this size respectively. The output data is available as a string; the colon character is added to the end of the string to serve as a separator.

3. Under the StyleList, add a new *StringAllSizes* variable and specify the following expression:

Remove([SizeList].Sum([StringSize]), len([SizeList].Sum([StringSize]))-2)

Such an expression generates a sum of StringSize values generated for all sizes; the summation of String type variables means the concatenation of all separate size strings. E.g. In a case of 36, 38 and 40 size, separate “36 :”, “38 :”and “40 :” strings will be concatenated into a “36: 38: 40 :” string. The Remove() function (see topic 4.5.4.3) is utilized to trim the last two characters from the concatenated string (space and colon).

The Remove () function provides you with the possibility to trim a string (provided as a first function argument) by removing characters starting from a certain position (the position starting from which all characters are removed, is provided as a second argument of the function). Since it is necessary to remove two characters, the position from which the removal starts is determined using the Len() function (see topic 4.5.4.3), which returns the length of the string, i.e. the amount of characters appearing within a string (which is provided as a function argument). So, the *len([SizeList].Sum([StringSize]))-2* equation returns you the length of the concatenated string reduced by 2. The resulted value is used as a secondary argument of the Remove function.

It is extremely important to preserve the correct syntax of the entire equation. Key points to pay attention to are:

- Correct use of parentheses used to specify arguments of functions used through the expression.
- Correct use of variables and lists.

4. In a similar manner define a new *StringAllQuant* variable with the following expression:

Remove([SizeList].Sum([StringQuantity]), len([SizeList].Sum([StringQuantity]))-2)

This expression allows you to define similar concatenated strings containing all quantities (separated with the colon character) and remove the last two characters, space and colon from the generated string.

5. Finally, when both *StringAllSizes* and *StringAllQuant* variables are added and fully supplied with expressions as described above, create a label in the Design area and specify the following expression for this label:

[StringAllSizes]+' = '+[StringAllQuant]

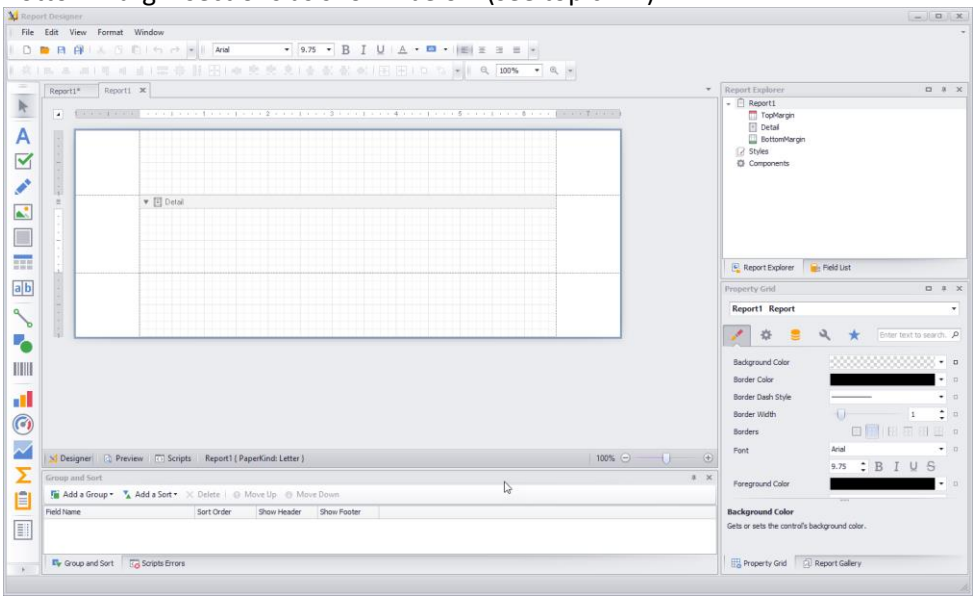
This expression allows you to compose an initially required string consisting of size data and quantity data separated with the “=” character.

5. Create Your First Marker Report

This topic provides you with a guided exercise of the sample report template creation in Marker.

To create a report template, perform the following steps:

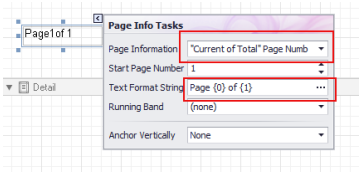
1. Launch Report Designer from the Marker application.
2. Create a new report template. An empty template provides you with the TopMargin, Detail and BottomMargin sections as shown below (see topic 4.2).



3. Add a new Page Info control (see topic 4.3.1 for more information) to the TopMargin section.

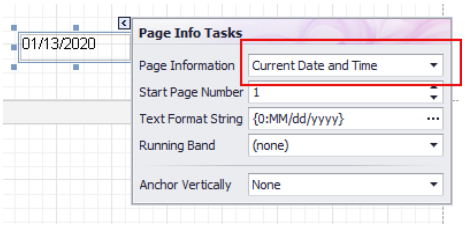
   Once the control is added, use the smart tag to display the Page Info Tasks panel.

   In the panel, choose the “Current of Total” Page Number option for Page Information to specify the type of data that should be displayed by the control. In the Text Format String edit box, specify *Page {0} or {1}* to define the output format. In this format string {0} specifies a number of the current page and {1} is used to specify the total number of pages.

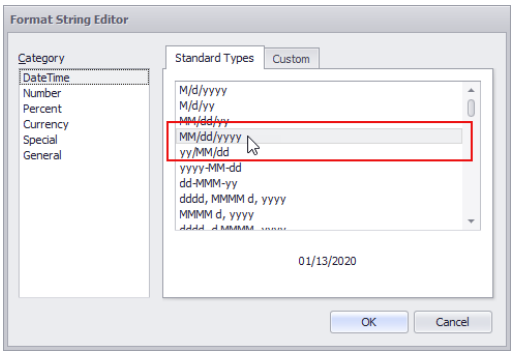


4. In a similar manner, add a new Page Info control to the TopMargin section to display the report generation date.

   In the Page Info Tasks panel, choose the Current Date and Time option for the Page Information.



In the Text Format String edit box, proceed to the Format String Editor and select the MM/dd/yyyy format for the date display as shown below.

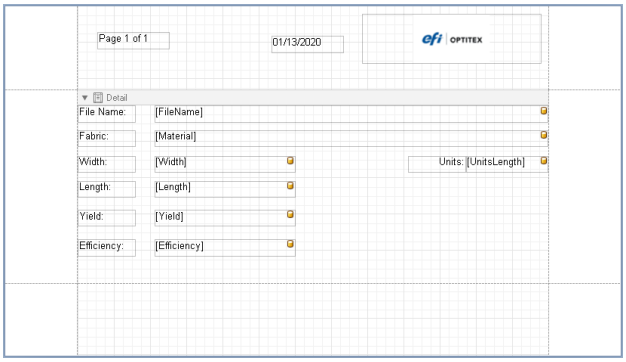


5. Add new Picture Box control to the TopMargin section (see topic 4.3.3 for more information) and select a logo image to be displayed at the top of the report page. Use an appropriate Sizing option to display the image with necessary scale.
6. Once created, a report template created via Marker is automatically bound to the top level of the data source which describes a Marker file itself.

   Drag the following parameters from the Untyped data source list available in the Fields List to the Detail section:

   - FileName
   - Material
   - Width
   - Length
   - Yield
   - UnitsLength
   - Efficiency

   Accompany each parameter with an appropriate label describing the parameter.



7. Round the Efficiency value. Launch the Expression Editor for the just added Efficiency parameter. In the Expression Editor, specify the following expression to round the evaluated efficiency value to the third decimal (see topic 4.5.4.3 for more information about rounding), and add the % character at the end.

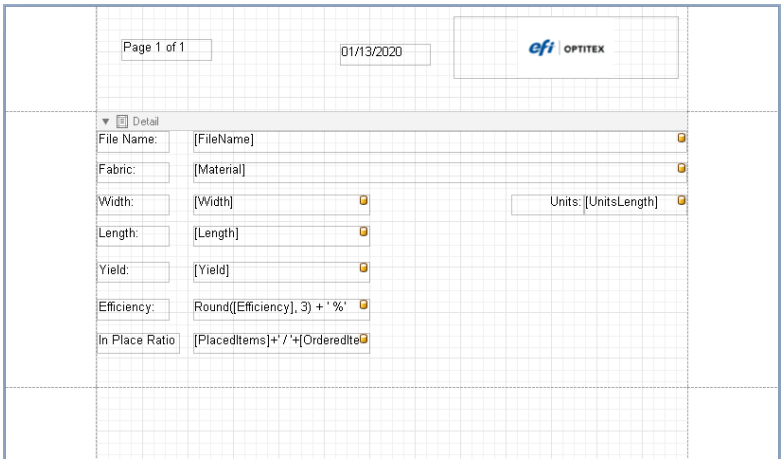
   Round([Efficiency], 3) + ' %'

8. Add a new label to the Detail section and launch the Expression Editor. In the Expression Editor, specify the following formula:

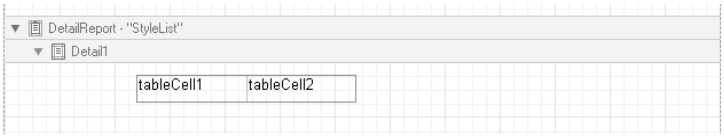
   [PlacedItems]+' / '+[OrderedItems]

   Such a formula provides you with the possibility to display the In-Place Ratio, which is a ratio between the number of placed bundles and the number of ordered bundles.

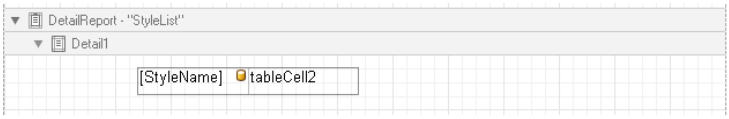
   Accompany the added expression with an appropriate label containing the description.



9. Add a Detail Report section bound to the StyleList object available in the Fields list. Once it is added, create a two-column table as shown below.



   Drag the StyleName variable from the Field List to the first cell of the table.



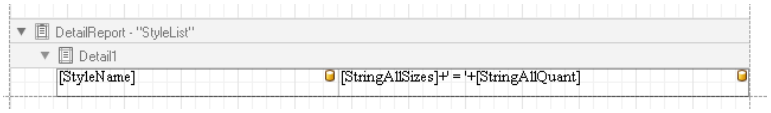
10. In the same manner as described earlier (see topic 4.5.5.4), add the size / quantity string expression to the second cell.

    Add *StringSize* and *StringQuantity* variables to the SizeList with the respective formulas.

    Add corresponding *StringAllSizes* and *StringAllQuant* variables to the StyleList with the respective formulas.

Associate the cell with the following expression:

[StringAllSizes]+' = '+[StringAllQuant]

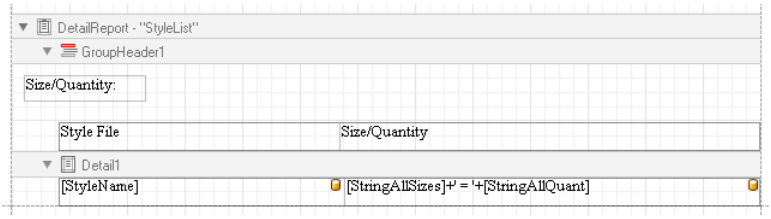


11. Add a new Group Header section to the Style List Detail Report.

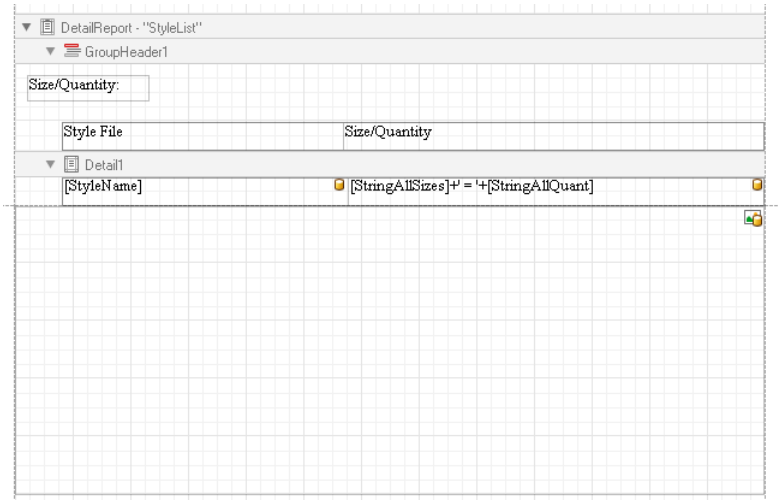
    Create a new two-column table within this section to represent a header row for the just added sizes/quantity table.

    Add labels to cells of the newly created table as shown below.

    Add a standalone label on top of the header table and specify the label text describing the entire table.



12. Add a marker image. The data source provides you with the Preview variable referring to the image of the current marker file layout. Drag the Preview variable from the Fields List to the BottomMargin section as shown below.



    Adjust the image width to occupy the entire width of the report page. Set the Sizing option for the picture box to the Zoom Image (see topic 4.3.3 for details) to fully display the marker layout image preserving its proportions.

13. At this stage the report template definition is completed. Save it and run the report generation with the just created template.

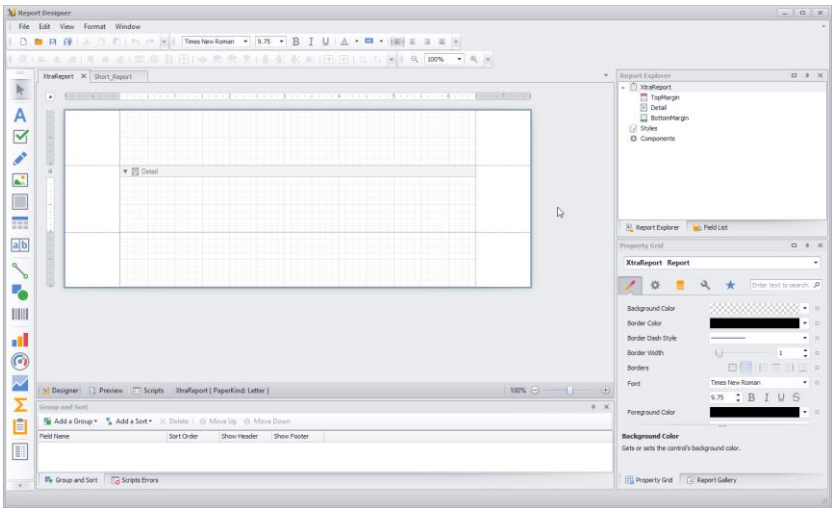

Page 1 of 1
1/13/2020
efi OPTITEX
File Name:
Fabric: T1
Width: 100cm
Units: cm
Length: 30.8cm
Yield: 0.89cm
Efficiency: 80.657 %
In Place Ratio: 24 / 24
Size/Quantity:
Style File
447743_YEHN0.PDS
Size/Quantity
36: 38: 40: 42: 44: 46 = 1: 1: 1: 1: 1: 1

6. Create your First PDS Report

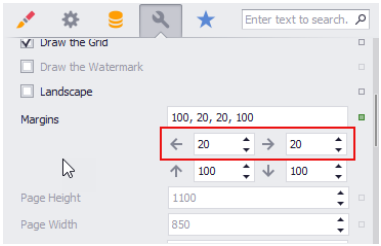
This topic provides you with a guided exercise of the sample report template creation in PDS.

To create a report template, perform the following steps:

1. Launch Report Designer from the PDS application.
2. Create a new report template; an empty template provides you with the TopMargin, Detail and BottomMargin sections as shown below (see topic 4.2).



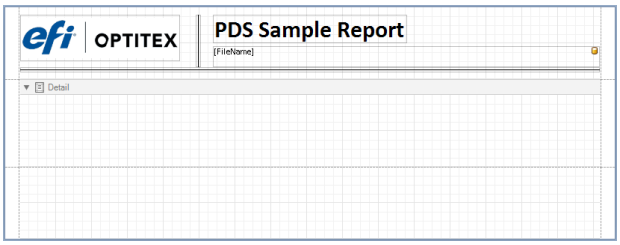
3. Define the report page margins. Select the entire report in the Report Explorer and switch to the Miscellaneous page in the Property Grid panel.
   Set the Right and Left margins to 20.



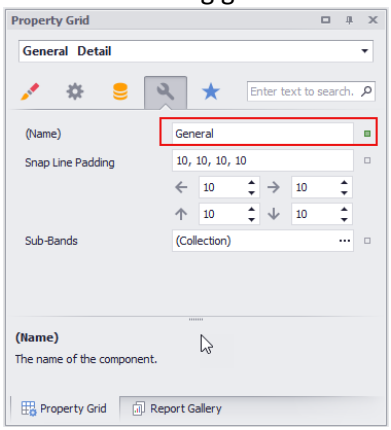
4. Add a new label to the TopMargin section to display the report name, and type the label text: PDS Sample Report.
5. Add a new Picture Box to the TopMargin section (see topic 4.3.3 for more information) to display the logo at the top of the report page. Use an appropriate Sizing option to display the image with the necessary scale and preserve proportions.
6. Once created, a report template created via PDS is automatically bound to the top level of the data source which describes a PDS file itself.

   Drag the FileName variable from the Untyped data source displayed within the Fields List to the TopMargin section. The name of the actual PDS file used for the report generation will appear on top of the report page.

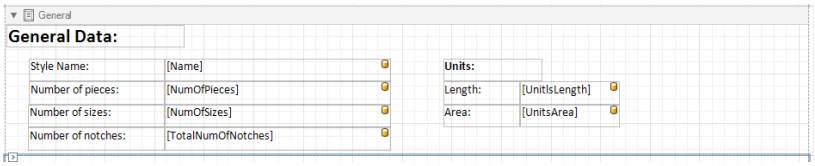
7. Use the Line tool to add two splitting lines to the TopMargin section as shown below, to separate the company logo and header.



8. In the Miscellaneous page in the Property Grid panel, rename the Detail section to General to emphasize the content of this section. This section is automatically bound with the Untyped data source containing general data of the current PDS file.



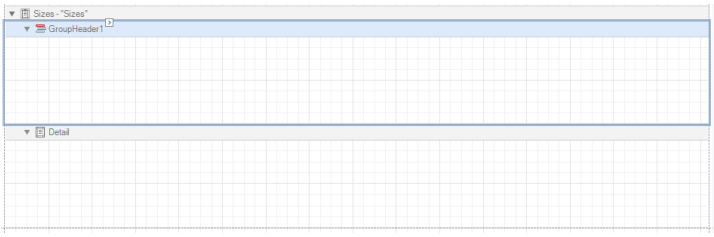
9. Using the Fields List, drag the following parameters to the General section:
   - Name
   - NumOfPieces
   - NumOfSizes
   - TotalNumOfNotches
   - UnitsLength
   - UnitsArea
10. Place labels for the section header as well as for the added variables description, as shown below.



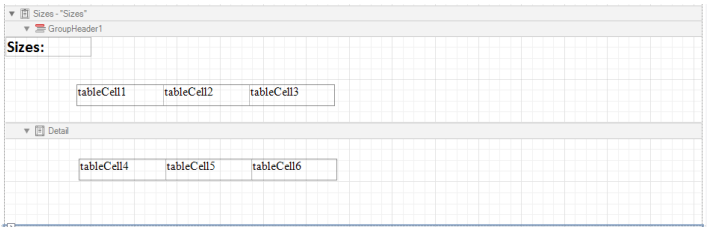
11. Insert a new Detail Report bound to the Sizes list available under the Untyped data source. Rename the added section to Sizes in order to describe the section content.

    The Added Sizes detail report section by default contains the Detail section, which is dedicated for size variables that will be evaluated for every size.

    Add a Group Header band to the Sizes section to place headers and data evaluated once for all sizes.



12. Add a Sizes header to the just added group header section.
13. Add two new three-column tables to the Detail and Group Header bands of the Sizes section, as shown below.



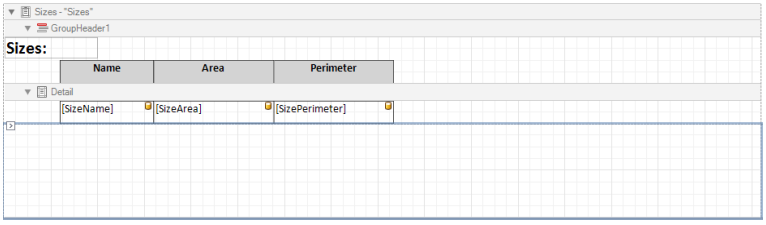
- The table in the Detail section is dedicated for the sizes data display; as mentioned above, this table will contain data evaluated for every size.
- The table located within the Group Header band is dedicated for the header preceding sizes data.

Using such a pattern of two tables (for data and for header) allows you to display evaluated data in a regular tabular manner.

14. Add header labels to the header table as shown below. Change the appearance of the header row accordingly using the Property Grid.
15. Drag SizeName, SizeArea and SizePerimeter variables from the Sizes List to the cells of the data table.

As mentioned above, the table located within the Detail band is evaluated for every size. In such a way, a template table will provide you with a single row within the generated report. To keep smooth transition and continuality between rows within the table, it is recommended to adjust the height of the Detail section and the height of the table. When these properties are equal, the final generated table provides you with rows appearing adjacently with no empty space between.

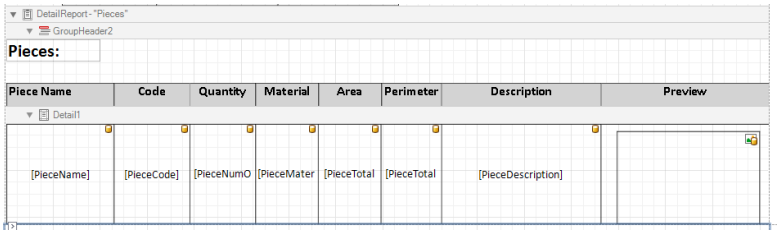
In the example below, the height of the Detail band and table are adjusted. In addition to it, the header row is aligned with the bottom boundary of the respective band. Such a layout will provide you with a continuous table.



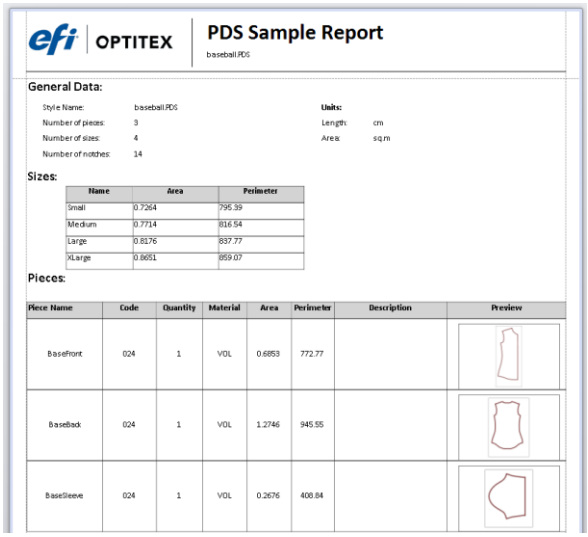
16. In a similar manner as described above, add a new detail report below the detail report just created for Sizes. Add a group header to the report and create header and data tables as shown below.

    Add the following variables from the Piece List to the data table.

    - PieceName,
    - PieceCode,
    - PieceNumOfInstances,
    - PieceMaterial,
    - PieceTotalArea,
    - PieceTotalPerimeter,
    - PieceDescription
    - Preview



17. At this stage the report template definition is completed. Save it and run the report generation with the just created template.
    The generated report will contain the general information about the current style file as well as a summary about sizes the pieces available within the current document.



7. Advanced Reporting

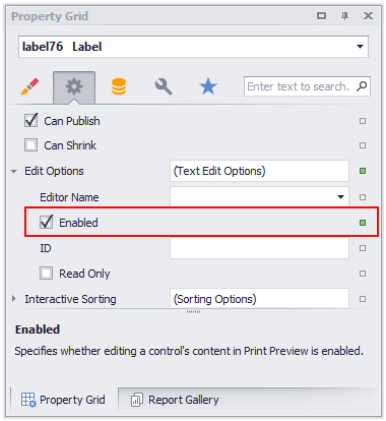
This chapter provides you with a description of the various advanced aspects of reporting use.

7.1. Customize Generated Parameter Value

As described in this document, the report generation procedure utilizes a report template and evaluates values for all variables available within the template. Once the report data is evaluated, the Preview window is displayed providing you with the WYSIWYG preview of the generated report. Typically, by default all values appearing within the report are displayed with disabled editing capabilities. However, Report Writer provides you with the possibility to allow you to customize the evaluated value of a certain variable directly during the preview.

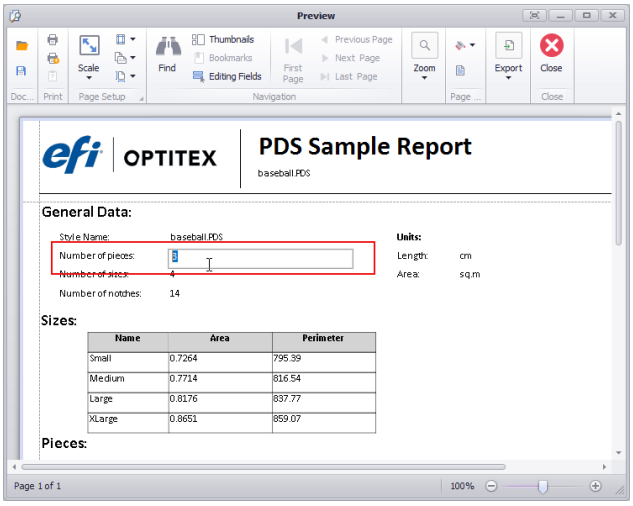
To allow the editing of a certain variable, perform the following actions:

1. With the variable control selected, switch to the Behavior tab in the Property Grid control as shown below and navigate to the Edit options group.
2. Turn on the Enabled option to activate the editing capabilities.



3. Save the report template and generate the report.
4. Once the report is generated and the Preview window is displayed, click on the generated value of the variable where the editing was enabled.

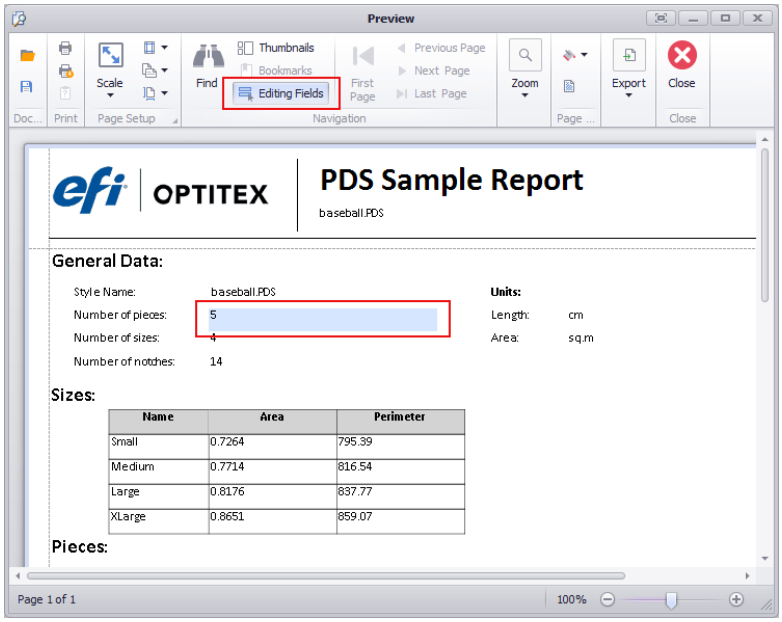
   The edit box allowing you to customize the value will be displayed in place of the value as shown below.



   Once confirmed, the customized value will be displayed within the report.

The editing capabilities for numeric and textual fields are available in the regular manner via edit boxes displayed on clicking the corresponding value within the Preview window. For images added via picture box control, clicking on the control provides you with the regular File Open dialog box which allows you to choose an image to be displayed.

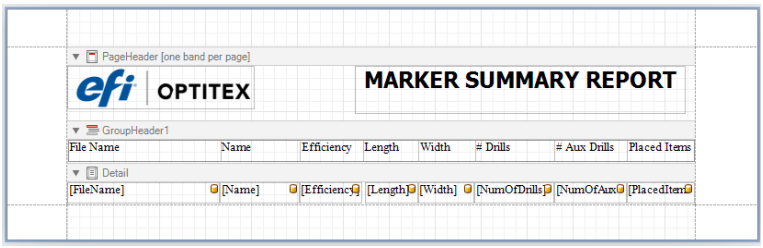
By default, when the Preview window is displayed, fields with enabled editing capability are not highlighted. Use the Editing Fields button from the toolbar as shown below to highlight these fields.



7.2. Aggregation Reports

In certain cases, it is necessary to generate an aggregated or summary report containing data from multiple data sources: either style files (*.pds) or marker files (*.mrk). When such a report is generated for multiple files, the Detail band of the report is evaluated for every data source in a sequence (see topic 4.2). In such a way, the Detail band of the report template should contain all the necessary variables that should be evaluated during the report generation.

The example shown below provides you with the layout of a template for the summary Marker report.



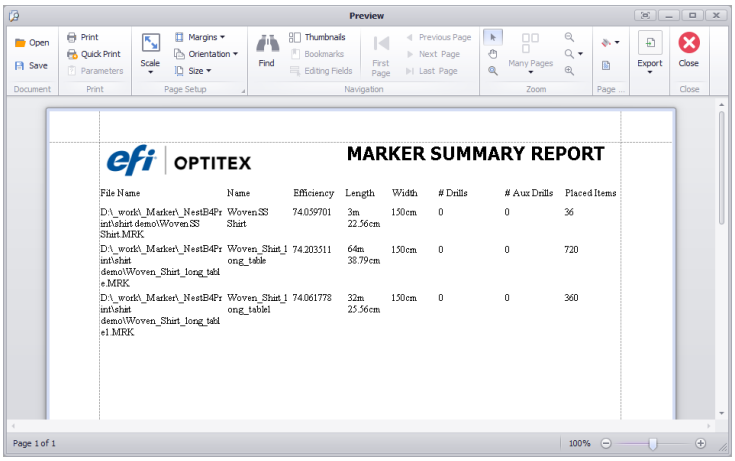
In this template, the evaluated data appears in a tabular format. The Detail band provides you with a data row containing the following variables:

- FileName
- Name
- Efficiency
- Length

- Width
- NumOfDrills
- PlacedItems

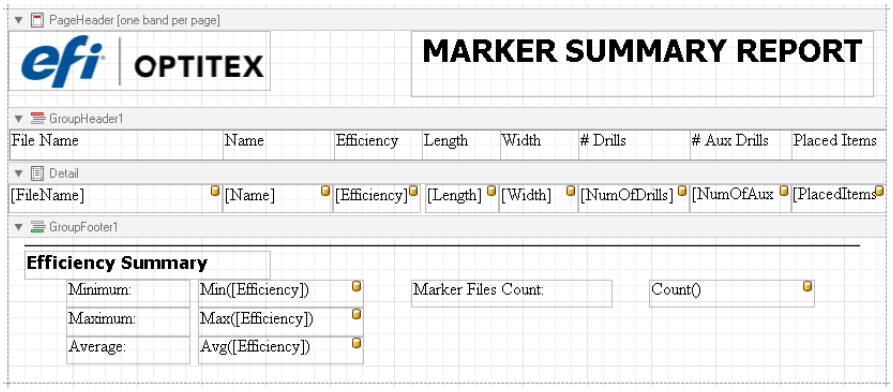
The table header row is defined using unbound Label controls within the Group Header band.

In such a way, the generated report will consist of a single header row and multiple data rows providing you with evaluated data for every Marker file participating in the report generation.

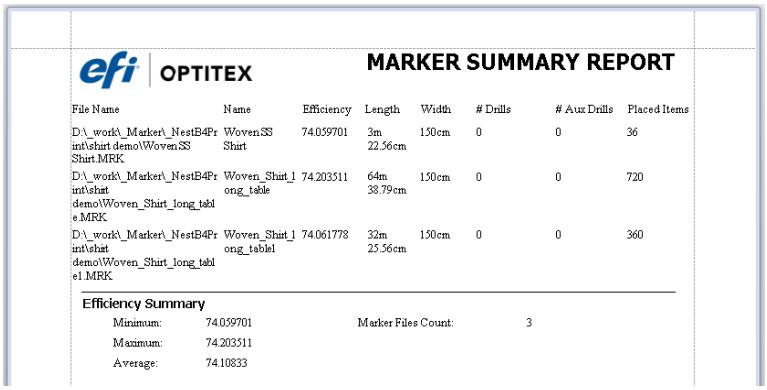


Statistical calculations could also be performed within a summary report. If such calculations are required, corresponding expressions should be added respectively to those bands that are generated once per report.

In the exercise below, minimum, maximum and average efficiency are evaluated as well as the amount of marker files participating in the report. Generation of such a report for all marker files processed during a shift allows you to get a basic overview of materials use.



Min(), Max(), Avg() and Count() functions (see topic 4.5.4.3) are used respectively to obtain necessary values from multiple data sources and perform calculations.



OPTITEX

MARKER SUMMARY REPORT

File Name	Name	Efficiency	Length	Width	# Drills	# Aux Drills	Placed Items
D:_work_Marker_NestB4Print\shirt demo\WovenSS Shirt.MRK	WovenSS Shirt	74.059701	3m 22.56cm	150cm	0	0	36
D:_work_Marker_NestB4Print\shirt demo\Woven_Shirt_long_table.MRK	Woven_Shirt_long_table	74.203511	64m 38.79cm	150cm	0	0	720
D:_work_Marker_NestB4Print\shirt demo\Woven_Shirt_long_table1.MRK	Woven_Shirt_long_table1	74.061778	32m 25.56cm	150cm	0	0	360

Efficiency Summary

Minimum: 74.059701

Maximum: 74.203511

Average: 74.10833

Marker Files Count: 3

7.3. Using Cross Tabs

Optitex Report Writer provides you with the Cross Tab control which allows you to create a pivot table for multi-dimensional data analysis. The Cross Tab summarizes large amounts of data in a cross-tabular format that can be sorted, grouped and filtered. These sorting, grouping and filtering are key capabilities of a Cross Tab, differentiating it from a regular table.

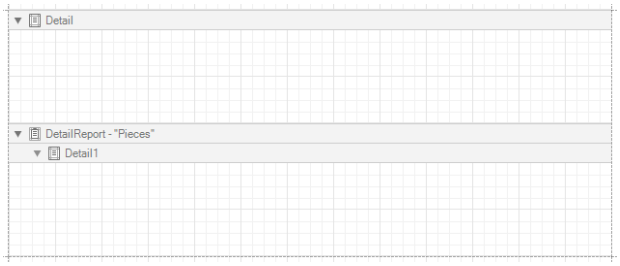
For example, pieces geometry appearing in a style file contains some number of various type notches. There are T, V and L notches where part of them should be cut in a regular manner and part of them should be punched. Dimensions of these notches also vary even for a certain type.

A regular band-based table (see topic 4.4) bound to Pieces.PieceNotches list (see topic 4.1.1) allows you to render data for all notches for every piece. It means that every certain notch will be represented by a single row in the table where the corresponding notch parameters / properties appear as well.

A pivot grid allows you to perform multi-dimensional grouping of data, i.e. all notches of a certain type could be grouped together. Within this type-based group all notches with the same plot/cut command (e.g. Punch) will be grouped as well. In the last turn, notches with the same type and cut command could be grouped by dimensions. Such a grouping allows you to display an aggregated data, you could display in the report the number of notches of certain type, cut command and dimension. A Cross Tab providing you with an aggregated notch data could also provide you with a drill-down to certain notch.

Perform the following actions to create a PieceNotches pivot table:

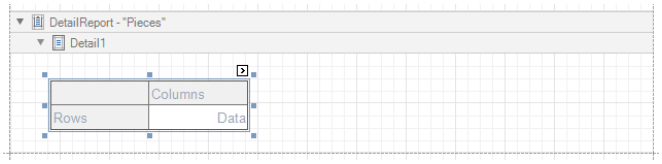
1. Add a Piece detail report to the report canvas as shown below. Make sure that binding to the data source objects is performed in the correct manner.



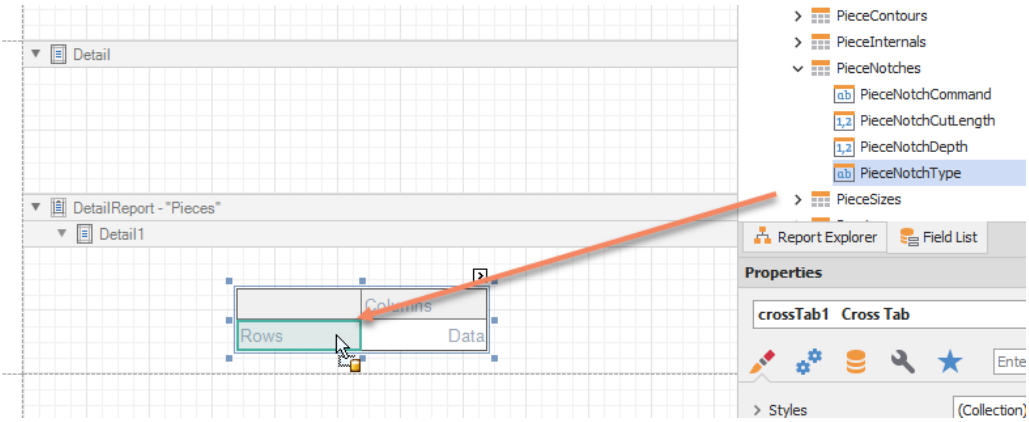
According to the band evaluation rules (see topic 4.2.1), the Pieces detail report will be evaluated for every certain piece within the style file. For a certain piece, the Pieces.PieceNotches object could be evaluated for every certain notch belonging to the current piece. This object provides with an entire set of parameters describing a certain notch.

The pivot grid is dedicated to displaying aggregated data; therefore, the pivot grid can be added to the Pieces detail report section and provide you with aggregated data for the underlying Pieces.PieceNotches list.

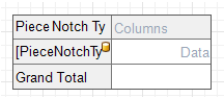
2. Choose the Cross Tab control from the Standard Controls toolbar (see topic 3.2.6) and click on an empty space within the Pieces detail report section. A new default cross tab will be added.



3. Expand the PieceNotches list in the Fields List tab and drag the PieceNotchType variable as shown below to the Rows cell in the newly created cross tab.



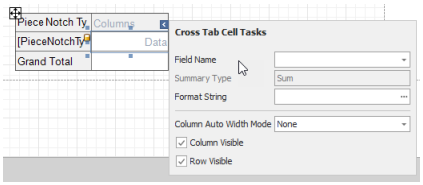
Once the variable is dragged, the current cross tab control is automatically bound to the PieceNotches list. The dragged variable name will appear in the target cell. In addition to it, the header cell will be automatically added and set to Piece Notch Type.



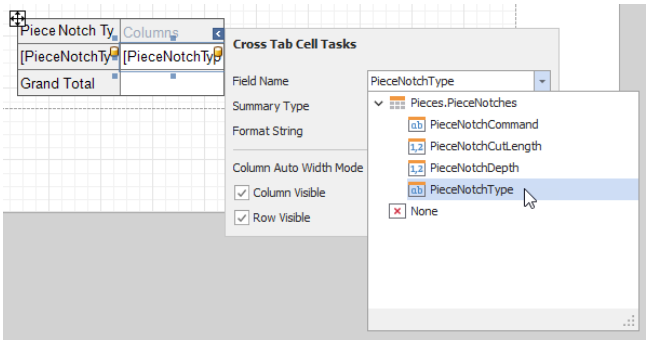
During the report generation, a new row will be populated for every unique notch type. Notch types will appear in the corresponding column as shown below.

Piece Notch Ty	
I	
T	
V	
Grand Total	

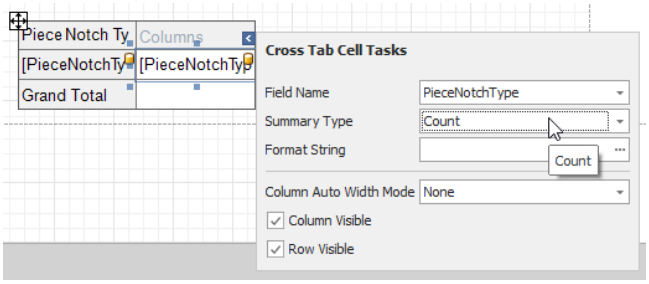
4. To add quantity of notches per certain type to the table, click on the Data cell of the cross tab control and then click on the smart tag adjacent to the selected cell.



In the Field Name drop down, choose the PieceNotchType variable from the list.



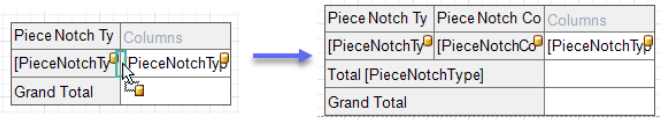
In the Summary Type drop down, select the Count option as shown below.



In such a way, the count of instances will be calculated for every certain notch type (populated using the PieceNotchType variable) and displayed within the corresponding cell. The Count header will be automatically added during the report generation.

Piece Notch Ty	Count
I	7
T	5
V	2
Grand Total	14

5. To add the notch command to the table, drag the corresponding PieceNotchCommand variable to the right border of the PieceNotchType cell. Dragging a variable to the right border allows you add a new column in the cross tab directly after the target border.



During the report generation, the Command will be added to the consideration and displayed in the corresponding cells for every notch type. Make a note that notch quantity displayed in the Count cells now will appear for every certain command.

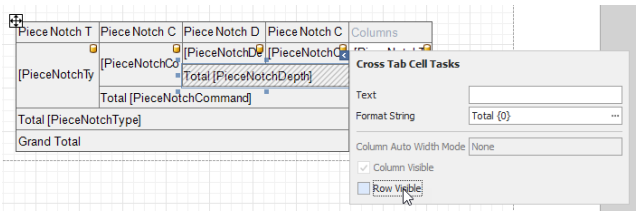
Piece Notch Ty	Piece Notch Co	Count
I	Cut	3
	Draw	4
Total I		7
T	Cut	3
	Punch	2
Total T		5
V	Cut	2
Total V		2
Grand Total		14

Total and Grand total cells

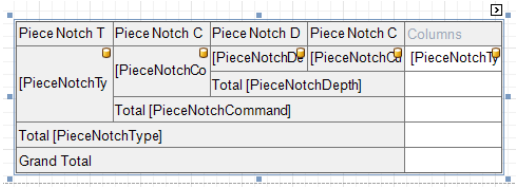
Total and Grand Total rows are automatically added to the cross tab control to handle the total data. The Total row is automatically added for every variable column added after the first one. In such a way, the generated report could provide you an intermediate total for the certain variable. In the current example case, the intermediate total provides you with the count of notches per certain notch type.

The Grand Total row is added as the last row of the cross tab. This row provides you with the data, aggregated for the entire cross tab. In the current example case, the Grand Total row provides you with the count of notches per all notch types.

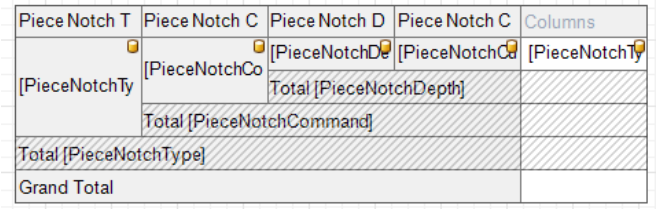
The visibility of total rows could be controlled with the Row Visible check box available in the Cross Tab Cell Tasks panel which is available with the smart tag adjacent to a total cell.



6. In the same manner, as explained above, drag PieceNotchDepth and PieceNotchCutLength variables from the Fields List to create additional columns in the cross tab.



7. Select intermediate total cells of PieceNotchType, PieceNotchCommand and PieceNotchDepth variables and turn off their visibility as explained above.



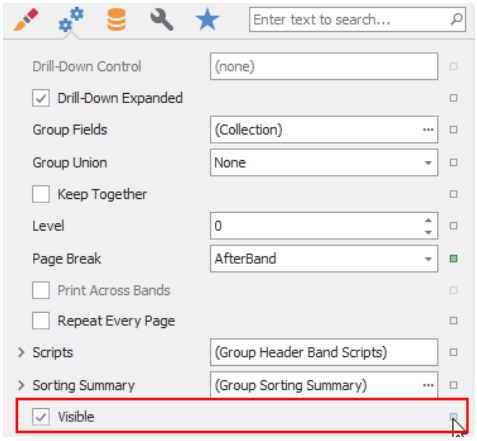
The intermediate total data will be hidden from the generated report.

Piece Notch T	Piece Notch C	Piece Notch D	Piece Notch C	Count
I	Cut	0.7	0.7	2
		0.8	0.8	1
	Draw	0.635	0.635	4
T	Cut	0.7	1.2	3
	Punch	0.6	1.1	1
		0.7	1.2	1
V	Cut	0.7	1.487	1
		0.8	1.676	1
Grand Total				14

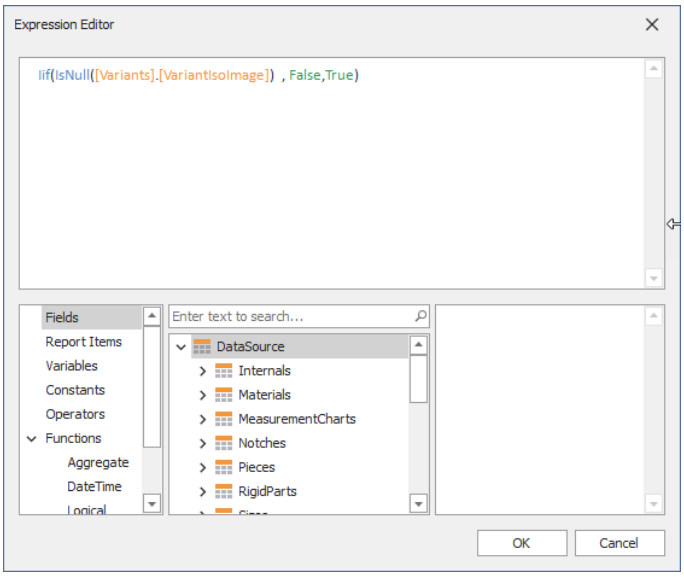
The resulted cross tab provides you with the aggregated notch data organized by type, command depth and cutting length. The notches count is displayed for every combination of these parameters. The Grand Total row provides you with the total count of notches for the actual piece.

7.4. Conditional visibility of data

The Visible check box available in the Properties panel for certain controls or entire report bands allows you to control over the control/band visibility.



The Expression icon □ located inline with the Visible check box provides you with the possibility to control over the check box state via expressions. Once the icon is clicked, the Expression Editor dialog box is displayed providing you with the possibility to define an expression intended to control the check box state.



The logical expression **Iif** provides you with the possibility to set the check box state conditionally depending on variables.

Expression format:

Iif(FirstClause, FirstClauseTrueValue, ..., LastClause, LastClauseTrueValue, FalseValue)

This expression returns either a value defined for the first Boolean condition that is True, or the last value if none of these conditions is True. This function can accept any number of arguments. The Clause parameter (e.g. FirstClause) should be represented as a logical expression. When such an expression is true, the value of ClauseTrueValue (e.g. FirstClauseTrueValue) is returned. In a case when the expression is false, the next clause expression is evaluated. The last argument of the Iif expression (FalseValue) is returned in a case when none of clause conditions is true.

In the example below, the Iif expression is used to control over the visibility of the band.

Iif(IsNull([Variants].[VariantIsoImage]) , False,True)

In a case when an image represented by the VariantIsoImage evaluated within the Variants list does not exist (is null), the False value is returned. The value of True is returned in the opposite case.

Returned True/False values define the status of the Visible check box, so the True value means the selected state of the check box causing the band display; the False values means the unselected check box causing hiding of the band.