List & Label® 15

www.combit.net

Table of Contents

1. Introduction

With the List & Label Designer, you can create or edit output forms which are used to present information taken either from a database or other data source.

The designer, print preview and export dialog are generally opened from within an application with a menu item (e.g. Configuration > Print > Labels or Output > Mail Merge > Labels).

1.1 General

The List & Label Designer considers output forms to be "**projects**". In a project, you will find all actual layout information, as well as the layout details including page size and orientation, fonts, colors, frames, circles, lines, pictures etc. if relevant to the project.

The List & Label Designer recognizes three types of projects - lists, labels, and cards (often complex label projects).

The individual elements of a project are called "**objects**". In this manner, a label project can, for example, consist of an object for the sender line, the address area, and the logo. These objects contain the information required for printing such as the actual contents, fonts, alignments, word wrapping, colors, etc.

The List & Label Designer provides different types of objects which can be freely placed and changed in size on your workspace. Depending on its type, an object can display information or have various different properties.

A	Texts can contain fixed text as well as the variable content of individual records.
╲	Lines for which layout options such as line width and color can be defined.
▭	Rectangles are rectangular frames for which layout options such as frame width, colors, filling pattern and shading can be defined.
⬭	Circles and Ellipses for which layout options such as frame width, colors and filling pattern can be defined.
🖼	Picture objects allow you to integrate graphics (fixed file or variable) into projects.
242	Barcodeobjects allow you to display fixed text or variable contents as barcodes.
▦	Report Container or Tables (depends of application): The report container contains tables, charts and crosstabs. Tables allow you to display fixed text and variable contents from various records.
📊	Charts object allow you to create diagrams.

	Crosstab allow you to analyze and display different kinds of data in several dimensions.
	Formatted Text objects can change the format within a line (unlike text objects).
	Form Controls permits input and changes within the Preview.
	HTML Texts can display the contents of web sites and other HTML formatted text.
	OLE Containers serves as a container for various OLE-Server-Documents. In this way, you can integrate documents created with Word, Excel, Visio or MapPoint into your project. .
	Form Templates can only be inserted by using Objects > Insert > Form Template. They are placed, as templates, in the background of the workspace and are used for the exact placement of objects. These templates are helpful in the design of complex forms. The form templates are exceptional in that they are not printed later.

You normally position the required objects in the workspace with your mouse, and then define the corresponding contents and layout properties. Alternatively, you can just drag a variable from the variable list per drag & drop, and place it on your workspace.

In order to edit an existing object you need to select it. Just click into the required object and the object will be selected. A selected object can be identified by its raised frame. When a new object is created, it is automatically "selected" and can be edited or changed immediately. The property dialog of an object can be activated with a double-click. Please be aware that the selection tool needs to be activated. See chapter "Object Bar".

The following subchapters provide you with a summary of the methods and procedures to be used for the creation and editing of projects. The following is a typical sequence of steps in this procedure:

1. Define the page layout
2. Set preferences and options (only necessary once)
3. Insert objects
4. Edit objects
5. Save the project

1.2 Help Functions

Context sensitive help

A context sensitive online help with comprehensive information about all the functions is available to you. You can open this directly with the "?" > Context sensitive menu. Alternatively, you can call up the help you need regarding a command or a dialog by pressing F1 where you need it. In dialog windows, there is also a help button at the bottom left corner. The online help regarding the relevant subject is opened with this. This context sensitive help is available for almost all commands and dialogs.

Online Help

Another way of finding information is to open the online help through "?" > Contents and then to use the search function to find the required topic.

Tooltips for dialog controls

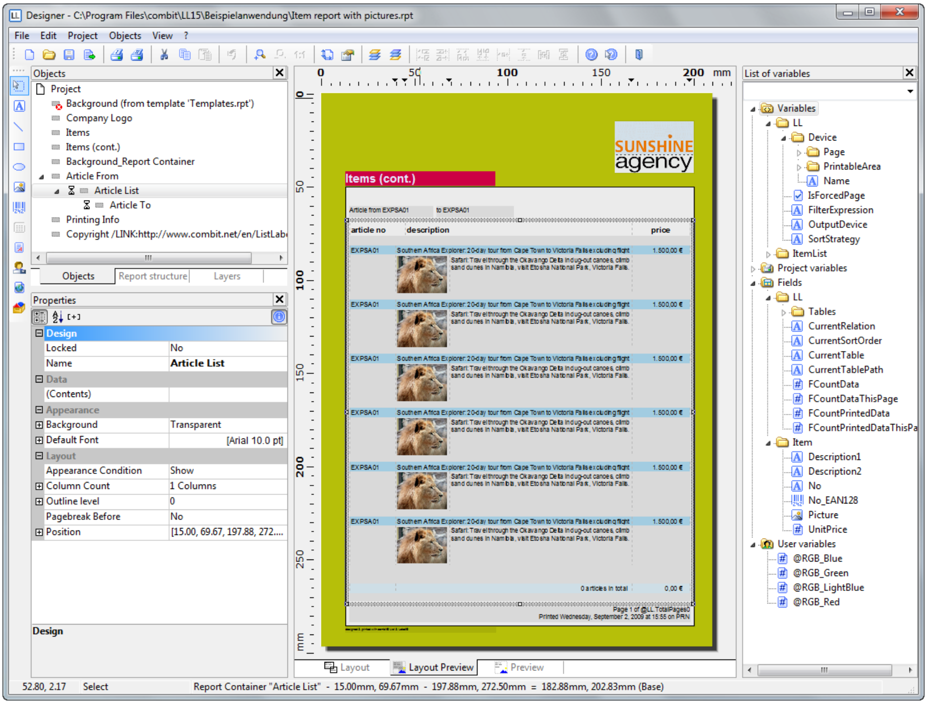
Many dialogs and buttons have tooltips. These are small help texts, which automatically appear when you keep the cursor still over a dialog element.

2. General Methods and Procedures

This chapter is an introduction to the List & Label Designer and the functions which are available to you.

In the appendix you will find a description of the functions which allow you to manipulate strings and numbers.

But, to start, let's take a look at the layout and the individual tools.



Designer - C:\Program Files\combit\LL15\Beispielanwendung\Item report with pictures.rpt

Design
Locked: No
Name: Article List
Data
(Contents)
Appearance
Background: Transparent
Default Font: [Arial 10.0 pt]
Layout
Appearance Condition: Show
Column Count: 1 Columns
Outline level: 0
Pagebreak Before: No
Position: [15.00, 69.67, 197.88, 272....

52.80, 2.17 Select Report Container "Article List" - 15.00mm, 69.67mm - 197.88mm, 272.50mm = 182.88mm, 202.83mm (Base)

2.1 The Designer Interface

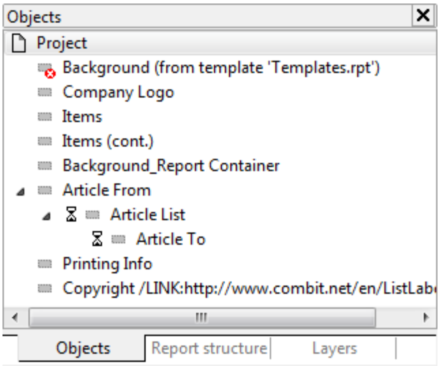
Fixed text appears in the workspace as defined in the project, variables and fields will be replaced with a predefined sample text. The representation of real data occurs in the preview, assuming that is supported by your application.

2.1.1 Dockable and Floating Tool Windows and Toolbars

You can freely define the way space is partitioned in the List & Label Designer. The windows with the most importance for the design of your layouts are the "tool windows". Tool windows allow themselves to be either integrated into the Designer window or opened as independent dialogs that stand as single windows on the desktop. The workspace assumes a special function, it is the only window that is constantly visible and is attached to the Designer window. The other tool windows, for example the Property List or the Object List, allow themselves to be docked to any part of the workspace.

Multiple tool windows can be docked together and, by using the tab strips, brought to the foreground.

To do this, select the tool window to be moved by depressing and holding the left mouse button in the title bar. You can now drag the tool window to any desired position. A rectangular frame will be presented which shows the position and size at which the tool window will be placed. If you drag it into the title bar of a previously docked tool window, it will be placed behind the window(s) at this position. If it is dragged into the upper portion of a previously docked window, then the position will be shared and the newly docked window will be in the upper portion. If you drag it into the lower portion of a docked window then the position will be shared and the new window will be in the lower portion. The same is true for the left and right portions of the (docked) tool windows.



If you place a tool window in a different position, it will not be docked but will be a movable "floating" window.

The workspace has the behavior of a previously docked window, with the differences that it cannot be selected and moved and other windows cannot be docked behind it.

Tip: When a tool window is dragged away from its position, it can be returned to that position but there will be no partitioning. To repartition, the window must first be placed in another position and then returned to share the space with its "colleague".

To set the size relationships of the docked windows, you can place them on the borders of their neighbors and adjust the size using the mouse. The position will then be partitioned anew. This functions in both the horizontal and vertical directions.

2.1.2 Toolbars

Both of the Designer's toolbars can be docked onto the four edges of the Designer window or presented as freely movable (floating) windows. Click with the mouse either in the narrow area at the left or top of the appropriate toolbar, or in the title bar of the toolbar if it is presented as an independent window.

Tool windows and toolbars can be turned on/off by using the command View > Windows >... .

Tip: Preferences, such as window position, size, display, etc. will be globally saved for every application List & Label is integrated into. They are then valid for all List & Label projects of the same type (lists, labels or file cards).

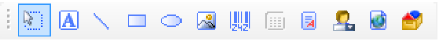
2.1.3 Object Bar

Some of the tools available in the Designer can be accessed via the button bar. With a mere button click you are offered a direct short-cut option.

These buttons are self-explanatory: if you remain on the button for a short time without pressing a mouse key, a tool tip appears. This bar can be placed/removed by choosing View > Windows >Toolbar 'Objects'.

2.1.4 Toolbar

In the same manner, menu items can also be directly selected via the toolbar (as a shortcut). Just click on the button you require.

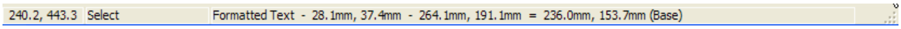


2.1.5 Context menu

The most important commands that are related to an object can be activated via a context menu. The title bar contains the object's name and type.

By clicking with the right mouse button in the margins, a small context menu opens which can be used for the quick opening of the Object List dialog.

2.1.6 Status Bar



240.2, 443.3 Select Formatted Text - 28.1mm, 37.4mm - 264.1mm, 191.1mm = 236.0mm, 153.7mm (Base)

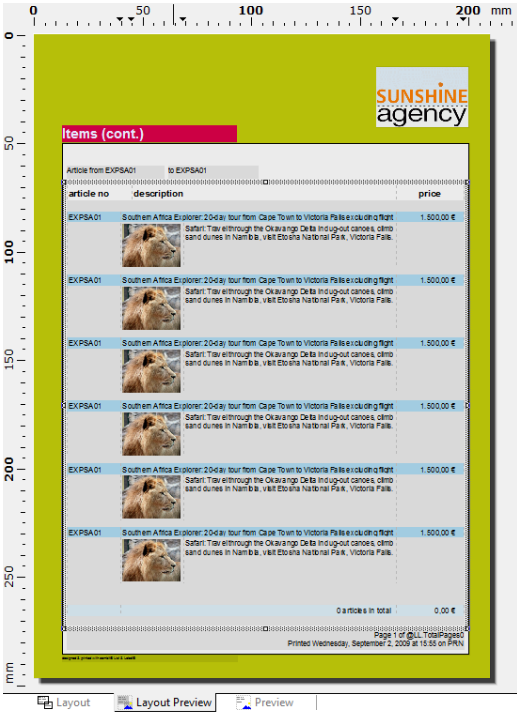
The status bar is divided into three sections:

- the left area indicates the current mouse position in millimeters or inches from the upper left workspace corner.
- the middle area indicates the current mode of operation. (for example, select, draw rectangle, etc.)
- the right area shows the name or type, relative appearance level, as well as the size of a selected object.

- the margin between the object's upper left corner in relation to the same corner of the workspace
- the margin between the object's lower right corner in relation to the same corner of the workspace

2.1.7 Workspace

The workspace is the area in which objects can be changed and defined. The shape and size of the workspace depends on the default values of the paper size and alignment (see: Project > Page Setup).



Items (cont.)

Article from EXPSA01 | to EXPSA01

article no	description	price
EXPSA01	Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight Safari: Travel through the Okavango Delta in dug-out canoes, climb sand dunes in Namibia, visit Etosha National Park, Victoria Falls.	1.500,00 €
EXPSA01	Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight Safari: Travel through the Okavango Delta in dug-out canoes, climb sand dunes in Namibia, visit Etosha National Park, Victoria Falls.	1.500,00 €
EXPSA01	Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight Safari: Travel through the Okavango Delta in dug-out canoes, climb sand dunes in Namibia, visit Etosha National Park, Victoria Falls.	1.500,00 €
EXPSA01	Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight Safari: Travel through the Okavango Delta in dug-out canoes, climb sand dunes in Namibia, visit Etosha National Park, Victoria Falls.	1.500,00 €
EXPSA01	Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight Safari: Travel through the Okavango Delta in dug-out canoes, climb sand dunes in Namibia, visit Etosha National Park, Victoria Falls.	1.500,00 €
EXPSA01	Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight Safari: Travel through the Okavango Delta in dug-out canoes, climb sand dunes in Namibia, visit Etosha National Park, Victoria Falls.	1.500,00 €
	0 articles in total	0,00 €

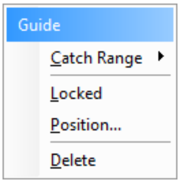
Page 1 of @LL.TotalPages0
Printed Wednesday, September 2, 2009 at 15:55 on PRN

Rulers and Scale

The rulers form a frame around the workspace. The position of the mouse is displayed on the rulers with the use of markers. You can always see the coordinates of the mouse, which are additionally displayed in the Status Bar.

The rulers can be turned on/off with the menu item View > Windows > Rulers.

To ease the entry of data and objects within the workspace, guides can be used in a project. Depress the right mouse button while on one of the rulers, drag into the workspace and let go. The new guide then has the same alignment as the ruler. All guides allow themselves to be relocated, and a snap function assists you in the exact placement of objects. The objects are not permanently attached to the guides; the guides only provide assistance in the placement.

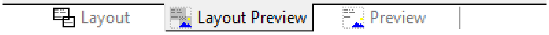


The guide options are defined with a context menu. You can set the "catch range" for each guide in pixel, and the objects will automatically be pulled to the guide when they enter the set catch range.

You can make the guides non-relocatable so that you do not accidentally move them within the workspace. The snap function is turned off when you depress the Ctrl-key while placing objects.

View Modes

At the bottom of the workspace you can select the view mode:



- In Layout mode you can see the object frames and the contents of the objects as formulas. This mode is ideal for exact positioning and is the fastest mode.
- Layout Preview mode shows the objects in WYSIWYG mode. If the system DLL MSIMG32.DLL is present (normally from Windows 98 or Windows 2000 upwards) objects will be painted transparently in the layer color.
- The Preview mode is identical to the separate preview window. The separate preview window is automatically closed when this mode is active. In contrast to the Preview Window, you can edit all objects.

Options in the view mode

In the view menu, you can find various options regarding the appearance of the designer. In order to work precisely even with complex projects, there are several levels of magnification available:

Whole Page (no zoom)	alt+1
Zoom 2x	alt+2
Zoom 4x	alt+3
Zoom 8x	alt+4

The current magnification is shown by a tick in the view menu.

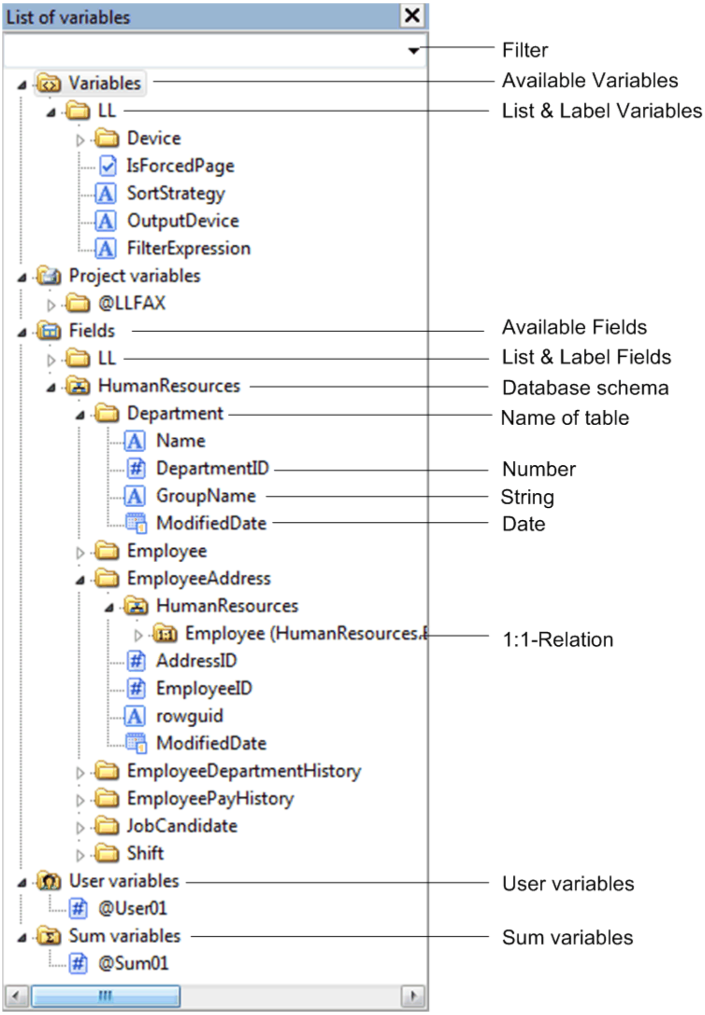
In the menu View > Windows you can hide/show the tool windows (except for the workspace), rulers and toolbars. If you wish, you can hide the preview window,

variable list, **object** window, **properties** window and **layers** window to gain more space.

2.1.8 Variable List

The Variable List displays all variables which are available in the current project; for list type projects, all available fields are additionally displayed.

Fields, variables, and internal List & Label variables are distinguished in the hierarchical list. Fields contain the data that changes from line to line in a table object, while variables usually only change from page to page.



If you wish to assign variables or fields to existing objects, you can simply drag the variable from the list and drop it on the respective object. List & Label does the pasting for you (drag & drop).

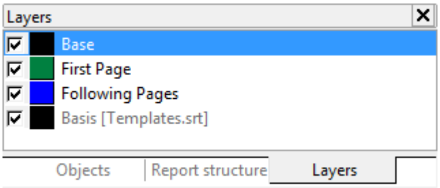
If you drag a variable to a free areain the project workspace, a new text object is automatically generated.

2.1.9 Layers Window

The Layer Window shows the various layers or levels of a project. This window can be repositioned on the workspace at your will.

You can improve the overview in complex projects by assigning objects to an individual layer. You can switch layers on and off at any time by using the Layers Window. You can give these layers their own appearance conditions. For example, one layer may only be valid for the first page of a multipage project, and the second layer for the remaining pages.

More detailed information on working with layers can be found in Chapter "Working with Layers".

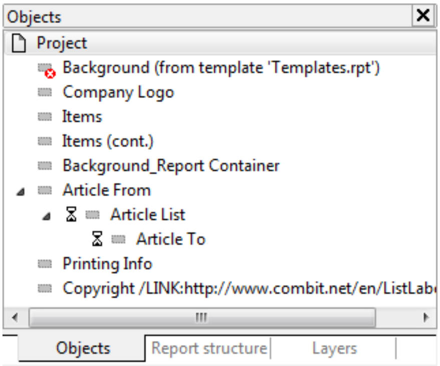


Layers

☑ ■ Base
☑ ■ First Page
☑ ■ Following Pages
☑ ■ Basis [Templates.srt]

2.1.10 Objects Window

Another tool window is the hierarchical Objects List. All objects in the project are displayed with a small rectangle and its name in the sequence that they will be printed. The name can be edited by clicking on the current name. The most important commands for the objects are contained in the context menu or in the workspace. Objects that are in a hidden layer have parentheses around the rectangle in front of their names. Locked objects are identified by a small X.



Objects

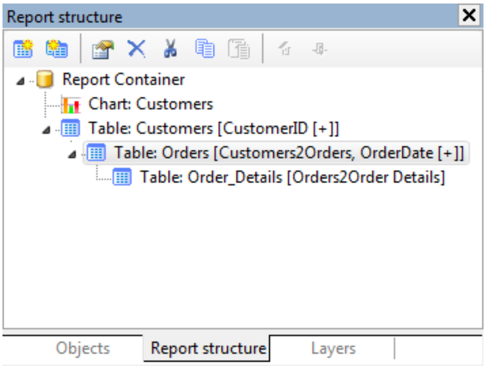
- Project
  - Background (from template 'Templates.rpt')
  - Company Logo
  - Items
  - Items (cont.)
  - Background_Report Container
  - Article From
    - Article List
      - Article To
  - Printing Info
  - Copyright /LINK:http://www.combit.net/en/ListLab

The sequence of the objects can be changed by pulling an object with the mouse into another position in the hierarchical list. If the Shift-key is depressed, the object will be attached to the object at that position, changing the hierarchy of the objects.

Hint: The special dependence of the objects upon one another, which is very important for printing, is displayed hierarchically. Further, detailed information is contained in Chapter "Object List".

2.1.11 Report Structure

The fifth window is the hierarchical Report structure, which is only available in some project types. All elements and their respective sub-elements displayed with their name in the order in which they will be printed. The most important commands for the various tables are available through context menus in the workspace, and a toolbar. More detailed information can be found in the chapter "Inserting Report Container".



Report structure
Report Container
Chart: Customers
Table: Customers [CustomerID [+]]
Table: Orders [Customers2Orders, OrderDate [+]]
Table: Order_Details [Orders2Order Details]
Objects
Report structure
Layers

2.1.12 Preview

The Preview Window is available for you in order to check the layout of your workspace.

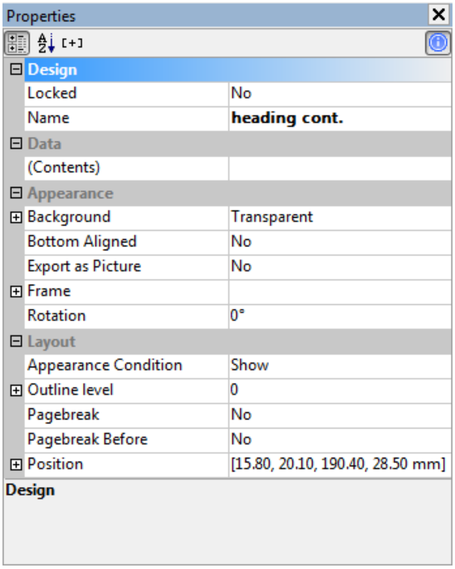
Click in the window to change between full size and normal window size. The size of the Preview Window can be changed by dragging the frame. These settings are stored globally for the project type, and are therefore valid for all List & Label projects of the same type. You can select an area to enlarge by dragging a rectangle. The zoom modes in this preview are independent of the zoom mode in the workspace. Alternatively, you can change the zoom modes by using the buttons in the toolbar. More detailed information can be found in "Real Data Preview" in Chapter "Projects".

2.1.13 Properties Window

The properties of the individual objects are displayed in this tool window, the Properties Window. The properties are sorted, using the appropriate method, by category. Each category has properties that can be combined into groups.

Categories and property groups can be, to improve view, opened and closed. It is also possible to abstain from the use of the categories, and display all property groups alphabetically instead.

[+] The sorting method is set using this button. You can sort either by "category"" or "alphabetically". Using the button "Hide or show constant expressions" it is possible to show formulas, e.g. "False" or "LL.Color.black".

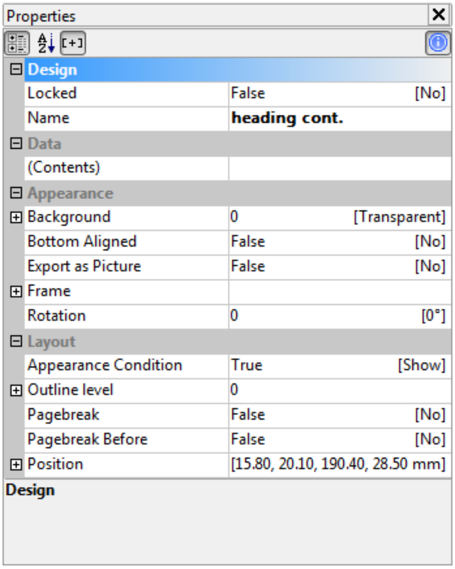


Properties

⊟ **Design**	
Locked	No
Name	**heading cont.**
⊟ Data	
(Contents)	
⊟ Appearance	
⊞ Background	Transparent
Bottom Aligned	No
Export as Picture	No
⊞ Frame	
Rotation	0°
⊟ Layout	
Appearance Condition	Show
⊞ Outline level	0
Pagebreak	No
Pagebreak Before	No
⊞ Position	[15.80, 20.10, 190.40, 28.50 mm]

Design

Properties with formulas



Properties

⊟ **Design**		
Locked	False	[No]
Name	**heading cont.**	
⊟ Data		
(Contents)		
⊟ Appearance		
⊞ Background	0	[Transparent]
Bottom Aligned	False	[No]
Export as Picture	False	[No]
⊞ Frame		
Rotation	0	[0°]
⊟ Layout		
Appearance Condition	True	[Show]
⊞ Outline level	0	
Pagebreak	False	[No]
Pagebreak Before	False	[No]
⊞ Position	[15.80, 20.10, 190.40, 28.50 mm]	

Design

Properties without formulas

To edit a property, click in the value column. When standard values are used, the input of the new value is accomplished using a combo box. Revised properties are shown in bold.

... An additional dialog is available for some property groups that eases the input of the property. You can open this dialog by either selecting the property group and depressing `ENTER` or clicking on the button "...".

It is also possible, for almost all properties, to enter the value as a formula. To do this, click during the input of the value on the combo box entry "Formula”. For values that can be entered directly, click on the formula-button and a formula dialog will open. You will find a detailed description of this function in Chapter "Variables, Formulas and Expressions".

ⓘ Below the property list, a short description of the selected property is displayed. This function can be turned on/off by using the i-button.

2.1.14 Using the Keyboard

The most important functions of the List & Label Designer can be activated by using the keyboard. Here are a few of the most important key combinations:

Tool Window

A key combination exists for every tool window. The window will be activated, or if it is docked behind another window, it will be brought to the foreground.

Tool Window	Shortcut
Workspace	Ctrl+1
Variable List	Ctrl+2
Layers	Ctrl+3
Objects	Ctrl+4
Preview	Ctrl+5
Properties	Ctrl+6
Table Structure	Ctrl+7

Properties

You can also work with key combinations within the property Lists.

Function	Shortcut
Change to next property or header	Cursor down/up
Open/Close category	+/-
Open all property groups of a category or sub-groups of a group	*
Edit property	Tab
End editing	Shift + Tab
Open combobox	Ctrl + F4 (in Edit mode)
ENTER	Depends on value: open content dialog, invert Yes/No, open combo box

2.1.15 Drag & Drop

The List & Label Designer is equipped with a broad range of interesting drag & drop functions, simplifying the project design process. It is possible (see above) to add new contents to existing objects by drag & drop, or create new objects in a free area of the project workspace. This system was also integrated into many dialogs: you may add new contents to the property dialogs of text- and table objects or move existing contents easily with the mouse.

2.2 Exiting the Designer

Select File > Exit to end the List & Label Designer session. If your layout definition has been modified since you last saved, you'll be prompted to save the file.

2.3 Variables, Formulas and Expressions

Information can be inserted into your project in two ways, by entering "fixed text" or using "fields" and "variables".

Information can be inserted directly into the project as "fixed text", e.g. a sender line on an address label or a column heading in a list. Fixed text is printed exactly the way you enter it into the project.

Information coming e.g. from a database is inserted into the projects as "variables" or "fields". Fields are the information in a table that changes from line to line, and variables the information that only changes from page to page. In this manner you can, for example, select the variable TELEPHONE for the contents of a list column. The various different telephone numbers of the database records will then be printed in this column. Variables take the place of information from a database, they're placeholders.

With these two types of information attractive projects can be designed which are sufficient for many purposes. The List & Label Designer, however, offers much more. By using formulas and expressions the information contained in variables and fixed text can be linked and even edited. To help you with this there are "formulas" (for calculations with numbers) and "expressions" (for the combination of text and numeric values and for logical conditions). In formulas and expressions you can insert fixed text and variables in "functions" and link them via "operators".

In the case of projects for printing address labels you can, for example, add the text "PO Box" automatically for a PO Box number saved in a variable POBOX by an expression. In this case (formula "PO Box " + POBOX) you wouldn't just have the bare PO Box number on the label but "PO Box 11 11 11".

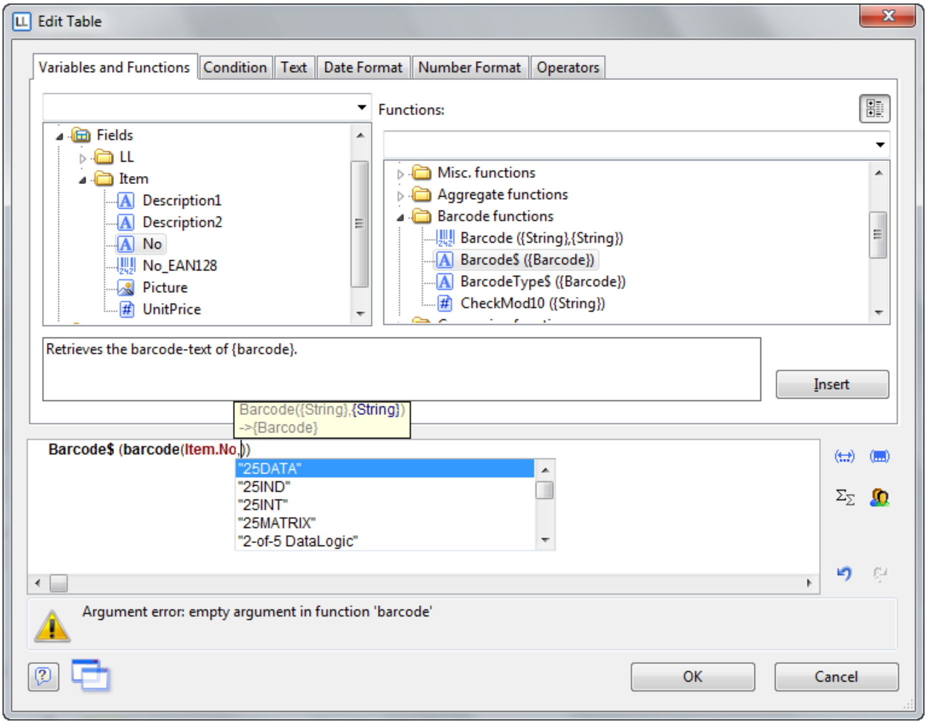
Or, imagine you have the net prices of articles in a variable PRICE, but you actually wanted the gross prices incl. VAT (sales tax) to be printed. Here you can use a formula which calculates and adds the VAT onto the net price. Of course it is the gross price which is then printed.

2.3.1 The Elements of an Expression

Fixed text, variables, formulas, functions, links etc. - all fall under the general name "elements of expressions". They can all be inserted and combined via the same dialog. This dialog contains an assistant who supports you in selecting the correct syntax, provides explanations, and help.

The formula wizard supports you with various input enhancements:

- Show Function Syntax: A tool tip appears in the formula assistant that describes the selected function, lists the necessary parameters and displays the result types.
- Autocomplete: After entering a character, all available functions, fields, parameters and variables that begin with this character will be shown and can be selected.
- Syntax Coloring: functions, parameters and operators are displayed in different colors.
- Automatic Type Conversion: Variables and fields will be automatically converted at the time of entry of the expression so the type of data conforms to expectations.



Retrieves the barcode-text of {barcode}.

Barcode({String},{String})
->{Barcode}

Barcode$ (barcode(Item.No,))

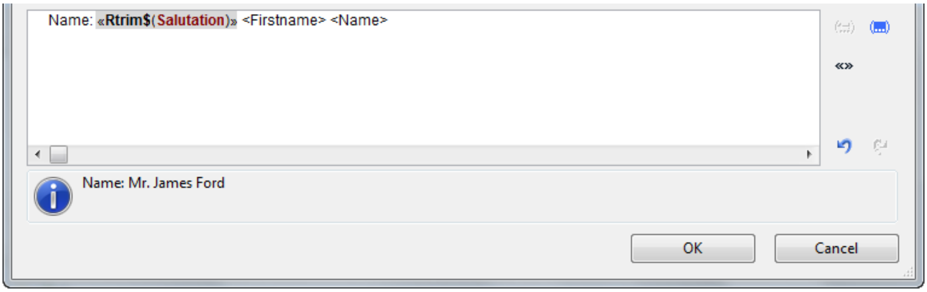
Argument error: empty argument in function 'barcode'

Expression Modes

Please note that there are two possible modes for expressions:

On one hand there is the normal expression mode, where variables and functions can be used without brackets or chevrons. Fixed text must be framed by quotation marks. Variables may be combined with the "+"-Operator.

«» On the other hand there is the extended mode, where fixed text may be stated as is without quotation marks. Variables are framed by "<" and ">" and functions are framed by chevrons "«" and "»". The chevrons can be inserted by clicking the button. The combining of variables with the "+" operator is not necessary. This mode is easier to use.



Name: «Rtrim$(Salutation)» <Firstname> <Name>

Name: Mr. James Ford

Register Cards

The dialog consists of a row of cards, each containing different editing elements.

Card	contains the elements
Variables and Functions	the variables and functions available for this object type
Condition	a special dialog for the definition of IF-THEN-ELSE conditions
Text	a dialog for the entry of fixed text
Date Format	a list of the available date formats
Number Format	a list of the available number formats
Operators	a list of the available link operators

On each of these cards you will find an "Insert" button with which you can insert the selected element into the expression field. You can also perform this by double-clicking or using drag & drop.

There are special rules for the syntax and linking of various individual elements of an expression (variables, text, functions, etc.). For this reason you should always insert the various elements into the expression field via the relevant card of this dialog. In order to insert variables you should use the "Variables and Functions" card, to enter text the "Text" card, etc.

Experienced users may wish to enter the expression directly into the expression field or to edit the expression (e.g. place brackets).

The Expression Field

The expression in the expression field is constantly checked during its "creation" for correct syntax. Any syntax errors are displayed in the info field beneath the input field, along with a hint as to the cause of the error. As long as the expression is incomplete the check routinely displays at least one syntax error. Once the expression is complete, however, and an error is still displayed, then the expression really does contain an error which you should correct.

To simplify more complex expressions, you may divide them into multiple lines. This does not affect the result.

The three buttons next to the input field are used to

- Mark the brackets belonging to the formula expression.
- Select the whole expression framed by a pair of brackets.
- Undo the last step.

2.3.2 Insert variables

The variables are placeholders which will later be filled from the application when configuring projects. The value type "String" (Text), "Number", "Date", "Boolean" (logical values), "Drawing", "RTF" and "Barcode" is important as soon as you want to use variables as parameters in functions, because usually only certain value types can be used as parameters. You can, for example, only multiply a numeric value with a numeric value and not with a picture.

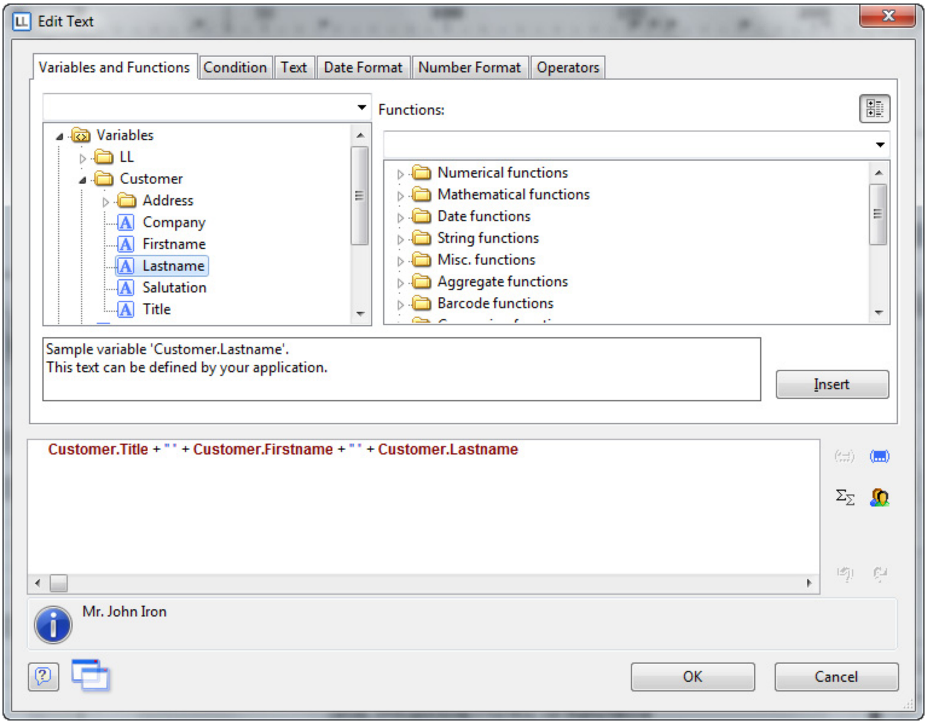
The "Variables and Functions" card contains a list of the variables which are available for the current object, including the data type in the form of an icon in front of the variable and the available functions.

Double-click (or use drag & drop) on the variable you'd like to transfer to the editing line.

The required variable is then transferred to the editing line with the correct syntax.

To include further variables in your expression just repeat the steps described above. If spaces should be between the individual variables, for example to separate FIRSTNAME and LASTNAME, then don't forget to enter these spaces in the editing line also: FIRSTNAME + " " + LASTNAME

You may also insert variables by dragging the required variable to the object on the workspace where it should be inserted. The variable is then added automatically to the object as a new line.



Sample variable 'Customer.Lastname'.
This text can be defined by your application.

```
Customer.Title + " " + Customer.Firstname + " " + Customer.Lastname
```

Mr. John Iron

2.3.3 Insert fixed text

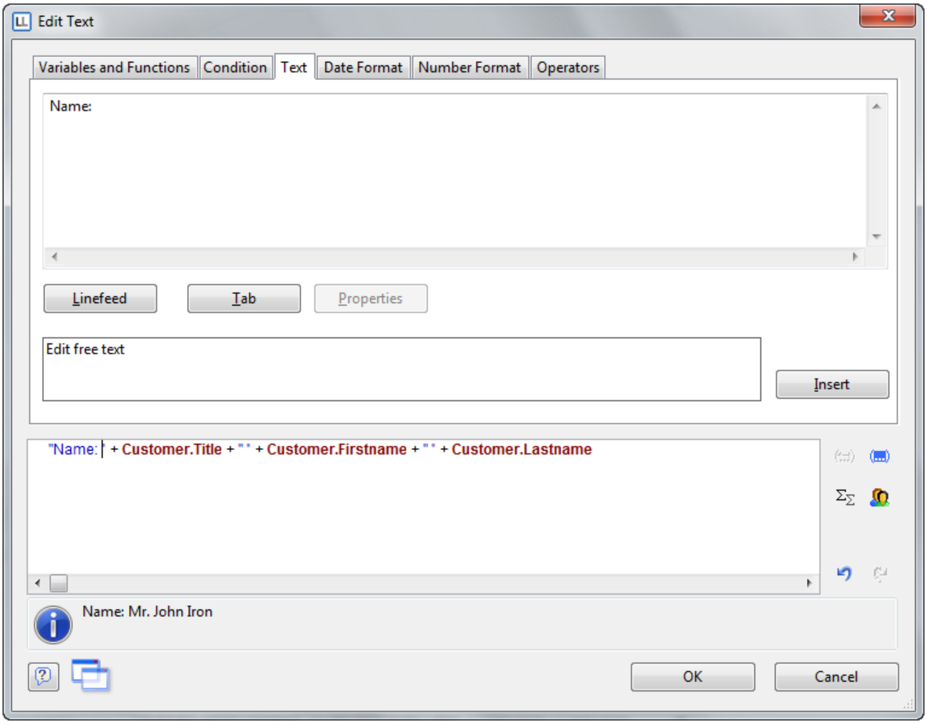
Another very important element in expressions is fixed text, for example as prefix for variables like "Phone Number " + PHONE

which will print: Phone Number 1-555-123-456

The card "Text" offers an easy way to insert text automatically, set tabs and set line breaks.

Enter the text and click on the "Insert" button to insert the text into the expression below. The text will be inserted automatically, depending upon the requirements, either in quotation marks (for function expressions) or without (for text expressions). For this reason you should always enter text using the "Text" card.

In this example, the fixed text "Name: " was entered using the "Text" card, and then the variable Customer.Firstname and Customer.Lastname using the "Variables and Functions" card. The text "Name:" will be printed first followed by the appropriate Firstname and Lastname from the database.



Name:

```
"Name: " + Customer.Title + " " + Customer.Firstname + " " + Customer.Lastname
```

Name: Mr. John Iron

Please note that spaces, for example separators between variables or between variables and text, are considered to be fixed text.

Inserting Linefeed

Linefeed To insert a linefeed, click the button "**Linefeed**". The linefeed is represented by a special character.

Linefeed are only active for those objects (a line in a text object or a column in a table) for which line breaks are set. In this way, words that do not fit into the line/column will be broken into the next line and all following text will be moved down. Please note that if the text doesn't consist of a number of words, but of a long word it will not be wrapped, but clipped.

The value for "Line Wrap" in the "Layout" category of the properties for the text object must be set to "True" for the appropriate paragraph.

For tables, the value for "Line Wrap" in the "Layout" category of the properties must be set to "True" for the appropriate column.

Inserting Tabs

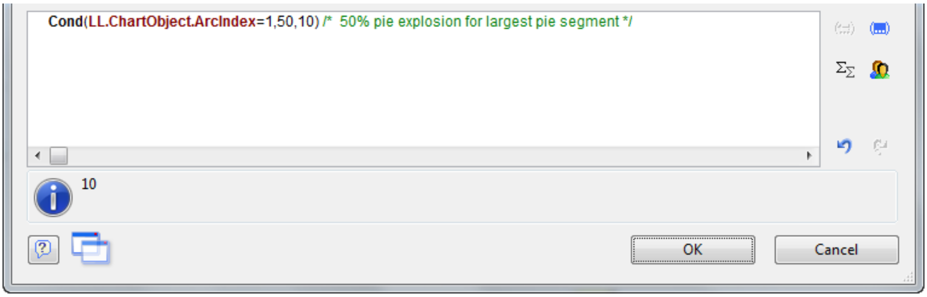
Tab Tabs are only available in text objects, thus the button will not appear in fields of table columns.

Only one tab can be inserted for each line. Its position and alignment can be adjusted using the button "Properties". For more information please refer to Chapter "Inserting Text Objects".

2.3.4 Inserting Comments

Depending that the application supports this feature, it is possible to include comments. This can be achieved using two methods:

- /* <text> */ for comments in the middle of a formula
- /* <text> for comments at the end of a formula. This way the remaining part of the formula will be converted to a comment, not just to the end of line!



```
Cond(LL.ChartObject.ArcIndex=1,50,10) /*  50% pie explosion for largest pie segment */
```

10

2.4 Working with Functions

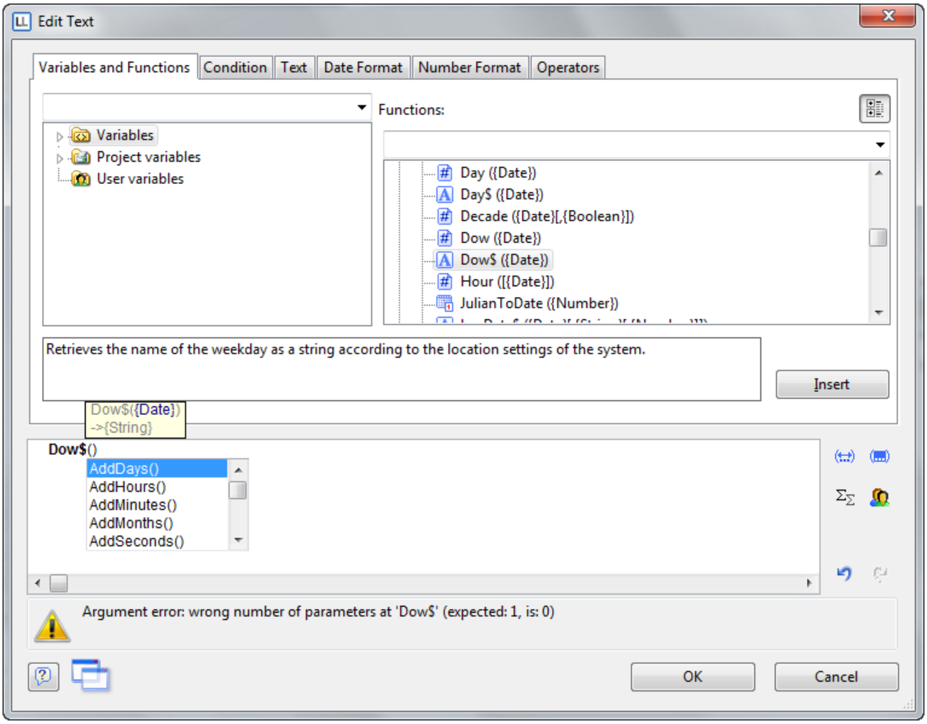
The built-in functions offer flexibility in defining expressions. Functions allow you to change the representation of fixed or variable text or numbers, recalculate and modify values and to set a display format for these.

You'll find a list of available functions and their parameters on the card "Variables & Functions." A detailed description of all functions and parameters can be found in the Appendix.

The functions are sorted by function group (e.g. "Numerical Functions", ...) or sorted alphabetically.

In this example, the user clicked the "Insert" button for the "Dow$()" function which returns the day of the week of the date that is passed in the parameter of the function, ex. "Thursday".

The function is inserted in the correct syntax (with chevrons, etc.) into the function expression field. Also inserted are place holders for the parameters which the function expects/allows. You'll be asked to replace the place holder with a valid value. It is recommended to set all values for the parameters of your function before you continue with the definition of the expression.



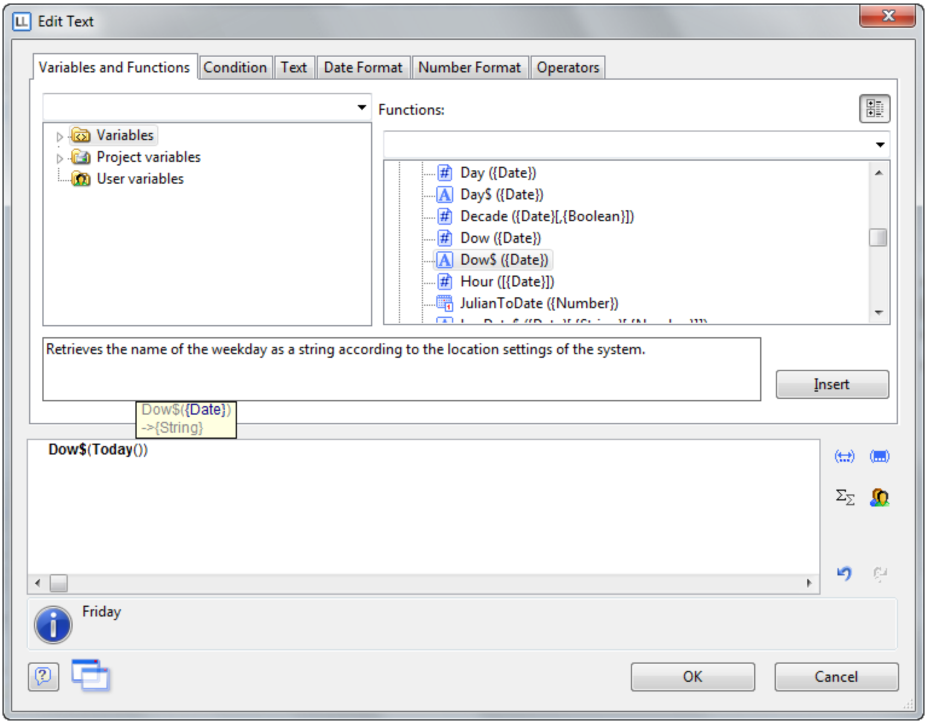
Retrieves the name of the weekday as a string according to the location settings of the system.

`Dow$()`

Argument error: wrong number of parameters at 'Dow$' (expected: 1, is: 0)

You can also use functions for some parameters. For example: Dow$(Today()). This will print the current day.

To insert a variable and a function simultaneously, drag the required variable on the required function folder (e.g. "Numerical Functions"). The folder will be opened automatically and you can select the desired function. If you drop the variable on a function, this function will be inserted into the editing line with the selected variable as first parameter.



Retrieves the name of the weekday as a string according to the location settings of the system.

```
Dow$(Today())
```

Friday

2.4.1 Function Syntax

The functions all use the same syntax, and look quite like those of the programming language BASIC:

Result = Function(Arguments)

Only the functions and arguments are entered. Upper or lower-case has no meaning in the function name, but is very important for the arguments, especially when using variables as arguments.

List & Label evaluates the expression and replaces the Function(Arguments) with the "return value". This means, when you enter a Function(Argument) the Return value is generated.

Definitions:

Element	Description
Function()	The name of the function in its correct syntax. The parentheses for the arguments are a part of the function name. Even if the function contains no arguments, the parentheses are required.
Arguments	Values used in the expression to calculate the return value. The arguments are entered in the parentheses without spaces. A function can have no,

	one, or multiple arguments. Often, functions require arguments of a specific value type (see below). It is important that the argument has the required value type.
Return value	The value List & Label calculates from the function and the arguments. The value type that is returned is dependent upon the value type of the function and argument.

2.4.2 Value Types

Type	Meaning
BOOLEAN	Logical values TRUE or FALSE. If the condition is met, the value is TRUE, otherwise FALSE.
String	Any combination of letters and numbers. If this is used as fixed text, it must be in single or double quotation marks so that they can be distinguished from variables.
Date	A number representing the date using a Julian calendar.
NUMBER	A string that consists of the digits 0-9, the decimal point and the minus (-) sign. Other characters are not allowed. Strings of this type need not be enclosed in quotation marks ("”).
Barcode	A string used as barcode text.
PICTURE	A supported graphic format.
RTF	A formatted text

2.4.3 Examples for the Use of Functions

The available functions are listed in the "Function" dialog page of the formula wizard along with a short description of their parameters and usage. Also displayed is the type of argument(s) (parameters) that is/are expected by the function. If no arguments are displayed for a function, then no arguments, except the empty parentheses, are expected. Otherwise, the function expects exactly the same number of arguments that are displayed. Arguments that are displayed in brackets ([]) are optional.

The argument type "ALL" means that the argument can be of any type, SAME means that this argument must be of the same type as the ALL parameter.

The function "FStr$(NUMBER, STRING)"

`"FStr$(NUMBER, STRING)"` formats a numeric argument (NUMBER) using the format string (STRING). It returns a string with the formatted number.

The format string can contain the following characters:

Format	Meaning
*	number or '*'-Prefix
$	number or '$'-Prefix

-	number or sign (when numerical argument negative)
+	number or sign
(	number or '('-Prefix (when numerical argument negative)
)	')'-Postfix (when numerical argument negative)
#	number or space
&	number or '0'
.	decimal point
,	1000-comma or space

The 'or' part will be taken when the number is too small to fill the position of that format character.

A prefix is a character which is written in front of the number. The formula `FStr$(1, "***")` has "`**1`" as result. In case of `FStr$(100, "***")`, the result would be "`100`". A postfix is put behind the number.

As an example, assume you want to format the number of software products you're selling and that number would be in the variable QUANTITY.

`Fstr$(QUANTITY, "######")`

formats the number to 6 significant digits. Every place except the smallest can be blank if the number is too small.

Value	Result
1	" 1"
255	" 255"

Imagine that you have the price of that software in PRICE and you need it with two decimals (rounded):

`Fstr$(PRICE, "#####&.##")`

The expression formats the content of the variable "price" to the desired format. The character # reserves the intended digits. The character & also reserves a position, though fills the position with "0" incase the formatted number at this position is "0" or does not have a digit.

Value	Result
999.55	" 999.55"
1100	" 1100.00"
1099.5	" 1099.50"
.55	" 0.55"
0.55	" 0.55"

You want the total price? No problem:

`Fstr$(QUANTITY*PRICE, "$$,$$$,$$&.##")`

We use the '$' here to show this feature:

Value	Result
2*999.55	" $1,999.10"
1100	" $1,100.00"
100*1099.5	" $109,950.00"

For some additional examples please refer to Chapter "List of List & Label Variables" in the "Appendix".

The function "LastPage()"

LastPage(returns TRUE if the current page is the last or FALSE if it is not. LastPage() has no arguments, still the parentheses have to be written to declare it as function usage.

As example we assume you want to have a 'sum' line at the bottom of a page which displays the current subtotal or total:

`If(LastPage(),"Subtotal: ","Total: ")+ FStr$(SUM,"$$$$$$$&.##")`

Note that objects using LastPage() in their appearance condition have to be linked to the table object.

The function "Left$(STRING, NUMBER)"

The function "`Left$(STRING, NUMBER)`" returns the first NUMBER of characters of the STRING.

Formula	Result
Left$("combit", 2)	"co"
Left$("combit", 10)	"combit"

2.4.4 Insert logical condition

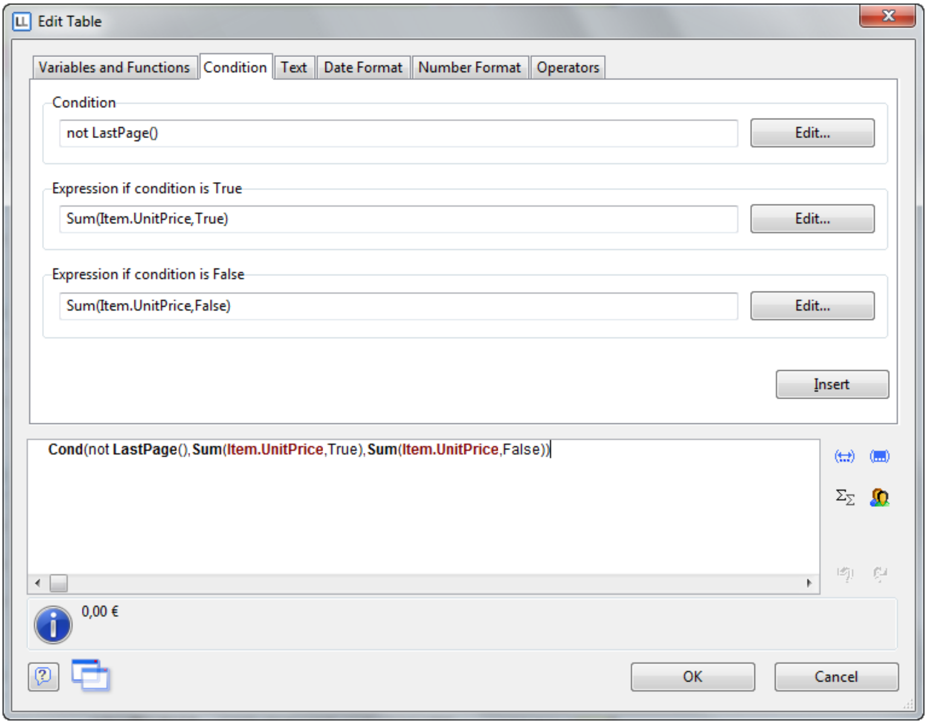
Another special function is the "Cond()" function. It represents an "IF - THEN - ELSE" condition: if the result of the expression in the first parameter is true, then the result of the second parameter will be returned, otherwise the result of the third parameter is used.

Enter the condition to be evaluated, or a Boolean variable, into the condition field. Enter the "true" part into the second line, and the "false" part into the third. Note that the "Edit" button starts an additional assistant to help you edit the appropriate part.

In our example, we used the function not LastPage() to evaluate if it is the last page. When you press "Insert", the parts will be wrapped into the "Cond(...)" function and inserted into the expression line below.

In the example the condition is true, assuming it is not the last page, in this case the page sum should be delivered (parameter of the function Sum is "True").

In the category “Expression if condition is false" enter the expression to be applied when the defined condition is false. This can also be achieved by using the standard "Edit" user interface. If nothing has to be modified, leave the field empty.



Condition: not LastPage()

Expression if condition is True: Sum(Item.UnitPrice,True)

Expression if condition is False: Sum(Item.UnitPrice,False)

```
Cond(not LastPage(),Sum(Item.UnitPrice,True),Sum(Item.UnitPrice,False))
```

0,00 €

In our example the condition is false, for the last page, in this case the sum total should be returned (parameter of the sum function is "False").

You can insert the established conditions into the expression field by using the "Insert” button. They can be stand-alone expressions or part of another expression.

2.4.5 Formatting

Formatting can be achieved via the property “Format” or by using functions FStr$() or Date$().

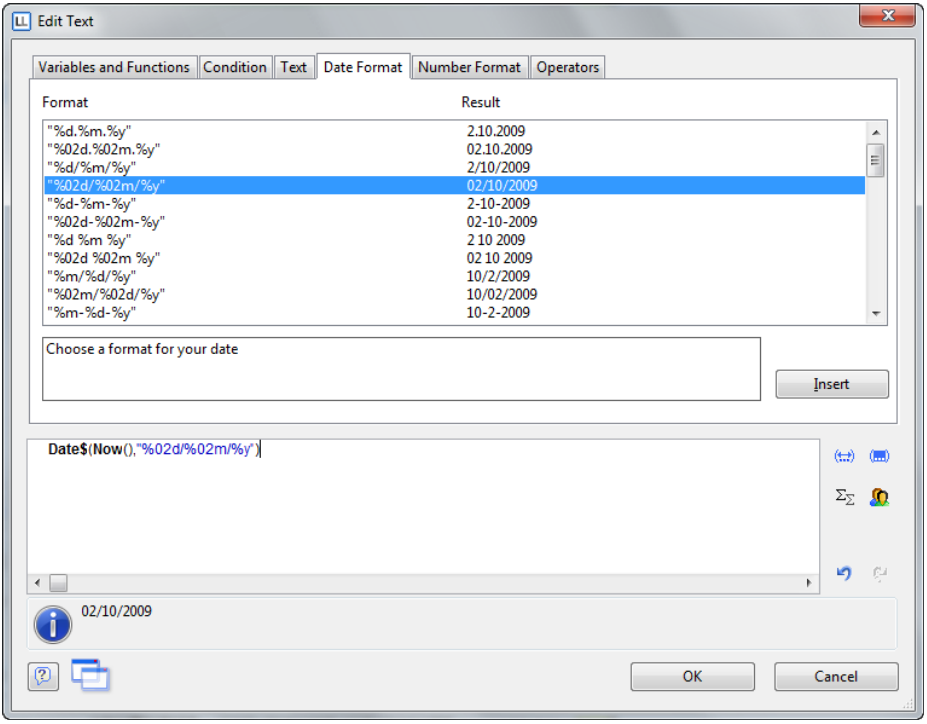
Formatting Dates with Date$()

Using the dialog with the card "Date Format" you can get assistance for date formatting. You can have full-text formatting for day and month values, 2- or 4-letter representation of the year and so on.

To the left is the format string, to the right the result with a date example: As default, the function Now() is selected as date parameter which will display the current date. Of course you can replace it with any date value you like:

Date$(Today() + 7, "%d.%m.%y")

This would display the date one week from today.



Format	Result
"%d.%m.%y"	2.10.2009
"%02d.%02m.%y"	02.10.2009
"%d/%m/%y"	2/10/2009
"%02d/%02m/%y"	02/10/2009
"%d-%m-%y"	2-10-2009
"%02d-%02m-%y"	02-10-2009
"%d %m %y"	2 10 2009
"%02d %02m %y"	02 10 2009
"%m/%d/%y"	10/2/2009
"%02m/%02d/%y"	10/02/2009
"%m-%d-%y"	10-2-2009

Choose a format for your date

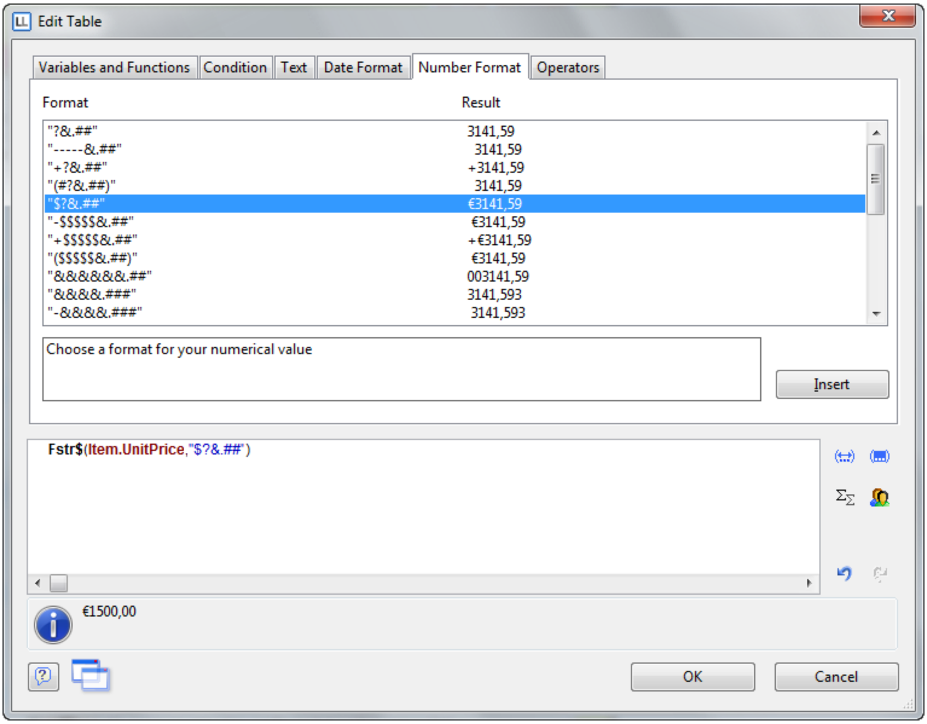
```
Date$(Now(),"%02d/%02m/%y")
```

02/10/2009

In the above example, the date function "Now()" has been formatted so that the result is displayed with the month and the day as a two digit number, and the year as a four digit number. More detailed information on the use of the Date$() function can be found in the Appendix.

Formatting Numbers with FStr$()

Using the card "Numner Format" you can define the format in which a number shall be presented. In this way number of digits, decimals, leading signs etc. can be defined. More detailed information on the use of the Fstr$() function can be found in Chapter "Examples for the Use of Functions" and in the Appendix.



Format	Result
"?&.##"	3141,59
"-----&.##"	3141,59
"+?&.##"	+3141,59
"(#?&.##)"	3141,59
"$?&.##"	€3141,59
"-$$$$$&.##"	€3141,59
"+$$$$$&.##"	+€3141,59
"($$$$$&.##)"	€3141,59
"&&&&&&.##"	003141,59
"&&&&.###"	3141,593
"-&&&&.###"	3141,593

Choose a format for your numerical value

```
Fstr$(Item.UnitPrice,"$?&.##")
```

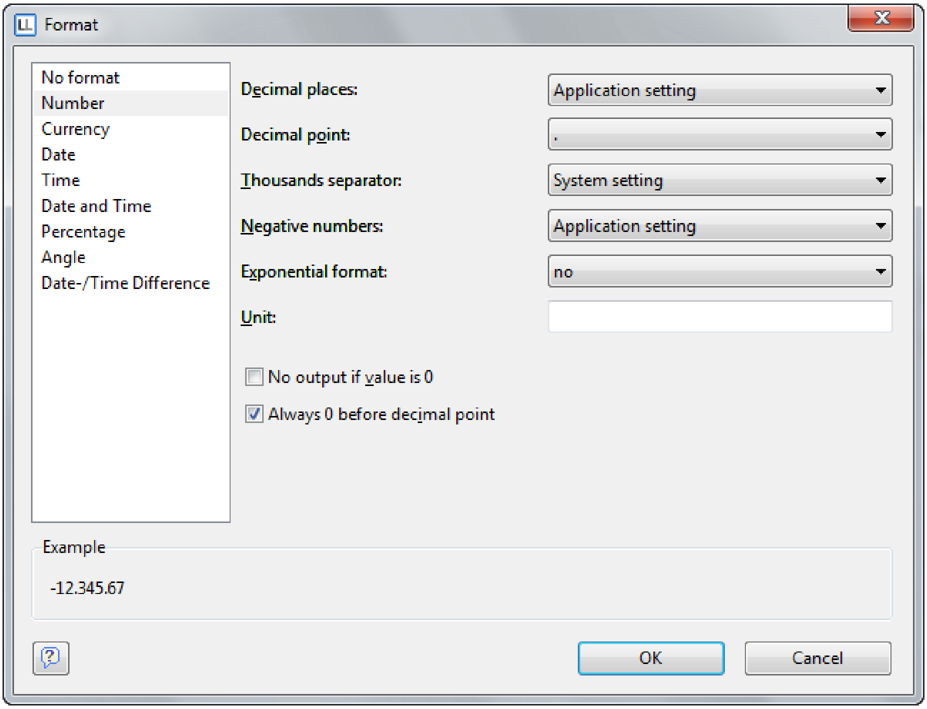
€1500,00

Formatting using the property "Format"

The format property is an alternative to formatting with the functions Date$() and FStr$() in the formula dialog. This property can be found, for example, in text, crosstab and table fields. Note that the formatting will affect the expression's result. If you only wish to format certain parts of an expression (e.g. for text and numbers within one expression) use the functions Date$(), LocCurrL$ or FStr$() in the formula dialog.

With the format editor you can set the format for numbers, currency, date, time, date and time, percentage, angle and date-/time difference.

By default, the respective application settings are used. Alternatively select the system setting or a custom setting. If no application setting is passed by the application, the application setting is the same as the system setting.



Decimal places: Application setting
Decimal point: .
Thousands separator: System setting
Negative numbers: Application setting
Exponential format: no
Unit:

☐ No output if value is 0
☑ Always 0 before decimal point

Example
-12.345.67

2.5 Operators

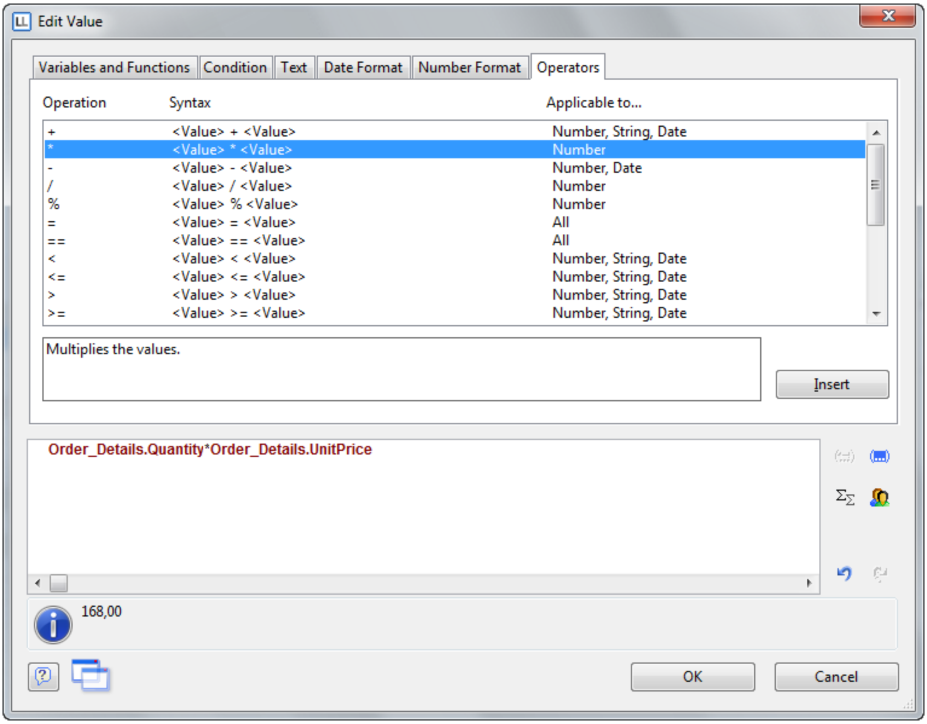
The card "Operators" displays a list of all available operators. These can be used to make calculations, combine variables and/or fixed text, as well as to perform logical comparisons.

The available operators can be categorized, in order of increasing priority, into logical operators (AND, OR, NOT), arithmetic operators (+, -, *, /, %) and relations (<, <=, =, >, >=, <>,!=).

The usual editing rules apply: logical operators will be calculated before arithmetical operators, which will in turn be calculated before relations.

There are three columns on the dialog page. The first describes the operator, the second is the syntax and the third the type(s) of value(s) it can be applied to.

The operator "+" has particular importance, because it can not only be used for addition (Value types "Number" and "Date"), but also for the concatenation of text (Value type "String") or fixed text.



Operation	Syntax	Applicable to...
+	<Value> + <Value>	Number, String, Date
*	<Value> * <Value>	Number
-	<Value> - <Value>	Number, Date
/	<Value> / <Value>	Number
%	<Value> % <Value>	Number
=	<Value> = <Value>	All
==	<Value> == <Value>	All
<	<Value> < <Value>	Number, String, Date
<=	<Value> <= <Value>	Number, String, Date
>	<Value> > <Value>	Number, String, Date
>=	<Value> >= <Value>	Number, String, Date

Multiplies the values.

```
Order_Details.Quantity*Order_Details.UnitPrice
```

168,00

Many operations can be performed in an expression. Operators, functions, parentheses - all in one formula. Thus it can be important to use parentheses to change the order of evaluation.

The order of priorities is

Priority	Operators
1	Parentheses ()
2	Functions
3	Logical operator NOT
4	Logical operators AND, OR, XOR
5	Arithmetical operator modulo (%)
6	Arithmetical operators (*, /)
7	Arithmetical operators (+ , -)
8	Relational operators (<, < = , = , > = , >, <>, ! =)

Operators combine two values to make a result, except for the negation operator NOT, which just takes one value. Operators are used to calculate (mathematical operators), compare (relational operators) or combine (logical operators).

2.5.1 Relational Operators

Relational operators consist of two values of the same data type that are compared, effectively, for whether the selected relational operator applies to both values or not. The result (return value) is the boolean value true if the relation applies and false if it does not.

Operator	Meaning	Data Types	Result Types
>	Greater than	STRING, NUMBER, DATE, RTF	BOOLEAN
> =	Greater than or equal	STRING, NUMBER, DATE, RTF	BOOLEAN
<	Less than	STRING, NUMBER, DATE, RTF	BOOLEAN
< =	Less than or equal	STRING, NUMBER, DATE, RTF	BOOLEAN
=	Equal	STRING, NUMBER, DATE, RTF, BOOLEAN	BOOLEAN
= =	Equal	STRING, NUMBER, DATE, RTF, BOOLEAN	BOOLEAN
<>	Not equal	STRING, NUMBER, DATE, RTF, BOOLEAN	BOOLEAN
! =	Not equal	STRING, NUMBER, DATE, RTF, BOOLEAN	BOOLEAN

2.5.2 Arithmetic Operators

Please note that for arithmetic operators the normal calculation hierarchy applies. First the "Modulo" operator is evaluated, then the "*" and "/" calculations followed by the "+" and "-" calculations. If you want a different calculation sequence, you must use brackets. The innermost brackets are calculated first.

Operator	Meaning	Data Types	Result Types
%	Modulo	NUMBER	NUMBER
+	Add	STRING, NUMBER, DATE	STRING, NUMBER, DATE
-	Subtract	NUMBER, DATE	NUMBER, DATE
*	Multiply	NUMBER	NUMBER
/	Divide	NUMBER	NUMBER

An example for the use of arithmetic operators is the expression

```
PRICENETTO+(PRICENETTO*0.19)
```

With this, the value of the PRICENETTO field would have 19% of the content of PRICENETTO added to it.

2.5.3 Logical Operators

The result (return value) of a logical expression is always a boolean value: true if the expression is true and otherwise false. Please note that for logical operators a calculation hierarchy similar to that for arithmetic operators applies. First the negations are evaluated, then the logical AND and lastly the logical OR. If you desire a different calculation sequence, you must use brackets. The innermost brackets are evaluated first.

Operator	Meaning	Data Types	Result Types
NOT or .NOT.	Not	BOOLEAN	BOOLEAN
AND or .AND.	Logical AND	BOOLEAN	BOOLEAN
OR or .OR.	Logical OR	BOOLEAN	BOOLEAN
XOR or .XOR.	Logical exclusive-OR	BOOLEAN	BOOLEAN

An example for the use of logical operators is the expression

```
ZIP>=70000 AND ZIP<=80000
```

With this, all records whose contents in the ZIP field are greater than or equal to 70000 and smaller than or equal to 80000 are selected.

2.6 Working with Sum Variables

Hint: sum variables are an alternative way of creating sums and counters. Sum variables are principally applicable to whole tables.

To create sum and counter values we recommend using the aggregate functions Sum() and Count(). Further information can be found in the chapter "List of Available Functions". Aggregate functions principally table specific.

Sum variables can be used to build sums/totals over the records, for example the sum of the column PRICE in a table. It is possible to optionally add all records on a particular page (page totals) or for the whole project (totals).

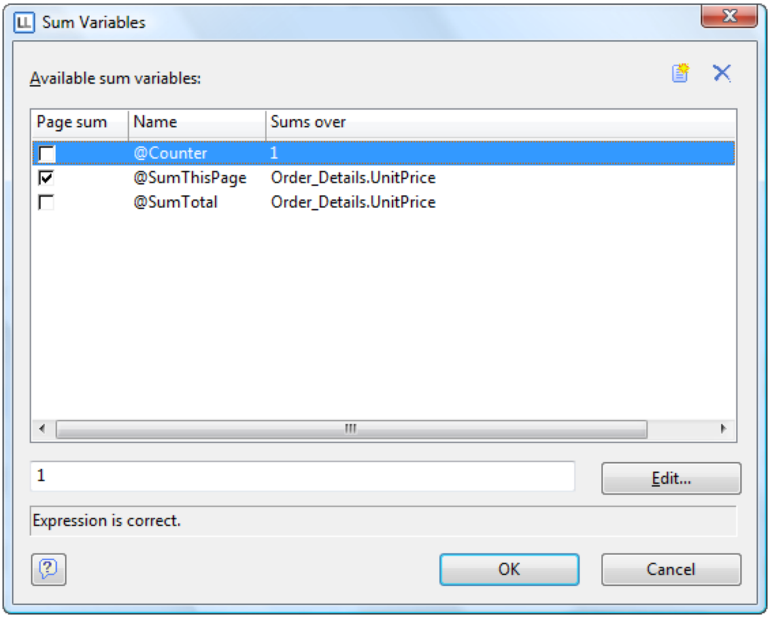
2.6.1 Defining Sum Variables

Sums of this type can be used for all numeric values, i.e. for numeric variables or for expressions which result in a numeric value. The following logic applies for these sum variables:

@SumXX = @SumXX + <numeric expression>

"@SumXX" is the sum variable in which the result of the addition is saved, "<numeric expression>" stands for the variable or the expression which forms the basis of the addition.

For each record which has been printed, List & Label calculates the value from "<numeric expression>" and adds it to the "@SumXX".



Page sum	Name	Sums over
☐	@Counter	1
☑	@SumThisPage	Order_Details.UnitPrice
☐	@SumTotal	Order_Details.UnitPrice

1

Expression is correct.

To use sum variables of this kind ("@SumXX") in projects they must be assign the required numeric expressions. This is done with the command **Project > Sum Variables**.

Click the "**Insert a new variable**" button to create a new sum variable first and give a descriptive name. When working with projects from older List & Label versions, 50 variables "@sum01"..."@sum50" are already predefined. To assign an expression to the new sum variable, click the "**Edit...**" button. This opens the assistant for defining expressions.

Example: The easiest way is to simply select a numeric variable, e.g. UnitPrice. The logical expression would be: @TotalPrice = @TotalPrice + UnitPrice. As the part of the expression "@TotalPrice = @TotalPrice + " is not predefined, simply predefine "UnitPrice".

On the other hand you can also build sums via complex expressions, as long as the result is only a numeric value.

Defining a Counter

With sum variables you can easily define the counter which is actualized for each printed record.

In the simplest case, a counter is increased for each record. The logical expression would be: @Counter = @Counter + 1. As the part of the expression "@Counter = @Counter + " is not predefined, simply predefine "1".

Page Totals and Other Totals

The option "page totals" lets you select whether the sum variables are reset at the beginning of each page or just once at the beginning of the document.

2.6.2 Using Sum Variables

Once you've defined a sum variable, you can use it in the objects of your project.

To do this, open the Contents dialog of the appropriate object and then open the dialog for the definition of expressions. (See Chapter "Variables, Formulas and Expressions"). In the card "Variables and Functions", you will also find the Sum variables listed. You can use these variables in your expression just like any other variables.

2.7 Working with User Variables

To avoid having to re-enter formulas that are required often, it is possible to store them in user variables. These are effectively formula building blocks. User variables are labeled with @<Name> (e.g. @User01). These are defined in Project > User Variables.

Hint: User variables can not be used in a Appearance Condition of Layers.

2.8 Using Filters

You can use filter conditions to select the data you'd like to print. You can assign such conditions to single objects or layers (so that they are printed or not) as well as to data records.

Filter conditions are logical expressions and the result determines whether the object or record will be printed. If the result of the logical expression is TRUE, the record is used (or the object is printed). If the result of the logical expression is FALSE, the record or object will not be printed.

2.8.1 Record Filters

If you wish to print a telephone list, for example, you can ensure through the use of filters that only those records that have contents in the Telephone Number field appear in your list.

Record filters are available for all types of projects. They are most commonly used in list projects. These filters can be globally defined for a complete project or individually for single variables.

2.8.2 Project Specific Filters for Records

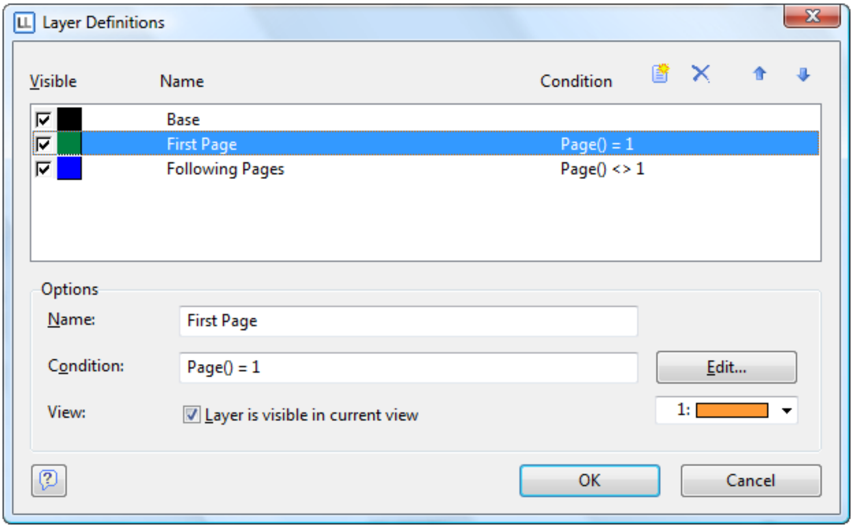
Project specific filters always apply to the complete project. They apply to all objects and variables, whether individual filters have been assigned or not.

To assign a project specific filter, use the command Project > Filter, which will open a dialog for the definition of filter conditions. This is similar to the dialog described in Chapter "Variables, Formulas and Expressions".

2.8.3 Layer Specific Filters for Objects

All objects assigned to a layer can have a common, layer specific appearance condition assigned. This is accomplished by using the command Project > Layer Definitions, doubleclick in the Layers tool window or using the context menu in the Layer Window.

Select the level, in the Layer Window, to which you wish to assign layer specific appearance conditions. Enter the desired condition directly into the field "Condition", or open the dialog for the definition of layer conditions (see Chapter "Variables, Formulas and Expressions").



Visible	Name	Condition
☑ ⬛	Base	
☑ 🟩	First Page	Page() = 1
☑ 🟦	Following Pages	Page() <> 1

Options

Name: First Page

Condition: Page() = 1

View: ☑ Layer is visible in current view 1: 🟧

2.8.4 Object Specific Filters for Objects

Appearance conditions can also be assigned to individual objects. The individual appearance conditions will be applied in addition to the display conditions, if conditions have been defined, of the layer to which the object is assigned. This means that both conditions for the object will be linked with a logical "AND".

2.8.5 Define Filters

To define object specific filters, enter the desired condition in the property "Appearance Condition" in the Property List.

3. Projects

Use the **File** menu commands to open existing projects in the List & Label Designer or create new ones.

3.1 Creating or Opening a Project

3.1.1 Create a new project

To start a new project, select **File** > **New**. If your current project has unsaved changes, you will be asked whether you would like to save them. List & Label then automatically opens the default project for the type of project you require.

Normally this is merely an empty workspace with a certain paper size and alignment. In the case of label projects you can also pre-determine a certain label format (size and the layout of individual labels on the page).

If you select "New project assistant" in the **Options** > **Workspace** dialog, you will be guided through the process of creating a new project by the project wizard (see below).

> **Hint:** A default project is a standard empty "copy" to aid in the creation of projects. It is however possible for you to change the default project to your own requirements and save it as a file called "default". The next time you use the command **File** > **New** your own - changed - default project will be opened.

3.1.2 The Project Wizard

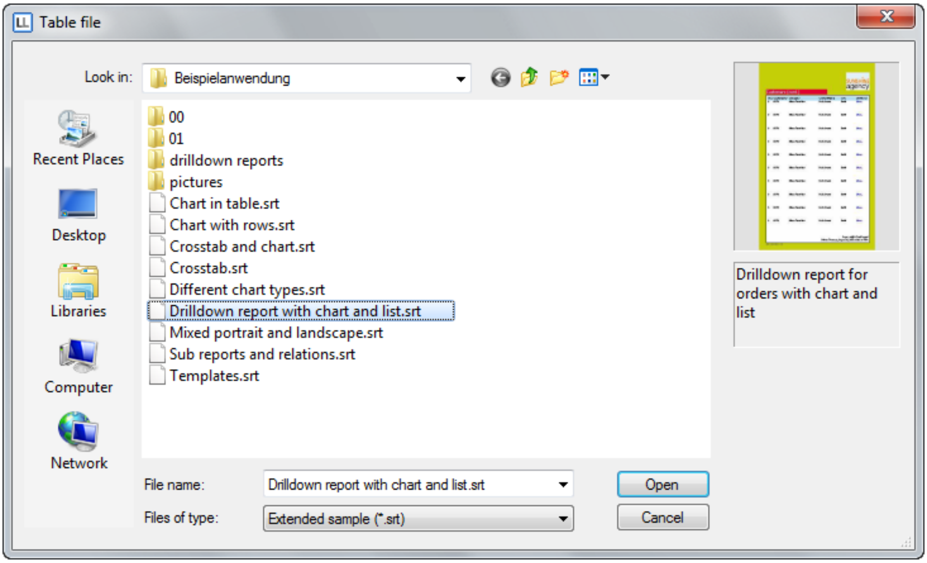
The project wizard will help you create new projects. In label / card projects you will be first lead through the layout options. Afterwards you can select a page title and a matrix of text objects. In list projects you can set the layout options as well as:

- page numbering
- page title (first page only or all pages)
- zebra pattern for the table
- create a summary
- select the data for the table object

3.1.3 Opening an Existing Project

To open an existing project select **File** > **Open**. If your current project has unsaved changes, you will be asked whether you would like to save them.

A dialog will open in which you can select the file to be opened.



Look in: Beispielanwendung

File name: Drilldown report with chart and list.srt

Files of type: Extended sample (*.srt)

Drilldown report for orders with chart and list

If a description has been given to a project (see Project Properties) the description will be displayed underneath the sketch field. In the "sketch" field you can see a sketch of each selected project, making it easier for you to find the project you are looking for. (The sketch is only shown if the corresponding option is set, see: Options > Compatibility > Sketch in File Dialog).

The Designer will issue a short warning if you select a project that was designed with an older version. Please note that projects designed with the new Designer will not be readable or editable with an older version. We advise you to make backups of your project files before editing them with the new Designer. Please check your layout after conversion carefully, as slight changes may occur.

3.1.4 Project Import

With the command File > Import you can insert all objects from another project into the current project.

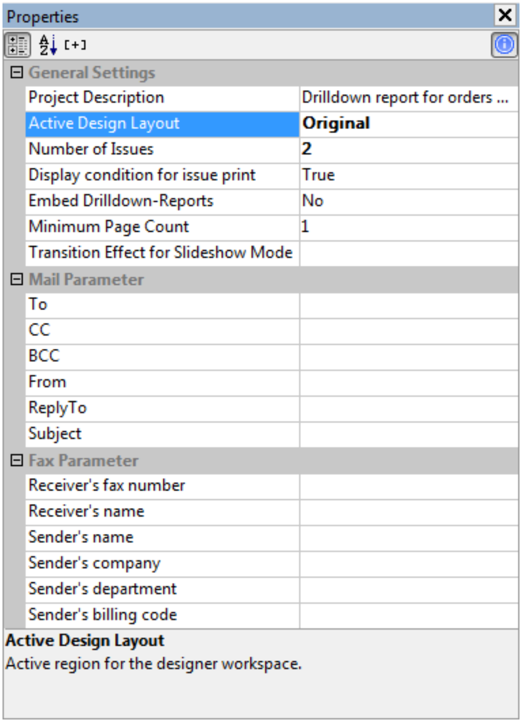
3.1.5 Project Types

The List & Label Designer is capable of different kinds of projects: labels, lists and cards. Please note that it is not possible to switch between the different project types in one Designer session. The Designer has to be closed and reopened in the new mode for this purpose.

It is however possible to design multiple projects of the same type in a single session. Simply open or create the projects in the order you want to design them. You may not open multiple projects at one time, but you can copy objects from one project to another using the clipboard.

3.1.6 Project Properties

The properties window of the project is displayed when no objects are selected in the workspace. The project parameter values are available with the function ProjectParameter$(). For available project parameters see "List of Fields".



General Settings	
Project Description	Drilldown report for orders ...
Active Design Layout	**Original**
Number of Issues	**2**
Display condition for issue print	True
Embed Drilldown-Reports	No
Minimum Page Count	1
Transition Effect for Slideshow Mode	
Mail Parameter	
To	
CC	
BCC	
From	
ReplyTo	
Subject	
Fax Parameter	
Receiver's fax number	
Receiver's name	
Sender's name	
Sender's company	
Sender's department	
Sender's billing code	

Active Design Layout
Active region for the designer workspace.

Project description

In the "Project Description" field you can enter a description of the project. This description is then displayed in the dialog File > Open and makes it easier to find the desired project. Alternatively, this description can also be displayed in the File > Save As... dialog.

Active Design Layout

For multi-page projects it can be sensible to define different layout setting (page size, orientation). Use the field "Active Design Layout" to define which layout-setting should be used in the designer workspace.

Number of Issues

Defines the number of issues for print and preview. Activates the function IssueIndex() for display and layout region conditions.

Display condition for issue print

Allows print conditions for the pages of the different issues, e.g. if the last page containing the GTC should be suppressed for the copy.

Example: If (IssueIndex()=2, not Lastpage(), True)

Embed Drilldown-Reports

Drilldown-Reports will be embedded in the preview file, so they can be shipped or saved completely.

Minimum page count

With card projects at least the number set in the "Minimum page count" field is automatically printed. For example you want to print out a four-page form with a different layout for the four pages, create a new layer for each of the four pages and place the objects on these layers corresponding to the desired output. Afterwards, set the minimum page count to "4".

With list projects the number set here defines the page number, from which the output of the table/report container is started. For example you need a cover sheet, you can assign the table to the layer "Following Pages" and design the layer "First Page" as desired. Afterwards, set the minimum page count to "2".

Transition Effect for Slideshow Mode

Select a transition effect as default for the preview´s slideshow mode.

Fax and mail parameter

List & Label documents are faxed using the installed fax (printer) driver, i.e. an installed fax program is a precondition for faxing.

If the fax should be sent using the Windows fax driver, the parameters (at least the fax number) need to be set in the project's properties. Use the "Fax parameter" section to enter the required data.

If you want to send the fax using another fax driver, you might be able to use special command sequences to pass the fax number and other field informations (e.g. DvISE commands for Tobit David). These command sequences can normally be placed in a text field. At print time, the fax dialog will then be suppressed, as the document contains all required data.

You can also send List & Label documents directly by eMail. The required eMail variables are also defined in the properties window of the project. The eMail settings (SMTP, MAPI, XMAPI) are changed in the control panel under "combit Mail".

3.2 Saving projects

Projects can be saved using any file name that is allowed by your operating system. It is recommended however to use the default file extension for each project type.

To save a project, select File > Save. When you want to save new projects using File > Save, the File > Save As dialog will be displayed and you can select a name for your project.

To save an existing project under a different name or path, select File > Save As. A dialog will open in which you can enter a name for the project, the device and the directory in which the project will be saved.

In addition to the file name, you can give a short description of the file which will be shown in the file-open dialog to help you to easily distinguish your projects.

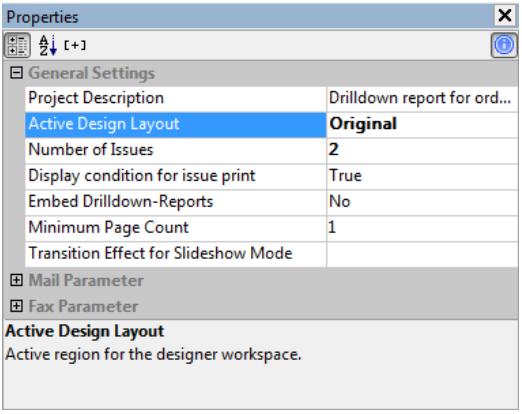
3.3 Defining Page Setup

Before you start to place objects in a project, please select your layout. With the command Project > Page Setup you can define properties such as printer selection, paper size and orientation. The layout options available depend on the type of project you are currently working with.

3.3.1 Regions

For multi-page projects it can be sensible to define different layout setting (page size, orientation).

In the field "Active Design Layout" of the project properties define the active region for the designer workspace.



⊟ General Settings

Project Description	Drilldown report for ord...
Active Design Layout	**Original**
Number of Issues	**2**
Display condition for issue print	True
Embed Drilldown-Reports	No
Minimum Page Count	1
Transition Effect for Slideshow Mode	

⊞ Mail Parameter

⊞ Fax Parameter

Active Design Layout
Active region for the designer workspace.

A new area can be defined over the new-button and then define its corresponding properties. A "condition" has to be defined when the Layout setting is used e.g. "Page()=1" when it should be used only for page 1. The first area will be used for valid for the set conditions.

The default-region "Standard layout" is always the last region and the condition is set to True. If you are sure that the standard layout is never used, you can suppress displaying the print dialog with the property "Show in print dialog".

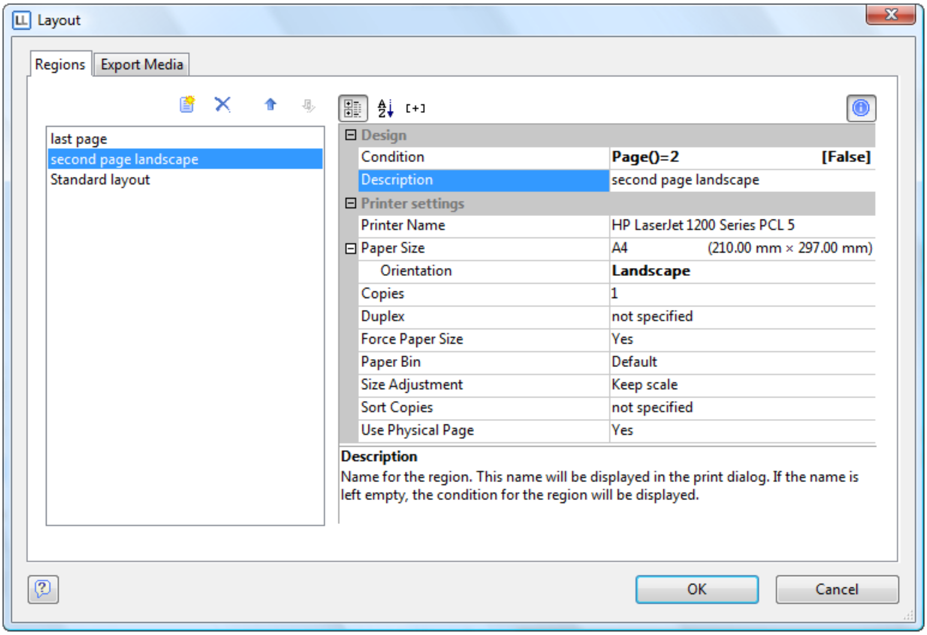
The format of the region, that includes the condition "Lastpage()", should not vary from the page layout which will be used when the condition Lastpage() is not true. This is due to List & Label not knowing at the start of a page if it is intended to be the last page, hence cannot select a region with the condition Lastpage() is true. Only after completing the page the printer can be assigned to this region. Should the page size or orientation change, this can lead to a scale change or data being cut off.

Please note: if the page size or orientation varies you need to adapt the width and height of the objects. Otherwise, data will be clipped. In order to dynamically position the report container you can use the LL.Device.Page variables, e.g.:

Position.Left, Position.Top: 10

Position.Width: LL.Device.Page.Size.cx-20

Position.Height: LL.Device.Page.Size.cy-20



last page
second page landscape
Standard layout

Design		
Condition	**Page()=2**	**[False]**
Description	second page landscape	
Printer settings		
Printer Name	HP LaserJet 1200 Series PCL 5	
Paper Size	A4	(210.00 mm × 297.00 mm)
Orientation	**Landscape**	
Copies	1	
Duplex	not specified	
Force Paper Size	Yes	
Paper Bin	Default	
Size Adjustment	Keep scale	
Sort Copies	not specified	
Use Physical Page	Yes	

Description
Name for the region. This name will be displayed in the print dialog. If the name is left empty, the condition for the region will be displayed.

Printer settings and page size

By setting the properties it is possible to define for the region the printer settings, paper size, orientation, copies, duplex, sort and paper bin (e.g. first page on paper with company header, following pages on standard paper.

Force Paper Size

If no printer definition file exists and this option is set, the paper format selected during design will be forced during print out. Prerequisite is that the respective format or a "user defined" format is supported by the printer. If not, the next size up will be selected.

Size Adjustment

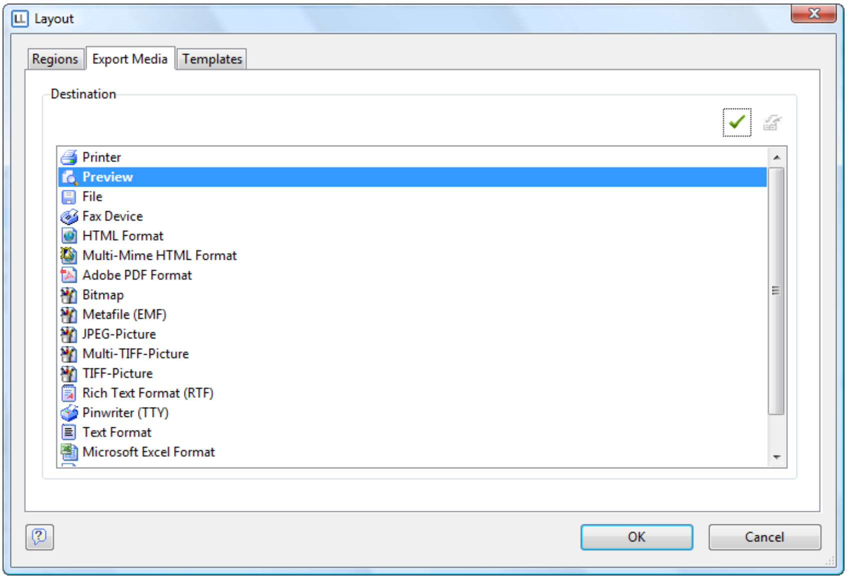
Sets whether the page will be fit if printed to a different printer or if the scale should be kept.

Use Physical Page and not only the printable area

If this option is set the whole physical page is available as workspace, including non-printable margins. Some label page layouts require this, as the first label starts right at the top left edge instead of at a specific distance. The unprintable margins are shown shaded in the full-page preview. This means that objects can also be placed in the margins which will be cut off during the print. If objects are placed on page edges you still should take notice of the unprintable area.

3.3.2 Export Media

Depending on the application, different export possibilities of List & Label are listed.



- Printer
- Preview
- File
- Fax Device
- HTML Format
- Multi-Mime HTML Format
- Adobe PDF Format
- Bitmap
- Metafile (EMF)
- JPEG-Picture
- Multi-TIFF-Picture
- TIFF-Picture
- Rich Text Format (RTF)
- Pinwriter (TTY)
- Text Format
- Microsoft Excel Format

You can configure the output into different export formats by the corresponding option dialogs. There are two buttons:

Select the format in the list and click on the options button

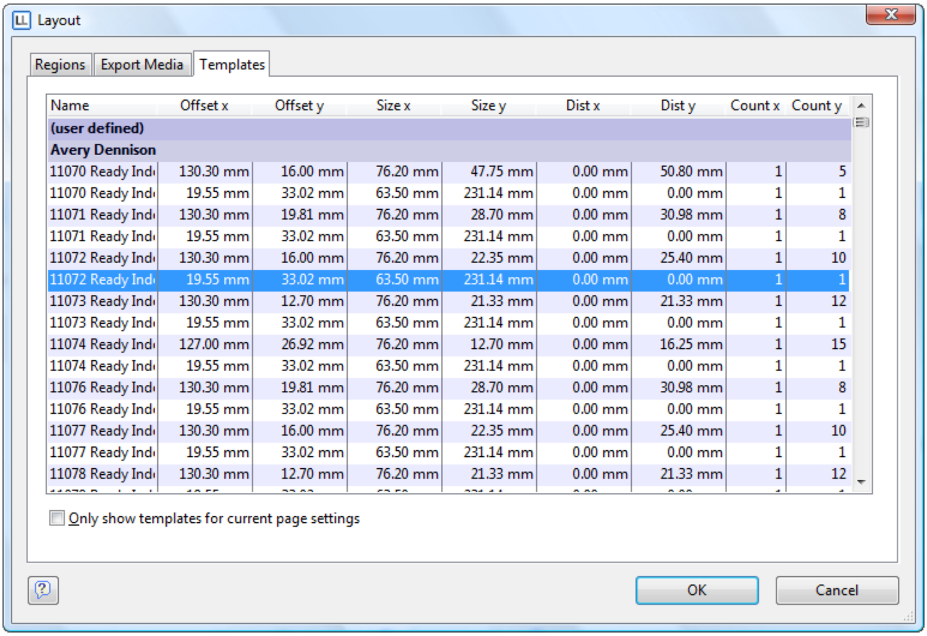
✓ You can select a target format as default for the later print

3.3.3 Templates and Page Layout for Labels

A previously or manually defined label format can be loaded in the page layout for labels.

Using Pre-Defined Label Formats

You can select from a range of pre-defined label formats in the "Templates" register in the page setup for labels. Now you won't have to figure out the size of a particular label and how many labels are on one page.



Name	Offset x	Offset y	Size x	Size y	Dist x	Dist y	Count x	Count y
(user defined)								
Avery Dennison								
11070 Ready Ind	130.30 mm	16.00 mm	76.20 mm	47.75 mm	0.00 mm	50.80 mm	1	5
11070 Ready Ind	19.55 mm	33.02 mm	63.50 mm	231.14 mm	0.00 mm	0.00 mm	1	1
11071 Ready Ind	130.30 mm	19.81 mm	76.20 mm	28.70 mm	0.00 mm	30.98 mm	1	8
11071 Ready Ind	19.55 mm	33.02 mm	63.50 mm	231.14 mm	0.00 mm	0.00 mm	1	1
11072 Ready Ind	130.30 mm	16.00 mm	76.20 mm	22.35 mm	0.00 mm	25.40 mm	1	10
11072 Ready Ind	19.55 mm	33.02 mm	63.50 mm	231.14 mm	0.00 mm	0.00 mm	1	1
11073 Ready Ind	130.30 mm	12.70 mm	76.20 mm	21.33 mm	0.00 mm	21.33 mm	1	12
11073 Ready Ind	19.55 mm	33.02 mm	63.50 mm	231.14 mm	0.00 mm	0.00 mm	1	1
11074 Ready Ind	127.00 mm	26.92 mm	76.20 mm	12.70 mm	0.00 mm	16.25 mm	1	15
11074 Ready Ind	19.55 mm	33.02 mm	63.50 mm	231.14 mm	0.00 mm	0.00 mm	1	1
11076 Ready Ind	130.30 mm	19.81 mm	76.20 mm	28.70 mm	0.00 mm	30.98 mm	1	8
11076 Ready Ind	19.55 mm	33.02 mm	63.50 mm	231.14 mm	0.00 mm	0.00 mm	1	1
11077 Ready Ind	130.30 mm	16.00 mm	76.20 mm	22.35 mm	0.00 mm	25.40 mm	1	10
11077 Ready Ind	19.55 mm	33.02 mm	63.50 mm	231.14 mm	0.00 mm	0.00 mm	1	1
11078 Ready Ind	130.30 mm	12.70 mm	76.20 mm	21.33 mm	0.00 mm	21.33 mm	1	12

☐ Only show templates for current page settings

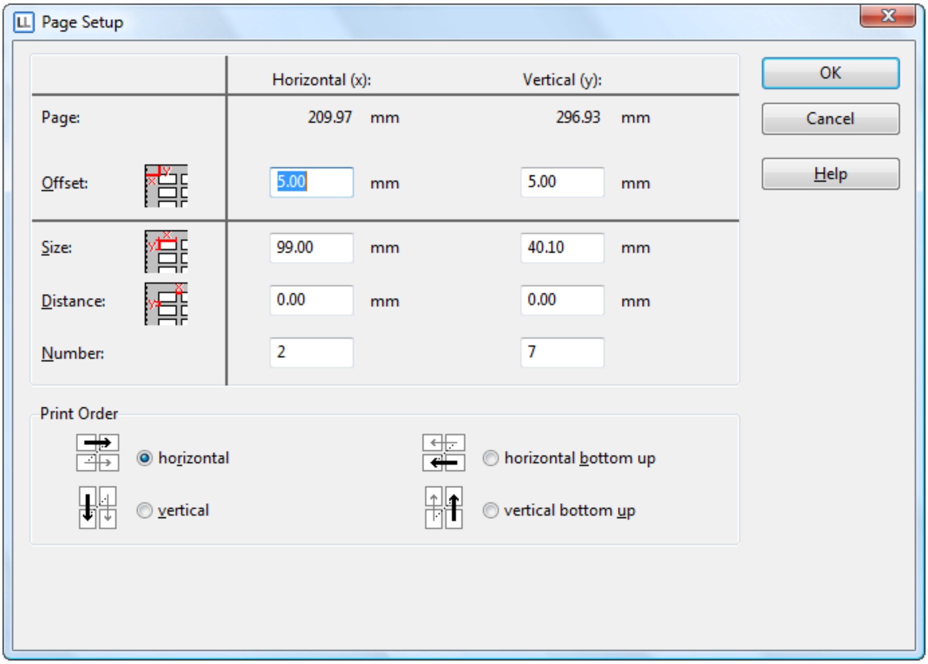
Defining Your Own Label Formats

It is also possible to define your own label layout if the one you require is not available. Use the following values for he property "Layout Definition":

Offset: The offset provides the horizontal and/or vertical distance of the top left-hand label to the edge of the workspace (depending on the "physical page size switch, this is relative to the paper edge or the printable area edge).

> The top left corner of the workspace always begins on the screen display with the coordinate 0/0, independent of the selected page size and offset. The effects of the offset can however be seen in the full-page preview or when you print.

Setting the Size: This provides the size (horizontal = width / vertical = height) of the label in inch or mm.



	Horizontal (x):		Vertical (y):	
Page:	209.97	mm	296.93	mm
Offset:	5.00	mm	5.00	mm
Size:	99.00	mm	40.10	mm
Distance:	0.00	mm	0.00	mm
Number:	2		7	

Print Order

● horizontal
○ horizontal bottom up
○ vertical
○ vertical bottom up

Setting the Distance: This option defines the distance from one label to the next (the space between two labels). For labels with only one column, only the vertical distance needs to be entered.

Setting the Number: This option defines the number of labels per page. (horizontal value = number of columns per page, vertical value = numbers of lines per page)

You can check the layout by selecting **Options > Preview > Page Preview**.

Defining Print Order: Sets the printing order if more than one label is printed on page. Possible values: 0 (horizontal, 1 (vertical), 2 (horizontal bottom up), 3 (vertical bottom up).

Labels are normally printed in lines from top left to bottom right. With partly used sheets of labels you may have already used the first line of labels, for example, which could cause the sheet to lose its stability at the top. Some printers have problems taking such sheets and react with a paper jam.

Here you have the option of printing sheets of labels from the bottom up. In this way the top line of labels will be printed last and the sheet remains stable, even when the critical area on the sheet is reached.

Store user defined Label formats in the Label template list

To save user defined label you can use the file "cmll1400.inf".

Composition of a Label (all units are in 1/1000 mm):

<A> <B>,<C> = <D> <E>,<F>, <G> <H>,<I>, <J> <K>

A: product code, B: product description, C: product size, D: label size (width), E: label size (height), F: space size between labels horizontally, G: space size between labels vertically, H: number of labels per column, I: number of labels per row, J: margin size left, right, K: margin size top, bottom

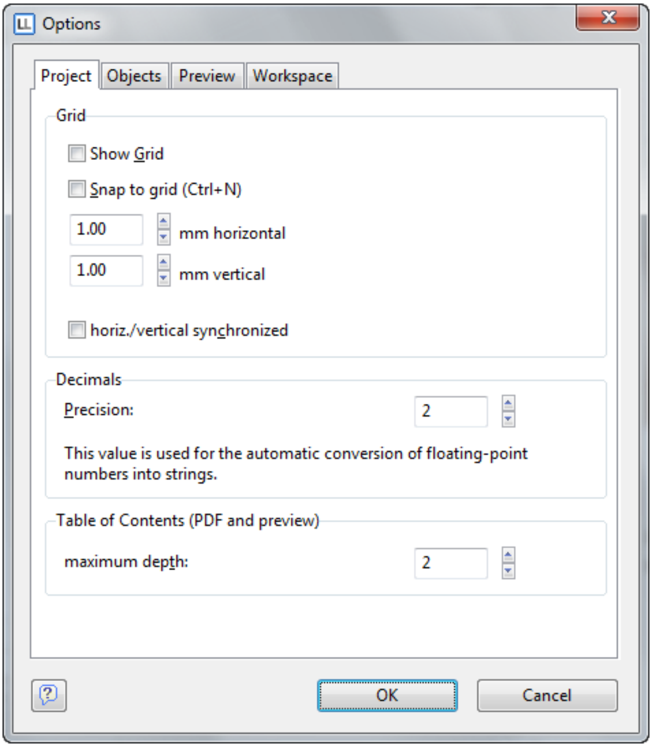
E.g. 3420 Universal Labels, 70 x 16,9mm = 70000, 16900, 0, 0, 3, 17, 0, 4850

3.4 Default Values and Options

Before you start to insert individual objects belonging to a project on your workspace, and set their properties, you need to define the default values you require with the command **Project > Options**. In this way you'll save time and effort later. Each option in this dialog is explained by a tool tip.

3.4.1 Options for the Project

The options set in the "**Project**" card are only valid for your current project.



☐ Show Grid

☐ Snap to grid (Ctrl+N)

1.00 mm horizontal

1.00 mm vertical

☐ horiz./vertical synchronized

Precision: 2

This value is used for the automatic conversion of floating-point numbers into strings.

maximum depth: 2

Defining Alignment Grid

Specify whether a grid net should be placed in the background of the workspace or not. You can set the distances between the grid lines. The option "horizontal - vertical - synchronized" enables the same grid spacing in both directions.

With the option "Snap to Grid" you ensure that objects are not freely inserted or moved on the workspace but only along the (sometimes invisible) grid lines.

Precision

Here you can define the default number of decimal places for numeric values, which will be used as long as they aren't printed with format specifiers (ex. FStr$).

Table of Contents

Here you can set the maximum chapter depth of lines and objects, which are used as bookmarks when exporting to PDF or preview.

3.4.2 Options for Objects

Default values set on the card "Objects" are valid for all new objects, until they are changed.

With the card "Objects" you can make various settings for objects. Newly created objects have these default values at the start. You can change these values later for each object. We recommend defining these values to a suitable level when beginning a new project so that the manual changes are later kept to a minimum.

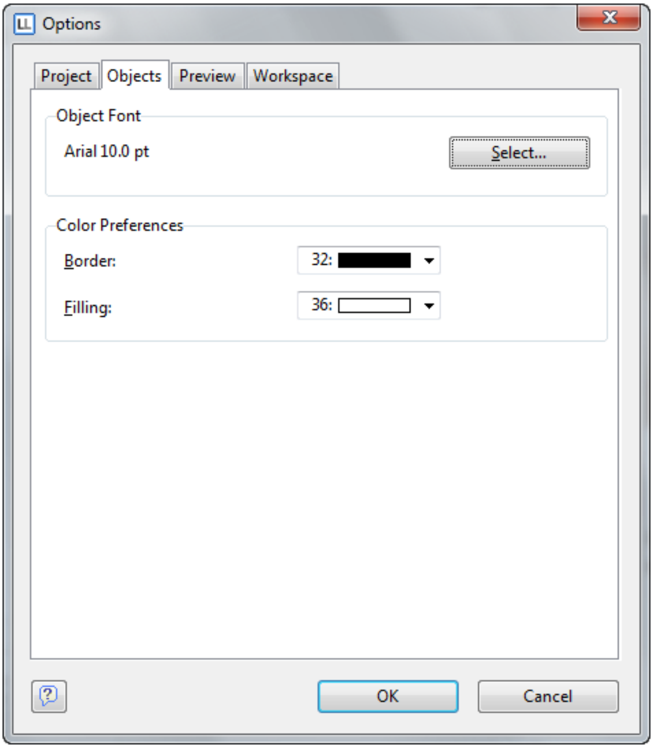
The default values are only valid for the current project.

Object Font

In the "Object Font" group you can define the font for each newly inserted object, using the "Select" button. With the button "default" in the object properties the font can be adjusted to a predefined system font. The settings also affect all objects which were not changed manually to a non default font.

Color Preferences

The category "Color Preferences" allows you to set options for the appearance of objects. With the combo boxes "Border" and "Filling" you can set different colors for different objects.



Object Font

Arial 10.0 pt

Color Preferences

Border: 32: ■

Filling: 36: □

3.4.3 Options for the Preview

On the "Preview" card you can define global settings for the preview.

Setting Colors for the Preview

In the "Colors" group you can define the color for the background of the preview window using the combo box "Background". With the combo box "Border" you can select the color of the simulated paper border in the preview.

Display of Label/Card Projects

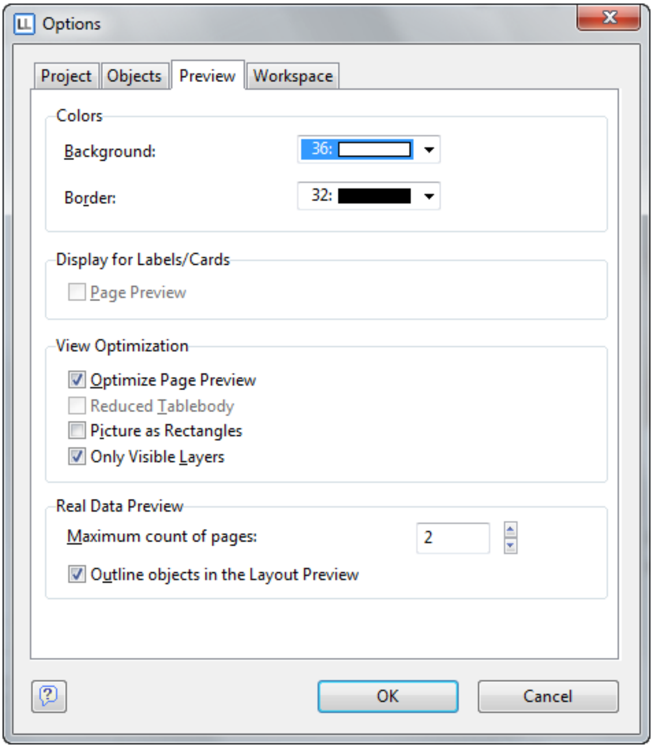
Select if the preview should contain only one label or the complete page.

Optimized View

Using the options from the "View Optimization" group you can reduce different preview details, which enables a faster preview.

Real Data Preview

Here you can set the maximum number of pages which are displayed in the Real Data Preview. Additionally objects can be marked by color (the color depend on the assigned layer).



Colors

Background: 36:

Border: 32:

Display for Labels/Cards

☐ Page Preview

View Optimization

☑ Optimize Page Preview

☐ Reduced Tablebody

☐ Picture as Rectangles

☑ Only Visible Layers

Real Data Preview

Maximum count of pages: 2

☑ Outline objects in the Layout Preview

3.4.4 Options for the Workspace

The workspace can also be adjusted to your needs in various ways.

Displaying Objects

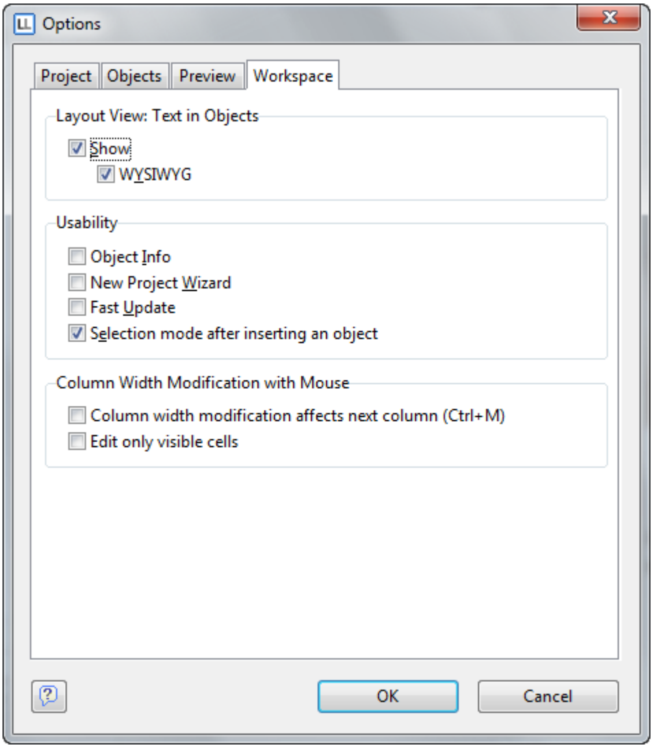
In the "Text in Objects" group you can define whether the text contained in objects should be displayed or not on the workspace. The "WYSIWYG"-option displays the selected fonts and formats the way they are printed.

Settings for Usability

The "Usability" options allow you to define various default values for handling the List & Label Designer.

- Check the option "Object Info" to get a tooltip with the object name.
- Check the option "New Project Wizard" to receive assistance when creating new projects.
- If the option "Fast update" is checked objects will be painted faster. On some systems light flickering may occur.

- Set the "Selection mode after object insertion" option in order to switch to the selection mode automatically after inserting an object. This prevents you, for example, from inserting multiple objects accidentally.



Layout View: Text in Objects

☑ Show
☑ WYSIWYG

Usability

☐ Object Info
☐ New Project Wizard
☐ Fast Update
☑ Selection mode after inserting an object

Column Width Modification with Mouse

☐ Column width modification affects next column (Ctrl+M)
☐ Edit only visible cells

Changing Column Widths Using the Cursor

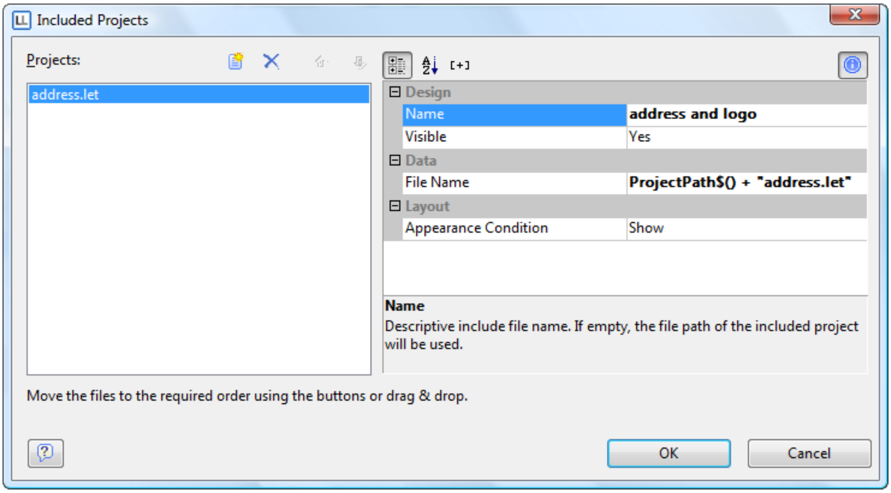
The width of a field or an entire column can be controlled precisely by adjusting the "width" property. You can also control the width directly with your cursor in the workspace by selecting the table object and moving the right hand border line of a column. This will affect all table columns, whose separators lie within a +/-2mm interval of the cursor.

- The adjustment will affect only the line on which the cursor is positioned if CTRL is held down.
- The line will snap to a separator mark if it is within a 10 pixel interval of it. By holding SHIFT, this function is turned off.
- With the option "Column width modification modifies next column", it is possible to change the width of the column while also changing the width of the next.

- To change the widths of columns that are currently invisible, deactivate the "Edit only visible cells" option.

3.5 Including Projects

When often using similar projects, it is practical not to design repeating elements over and over again, but to include other existing projects in your current projects.



address.let

Design	
Name	**address and logo**
Visible	Yes
Data	
File Name	**ProjectPath$() + "address.let"**
Layout	
Appearance Condition	Show

Name
Descriptive include file name. If empty, the file path of the included project will be used.

Move the files to the required order using the buttons or drag & drop.

This way e.g. a letter head can easily be included. In addition, changes just need to be edited in one place, if e.g. the design needs to be adapted. A change of address has to be made only in one single project instead of different projects.

Using Project > Include you can include existing projects in current projects.

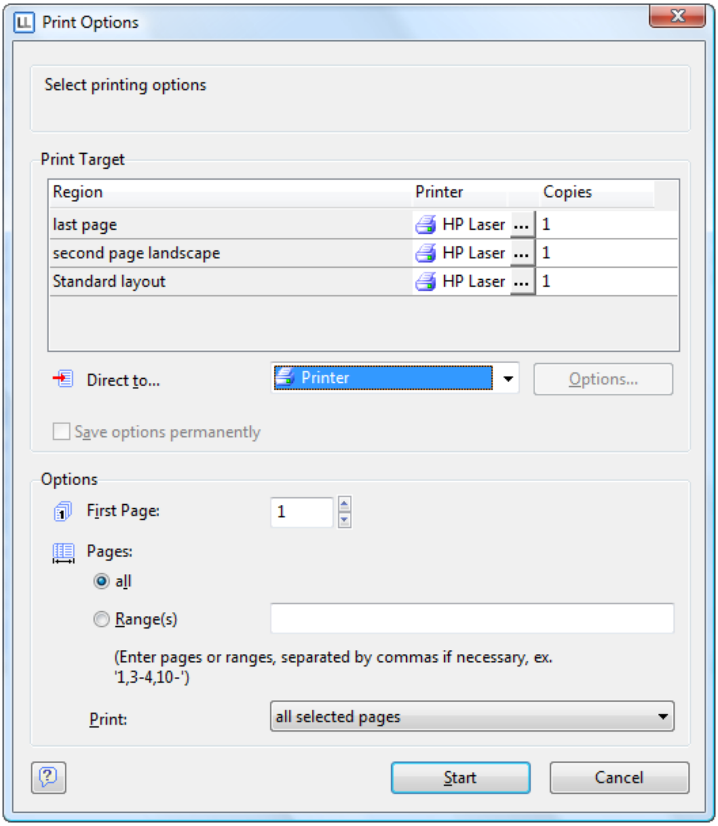
The objects of these included projects will be visible, but are locked against editing. Also included are sum and user variables of the original project. Please notice not to generate duplicate identifiers when designing, e.g. including an existing project that is using a sum variable which is already used within the current project. The sequence in which the projects are included can be defined here as well.

3.6 Printing Projects

There are several ways to print a project: printer call from application, real time data preview (assuming the feature is supported from your application) and a sample print with example data.

3.6.1 Print Options

Printing can be called directly from the upper level program via a menu option or from preview.



Select printing options

Print Target

Region	Printer	Copies
last page	HP Laser ...	1
second page landscape	HP Laser ...	1
Standard layout	HP Laser ...	1

Direct to... Printer

☐ Save options permanently

Options

First Page: 1

Pages:
● all
○ Range(s)

(Enter pages or ranges, separated by commas if necessary, ex. '1,3-4,10-')

Print: all selected pages

As long as the print call from the upper level program, it will be viewed depending on the selected template will be viewed using the dialog options selected.

Print Target

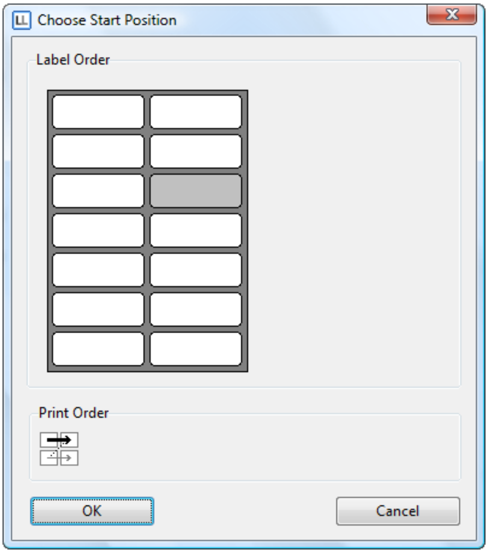
In the category "Print Target" you can, by using the "Change..." button select a different printer or different printer options than the defaults defined in your project in the Designer.

- Selecting the option "Direct to" defines the output format (e.g. preview, printer, Excel). Please take into account the range of options available for formatting that are accessible for the menu option "Option", e.g. the number of Excel rows and columns can be limited to a required number using the option "Exclusive data export from table objects".
- With the option "Preview" it is possible to send the print output to the screen for viewing of the real data and not to the printer. This way it is possible to view the real data and control the contents before printing.

- With the option "Save options permanently" this setting can be either one time or permanently saved to your project as the default.
- In the category "Options", you can enter the number of copies, the starting page number and the page range to be printed. If you have defined several regions, the copies are defined in the category "Print Target".

Label projects

For label projects, you can additionally select the position of the first label on the first page with the button "Select".



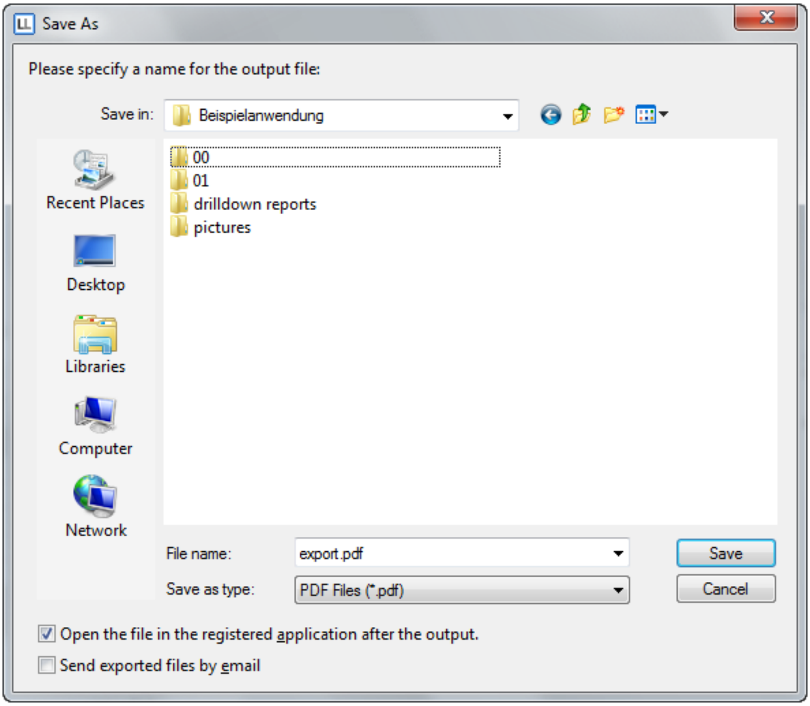
This is useful when printing on pages where labels have already been used.

The dialog shows the layout of the label page. Click on the label you want to use first during printout, taking into account the print order noted below. You can print not only from the upper left to the lower right, but you can reverse this order or print by columns. Printing will start from the label that you chose.

Output to other formats (PDF, RTF, Excel, ...)

Print outputs can also be output into other file formats. For this, select the corresponding output medium in the print options dialog. Please note that the layout may not always be transferred 1:1 depending of the format, since some have specific limits regarding the conversion.

If you have chosen a format other than printer or preview, select "Save As" in the following dialog and enter a name for the file being created in the "File name" field.



Save As

Please specify a name for the output file:

Save in: Beispielanwendung

00
01
drilldown reports
pictures

File name: export.pdf

Save as type: PDF Files (*.pdf)

☑ Open the file in the registered application after the output.

☐ Send exported files by email

In addition, you have the option of displaying the file directly in the corresponding program or to send it by e-mail. Activate the corresponding options if required.

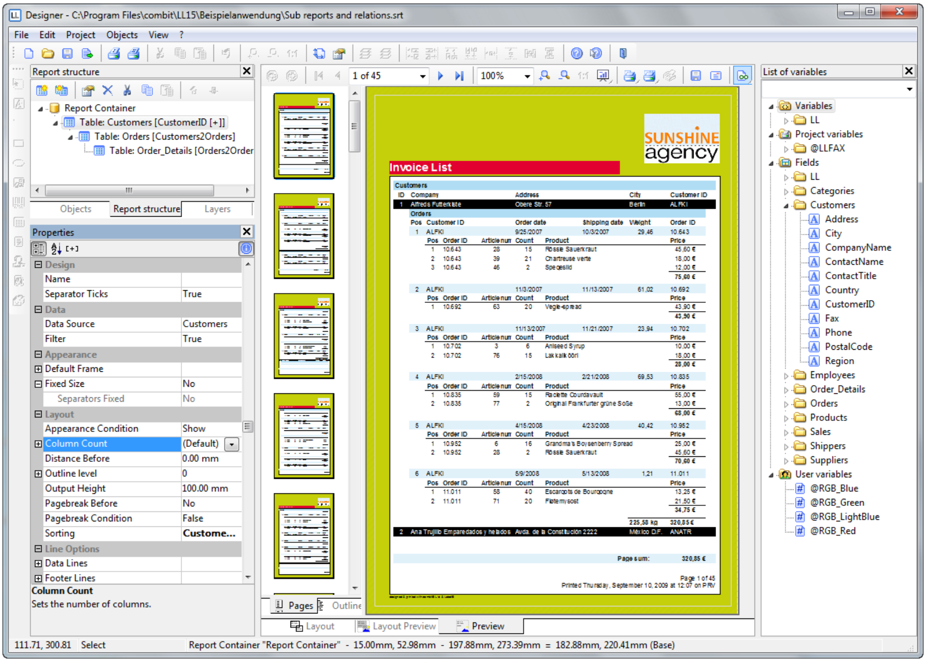
- With the option "Open the file in the registered application" you can open the file directly in the relevant program (e.g. Excel) after creating it.
- With the option "Send exported file by email" you can send the file directly by mail.
- With the option "Digitally sign created files" you can digitally sign the file.

3.6.2 Real Data Preview

Assuming the feature is supported from your application, the real data preview is available in the designer. Otherwise the preview is available in the application. Refer to your application's manual for hints on how to open the preview print.

All print jobs can be viewed prior to printing in a preview window on the monitor. In this way, you can review the layout of your project without wasting paper. The output to the monitor takes place as it will during the actual printing. After reviewing the layout, you can start printing directly from the preview without again having to use the print command.

- The entire print data can be viewed and printed in a preview window.



Designer - C:\Program Files\combit\LL15\Beispielanwendung\Sub reports and relations.srt

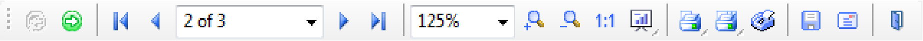
Design	
Name	
Separator Ticks	True
Data	
Data Source	Customers
Filter	True
Appearance	
Default Frame	
Fixed Size	No
Separators Fixed	No
Layout	
Appearance Condition	Show
Column Count	(Default)
Distance Before	0.00 mm
Outline level	0
Output Height	100.00 mm
Pagebreak Before	No
Pagebreak Condition	False
Sorting	Custome...
Line Options	
Data Lines	
Footer Lines	

Column Count
Sets the number of columns.

111.71, 300.81 Select Report Container "Report Container" - 15.00mm, 52.98mm - 197.88mm, 273.39mm = 182.88mm, 220.41mm (Base)

- The preview is displayed during printing (depending on application). The current print progress is shown in percentage in the head line, and the progress bar at the bottom right.
- The non-printable margins will be shaded.
- On the left side of the Preview window, you will find the minimized pages and the outline.
  - The minimized pages of the register "Pages" allows a very fast, direct navigation through the document. The number of copies set in the printing options window is ignored in the preview and only becomes relevant when printing begins.
  - The register "Outline" contains the headlines.

The preview window contains a toolbar that you can use to utilize the various functions of the preview.



- In Drilldown-Reports the green arrow buttons allow you to change to the next/previous report tab.
- The blue arrow buttons allow you to change to the next/previous page or jump to the beginning/end of the document.

- The magnifiers, as well as the 1:1 tool, represent various zoom levels that can be used to see fine details in the document. Alternatively, you can use the mouse to select an area to zoom by drawing a rectangle.
- With the Slideshow, you can generate a complete PPT presentation with slide transition effects. If you right click on this button, you can select the transition effect. Many keyboard features that are familiar from Microsoft PowerPoint (e.g. screen black and white, slide navigation) are also supported:
    - <number> ENTER: Go to Slide <number>
    - B or Period: Black/Unblack Screen
    - W or Comma: White/Unwhite Screen
    - A: Show/Hide Pointer
    - ESC, Minus, END: End Show
    - Spacebar, N, Right Arrow, Down Arrow, Page Down: Advance to Next Slide
    - Backspace, P, Left Arrow, Up Arrow, Page Up: Return to Previous Slide
- Using the button "Print Current Page", you can individually print the currently displayed page from the preview. If you right click on this button, a Printer Assignment dialog window is opened.
- Using the button "Print All Pages" will send all pages directly from the preview window to the printer. The file will be printed exactly as shown in the preview. The Printer Assignment dialog can also be opened using the right mouse button.
- Use the button "Send To" to send the current preview file e.g. through your Standard-Mail-Application to an email recipient. The preview-file can be send in different formats, e.g. LL, PDF, XPS, EMF, TIFF, ANSI, UNICODE. In order to view the *.LL-file, the recipient must use the List & Label Viewer, which is often included with the application. The List & Label Viewer can also be used to print the preview file.
- Using the button "Save As” you can save the current preview file with the *.LL file extension.
- The preview file can be saved in different formats, as e.g. data exchange format PDF. Depending on the chosen format, further corresponding properties are possibly available. Please notice that saving in another format does not guarantee that all information is copied.
- Using the "Exit Preview" button, you will leave the preview window and return to your application. The displayed print preview will not be printed, but discarded. Any changes will be lost unless previously saved with "Save As".

3.6.3 Print Sample from the Designer

Using **File** > **Print Sample** you can generate a sample printout of your current project. You can use this sample to check the layout of your project.

Fixed text appears as in the final print, but variables are substituted by sample data which is constant throughout the sample. Printing real data can not be done from the Designer.

Printing layers

With the option "**Only Visible Layers**" on the "**Preview**" card in the options dialog, which is opened with **Project** > **Options,** you define the way layers are printed in the sample print. When this option is not activated, all layers will be printed during the sample print. When this option is activated, only the layers visible in the Designer will be printed. The sample print is identical to the project in the Preview Window.

Various options are available for the test print depending upon the type of project being created.

Print Sample in Labels

By selecting the command **File** > **Print Sample** > **Print Sample with Frames**, your labels will be printed on plain paper with a frame, which represents the borders of the labels. These frames do not appear when printing real data. By using these frames, you can determine if your labels have been printed in the correct size and position.

By selecting the command **File** > **Print Sample** > **Print Sample without Frames**, a page of your project will be printed, without frames, in the way that it was defined with the exception that the variables have been replaced with sample data.

Print Sample in List Projects

The command **File** > **Print Sample** offers you the options **Print Sample 1st Page** and **Print Sample Following Page** for your list project.

With multipage projects, you can define the first page, which may have a special title or header, the following pages (which are between the first and last pages), and the last page. These three parts of a multipage project can be sample printed separately.

To print the first page of your list project, select the menu command **File** > **Print Sample** > **Print Sample 1st Page**. This will generate a sample print of the first page only.

To print the following page of your list project, select the menu command **File** > **Print Sample** > **Print Sample Following Page**. This will print the second page of your project.

Print Last Page

Depending upon the number of records to be printed in your list project, the first, or one of the following pages, can be the last page. Objects can be defined for the last page, for example a closing text, by using the appearance condition "Lastpage()”. These objects will appear on the last page only, in addition to objects that are normally assigned to the page.

When you use one of the commands for a sample print, a dialog will appear in which you will be asked if the objects with the appearance condition "Lastpage()” should be printed. If you answer the question with "YES” all objects with the condition "Lastpage()” will be printed. If you answer with no, these objects will not be printed.

This distinction is irrelevant if no objects have been created with the appearance condition "Lastpage()”.

Please note that the condition "Lastpage()" is only reasonable for use with linked objects. Detailed information about linking of objects can be found in Chapter "Interlinking objects".

4. Objects

Objects usually have a rectangular shape and are surrounded by a frame in which their size and position can be changed. This frame indicates the area which the object occupies in the workspace and consequently the maximal scope that the contents of an object can have. Objects may, however, overlap slightly or fully whereby the overlapped object may sometimes be concealed completely.

4.1 Inserting Objects

Objects may be inserted in the project workspace in different ways: using the menu Object > Insert, the toolbar, shortcut keys or per drag & drop function of the variable list. Text objects are inserted most comfortably and efficiently per drag & drop from the variable list. Simply select the desired variable and drag it to a free area in the project workspace. The easiest way to insert all other objects is via the toolbar.

1. Select the desired object type. The mouse pointer will change to a crosshair.
2. Place the crosshair on the point at which a corner of the object should be placed. It is best to use the left upper corner of the planned object. Depress the left mouse button and pull – while keeping the mouse button depressed – to the opposite corner of the planned object. If you started in the upper left corner, pull the crosshair to the lower right corner of the planned object. A dashed frame will appear which represents the size that the object will assume upon release of the left mouse button.
3. Release the mouse button when the object (dashed frame) has the desired size.

4.2 Editing Objects

Select the object that you would like to edit. The object will be marked with a selection frame. You can now:

- change the size of the object
- reposition the object
- define the exact position of the object on the workspace via the position dialog in 1/10 increments of the current unit.
- edit the contents (properties) of the object. These contents are different according to the type of object selected (text, picture, table, etc.). You can find a detailed description of the properties of objects in the subchapters.
- define appearance conditions for the object. With an appearance condition you define the condition(s) that must be met for the object to be printed. In this way

you can allow a logo to appear on the first page of a project only and not on the following pages

- name the object. Using the name you can easily find the object you are looking for
- assign the object to a (display)-layer or copy it to a layer. Multipage and/or complex projects become clearer if objects which belong together are put on a mutual layer. So, in a multipage project all objects which belong to one page can be put on a common layer. You only need to switch the visibility of this layer when you want to edit it
- copy the object. If you want to place several, similar objects with the same distance on the workspace then you can use the function Create Multiple Copies.
- If you have selected more than one object you can combine the selected objects into a group (grouping), align them or adapt their size.

These editing possibilities will now be described individually in detail.

4.2.1 Selecting Objects

You must be in the selection mode before you can select an object. The currently active mode is displayed in the middle area of the info bar.

To select an object in the selection mode just click into the object you wish to select. If you click into the object with the right mouse button, the object will be selected and a context menu will be simultaneously displayed. An object can also be selected by dragging a frame around the object by using the left mouse button. Release the mouse button when the desired object is completely enclosed with the frame. All objects within the frame will be automatically selected.

To select multiple objects press the shift-key and click with the left mouse button into the objects you wish to select. Or drag a frame completely surrounding the objects you wish to select. All objects that are completely enclosed by the frame will be selected.

When the command **Objects** > **Select** is selected, a sub-menu opens with which you can select from various select modes.

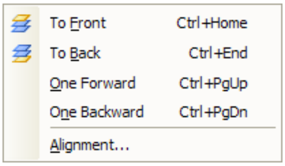
- Use the command Objects > Select > Select All or the shortcut Ctrl+a to select all objects in the workspace.
- Use the command Objects > Select> Invert Selection or the shortcut Ctrl+y to invert the selection. All selected objects become unselected and all unselected objects become selected. This is valid only for objects on visible layers.
- The order in which the objects were created is important for the command Objects > Select > Next Object. This command, or the plus key on the numeric

keypad, selects the object that is next in the object list (that is, the object created later) than the currently selected one.

- This command has the opposite effect to Objects > Select > Next Object.

4.2.2 Arrange

Objects on the workspace may overlap or completely cover each other. You can imagine that each of your objects will be printed on a transparency. When you place your objects on the workspace, place these transparencies in the proper order on top of each other. Each object is now in its own plane.



You can look at the order using the object list (Objects > Object List).

Using Objects > Arrange, you can rearrange the planes of the selected objects so that they have the order you require.

Please note that these "planes" (just a term in this case) have nothing to do with the layers. Objects that have been edited using Objects > Arrange will not have their layer assignment changed.

Select the object that you would like to arrange on the workspace, and select one of the following options.

- To Front: All selected objects are moved on top of the stack of sheets, that is, they have priority over the objects lying deeper. Use the command Objects > Arrange > To Front to bring the selected object to the top of the stack, or use the shortcut CTRL+POS1.
- To Back: Puts the selected objects to the bottom of the stack of sheets. Use the command Objects > Arrange > To Back place the selected object at the bottom of the stack, or use the shortcut CTRL+END.
- One Forward: Using the command Objects > Arrange > One Forward or the shortcut CTRL+PAGEUP, the selected objects will be moved one position up in the stack of sheets.
- One Backward: Using the command Objects > Arrange > One Backward or the shortcut CTRL+PAGEDOWN, the selected objects will be moved one position down in the stack of sheets.

4.2.3 Alignment

Using the command Objects > Arrange > Alignment or the toolbar you can align multiple objects in relation to each other. A minimum of two objects must be selected to use this command.

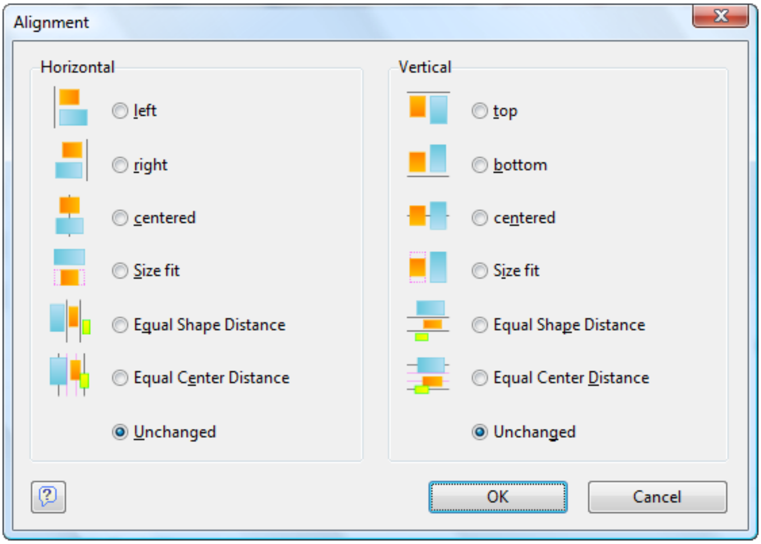
Left / Right / Top / Bottom: The functions reposition the selected objects at the corresponding border of the selection rectangle. They will not change in size.

Centered: The objects will be placed at the center of the selection rectangle. They will not change in size.

Size Fit: The objects will be resized to fill the selection rectangle in the corresponding direction (horizontal or vertical). Thus they will all get the same width or height.

Equal Shape Distance: The selected objects will be repositioned so that the distances between their frames are constant. If the objects are equal in size, this is the same as Equal Center Distance.

Equal Center Distance: The selected objects will be repositioned so that the distances between their centers are constant.



Alignment

Horizontal

○ left
○ right
○ centered
○ Size fit
○ Equal Shape Distance
○ Equal Center Distance
● Unchanged

Vertical

○ top
○ bottom
○ centered
○ Size fit
○ Equal Shape Distance
○ Equal Center Distance
● Unchanged

4.2.4 Moving and Sizing Objects

You can reposition any selected object or change its size. While in editing mode, these steps can be undone with the menu command **Edit > Undo**. This can also be accomplished by using the shortcut CTRL+Z or ALT+BACKSPACE.

> **Hint:** If multiple objects are selected these can be changed simultaneously, as with a single object.

Changing Size

1. Select the desired object.

2. When the mouse pointer is placed over the frame, it changes into a bi-directional arrow. The size can be changed by depressing the left mouse button and pulling in one of the displayed directions. To simultaneously change the objects size in both the horizontal and vertical, the mouse pointer must be placed in one of the corners of the object frame.
3. While dragging the mouse pointer, a frame is created, which reflects the new size of the object.
4. Release the mouse button when the object has reached the required size.

Repositioning an Object

1. Select the object you require.
2. Click into the object to reposition it. If the mouse pointer is positioned over the selection frame, a four directional arrow will be displayed.
3. If the shift key is pressed, objects can be shifted only horizontal or vertically. The adjustment remains.
4. Release the mouse button when the object has reached the position you require.

Repositioning and Changing Size via the Keyboard

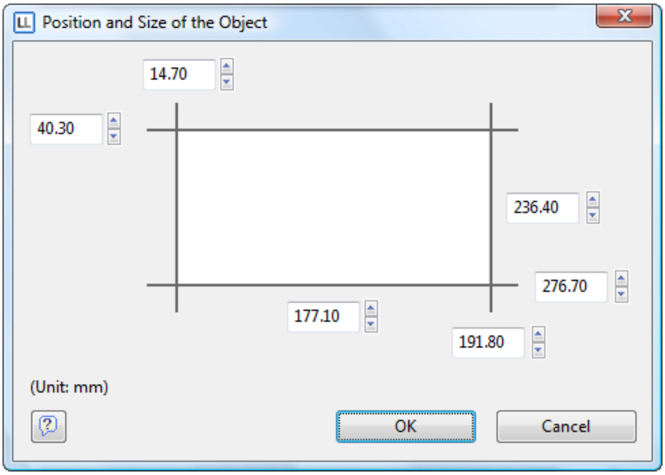
In addition to the mouse and position dialog, objects can also be repositioned on the workspace via the keyboard.

- Select the object you wish to reposition.
- Use the cursor-keys to move the object in the direction you require. Press the key once and the object moves 1/10 millimeter, if the shift-Key is also held, the object is moved by 1 millimeter.
- Use the ctrl and cursor keys for fine adaptation of the object's size
- Use the shift, ctrl and cursor keys for a quick adaptation of the object's size.

Changing Size and Position via Dialog

You can change the size and/or position of an object in the Property List. Please note that changing an entry will cause the automatic recalculation of all other related values.

If you double-click on the subcategory "Position" in the Property List a position and size dialog opens. Using this dialog, the defining of the size and position of an object using the keyboard is simplified.



Position and Size of the Object

14.70

40.30

236.40

276.70

177.10

191.80

(Unit: mm)

4.2.5 Grouping Objects

Multiple objects belonging together can be combined into a group in order to be treated like a single object. Please note that an object can only be a member of one group. It is therefore not possible to combine groups into a larger group.

To combine two or more objects use the following procedure:

- Select the objects you require.
- Select the command Group from the Objects menu or from the context menu.
- To undo a grouping select the command Ungroup from the Objects menu.

4.3 Working with Layers

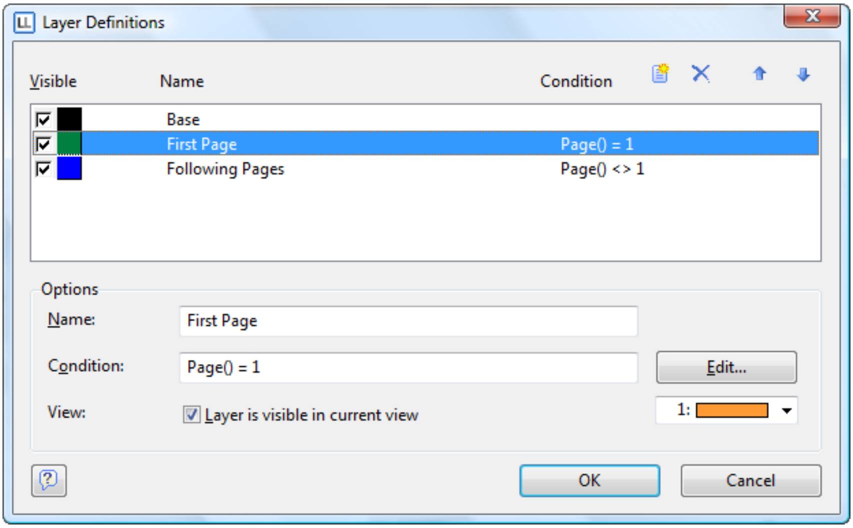
These layers are especially helpful with complex or larger projects. For larger projects, it is suggested to relate the objects on each page (first page, following pages, last page) to their own layer. Because you can fade these layers in or out as required, every page of your project can be handled separately. Otherwise the objects of different pages would be overlapping and would make the treatment of single objects more difficult.

Selected objects can also be given appearance conditions in the layers window.

It can also be helpful for complex projects to relate objects with the same appearance condition to one layer. Even the most complex forms can be created without losing control of your project.

4.3.1 Defining Layers

Layers can be defined using the command **Project > Layer Definitions**, doubleclick in the Layers tool window or by using the command **Layer Definitions** in the context menu of Layers tool window. The defined layers apply to the current project.



Visible	Name	Condition
☑ ⬛	Base	
☑ 🟩	First Page	Page() = 1
☑ 🟦	Following Pages	Page() <> 1

Options

Name: First Page

Condition: Page() = 1

View: ☑ Layer is visible in current view   1: 🟧

Inserting, Deleting and Editing Layers

In the layers window of the List & Label Designer at least one layer is defined which is called "**Basic**". When new projects are created, the layers "**Basic**", "**First Page**" and "**Following Pages**" are automatically defined.

- By using the "**New**" button you can define further layers.

Each new layer is initially presented with the name "**Layer**". You should enter a name for the new layer in the "**Name**" field that is easily identifiable. By using the Option "**Layer is visible in current view**" you can define whether the new layer is visible in the workspace or not.

To easily distinguish between the various layers on your workspace we recommend that you give the various layers different colors. All objects on one layer are then displayed in that color. The color has, of course, no effect on the actual print.

- To remove a layer that is no longer required select the corresponding layer and press the "**Delete**" button.

If the layer you wish to delete contains objects, they are automatically moved to the basic layer. In this manner you cannot lose any objects by deleting layers.

Appearance Conditions for Layers

You can define appearance conditions for each layer that define under which conditions the layer should be printed. These appearance conditions correspond to all objects of a particular layer.

Appearance conditions for single objects are given via the command **Objects > Appearance Condition**.

The appearance condition for a layer can be entered directly in the field "**Condition**", as long as you are familiar with the syntax. With the "**Edit**" button, however, it is possible for you to open the formula wizard, in which you can combine variables, text, and functions with any expressions. Further information can be found in Chapter 2.8 "Variables, Formulas, and Expressions". Typical appearance conditions for layers are those that allow objects of a layer to appear only on certain pages.

Condition	Description
no condition	The objects on this layer always appear
Page()=1	The objects on this layer appear only on the first page.
Page()>1	The objects on this layer appear on all except the first page.

Additionally, logical expressions can be used as appearance conditions.

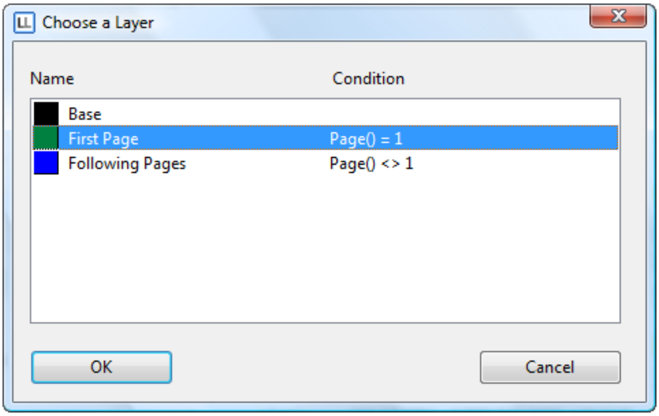
> The function LastPage() can only be correctly analyzed when it is used within the footer of a table object or in objects that are linked to a table. Within the data line the result of LastPage() is always "False".

Where will objects be printed?

- Basic objects on every page (line, rectangle, ellipse/circle, drawing, barcode, ...)
- Text and formatted (RTF-) text: Every page or continuous, starting on first page when Property Page Wrap = true.
- Table, Report Container, Crosstab: Continuous, starting on first page

4.3.2 Assigning Layers

The currently selected objects can be assigned to the different layers on the workspace.



Name	Condition
⬛ Base	
🟩 First Page	Page() = 1
🟦 Following Pages	Page() <> 1

Select the objects which you wish to assign to a particular layer, and then select the command **Assign to Layer** from the **Objects** menu or the context menu. A list with the defined layers will appear. Select the desired layer.

Double-click on the required layer to assign the selected objects to it. Alternatively, you can also click on the required layer and then confirm the dialog with "OK".

The assigned objects automatically appear in the color of the corresponding layer on the workspace. This however only applies to the appearance on the workspace.

4.3.3 Copying into Layers

The Designer allows you not only to assign objects to layers but also to copy existing objects into layers.

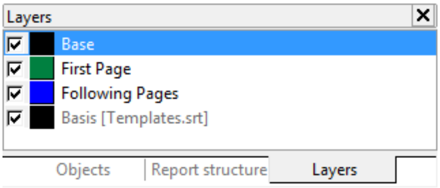
The original object stays on its layer. A copy of the object is made on the target layer.

To copy one or more selected objects into a certain appearance layer, select the command **Copy to Layer** from the **Objects** menu or from the context menu. This is practical, for example, when you wish to use many similar objects in various layers.

4.3.4 Switching Layers On/Off

The only objects that are displayed on the workspace are those that are assigned to layers that are turned "on".

You can turn layers on and off by double-clicking on the corresponding layer in the layers window or by clicking directly on the checkbox corresponding to the layer.



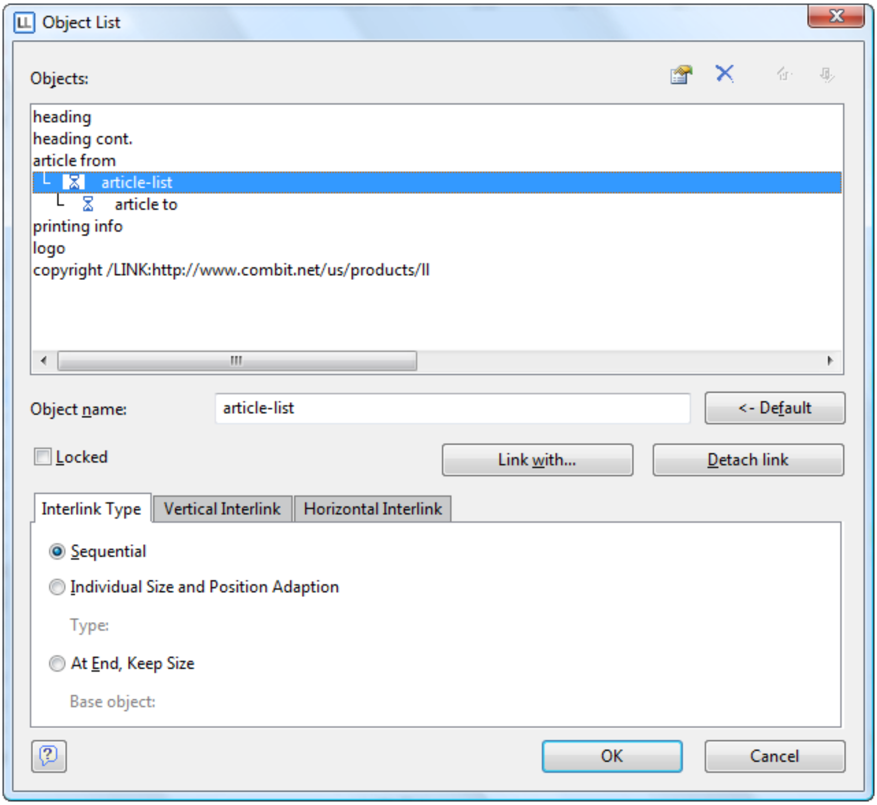
Depending on which option you have set on the card "**Preview**" (in the Options dialog via the option "**Only Visible Layers**"), either only those layers which are turned on will be displayed in the preview window, or all layers.

4.4 Object List

By using the command **Objects** > **Object List** or the shortcut NUM* (key * in the number pad) a dialog with a list of all available objects is opened.

The object list creates an implicit print order of the objects, so it automatically creates a sequential interlink. Unlinked objects are printed first, followed by any objects that are linked in any way.

This dialog contains the Object List, the appropriate to window, and features added functionality.



heading
heading cont.
article from
  └ article-list
    └ article to
printing info
logo
copyright /LINK:http://www.combit.net/us/products/ll

Object name: article-list

☐ Locked

● Sequential

○ Individual Size and Position Adaption

Type:

○ At End, Keep Size

Base object:

- Editing or Deleting Objects: This can be done using the "Edit" and "Delete" buttons. You may also start the object properties dialog by double-clicking on an object in the list. If multiple objects are selected when clicking the "Delete" button, all objects are deleted.
- Lock objects: If the option "Locked" has been selected for an object, the object can be neither selected nor edited on the workspace. It can, however, be edited using the Object List. If the option is not selected, the object can be selected and edited on the workspace.
- Naming Objects: The edit control allows you to change the name of the selected object. The name is of course the same name that can be changed using the property list.
- Resetting the Name: The "<-Default" button will set the default text (type and coordinates) as the name of the object.

4.5 Interlinking objects

By interlinking objects, you can set that some objects are printed after other objects and that they superpose when they overlap ("sequential interlinking") or that some objects adjust in size and position responding to changes in other objects ("spatial interlinking").

There are three types of interlinking:

- Sequential
- Individual Size and Position Adaption
- At End, Keep Size

When interlinking the objects there is a hierarchy: the parent object and the attached (interlinked) object.

Select Objects > Object List to create a interlink. With the button "Link with...", you can interlink other objects to the object selected in the object list. A list of objects available to be linked will appear. Select the object that you wish to attach to the object selected in the object list. The tree structure visualizes the interlinks. To attach several objects to an object, repeat the steps described above.

As soon as an interlinked object in the object list is selected - e.g. by clicking it with the mouse – various types of interlink are made available in the "Interlink" field, which will be described more closely hereafter.

4.5.1 Detaching Interlinks

If you have selected an interlinked object in the object list, the button "Detach Link" is available. Click "Detach Link" to break an existing link. The linked object will be displayed as an regular object in the object list.

4.5.2 Individual Interlinking (Size and Position Adaption)

Both types of interlinking are relating to the used space of the parent object, i.e. the linked object will be subordinated to the parent object spatially, which means it will be subordinated in position and size. This automatically creates a sequential link too.

If the parent object changes in size or position , e.g. because the variables contained within it take up less space than is available, the interlinked objects will automatically adjust to these changes.

Spatial interlinks are only effective if the size and position of the parent object change when the variables are replaced by the respective field contents. The objects can shrink, but not enlarge. Manual changes in the size and position of the parent object in the workspace have no effect on the interlinked objects.

Spatial interlinks are marked with a square symbol in the object list. For clarity, the type of interlink selected is displayed on the "Interlink Type" tab.

Example Invoice Print: An invoice generally consists of a table that shows the individual invoice line items on any number of pages. The first page should of course differ from the others because it also has a letterhead. Create an invisible frame on the base layer, which starts where the invoice table is to start on the following page and ends where the invoice table is to start on the first page. Assign the display property "Page()=1" to the rectangle. The table is also created on the base layer. This starts below the rectangle object. Now link the table with the rectangle (parent object) and set the link types "Position Adaption: Vertical Relative to End" and "Size Adaption: Vertical Inverse Proportional".

The interlink type is selected under the "Vertical Interlink" and "Horizontal Interlink" tabs.

Vertical Interlinking

The interlinked object adjusts in its position or height depending on the vertical position or size changes of the parent object. Which corner of the parent object is applicable for the position adaption of the linked object, depends on the option selected ("relative to Begin", "relative to End" or "to End" for the position, "proportional" or "inverse" for the size). If the parent object moves vertically, the interlinked object will move by the same distance in the same direction.

Horizontal Interlinking

The interlinked object adjusts in its position or width depending on the horizontal position or size changes of the parent object. Which corner of the parent object is applicable for the position adaption of the linked object, depends on the option selected ("relative to Begin", "relative to End" or "to End" for the position, "proportional" or "inverse" for the size). If the parent object moves horizontally, the interlinked object will move by the same distance in the same direction.

If both options "horizontal" and "vertical" are on, the linked object will adjust to both types of change of position of the parent object.

In any case, two completely different types of interlinking are available to you:

Position adaption

This adaption depends on the position of the parent object. If the parent object changes its position, the position of the linked object changes in the same way (depending on the additional options).

- Option "relative to Begin": The child object moves like the top left corner of the parent object.
- Option "relative to End": The child object moves like the bottom right corner of the parent object.
- Option "To End": The top edge of the child object starts at the end of the parent object, independent of its original position. This creates an implicit size change on the first page on which the child page is printed.

Size adaption

This is similar to the position adaption, but as the name suggests, this interlink depends on the size of the parent object, i.e. when the parent object changes in size, the interlinked object will also change its size (also dependent on the additional options).

- Option "proportional": The size of the child object changes in the same way as the parent object. If it becomes 1cm shorter, the child object also becomes 1cm shorter.
- Option "inverse": The size of the child object changes inversely proportional to the size of the parent object. This is an especially useful option: if the parent object becomes 1cm shorter, the child object becomes 1cm longer.

4.5.3 The "At End, Keep Size" Interlink

This type of interlinking is similar to the position adaption, but here the available space of the parent object is taken into consideration and the size of the linked object remains fixed. If you want to output a chart object after a table and know exactly what the chart object should look like, then you place it directly behind the table and link it "At End, Keep Size" with the table. Regardless of where the table ends, the chart object will always be output in its chosen size after the table. If there isn’t enough space behind the table on the last page, then the chart object is printed on the next page to leave its size unchanged.

In this way, you can place several objects beneath each other (such as diagrams, pictures, etc.); the base object of the interlink is the first object in the interlink hierarchy with activated page break.

The linked object must overlap the parent object in the designer. It is important that the parent object is larger than the linked object in every situation, since the linked object always tries to take up the space that is left over from the original size of the parent object. If, in our example, the chart takes up more vertical space than the table, List & Label will make a page break after the table and try to print the chart on the next page, but still within the original area of the table. The chart will still not fit, and another page break will be triggered, which leads to a neverending loop when printing.

4.5.4 The "sequential" interlink

Sequential interlinks are useful when the linked object can only be filled with information once the parent object has been printed.

Example 1: You are writing text and want to have the number of letters and words written on every page counted. The linked object only knows this when the text (text object) has been printed. The sum can then be printed at the end of the page. Example 2: You want a final text to be printed after a table, and give it the display condition "LastPage()". Because it is only certain what size it is after the table has been output, the final text can therefore only be printed once the table is finished. For this, you must sequentially link the table with the final text and the final text will then only print after the table.

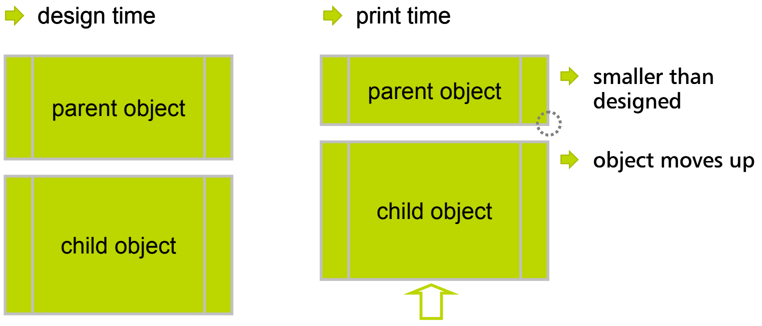
Sequential interlinking is the default for linking objects. It is denoted with an hourglass symbol in the object list. When the sequential interlink is selected, none of the options for position and size adaption are available in the interlink field.

> The object list creates an implicit print order of the objects, so it automatically creates a sequential interlink. Unlinked objects are printed first, followed by any objects that are linked in any way. This means sequential interlinking is only necessary in very specific cases, e.g. when an unlinked object is to be printed after the linked objects.

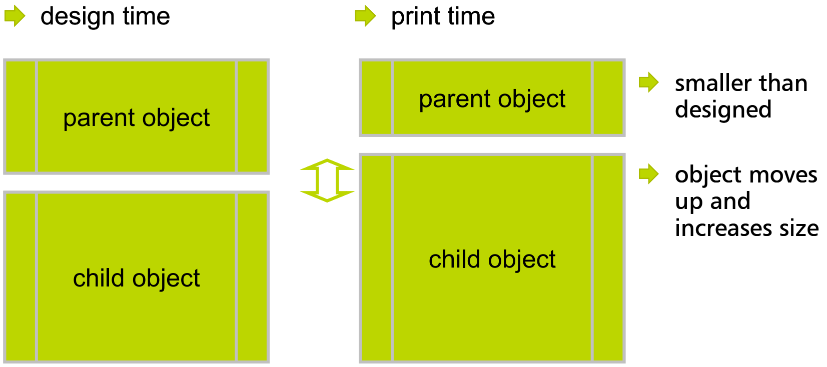
4.5.5 Examples for Individual Interlinks

How and when to use the various interlinks is described in the following pages with examples.

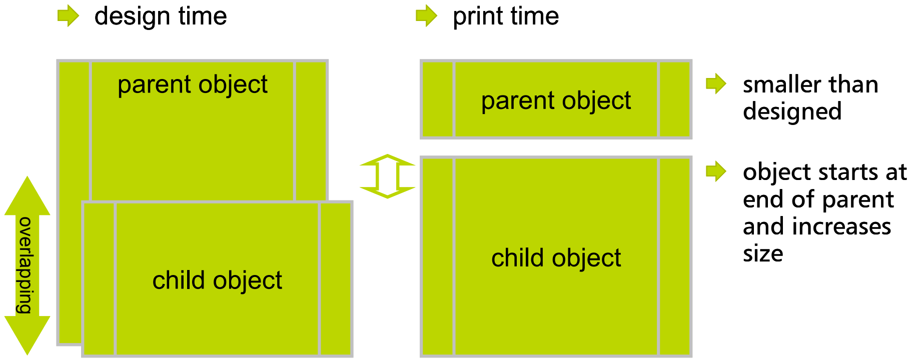
Example 1: Position Adaption: Vertical Relative to End
The linked object (child object) changes its position depending on the lower right corner of the parent object. If the parent object moves up, the child object moves upward proportionally too.



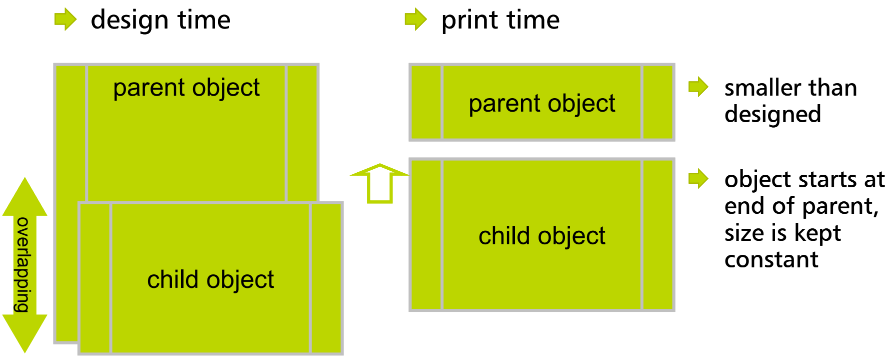
Example 2: Position Adaption: Vertical Relative to End & Size Adaption: inverse
The linked object changes its position depending on the lower right corner of the parent object and increases size proportionally.



Example 3: Position Adaption: Vertical To End
The top edge of the child object starts at the end of the parent object, independent of its original position. This creates an implicit size change, since the top edge of the linked object will be changed because of the parent object but the position of the parent object does not change. The linked object must overlap the parent object.

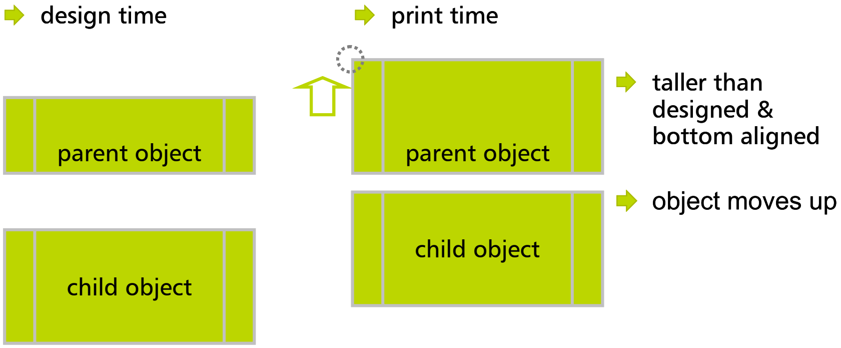


Example 4: Interlink Type: At End, Keep Size

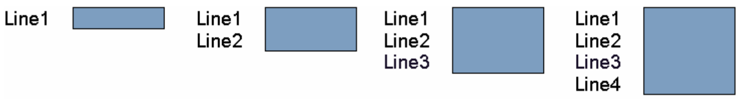


Example 5: Position Adaption: Vertical Relative to Begin

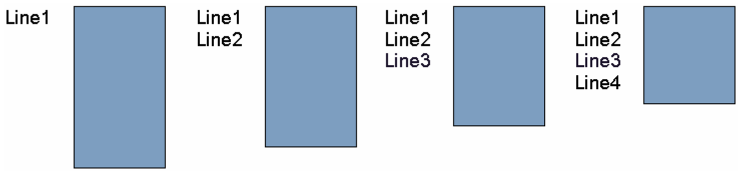
The linked object changes its position depending on the top left corner of the parent object. With this link, it behaves exactly opposed to the "vertical relative to end" interlink: The parent object moves its position upwards because of the setting "Bottom aligned = True" in the designer and the linked object follows this position adaption from bottom to top.



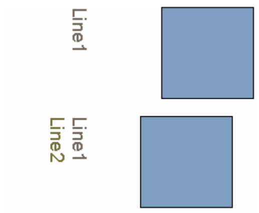
Example 6: Size Adaption: Vertical Proportional
The linked object changes proportionally (in the same way) to the parent object in size. If the parent object gets a new row, the linked object becomes larger by the same dimension.



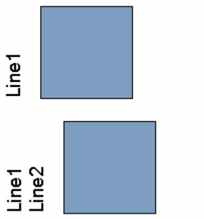
Example 7: Size Adaption: Vertical Inverse
The linked object, behaves inversely (in the opposite direction) to the parent object, i.e. if the parent object gets a new row, the linked object will become smaller by the same dimension.



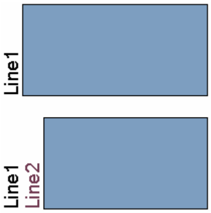
Example 8: Position Adaption: Horizontal, Relative to Begin
The linked object changes in position depending on the top left corner of the parent object.



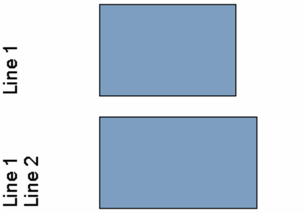
Example 9: Position Adaption: Horizontal Relative to End
The linked object changes in position depending on the bottom right corner of the parent object.



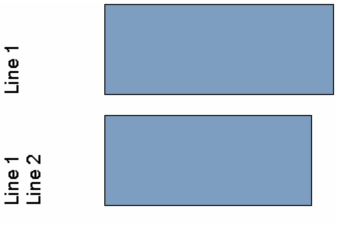
Example 10: Position Adaption: Horizontal To End
The linked object changes its position depending on the parent object. This brings about a change in size because the top edge of the linked object changes depending on the parent object, but the linked object does not change position. Therefore the linked object becomes smaller. The two linked objects must overlap (in the example, the text object is in the foreground), or this type of interlink has no effect.



Example 11: Size Adaption: Horizontal Proportional
The linked object changes its size proportionally (in the same way) to changes in the parent object. If the parent object becomes larger, the linked object will become larger by the same factor.



Example 12: Size Adaption: Horizontal Inverse
The linked object changes inversely (in the opposite direction) to the parent object. If the parent object becomes larger, the linked object becomes smaller.



4.6 Common Object Properties

Most properties can be defined using the Property list or additional dialogs. The properties for every object are different, but there are a series of characteristics that all object have in common, for example size, position, name and appearance conditions. The properties are described here once, and are not listed in the following chapters. The properties for individual objects are described in detail in the relevant subchapter of this Chapter.

When a fixed list of values are available for a property, a button ▼ will be available in the second column of the list. This button opens the list of predefined values for the property.

As long as a "Formula" value is provided, you can set the property value by using a function or formula.

Example 1: Bold font if a sum exceeds "1000"

1. Make sure the "Default" property of the font is set to "False"
2. Choose "Formula" as value for the "Bold" property
3. Enter a condition, e.g. "Sum>1000"

Example 2: Changing the position of a text depending on the page number

Assume you want to reserve space for the address label on the first page of a letter. On the following pages, the letter text should span the whole page, of course. You may use a formula for the Position.Top and Position.Height properties of the letter text object.

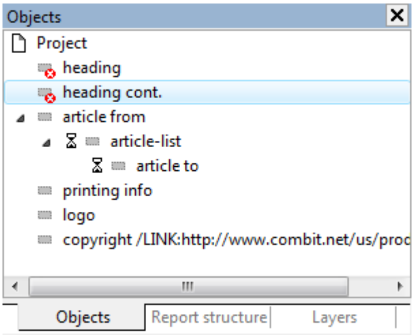
1. Select the formula-button at the Position property "Top" and enter a condition, e.g. "Cond(Page()>1,20.0,120.0)".
2. Select the formula-button at the Position property "Height" and enter a condition, e.g. "Cond(Page()>1,254.0,154.0)".

4.6.1 Locked

Protects an object from being selected accidentally by a mouse click. This property is only relevant for the design and has no effect upon the print. If an object is locked it can no longer be selected in the workspace and will be marked with a small red icon in the Object List. A locked object can, still, be selected in the Object List for editing.

Since "Locked" is only relevant for the design phase of a project, there is no possibility to define the value of this property using formulas.

Property	Description	Value	Description
Locked	Protects the object from accidental selection in the workspace through a mouse click	True	Locked
		False	Unlocked



4.6.2 Name

When you insert a new object onto the workspace, an object description appears in the right segment of the status bar. The description is made up of the object's type and coordinates. This is the default name for the object.

If, however, you have a large number of similar objects in your project, these descriptions can quickly become confusing. For this reason, you can give the object a meaningful name using the tool window **Objects** or the object's Property List. In the Property List, click once on the current name to change it.

Alternatively, you can change the object's name by using the command **Objects > Object List** or in the Property field.

Property	Description	Value	Description
Name	Object Name	Name	

The object name is – unlike most properties – a fixed string, thus needing no quotation marks for a fixed text. If you have activated the option **Options > Workspace > Object Info**, the object name will be displayed in the tool tip, when the mouse is over the object.

4.6.3 Appearance Condition

An appearance condition can be set for every object, which defines under which conditions the object will be printed. These appearance conditions are listed in the Property List under the category "**Layout**". You will find guidance for the definition of these conditions in Chapter "Variables, Formulas and Expressions".

Property	Description	Value	Description
Appearance Condition	Appearance condition for printing. The object will not be printed in the result is "False".	True	Display
		False	Do not display
		Formula	Formula dialog

4.6.4 Outline level

Sets the index level for the outline entry (0=do not add to outline).

Property	Description	Value	Description
Outline level	If the condition returns "True", a page wrap will be initiated before printing the object.	True	Break
		False	No break
		Formula	Formula dialog
Text	Text for the outline entry.	Text	Formula dialog

4.6.5 Pagewrap Before Object Output

Every object may initiate a page wrap prior to being printed; this means that the object will be printed on a new page.

Property	Description	Value	Description
Pagewrap Before	If the condition returns "True", a page wrap will be initiated before printing the object.	True	Break
		False	No break
		Formula	Formula dialog

4.6.6 Export as Picture

To export objects in picture format, if a vector export cannot be used for this purpose. (e.g. complex vector graphics with LL.OutputDevice="PDF").

Property	Description	Value	Description
Export as Picture	Sets if the object should be forced to export as picture.	True	Yes
		False	No
		Formula	Formula dialog

4.6.7 Position

The property group "Position" of an object defines the x- and y- coordinates of its upper left corner, the object width and the object height.

Property	Description	Value	Description
Position	Position and size of the object. All data is entered in the units used on the workspace	...	Position dialog
Left	Horizontal offset of the upper left corner of the object relative to the upper left corner of the workspace	Number	
		Formula	Formula dialog
Top	Vertical offset of the upper left corner of the object with the upper left corner of the workspace	Number	
		Formula	Formula dialog

Width	Object width	Number	
		Formula	Formula dialog
Height	Object height	Number	
		Formula	Formula dialog

4.6.8 Font

The font properties can be defined in a dialog. If **Default** is set to True, the default font is used. You can define the default font by selecting the command **Project > Options**. Default values set on the card "**Objects**" are valid for all new objects, until they are changed. In the "**Object Font**" group you can define the font for each newly inserted object, using the "**Select**" button. With the button "default" in the object properties the font can be adjusted to a predefined system font. The settings also affect all objects which were not changed manually to a non default font.

Property	Description	Value	Description
Font	The font properties can be defined in a dialog. If Default is set to True, the default font is used.	A	Font dialog
Default	The default font is used instead of the one set	True	Default font
		False	No
		Formula	Formula dialog
Name	Selection of font. All installed fonts are displayed.	List	Font
		Formula	Formula dialog
Character set	Sets the country version of the character set. All available character sets are displayed.	Number	Character set
Size	Size of the font in points. Shows a list of the available sizes of the selected font.	Number	Default size
		Formula	Formula dialog
Width	Sets the width of the font 0 means standard width, otherwise the average character width will be used.	Number	Width
		Formula	Formula dialog
Bold	Turns on or off the text setting "Bold"	True	Yes
		False	No
		Formula	Formula dialog
Italic	Turns on or off the text setting "Italic"	True	Yes
		False	No

		Formula	Formula dialog
Underline	Turns on or off the text setting "Underlined"	True	Yes
		False	No
		Formula	Formula dialog
Strikeout	Turns on or off the text setting "Strikeout"	True	Yes
		False	No
		Formula	Formula dialog
Color	Font color		Color dialog
			Selection of predefined colors

4.6.9 Colors

Colors are always defined in the Property List in the same way.

Property	Description	Value	Description
Color	A color dialog is available.	…	Color dialog
	The color can be selected from either the list of predefined colors or by using a formula. (1) The color is selected using the HSL function and consists of a Hue (0-360), Saturation (0-1) and Lightness (0-1) portion. (2) The color consists of a red, green and blue portion and is selected using the RGB function. Each color portion can assume a value between 0 and 255. 0 denotes no portion of this color, 255 denotes full color saturation.	▼	Choice of predefined colors and the Formula dialog

4.6.10 Filling / Background

With the property Filled a color gradient can be set for the object fill. This property can be found, for example, for the crosstab background, table background, ellipse and rectangle.

Property	Description	Value	Description
Filled / Background	Select the type of gradient and set its color, end color and mid-color. The property color, end color, mid-color is only used when "Filled" or "Background" are set to a value [2...5].	0	Transparent
		1	Pattern/Block Color
		2	Horiz. Gradient

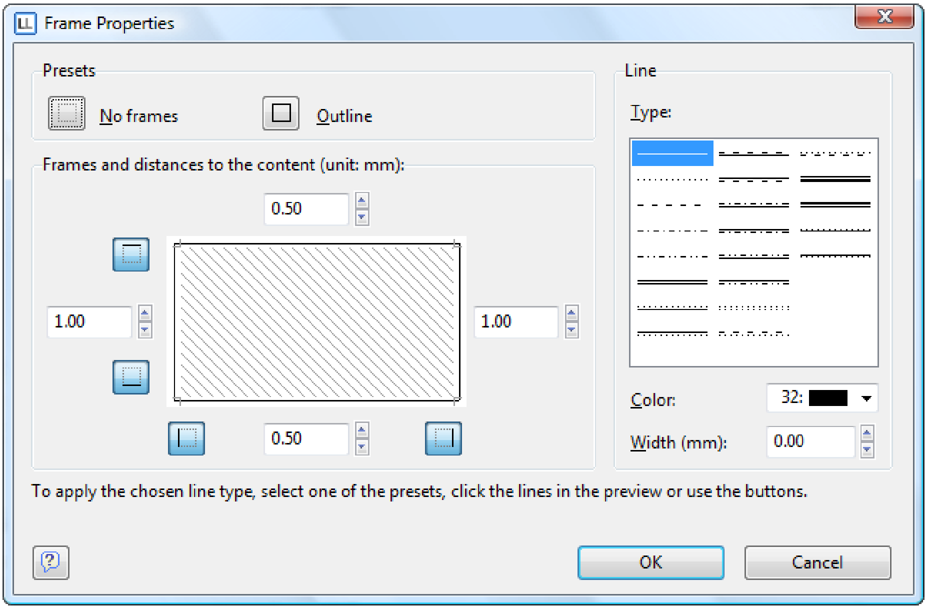
		3	Vert. Gradient
		4	Horiz. Two-Part Gradient
		5	Vert. Two-Part Gradient
		6	Partly Transparent
		7	Drawing
		Formula	Formula Editor

4.6.11 Frame

The property group "Frame” defines the frame properties and the content's distances from the frames. It is available for many objects and sub-objects (e.g. table columns).

Property	Description	Value	Description
Frame	Frame properties and offset can be defined in a dialog. To apply the chosen line type, select one of the presets, click the lines in the preview or use the buttons.	...	Frame Dialog
Default (in sub-objects)	When true, the default frame is used, which is defined in the object's base properties.	True	Use default
		False	Use custom
		Formula	Formula Dialog
Layout	Describes the type of line draw order being used with frames with more than one line on it's sides.	0	Surrounding
		1	Horiz. Priority
		2	Vert. Priority
		Formula	Formula Dialog
left/top right/bottom	Settings for the individual frame lines.		
Distance	Distance between the contents and frame.	Number	
		Formula	Formula Dialog
Line	Visibility of the frame line.	True	Line
		False	No Line
		Formula	Formula Dialog
Color	Choice of the line color.		
Line Type	Choice of the line type.	Choice	Choice of pre-defined lines (20)

		Formula	Formula Dialog
Width	Line Width	Number	
		Formula	Formula Dialog



Frames and distances to the content (unit: mm): 0.50, 1.00, 1.00, 0.50

Color: 32:

Width (mm): 0.00

To apply the chosen line type, select one of the presets, click the lines in the preview or use the buttons.

4.6.12 Pattern

With the pattern property you can define the texture of a color. This property is available with the background colors of table rows and drawn objects and is shown in combination with the color property.

Property	Description	Value	Description
Pattern	Choose a pattern from many predefined patterns. A value is always defining a pattern.	Number	Predefined sample
	This property is only analyzed if "Filled" is set to "True".	Formula	formula dialog

4.6.13 "Contents" of Objects

Objects that contain sub-objects, for example a text field that contains multiple paragraphs, can have additional properties defined that may not appear in the Property List window.

To edit the contents of these objects, select the object and use the command **Contents** in the **Objects** menu or in the **Context menu**, or by using the properties button in the content property of the Property List. Often, you can also double-click on the object. These commands can not be used if a Contents dialog is not available for the selected object.

If available, a dialog window will open that may have a different appearance depending on the type of object.

You will find further information about the Contents dialogs for individual objects in the following Chapters.

4.7 Inserting Text Objects

Text objects are used to place text in the workspace. In addition to fixed text, you can also use place holders (variables) such as page number, date, company name, etc. that are made available by your application. These variables are then replaced during printing by their assigned contents.

Text objects can be inserted into your project using the command **Objects > Insert > Text** or the shortcut CTRL+T.

4.7.1 Properties

See also Chapter "Common Object Properties".

Property	Description	Value	Description
Bottom Aligned	Alignment at the bottom of and within the object frame. If this option is activated the object's text will be aligned on the lower edge, if not, on the upper edge. Requirement is that the paragraphs are not larger than the object. If so, the text will be clipped. This option is useful e.g. when text of an unknown length is to be placed at the bottom of a page.	True False Formula	Bottom Aligned Top Aligned Opens Formula dialog
Rotation	Rotates the object counter-clockwise. Please note that only TrueType fonts can be rotated.	0 1 2 3 Formula	0° 90° 180° 270° Formula dialog
Page break	Defines if the object can initiate a page break. If the property is activated contents that exceed the size of the object will automatically appear on the next page. This is useful for text objects that occupy multiple pages. With labels, the next label will not be started until all objects with this option have been printed on the	True False Formula	Page break No Page break Formula dialog

	current label. It is possible that this option cannot be used if page break is not supported by your application. If the expression's result is FALSE, the text will be reprinted on every page.

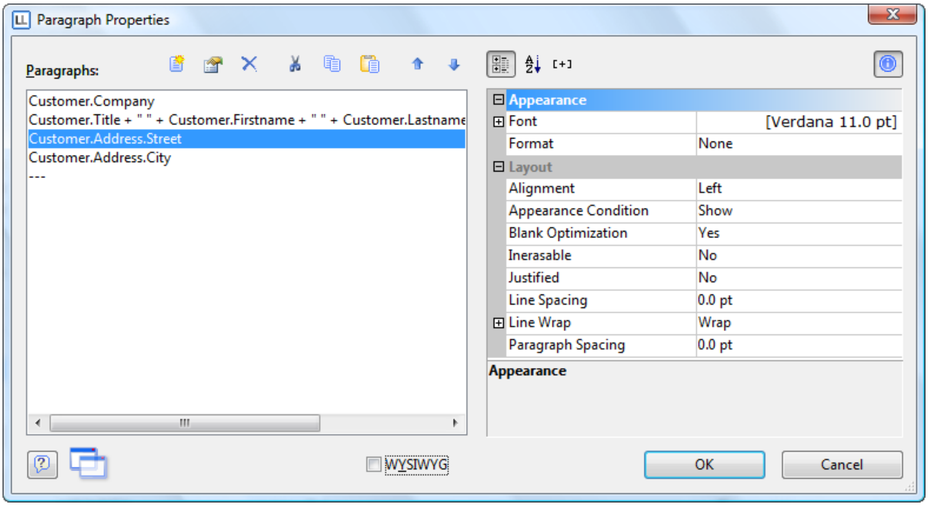
4.7.2 Text Contents

A text object can contain many paragraphs that may have totally different display properties. These paragraphs and their properties are the contents of a text object.

The paragraphs can be individually edited in the Paragraph Properties dialog.

4.7.3 Editing Paragraphs

Text objects are paragraph oriented, this means that each paragraph can be individually edited and formatted. By selecting multiple paragraphs, you can edit their properties simultaneously.



Customer.Company
Customer.Title + " " + Customer.Firstname + " " + Customer.Lastname
Customer.Address.Street
Customer.Address.City

Appearance	
Font	[Verdana 11.0 pt]
Format	None
Layout	
Alignment	Left
Appearance Condition	Show
Blank Optimization	Yes
Inerasable	No
Justified	No
Line Spacing	0.0 pt
Line Wrap	Wrap
Paragraph Spacing	0.0 pt

Appearance

With these buttons you can:

- insert a paragraph
- edit the selected paragraph(s)
- delete the selected paragraph(s)
- copy the paragraph to the clipboard and then delete
- copy the paragraph to the clipboard
- insert paragraphs/text from the clipboard
- move paragraphs up

- move paragraphs down

You can also insert new paragraphs using drag & drop from the variable list.

Such an expression can contain fixed text, a variable, a function or any combination of these elements.

WYSIWYG: If the option "**WYSIWYG**" is selected, the fonts for the individual lines will be displayed in the dialog as you have formatted them. If the option is not selected, then a standard font will be used and special formatting such as size or color will not be displayed in the paragraph list.

Edit Paragraph: A paragraph can be edited by clicking on the appropriate button or by double-clicking with the mouse on a paragraph in the paragraph list

This opens the Formula Editor that has an additional card "Tab" (see below). After you have defined the expression for the line, you can accept the line into your text object with the button "OK". You then automatically return to the dialog "Text Properties".

4.7.4 Inserting tabulator

When a text contains a tab, e.g. by inserting it in the "Text" card, the button "Properties" specifies the tab alignment options. You can set both the alignment and position of the tabulator.

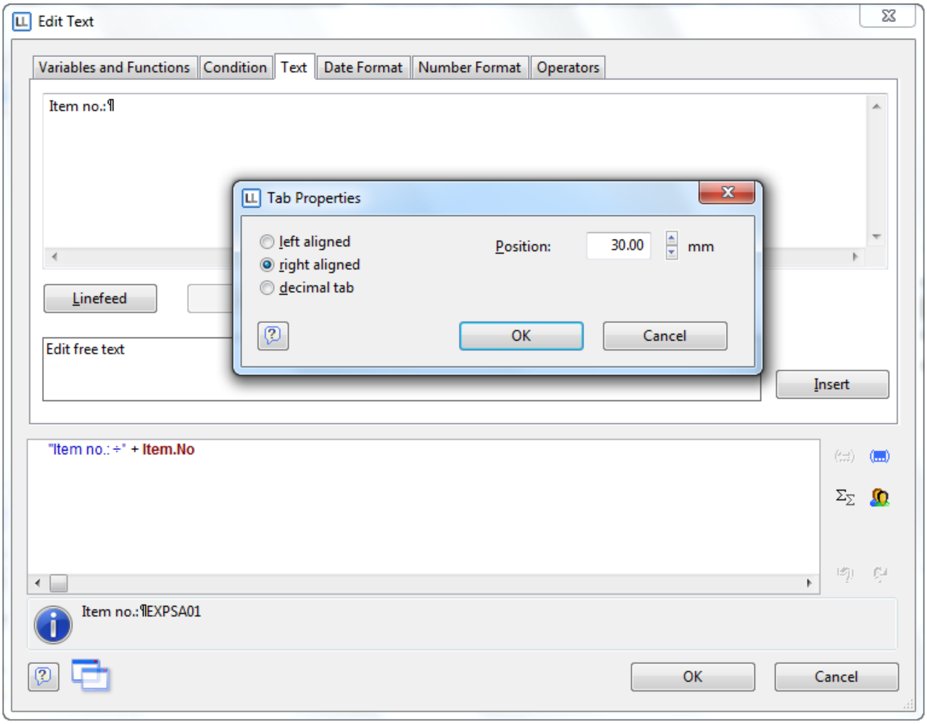
Tabulator Alignment: A tabulator causes the preceding text to run, to a maximum, to the tab stop. If the option "**Line Wrap**" in the Paragraph Properties is selected, the text will be wrapped into the next line, if necessary. Otherwise, the text will be cut.

The text after the tabulator will be wrapped, dependent on the type of tab stop, if the paragraph property "Line Wrap" is selected:

- left aligned: the text is left justified in the area between the tabulator and the right edge of the object.
- right aligned: the text is right justified in the area between the tabulator and the right edge of the object.
- decimal tab: the number after the tabulator will be aligned on the decimal tabulator. (Note: Text or numbers in front of a decimal tabulator will not be wrapped!).

Tabulator Position: The tabulator's position within the selected object is given in the current unit. A positive value signifies that the position is calculated from the left edge of the text object. A negative value defines the position relative to the right edge of the text object.

A position of "50" indicates that the text in front of the tabulator will run to a maximum of 50 in the current unit from the left edge of the text object, after which it will be either wrapped or cut. Text following the tabulator will start at a distance of 50 in the current unit from the left edge of the object and run to a maximum to the right edge of the object.



Item no.:¶

○ left aligned
● right aligned
○ decimal tab

Position: 30.00 mm

"Item no.:÷" + Item.No

Item no.:¶EXPSA01

4.7.5 Paragraph Properties

A paragraph's properties are defined using a Property List whose functioning is identical to the tool window "Properties".

See also Chapter "Common Object Properties".

Property	Description	Value	Description
Paragraph spacing	Space between paragraphs ("paragraph end spacing"), vertical blank space The spacing is entered in points: To achieve a 1.5 line spacing with a 10 point font, enter 5 points. Negative values are allowed.	Number	
		Formula	Formula Dialog
Alignment	Here you can, exactly as in word processing, define the alignment of individual lines of text.	0	left
		1	centered
		2	right
		Formula	Formula dialog

Block	Block text is both left and right justified. This property is only functional when the selected line is wrapped. This means that the line must be longer than the available space, consist of multiple words and the option "Line Wrap" selected. The last line will be displayed in accordance with the property "Alignment".	True False Formula	Yes No Formula dialog
Format	The format editor is available in text object paragraphs and in table columns and is an alternative to formatting with the functions Date$() and FStr$() in the formula dialog. Using the format editor, you can set the format for numbers, currency, dates, times, percentages and angles. By default, the computer's system settings are used. Note that the format settings affect the entire expression. If you only wish to format certain sections of the expression (e.g. some text and numbers within the expression) use the functions Date$() and FStr$() in the formula dialog.	True False	Dialog
Blank Optimization	Blank-optimization can remove undesired spaces (leading, enclosed and remaining). Imagine that you wish to print a label project using the following variables which are to be separated with a space: <TITLE> <FIRSTNAME> <LASTNAME> If there is a record in which no title exists, this variable would remain empty but the following space would be printed. First name and Last name appear with a leading space. If, for example, the variable "Firstname" was empty, there would be two spaces between "Title" and "Lastname" (enclosed spaces). If all three variables were empty, both spaces would remain (remaining spaces). This line is then not empty and would not be automatically ignored. In these cases, the option "Space-Optimization" assists in the automatic removal of leading, enclosed and remaining spaces. Multiple enclosed spaces will be automatically reduced to one space.	True False Formula	Yes No Formula dialog

Ineraseable	Lines that remain completely empty will be ignored. The following lines will be moved up. This usually comes handy, but at times can be undesirable. With the option "Ineraseable" the empty line will remain if it would be empty after use of the variable.		True False Formula	Yes No Formula dialog
Line Spacing	Line spacing defines the space between individual lines of a paragraph. The spacing is entered in points: To achieve a spacing of 1.5 lines with a 10 point font, enter the value 5. Negative values are also allowed.		Number Formula	 Formula dialog
Line Wrap	Determines the behavior, if the text becomes too long for a line. A long word is cut off at value "1" (Wrap). With the option "Force Wrap" you can force a wrap after the last suitable character. Value "3" (compress) reduces the character separation and should only up to a certain limit be used around the legibility to ensure. Optimal Fit (value "4") causes the font size to be varied in a way to fill out the object as completely as possible.		0 1 2 3 4 Formula	Truncate Wrap Shrink Compress Optimal Fit Formula dialog
	Pagebreak	With this property you define whether the paragraph will be kept together during a page break.	True False Formula	Wrap Keep Together Formula dialog
	Force Wrap	Sets if a long word should be forced to wrap after the last suitable character.	True False Formula	Yes No Formula dialog

4.8 Inserting Lines

To create a line object, select **Objects** > **Insert** > **Line** (CTRL+L).

4.8.1 Properties

See also Chapter "Common Object Properties".

Property	Description	Value	Description
Width	Line width in the measuring system used in workspace	Number Formula	 Formula dialog

4.9 Inserting Rectangles

To create a rectangle object, select **Objects** > **Insert** > **Rectangle** (CTRL+R).

4.9.1 Properties

See also Chapter "Common Object Properties".

Property	Description		Value	Description
Border	Define whether the rectangle should have a border.		0 1 Formula	transparent pattern / block color Formula dialog
	Color	Color of the frame.		
	Width	Width of the frame in the measuring system used in workspace	Number Formula	 Formula dialog
Rounding	Rounding factor for the corners of the rectangle in % (0=rectangular, 100=elliptical)		Number Formula	 Formula dialog
Shadow	Define whether the rectangle should have a shadow.		0 1 Formula	transparent pattern / block color Formula dialog
	Pattern	Shadowpattern.		
	Color	Shadowcolor.		
	Width	Width of the shadow in the measuring system used in workspace	Number Formula	 Formula dialog

4.10 Inserting Ellipses

Ellipse objects are also available, as a special type of ellipse, for the display of circles.

To create an ellipse object, select **Objects** > **Insert** > **Ellipse** (CTRL+I).

4.10.1 Properties

See also Chapter "Common Object Properties".

Property	Description	Value	Description

Filled	Define if the ellipse object should be empty or if it should be filled with a pattern and/or color.			
Circle	Define if the ellipse should always be displayed as a circle centered in the.		True	Yes
			False	No
			Formula	Formula dialog
Border	Define if the ellipse should have a border.		0	transparent
			1	pattern / block color
			Formula	Formula dialog
	Color	Color of the frame.		
	Width	Width of the frame in the measuring system used in workspace	Number	
			Formula	Formula dialog

4.11 Inserting Picture Objects

To create a drawing or picture object, select **Objects** > **Insert** > **Picture** (CTRL+D). All available formats are shown in the files dialog. If you evaluate a file name or a variable as data source, you can do these also by doubleclick on the object to select.

4.11.1 Properties

See also Chapter "Common Object Properties".

Property	Description		Value	Description
Data source	Select how the data source for the graphic to be displayed is defined.		File name	
			Formula	Formula dialog
			Variable	
	File name	Existing file name, will be evaluated if file name was selected in the property "Source”. Select the desired graphic in the Windows familiar dialog. In this dialog, by using the option "Imbed in Project", you have the capability to imbed the graphic file into your project. The file is then copied into your project and is available even though the file may be absent.	📂	File > Open dialog

		In this case, embedded will be displayed instead of the file name.			
		Relative Path	The path is relative to the project path, e.g. "..\pics\..."	True False	Yes No
	Formula	The file name is the result of a formula, if formula was selected in the property "Source”. The result of this formula must be a value of the type "Drawing”. You can also enter a valid file name, but it must first be converted to a value of the type "Drawing” with the function Drawing().		√α	Formula dialog
	Variable	The file name is the result of a variable, if variable was selected in the property "Source”. Select the desired variable from the list box. In this box you will find all variables of the type "Drawing” that were defined in your application.		List of all variables of the type Drawing	
	Properties	Dependent upon your application, a further dialog can be opened in which more properties can be defined.		📋	Opens dialog
Original Size	Sets wether the drawing is rendered in its original size or fit to the object.			True False not defined	Yes No not defined
	Keep proportions	Using this option, you define whether the graphic should be inserted so that the relationship between height and width is kept (True) or if the graphic should be adapted to fit the frame of the object (False).		True False Formula	Yes No Formula dialog
	Alignment	Sets the alignment of the drawing in the object´s frame.		0 1 2 3	Centered Tiled Left Top Left Bottom

		4	Right Top
		5	Right Bottom
		6	Left
		7	Right
		8	Top
		9	Bottom

4.12 Inserting Barcode Objects

Barcodes can be used for product labels, price stickers, serial numbers and many other purposes. A barcode normally consists of a series of differently sized bars and spaces. The minimum bar size should be 0.3 mm, in List & Label the bar width relationship is set at 1:3.

You can find a description of the barcode formats in Chapter "List of Available Barcodes". To create a barcode object select Objects > Insert > Barcode (CTRL+B).

4.12.1 Properties

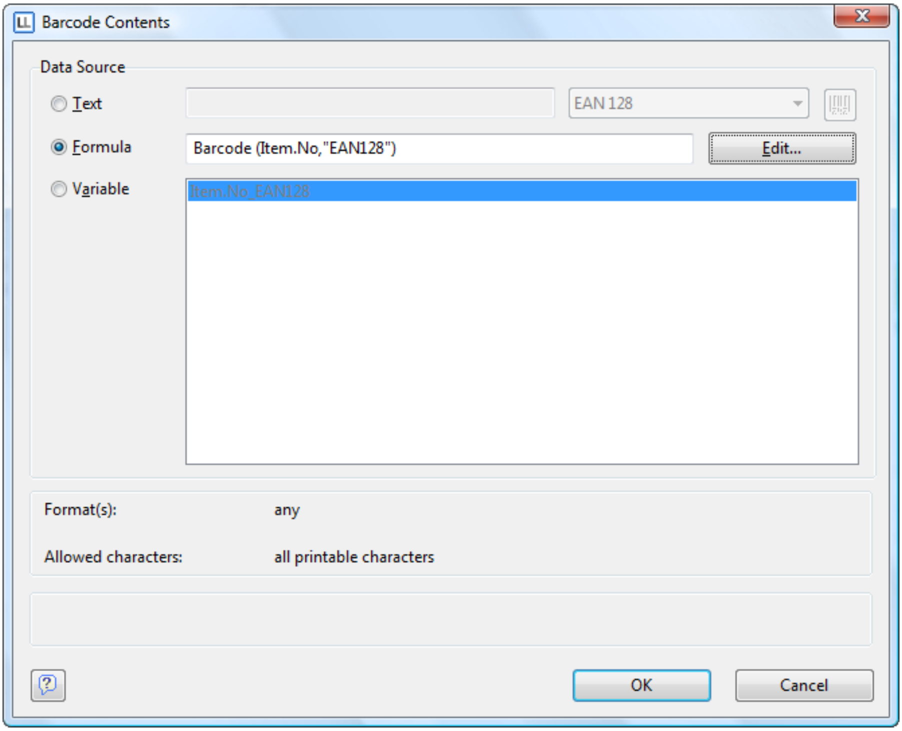
See also Chapter "Common Object Properties".

Property	Description	Value	Description
Bar Color	Color for the selected Barcode.		
Bar Width	The narrowest bar width in SCM units (1/1000 mm). 0=automatic adjustment. Not supported by all barcode types.	Formula	Formula dialog
	Orientation Orientation of the barcode within the object frame.	0	left
		1	centered
		2	right
		Formula	Formula dialog
Bar Width Ratio	The ratio between the bars or spaces. Not supported by all barcode types.	Formula	Formula dialog
Rotation	Rotates the object counter clockwise	0	0°
		1	90°
		2	180°
		3	270°
		Formula	Formula dialog
Optimum	Sets the optimum size for the Barcode. This property could affect size changes with the	True	Yes

Size	following formats: EAN 13 (all), DP-Leitcode, DP-Identcode, German Parcel, Postnet and FIM.	False	No
		Formula	Formula dialog
Font	Font for the Barcode text. Will only be evaluated if the "Show Text" property is activated. If the default is set the default font will be used. The font definitions dialog can be opened by using the button. You can set all properties in this property group in a single dialog.		Font dialog
Show Text	Define whether the contents of the barcode should also be displayed as clear text.	True	Yes
		False	No
		Formula	Formula dialog

4.12.2 Barcode Contents

You can further define the barcode object using the contents dialog.



Barcode Contents

Data Source

○ Text

● Formula: Barcode (Item.No,"EAN128")

○ Variable: Item.No_EAN128

Format(s): any

Allowed characters: all printable characters

- If you wish to print fixed text in barcode form, select the option "Text". Enter the value to be printed in the first field. Select your desired type of barcode in the second field. Please note that certain requirements (number of characters, or spaces) must be met for the selected type of barcode. If the requirements are not met you will receive an error message.

- For some barcode types, for example Maxicode, additional options exist that can be edited in a further dialog. This dialog is opened with the button "Options".
- If you wish to use a formula as a barcode, select the option "Formula" and define a valid formula expression using the "Edit" button. The return value of the formula must be of the "Barcode" type. You can also enter variables, but they must first be converted to variables of the barcode type using the function Barcode().
- If you wish to print a variable in barcode form, select the option "Variable". All variables of the type Barcode will be displayed in a list box.

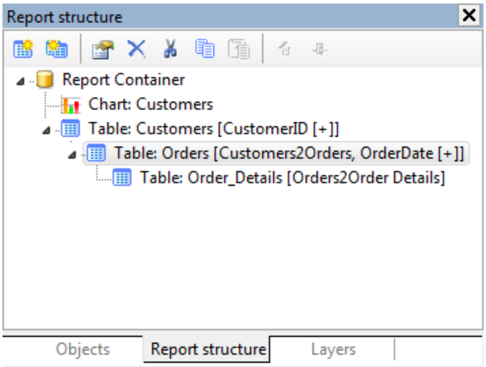
4.13 Inserting Report Container

To create a Report container object select Objects > Insert > Report Container (not in any application available).

The report container can contain several objects such as tables, crosstabs and charts. All elements and respective sub-elements of a report structure are represented in the Report Structure tool window.

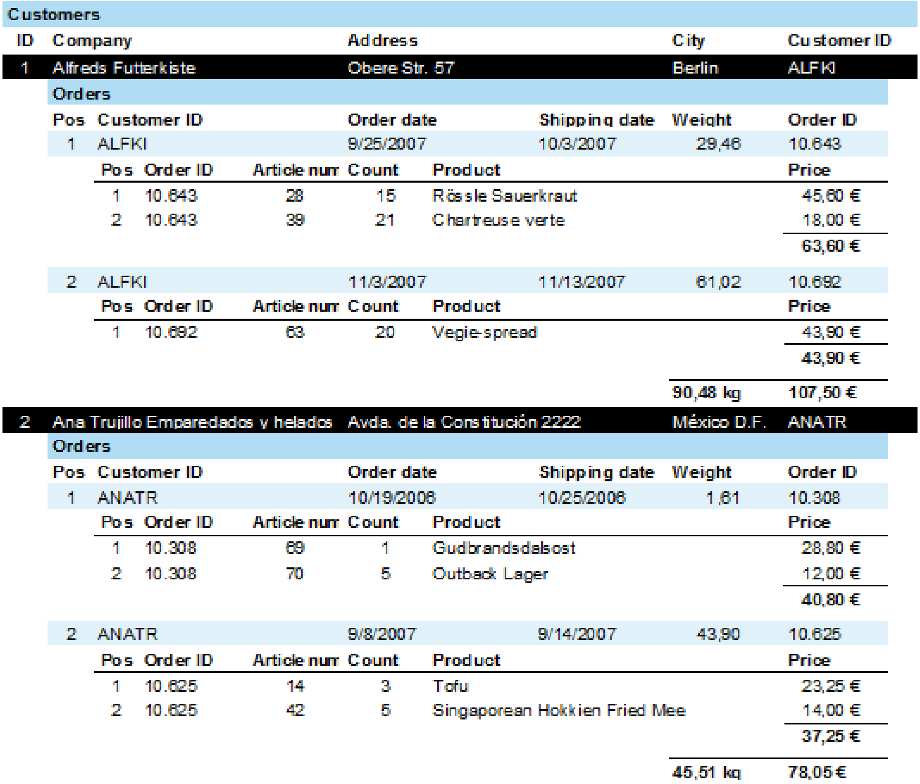
Tables, charts and crosstabs can be inserted in any order, also as sub-elements of tables. This means that sub-reports can be defined with almost any relations between tables. However, there cannot be any separate tables, crosstabs or charts in addition to a report container, and a second report container cannot be inserted.

Further elements and sub-elements and the desired hierarchical structure can be defined in the tool window "Report structure". The table description is composed of the table name [relation name, sort name].



- Report Container
  - Chart: Customers
  - Table: Customers [CustomerID [+]]
    - Table: Orders [Customers2Orders, OrderDate [+]]
      - Table: Order_Details [Orders2Order Details]

Example for relationally linked tables: You wish to create a list of all customers in which the orders of the customers and all ordered items are visible (relationally linked tables Customers > Orders > OrderDetails).



Customers

ID	Company	Address	City	Customer ID
1	Alfreds Futterkiste	Obere Str. 57	Berlin	ALFKI

Orders

Pos	Customer ID	Order date	Shipping date	Weight	Order ID
1	ALFKI	9/25/2007	10/3/2007	29,46	10.643

Pos	Order ID	Article num	Count	Product	Price
1	10.643	28	15	Rössle Sauerkraut	45,60 €
2	10.643	39	21	Chartreuse verte	18,00 €
					63,60 €

Pos	Customer ID	Order date	Shipping date	Weight	Order ID
2	ALFKI	11/3/2007	11/13/2007	61,02	10.692

Pos	Order ID	Article num	Count	Product	Price
1	10.692	63	20	Vegie-spread	43,90 €
					43,90 €

90,48 kg 107,50 €

ID	Company	Address	City	Customer ID
2	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.	ANATR

Orders

Pos	Customer ID	Order date	Shipping date	Weight	Order ID
1	ANATR	10/19/2006	10/25/2006	1,61	10.308

Pos	Order ID	Article num	Count	Product	Price
1	10.308	69	1	Gudbrandsdalsost	28,80 €
2	10.308	70	5	Outback Lager	12,00 €
					40,80 €

Pos	Customer ID	Order date	Shipping date	Weight	Order ID
2	ANATR	9/8/2007	9/14/2007	43,90	10.625

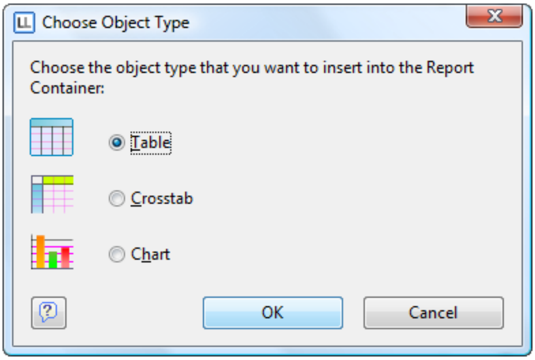
Pos	Order ID	Article num	Count	Product	Price
1	10.625	14	3	Tofu	23,25 €
2	10.625	42	5	Singaporean Hokkien Fried Mee	14,00 €
					37,25 €

45,51 kg 78,05 €

4.13.1 Inserting a new element

Further elements and sub-elements and the desired hierarchical structure can be defined in the tool window "Report structure". The most important commands for the various tables are available through a toolbar. Using the buttons in the tool window, you can attach a new element, attach a new sub element, delete, cut, copy, paste elements and change the sequence of the elements of the same hierarchical level.

- Select the button "Add an element" or "Add a sub-element".
- In the following dialog, choose the object type that you want to insert into the report container: Table, Crosstab or Chart.



Choose Object Type

Choose the object type that you want to insert into the Report Container:

● Table

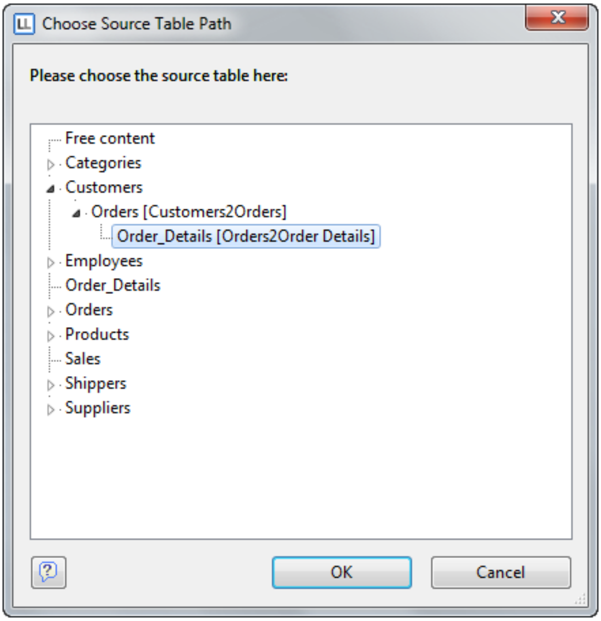
○ Crosstab

○ Chart

See also the chapters "Inserting Table Objects", "Inserting Chart Objects" and "Inserting Crosstab Objects".

- In the following dialog, choose the Source Table Path. In the "Choose Source Table Path" dialog all available tables are displayed hierarchically, i.e. you will find the respective relationally linked tables under the tables.

  "Free content" can also be selected as the data source, provided this is supported by the application. This means that data lines with fixed content or variables can be created and thus e.g. texts, letters, images or barcodes can be integrated into the "Report Script".



Choose Source Table Path

Please choose the source table here:

- Free content
- Categories
- Customers
  - Orders [Customers2Orders]
    - Order_Details [Orders2Order Details]
- Employees
- Order_Details
- Orders
- Products
- Sales
- Shippers
- Suppliers

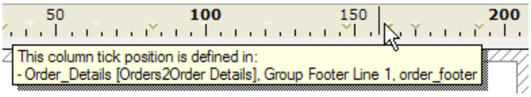
4.13.2 Properties of the Report container

See also Chapter "Common Object Properties".

Property	Description	Value	Description
Pagebreak before	Selects if a page break is required before the report container is printed.	True False Formula	Page break No page break Formula dialog
Column Count	Sets the default number of columns for all sub items.	1 2 3 4 5 Formula	1 Columns 2-Columns 3-Columns 4-Columns 5-Columns Formula dialog
	Distance Column distance.	Number Formula	 Formula dialog

4.13.3 Characteristics of Elements

See also Chapter "Common Object Properties".

Property	Description	Value	Description
Name	Name of the element (stored in the variable "LL.CurrentContainerItem").	Name	
Separators	To easier align (sub)tables and columns accurately, you can show separator tick marks to the ruler.	True False	Yes No
	 This column tick position is defined in: - Order_Details [Orders2Order Details], Group Footer Line 1, order_footer		
Distance before	Vertical distance from the previous element (does not apply when the element is at the beginning of a page).	Number Formula	 Formula dialog
Sorting (only table and chart, depends of application)	The order in which the table is presented. All available sorting choices are displayed in a list.	List of available sorting options.	
Pagebreak before	When the value "true" is returned, a page break will be initiated before the object is printed.	True False Formula	Wrap None Formula dialog

Pagebreak Condition (only table)	General page break condition. If "True" while printing a data line, a page break is triggered.		True	Wrap
			False	None
			Formula	Formula dialog
Column Count	Sets the number of columns.		0	Default
	If there are several multi-column objects, a page break is automatically triggered after an object if the number of columns for the objects is different (e.g. 2-column table followed by a 3-column table) and no more space is available for the following object.		1	1 Columns
			2	2-Columns
			3	3-Columns
			4	4-Columns
			5	5-Columns
			Formula	Formula dialog
	Distance	Column Distance.	Number	
			Formula	Formula dialog
	Column Break Condition (only table)	General column break condition. If "True" while printing a data line, a column break is triggered.	True	Wrap
			False	None
			Formula	Formula dialog
	Column Break Before	When the value "true" is returned, a column break will be initiated before the object is printed.	True	Wrap
			False	None
			Formula	Formula dialog

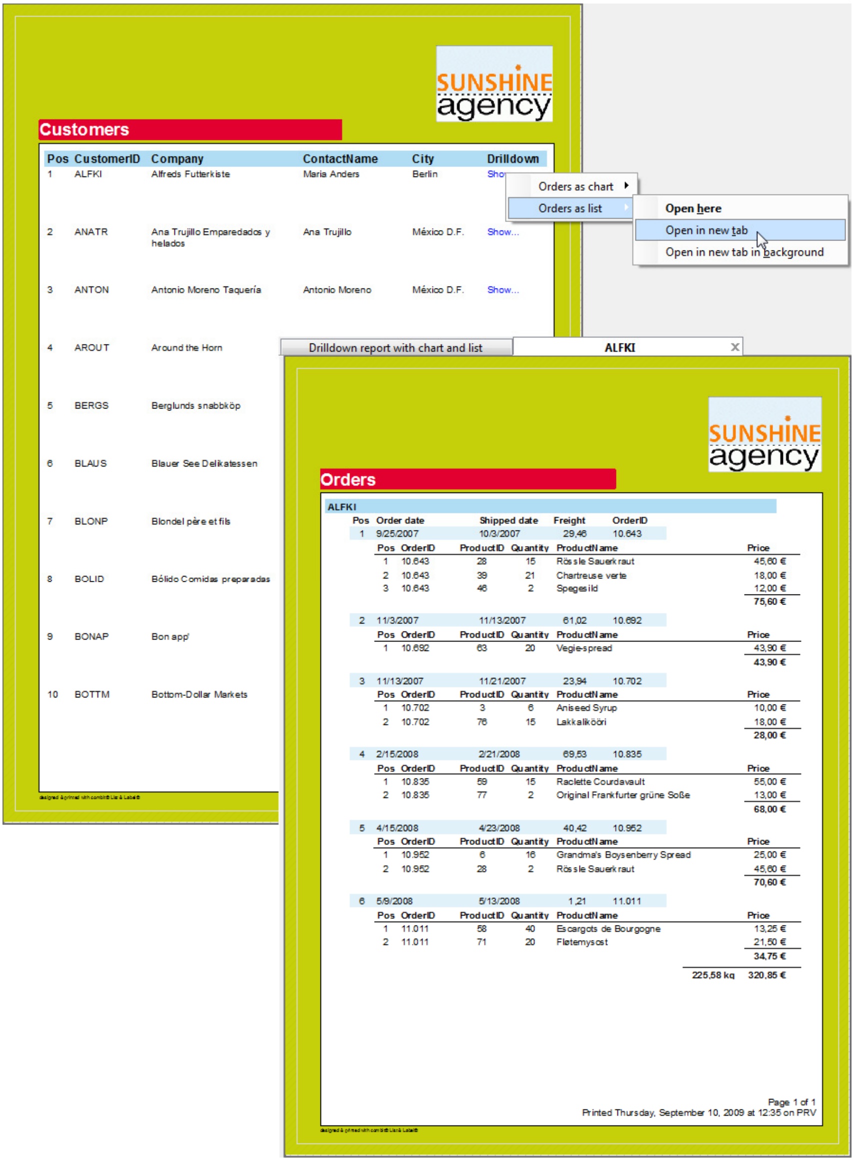
4.13.4 Drilldown-Reports

Drill-down means navigation in hierarchical data. Drill-down allows a "zooming" in which the existing data in different levels of detail can be seen. Even for very large and complex data sets so you can quickly prepare all the required information.

First, only a top level is printed (e.g. Customers). A click on a customer then opens a new detail report (e.g. Orders).

The context menu can be used to choose if the drilldown report should open in the same window (in this case, you can navigate back using the green arrow buttons in the preview's toolbar), in a new tab or in a new tab in the background.

Only the preview supports drilldown functionality. If you need to export a certain drilldown report to e.g. PDF, you can do so from within the preview window. Drilldown reports can be embedded into the preview file enabling you e.g. to send the full report via email. You find the respective option "Embed Drilldown Reports" in the project's settings.



Customers

Pos	CustomerID	Company	ContactName	City	Drilldown
1	ALFKI	Alfreds Futterkiste	Maria Anders	Berlin	Sho
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	México D.F.	Show...
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	México D.F.	Show...
4	AROUT	Around the Horn			
5	BERGS	Berglunds snabbköp			
6	BLAUS	Blauer See Delikatessen			
7	BLONP	Blondel père et fils			
8	BOLID	Bólido Comidas preparadas			
9	BONAP	Bon app'			
10	BOTTM	Bottom-Dollar Markets			

Orders

ALFKI

Pos	Order date	Shipped date	Freight	OrderID
1	9/25/2007	10/3/2007	29,46	10.643

Pos	OrderID	ProductID	Quantity	ProductName	Price
1	10.643	28	15	Rössle Sauerkraut	45,60 €
2	10.643	39	21	Chartreuse verte	18,00 €
3	10.643	46	2	Spegesild	12,00 €
					75,60 €

Pos	Order date	Shipped date	Freight	OrderID
2	11/3/2007	11/13/2007	61,02	10.692

Pos	OrderID	ProductID	Quantity	ProductName	Price
1	10.692	63	20	Vegie-spread	43,90 €
					43,90 €

Pos	Order date	Shipped date	Freight	OrderID
3	11/13/2007	11/21/2007	23,94	10.702

Pos	OrderID	ProductID	Quantity	ProductName	Price
1	10.702	3	6	Aniseed Syrup	10,00 €
2	10.702	76	15	Lakkalikööri	18,00 €
					28,00 €

Pos	Order date	Shipped date	Freight	OrderID
4	2/15/2008	2/21/2008	69,53	10.835

Pos	OrderID	ProductID	Quantity	ProductName	Price
1	10.835	59	15	Raclette Courdavault	55,00 €
2	10.835	77	2	Original Frankfurter grüne Soße	13,00 €
					68,00 €

Pos	Order date	Shipped date	Freight	OrderID
5	4/15/2008	4/23/2008	40,42	10.952

Pos	OrderID	ProductID	Quantity	ProductName	Price
1	10.952	6	16	Grandma's Boysenberry Spread	25,00 €
2	10.952	28	2	Rössle Sauerkraut	45,60 €
					70,60 €

Pos	Order date	Shipped date	Freight	OrderID
6	5/9/2008	5/13/2008	1,21	11.011

Pos	OrderID	ProductID	Quantity	ProductName	Price
1	11.011	58	40	Escargots de Bourgogne	13,25 €
2	11.011	71	20	Fløtemysost	21,50 €
					34,75 €

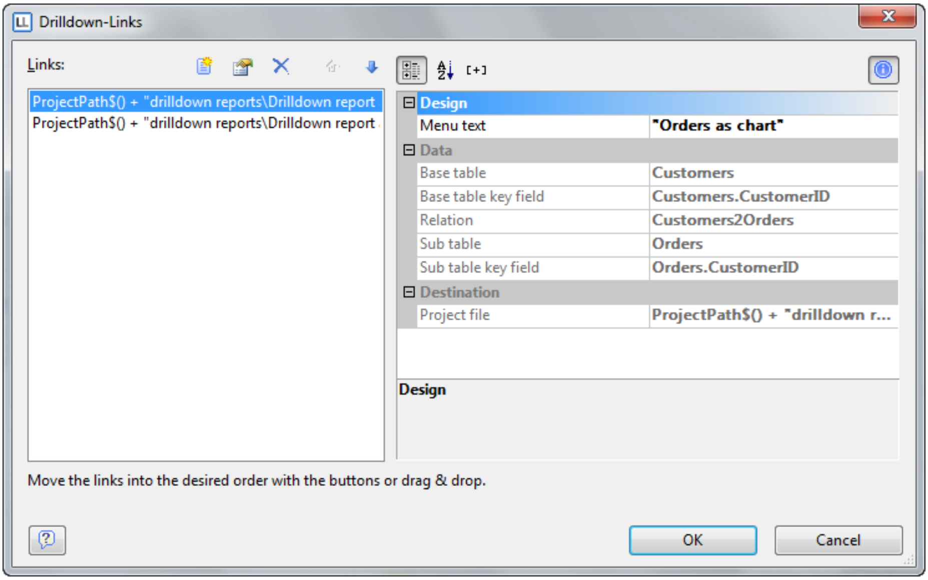
225,58 kg **320,85 €**

Page 1 of 1
Printed Thursday, September 10, 2009 at 12:35 on PRV

Create and edit Drilldown-Links

A drilldown link refers to a single field or a whole table line. Each of these elements can contain any number of drilldown links to enable different views on the related data.

- The "Drilldown Links" property of a field or table line can be used to open an edit dialog for Drilldown-Links.



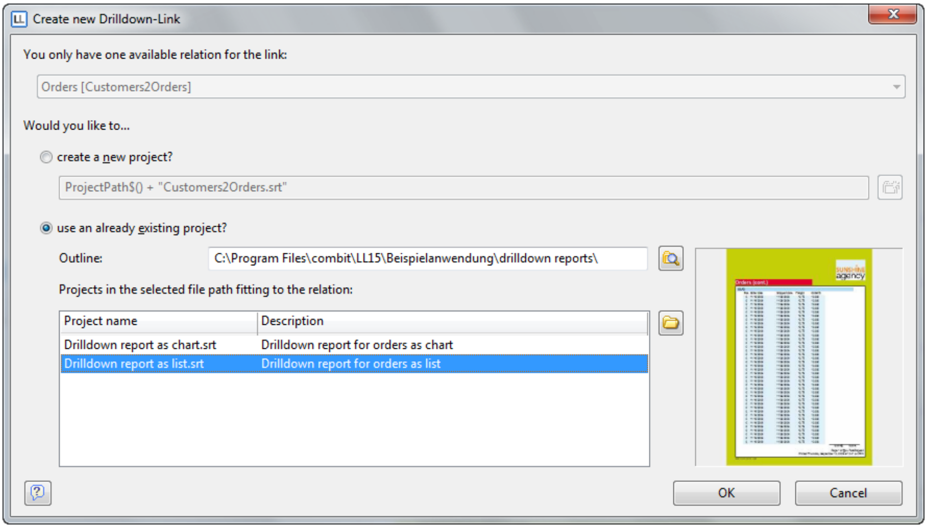
ProjectPath$() + "drilldown reports\Drilldown report
ProjectPath$() + "drilldown reports\Drilldown report

Design	
Menu text	"Orders as chart"
Data	
Base table	Customers
Base table key field	Customers.CustomerID
Relation	Customers2Orders
Sub table	Orders
Sub table key field	Orders.CustomerID
Destination	
Project file	ProjectPath$() + "drilldown r...

Design

Move the links into the desired order with the buttons or drag & drop.

- With the "New" button, you can create a new drill-down link.



You only have one available relation for the link:

Orders [Customers2Orders]

Would you like to...

○ create a new project?

ProjectPath$() + "Customers2Orders.srt"

● use an already existing project?

Outline: C:\Program Files\combit\LL15\Beispielanwendung\drilldown reports\

Projects in the selected file path fitting to the relation:

Project name	Description
Drilldown report as chart.srt	Drilldown report for orders as chart
Drilldown report as list.srt	Drilldown report for orders as list

- Choose the required relation from the drop down list in this dialog.

- Either choose an existing project file or create a new one. When creating a new project, a second instance of the designer opens. Create the report as usual.

4.14 Inserting Table Objects

A table object can be inserted in the tool window "Report structure" or select Objects > Insert > Table (depends of application).

4.14.1 Properties

See also Chapter "Common Object Properties".

Property	Description			Value	Description
Data lines					
	Force Sums	Sums will be calculated even with the suppression of the data lines.		True False Formula	Yes No Formula dialog
	Keep Together	If this property is activated, data lines will be kept together (if this is possible) after a page break. The lines will be printed on the next page.		True False Formula	Yes No Formula dialog
	Suppress	If the option "Data Lines Suppress" is selected, all data lines will be completely suppressed. This option is especially useful in combination with the option "Force Sums". The last option defines that the sum will be calculated when the data line is not printed. With the combination of both options and the use of groups and sum variables, the printing of interesting statistics is possible.		True False Formula	Yes No Formula dialog
	Zebra Pattern	With the option "Zebra pattern" in the "Data Lines" group, you can assign alternating colors to the data lines. This will improve the clarity of large tables.		True False Formula	Yes No Formula dialog
		Color	Color of the selected pattern.		
		Pattern	Pattern, with which every other data line is assigned. Select one of	Number	Predefined pattern

		the predefined patterns. Each number represents a different pattern.	Formula	Formula dialog
Footer lines	When using multi-tables, this setting is only available for sub-tables and not for the Report Container.			
	Keep Together	With the property Footer Lines "Keep Together", you can define, simultaneously with the option Lines "Keep Together", that footer lines will be kept together during a page break, as far as is possible.	True False Formula	Yes No Formula dialog
Group Footer Lines	Options for the Group Footer Lines. When using multi-tables, this setting is only available for sub-tables and not for the base object.			
	Also Empty Groups	Group Footer Lines will also be printed for empty groups	True False Formula	Yes No Formula dialog
Group Header Lines	Options for the Group Header Lines. When using multi-tables, this setting is only available for sub-tables and not for the base object.			
	Keep Together	Using this option, you can define that group lines (intermediate headers) will not be separated from their data lines by a page break, if possible.	True False Formula	Yes No Formula dialog
Fixed Size (not available for sub-tables)	If the property "Fixed Size" is selected, the table will not automatically adjust its size if less space is used than available. If the property is not selected, the bottom edge of the table object will automatically move up and will be printed directly underneath the last data line. If "Fixed Size" is false, the footer line will be located at the very bottom of the table objects rectangle. Please note: objects which may be linked to the table can only change their position automatically when "Fixed Size" is not selected.		True False Formula	Yes No Formula dialog
	Separators Fixed	If this property is selected, the column separators will be continued from the last data line to the footer line. If this option is not selected, the column separators will reach only to the	True False Formula	Yes No Formula dialog

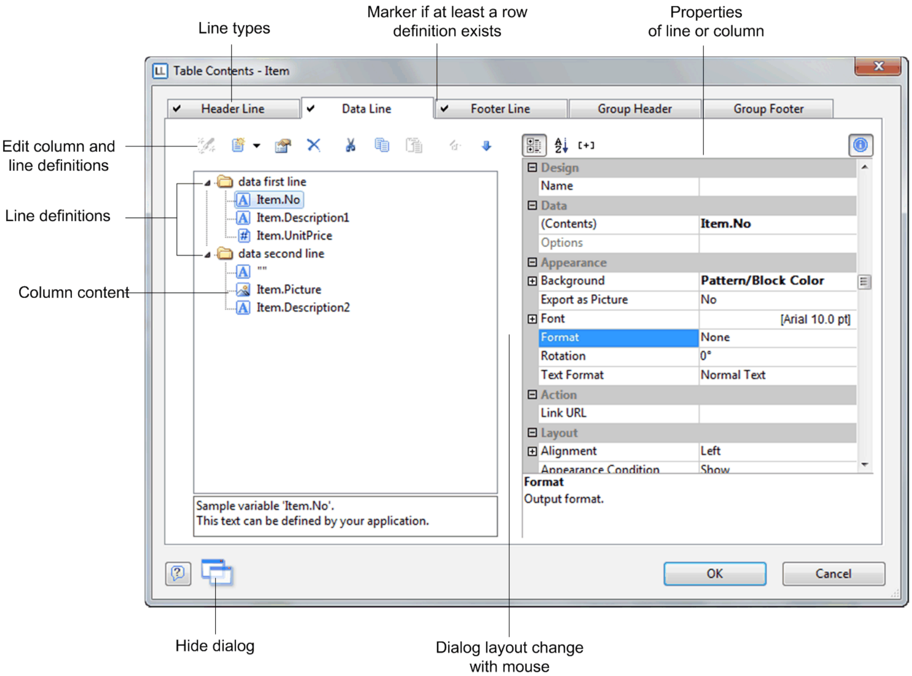
		last data line. This option is only available for tables of "Fixed Size”.		
Default Font		With this property you can assign the default font for the table. The Font dialog can be opened using the appropriate button. (not available for sub-tables)		Font dialog
Default Frame		With this property you can assign the default frame for the table.		Font dialog
Pagewrap Before		If the condition returns "True", a page wrap will be initiated before printing the object.	True False Formula	Break No break Formula dialog
Page wrap Condition		Additionally, you can enter a condition which will cause a page wrap to be performed as soon as the condition is met. If Page wrap = True, a page wrap will be performed after each line. If Page wrap = False, the page wrap will only be performed when necessary.	True False Formula	Yes No Formula dialog
Column Count		Sets the number of columns. If there are several multi-column objects, a page break is automatically triggered after an object if the number of columns for the objects is different (e.g. 2-column table followed by a 3-column table) and no more space is available for the following object.	0 1 2 3 4 5 Formula	Default of Container 1 Columns 2 Columns 3 Columns 4 Columns 5 Columns Formula dialog
	Distance	Column Distance.	Number Formula	 Formula dialog
	Column Break Condition	General column break condition. If true while printing a data line, a column break is triggered.	True False Formula	Yes No Formula dialog
	Column Break Before	When the value "true" is returned, a column break will be initiated before the table is printed.	True False Formula	Wrap None Formula dialog

4.14.2 Table Contents

A table consists of various types of lines that can be defined independently of each other. There are header lines, data lines, footer lines, group header lines and group footer lines.

- Headers lines are most commonly used as a description for table columns.
- Data lines contain the formatting information for the data that is to be presented in the table.
- Footer lines are located at the end of the page/table and can contain further information about the data displayed in the table.
- Group header and footer lines can be used to further structure the data lines through the use of "intermediate headers" and "intermediate footers".

All line types can be defined independently of one another. The columns for a header line can have a different appearance than the data lines or footer lines. The header lines can have a completely different appearance than the following data or footer lines.



Line types

Marker if at least a row definition exists

Properties of line or column

Edit column and line definitions

Line definitions

Column content

Hide dialog

Dialog layout change with mouse

Sample variable 'Item.No'.
This text can be defined by your application.

Design	
Name	
Data	
(Contents)	**Item.No**
Options	
Appearance	
Background	**Pattern/Block Color**
Export as Picture	No
Font	[Arial 10.0 pt]
Format	None
Rotation	0°
Text Format	Normal Text
Action	
Link URL	
Layout	
Alignment	Left
Appearance Condition	Show

Format
Output format.

Various line layouts or definitions can be defined for the individual types of lines. Special appearance conditions can be activated, if needed, for the various line definitions.

All line definitions are defined in the identical manner and consist of columns that can all be individually edited and formatted.

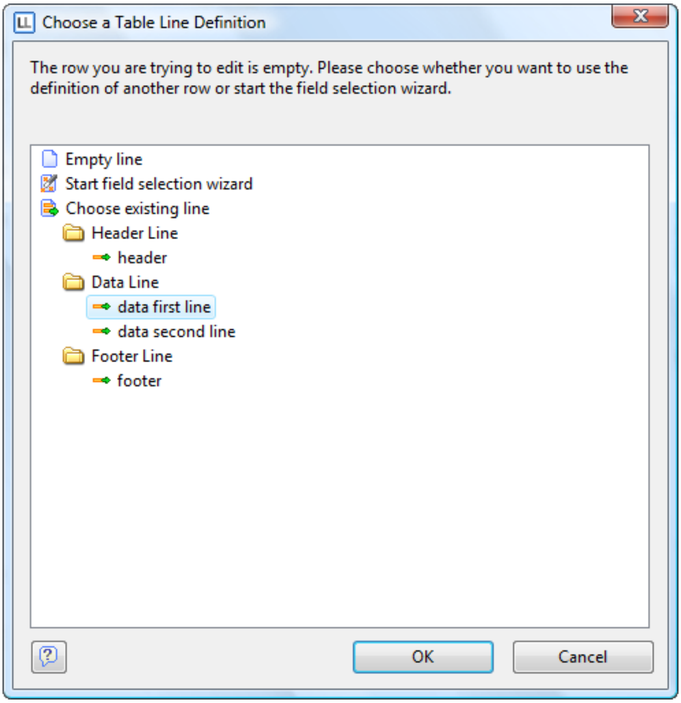
The individual lines types with their definitions and columns can be edited in the dialog "Table Contents".

A card is provided for the definition of every line type and the columns.

4.14.3 Definition of Table Lines

The procedure for the definition of the various table lines is always the same. Appropriately, the cards for the header, data, footer, group header and group footer lines have nearly the same appearance.

First, select the type of line to be edited by clicking on the appropriate card. If nothing has been defined for this type of line, you will be asked if you would like to use an existing line definition for the new line type.



Choose a Table Line Definition

The row you are trying to edit is empty. Please choose whether you want to use the definition of another row or start the field selection wizard.

- Empty line
- Start field selection wizard
- Choose existing line
  - Header Line
    - header
  - Data Line
    - data first line
    - data second line
  - Footer Line
    - footer

If line definitions have been created, you can select the definitions to be used in the new line type.

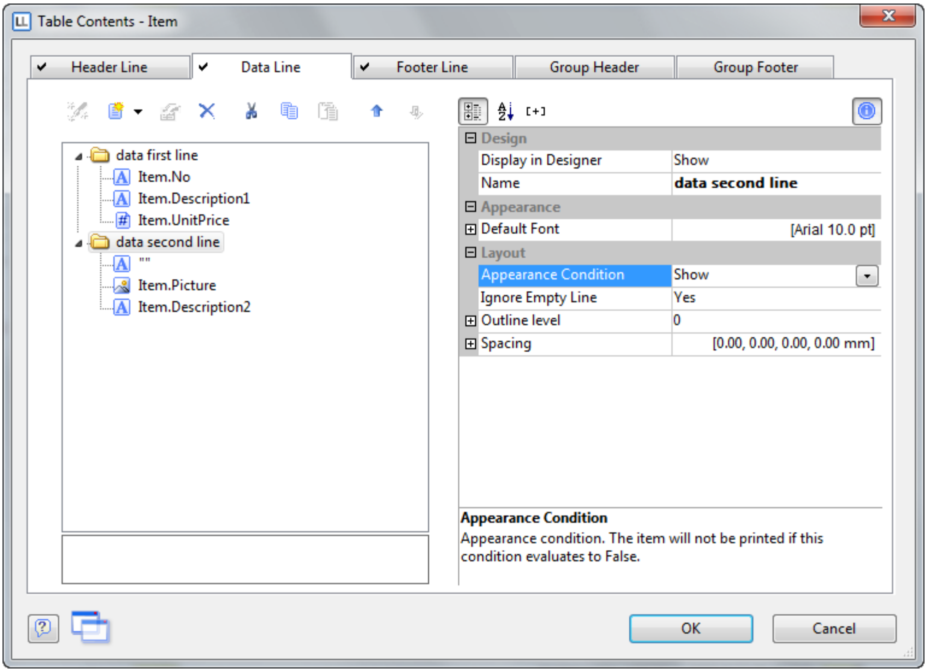
In the example pictured above, the definition for line definition "data first line" was selected for use in the data line.

If you are creating a new table and no lines have been defined, this dialog will not appear.

You also have the possibility to copy one or more columns from one table to another or from one area of a table to another, using the clipboard.

Defining multiple line definitions

Various layouts can be defined for each line type. In connection with the appearance conditions, and dependent upon the situation, the appropriate layout will be used. In this way, table lines for some records will appear in bold print while others are displayed normally. Or, under certain conditions, the table lines contain other columns.



- data first line
  - Item.No
  - Item.Description1
  - Item.UnitPrice
- data second line
  - ""
  - Item.Picture
  - Item.Description2

⊟ Design	
Display in Designer	Show
Name	**data second line**
⊟ Appearance	
⊞ Default Font	[Arial 10.0 pt]
⊟ Layout	
Appearance Condition	Show
Ignore Empty Line	Yes
⊞ Outline level	0
⊞ Spacing	[0.00, 0.00, 0.00, 0.00 mm]

Appearance Condition
Appearance condition. The item will not be printed if this condition evaluates to False.

- To create multiple line layouts for a line, select a line definition and click on the button "Append the line definition".
- If you have defined a layout for at least one line, you have the possibility of using the existing layout as a template for the new line layout.
- As described in Chapter "Define Column Contents", you can now define the columns, their contents and properties for the new layout. Using the button "Layout" you can define the complete appearance for the new line layout. You can define, for example, special frames.
- You can assign names to the line definitions. This will make it easier for you to find the line in complex layouts.

- The order of the line definitions in the list can be changed using the arrow buttons or per drag & drop.
- With the "Display in Designer" property you can hide single lines in the workspace. This is very useful when you have a lot of line definitions.

Format Lines

Select a line to define the appearance of the current table line.

In the category "Spacing", the margins that the table line should have within the table object can be set.

The margins "Top" and "Bottom" define the space between the individual lines of the table. A bottom margin of 3.0mm sets a space of 3mm between a line of the selected type and the following lines. When a top margin for the lines is additionally defined, the space between the lines will be resized appropriately. Each data line will be printed with the frame line settings in category "Frame".

With the margins "Left" and "Right", margins relative to the table object or to other table lines can be set. If you have, for example, defined a left margin of 10.0 for the header lines and a margin of 20.0 for the data lines, the data lines will be indented 10mm in relation to the header line.

Appearance Conditions for Table Lines

Appearance conditions for the table lines can be assigned using the property "Appearance Cond." Using the formula button opens the familiar dialog for the definition of logical expressions (see Chapter "Variables, Formulas and Expressions").

These appearance conditions are valid in addition to the project specific appearance conditions assigned defined with Project > Filter.

Appearance conditions for table lines are especially meaningful when you define more than one layout for a table line. You can use appearance conditions to change between the various layouts.

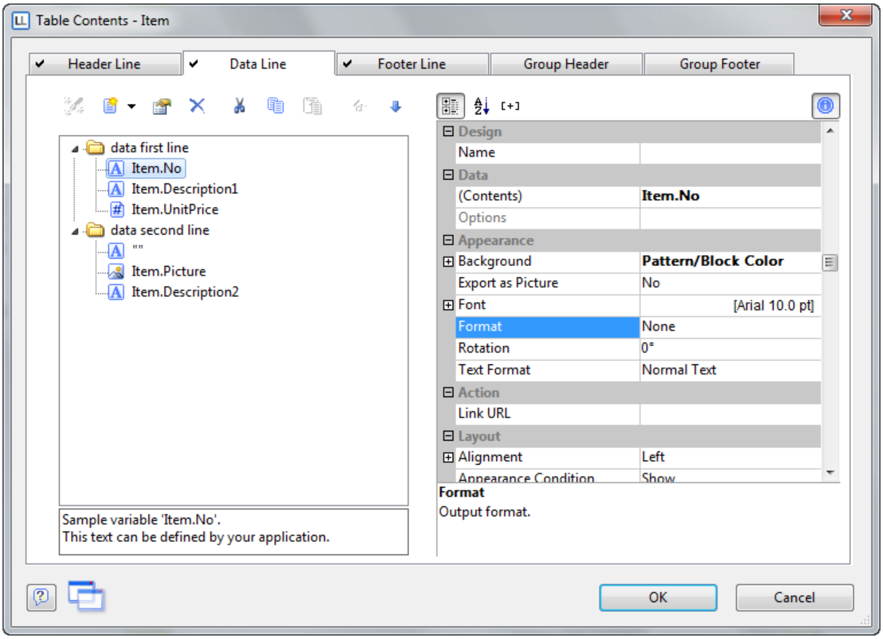
If each table line is to be displayed with a single line layout, ensure that the various appearance conditions assigned to the individual lines cancel each other out. Otherwise the same record will appear repeatedly in the table, which means once for every line layout that fits the appearance conditions. Sometimes this may be desired, especially when the fields of a table are to be displayed over multiple lines.

4.14.4 Define Column Contents

Table lines are column-oriented, which means that every line definition can contain many columns and every column can be separately edited and formatted.

- Every column also has a definite type. The column types that are available are: Text, Drawing, Barcode, RTF Text, Chart, HTML Text and OLE Container.
- Every column of a table object is represented by a node in the list. Multiple expressions or variables can appear within a column.

- In order to insert a new column, click on the ▾ button next to the "Append column" button and select the relevant column type.
- You can edit, delete, cut, copy or insert the selected columns using the buttons. Columns can also be moved across line definitions by Drag & Drop.



data first line
- Item.No
- Item.Description1
- Item.UnitPrice

data second line
- ""
- Item.Picture
- Item.Description2

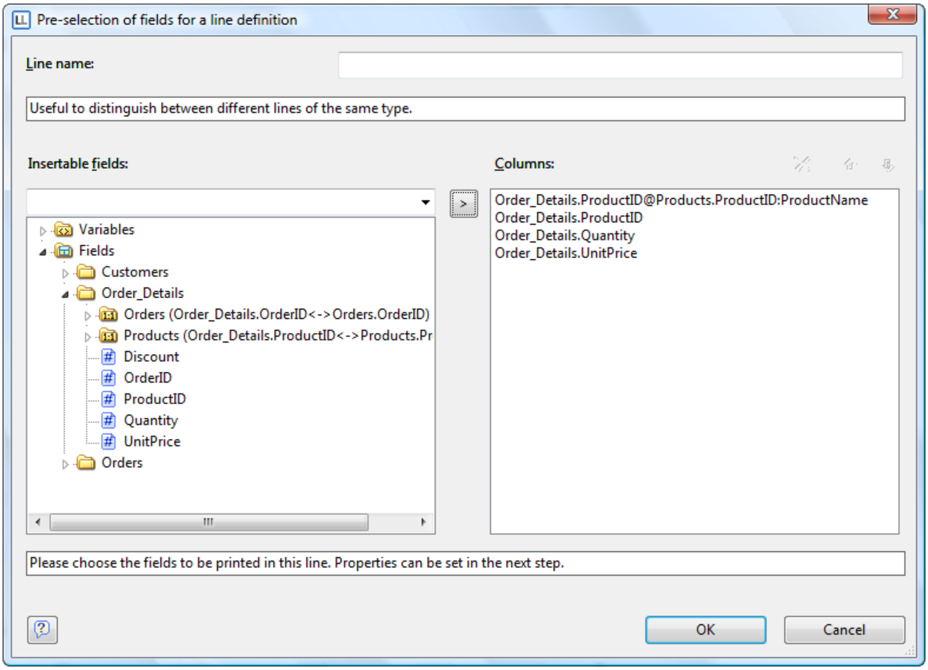
Sample variable 'Item.No'.
This text can be defined by your application.

Design	
Name	
Data	
(Contents)	**Item.No**
Options	
Appearance	
Background	**Pattern/Block Color**
Export as Picture	No
Font	[Arial 10.0 pt]
Format	None
Rotation	0°
Text Format	Normal Text
Action	
Link URL	
Layout	
Alignment	Left
Appearance Condition	Show

Format
Output format.

- The order of the column definitions can be changed with the arrow buttons or by Drag & Drop.
- You can give the column definitions new names, such as e.g. "Article No.", "Customer Number" etc. This will enable you to quickly find this column in complex layouts.
- You can also insert new columns in this dialog window by Drag & Drop from the variable list, provided it is not linked. The column type is then adopted from the variable type.
- Editing of the column is done in the familiar Formula Editor, in which expressions can be defined as column contents, see Chapter "Variables, Formulas and Expressions" for more information.
- The field selection wizard makes it easier to add new columns. When defining the table rows you will be assisted by the field selection wizard, in which only the fields of the selected table and its parent tables are available. It is

possible to apply changes to multiple columns in one step. The representation of the fields in the variables window is automatically hierarchical, i.e. all fields of a element can be found in its own subfolder. Provided you have already defined a layout for at least one line, a further line definition is created with the field selection wizard.


Pre-selection of fields for a line definition
Line name:
Insertable fields:
Variables
Fields
Customers
Order_Details
Orders (Order_Details.OrderID<->Orders.OrderID)
Products (Order_Details.ProductID<->Products.Pr
Discount
OrderID
ProductID
Quantity
UnitPrice
Orders
>
Columns:
OK
Cancel

Useful to distinguish between different lines of the same type.

Order_Details.ProductID@Products.ProductID:ProductName
Order_Details.ProductID
Order_Details.Quantity
Order_Details.UnitPrice

Please choose the fields to be printed in this line. Properties can be set in the next step.

4.14.5 Column Properties

The properties of a column are defined using a Property List, which is similar to the tool window "Properties". The Property List may be different from column to column, depending on the type of column.

If more than one column is selected, the common properties can be edited all at once. Properties that are not common to all marked columns are not available for editing.

The column properties correspond to, with some table related restrictions, the properties of the appropriate object type.

Columns of the type "Text" and "RTF-Text" possess a special characteristic. These text variants can be, also using a column property, intertwined. Dependent upon this property, the property list is changed appropriately.

In addition to the object properties, several properties are available for columns:

Property	Description	Value	Description
Link URL	Link target (only effective for preview, PDF and HTML export.	Link	
		Formula	Formula dialog
Vertical Alignment (Text, Barcode)	With this option you define the vertical alignment of the selected column.	0	Top
		1	Centered
		2	Bottom
		Formula	Formula dialog
Alignment (Picture)	Sets the alignment of the drawing.	0	Centered
		1	Tiled
		2	Left Top
		3	Left Bottom
		4	Right Top
		5	Right Bottom
		6	Left
		7	Right
		8	Top
		9	Bottom
Alignment (Text)	With this option, you define the text alignment. "Decimal" signifies that number values will be aligned at the decimal point.	0	Left
		1	Centered
		2	Right
		3	Decimal
		Formula	Formula dialog
	Decimal-position The position of the decimal point with the field, measured from the left corner of the preceding frame.	Number	
		Formula	Formula dialog
Width	Width of the selected column. If the sum of the column widths exceeds the width of the table, you will receive an error notice. The width of a field or column can also be changed in the workspace using the mouse by selecting the table object and moving the right border. The property list is recommended for precise resizing.	Number	
		Formula	Formula dialog
Bar color (only with barcodes)	With this option, you define the color of barcodes.		

Bar Width (only with barcodes)		The narrowest bar width in SCM units (1/1000 mm). 0=automatic adjustment. Not supported by all barcode types.	Formula	Formula dialog
	Orientation (only with barcodes)	Orientation of the barcode within the object frame.	0	left
			1	centered
			2	right
			Formula	Formula dialog
Bar Width Ratio (only with barcodes)		The ratio between the bars or spaces. Not supported by all barcode types.	Formula	Formula dialog
Show Text (only with barcodes)		Define whether the contents of the barcode should also be displayed as clear text.	True	Yes
			False	No
			Formula	Formula dialog
Rotation		Rotates the object counter-clockwise. E.g. turn column titles by 90°	0	0°
			1	90°
			2	180°
			3	270°
			Formula	Formula dialog
Fit		Determines the behavior when the content is too long for the line. A long word is cut off at value "1" (Wrap). With the option "Force Wrap" you can force a wrap after the last suitable character. Value "3" (compress) reduces the character separation and should only up to a certain limit be used around the legibility to ensure.	0	Truncate
			1	Wrap
			2	Shrink
			3	Compress
			Formula	Formula dialog
	Force Wrap	Sets if a long word should be forced to wrap after the last suitable character.	True	Yes
			False	No
			Formula	Formula dialog
Format		The format editor is available in text object paragraphs and in table columns and is an alternative to formatting with the functions Date$() and FStr$() in the formula dialog. Using the format editor, you can set the format for numbers, currency, dates, times, percentages	True	Dialog
			False	

	and angles. By default, the computer's system settings are used. Note that the format settings affect the entire expression. If you only wish to format certain sections of the expression (e.g. some text and numbers within the expression) use the functions Date$() and FStr$() in the formula dialog.		
Background	The background color for the column can be defined.		
Height (not for Text, RTF Text)	Height of the selected column in mm. The actual height of a table line is the height of the tallest column.	Number	
		Formula	Formula dialog
Options (for Chart, HTML, OLE)	Opens the contents dialog for the appropriate object type.		Opens dialog
Frame	This sets the frame properties and the spacing between the frame and the individual cells of a table. Along with the font size, the cell edges "top" and "bottom" affect the height of the table row in text objects.	...	Opens dialog

4.14.6 Footers and Headers

When a table is wrapped to the next page, the headers of the currently active table and the outermost table are printed again. To prevent the headers from being reprinted on the next page, use the function FirstHeaderThisTable() as appearance condition.

The same functionality exists with footers. Use the function LastFooterThisTable() as the appearance condition. When using this, the footer will only be printed on the last table page if the table is wrapped to the next page.

Additional information about this functions can be found in Chapter "List of Available Functions".

4.14.7 Defining Group Lines

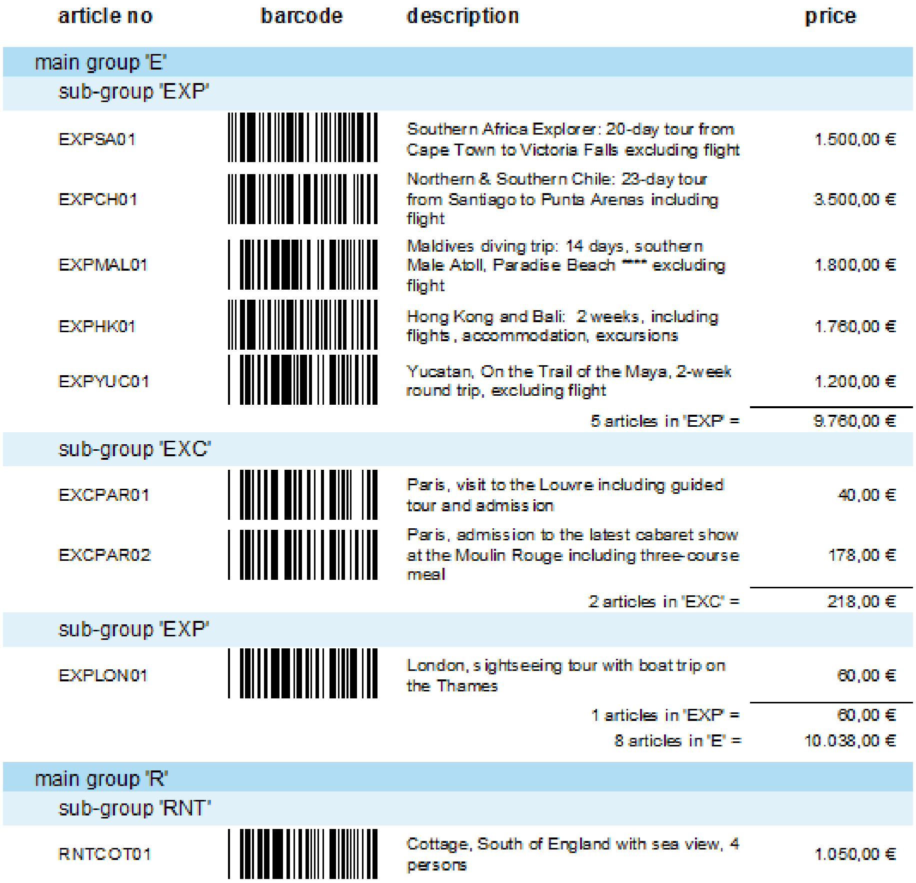
The group lines are another special line type. Their purpose is to combine the data lines to be printed into groups. An example would be an alphabetical directory in which letters are used as intermediate headers. The data lines would be grouped and printed according to the first letter of the name.

Whenever the string defined in the property "Group by" changes, a group change is triggered and the assigned column definition for the group line will be printed as an intermediate header.

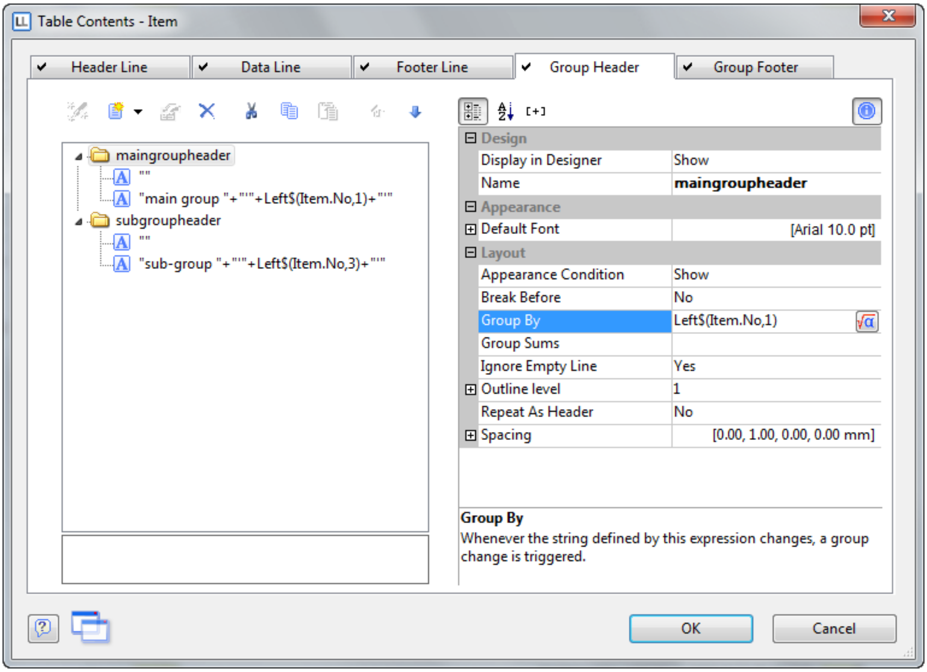
Example: Group by the first Letter of the variable "Item.No".

Use the expression "Left$(Item.No,1)" for the property "Group by". The result is the first letter of the variable "Item.No". Whenever the string defined by this expression changes, a group change is triggered.

The assigned column definition for the group line could be "'main group ' + Left$(Item.No,1)". For every new beginning letter in the variable Item.No, the group line will be printed. More than one line layout is also possible. Hierarchically indented intermediate headers can be created. You can define a line layout, like in the above example, which will generate an intermediate header using the first letter of the variable Item.No. Additionally, you can define a second line layout which generates an intermediate header using the first three characters of the variable Item.No. The expression for this would be "Left$(Item.No,3)", a suitable column definition would be "'sub-group ' + Left$(Item.No,3)". The result could look like:



article no	barcode	description	price
main group 'E'			
sub-group 'EXP'			
EXPSA01		Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight	1.500,00 €
EXPCH01		Northern & Southern Chile: 23-day tour from Santiago to Punta Arenas including flight	3.500,00 €
EXPMAL01		Maldives diving trip: 14 days, southern Male Atoll, Paradise Beach **** excluding flight	1.800,00 €
EXPHK01		Hong Kong and Bali: 2 weeks, including flights, accommodation, excursions	1.760,00 €
EXPYUC01		Yucatan, On the Trail of the Maya, 2-week round trip, excluding flight	1.200,00 €
		5 articles in 'EXP' =	9.760,00 €
sub-group 'EXC'			
EXCPAR01		Paris, visit to the Louvre including guided tour and admission	40,00 €
EXCPAR02		Paris, admission to the latest cabaret show at the Moulin Rouge including three-course meal	178,00 €
		2 articles in 'EXC' =	218,00 €
sub-group 'EXP'			
EXPLON01		London, sightseeing tour with boat trip on the Thames	60,00 €
		1 articles in 'EXP' =	60,00 €
		8 articles in 'E' =	10.038,00 €
main group 'R'			
sub-group 'RNT'			
RNTCOT01		Cottage, South of England with sea view, 4 persons	1.050,00 €



maingroupheader
- ""
- "main group "+"'"+Left$(Item.No,1)+"'"

subgroupheader
- ""
- "sub-group "+"'"+Left$(Item.No,3)+"'"

Design	
Display in Designer	Show
Name	**maingroupheader**
Appearance	
Default Font	[Arial 10.0 pt]
Layout	
Appearance Condition	Show
Break Before	No
Group By	Left$(Item.No,1)
Group Sums	
Ignore Empty Line	Yes
Outline level	1
Repeat As Header	No
Spacing	[0.00, 1.00, 0.00, 0.00 mm]

Group By
Whenever the string defined by this expression changes, a group change is triggered.

It is also possible to use appearance conditions for group header lines. Further information can be found in the section "Appearance Conditions for Table Lines" in this chapter. The options of the group header are defined in a property list:

Property	Description	Value	Description
Group by	Whenever the string defined by this expression changes, a group change is triggered.	Formula	Formula dialog
Group Sums	The selected sum variables are reset each time the group line is printed. We recommend to use the function Sum() for aggregations.		Dialog
Repeat as Header	Prints the group header again after a page break.	True False Formula	Yes No Formula dialog
PDF Index Level	Sets the index level for the PDF-TOC (0=do not add to TOC).	Number Formula	 Formula dialog

Break Before	If the group line is printed, a pagebreak or column break is triggered before. This means every group begins on a new page. When several group lines are printed with this option on at the same time, they will appear underneath each other on the new page. In the formula dialog you can set specific circumstances that trigger a page break before group headers, e.g. "Page break before there is only 5% space left". Further information can be found in the appendix under the description of the function "RemainingTableSpace".	True False Formula	Yes No Formula dialog

4.14.8 Define Group Footers

A further special line type is the group footer. In principle, it works in exactly the same way as the group header, but appears only after the conditions have changed.

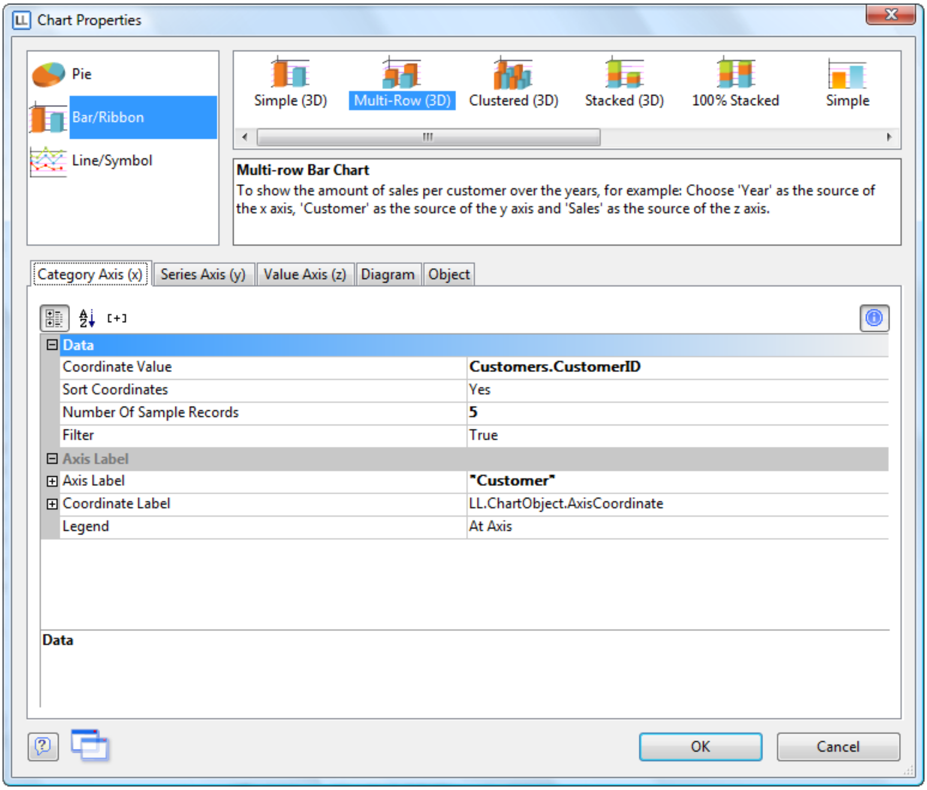
The options of the group footer are defined in a property list:

Property	Description	Value	Description
Group by	Whenever the string defined by this expression changes, a group change is triggered.	Formula	Formula dialog
Group Sums	The selected sum variables are reset each time the group line is printed. This setting is useful to create group sums, for example to add the price of all items of a certain item group. We recommend to use the function Sum() for aggregations.		Dialog
Pagewrap After	After a group footer is printed, a page break is triggered. This means the next group begins on a new page. When several group lines are printed with this option on at the same time, they will appear underneath each other on the "old" page. In the formula dialog you can set specific circumstances that trigger a page break after group footers, e.g. "Page break when there is only 5% space left". Further information can be found in the appendix under the description of the function "RemainingTableSpace".	True False Formula	Yes No Formula dialog

4.15 Inserting Chart Objects

A chart object can be inserted in the tool window "Report structure" or select Objects > Insert > Chart (depends of application).

This object is used for analyzing and displaying different types of data. You can visualize the range of sales for example, the percentage of different sources or, of course, "simple" bar charts. A large number of different chart types are available and can be used for diverse applications. Most chart types have different sub-types.



Multi-row Bar Chart
To show the amount of sales per customer over the years, for example: Choose 'Year' as the source of the x axis, 'Customer' as the source of the y axis and 'Sales' as the source of the z axis.

⊟ **Data**	
Coordinate Value	**Customers.CustomerID**
Sort Coordinates	Yes
Number Of Sample Records	**5**
Filter	True
⊟ Axis Label	
⊞ Axis Label	**"Customer"**
⊞ Coordinate Label	LL.ChartObject.AxisCoordinate
Legend	At Axis

Data

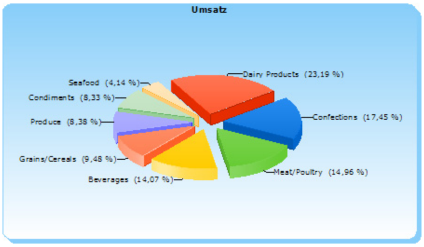
In general, a distinction is made between three and two axis charts. Three axis charts have three data axes and can be used to visualize the sales range per month and employee, for example. A two axis chart has only two data axes and shows e.g. the sales range for the whole company over a number of months.

4.15.1 Properties

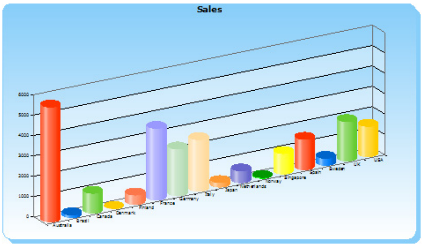
See chapter "4.13.3 Characteristics of Elements" in the report container.

4.15.2 Selecting the Chart Type

Select the chart type from the list on the left hand side. The following types are available:

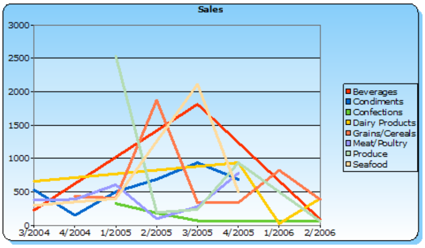


Pie Chart: a typical pie chart.



Cylinder/Bar/Pyramid: various types of bar charts that differ mainly in the form of the bars.

Ribbon: a three dimensional ribbon chart, e.g. to represent processes. Select the sub-type "Multi row" and choose "Ribbon" for the property "Presentation" on the tab "Value Axis".

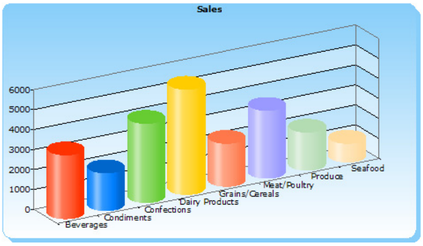


Line/Symbols/Line & Symbols: two dimensional charts.

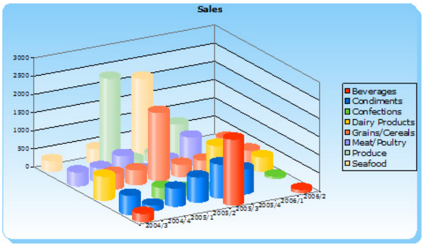
4.15.3 Selecting the Sub Type

Depending on the chart type selected, up to eight different sub types may be available.

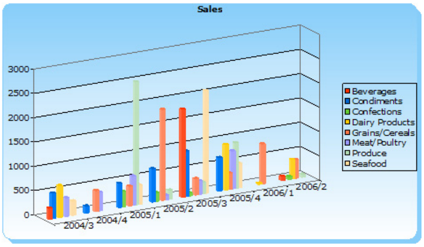
Sub types for cylinder, bar and pyramid charts:



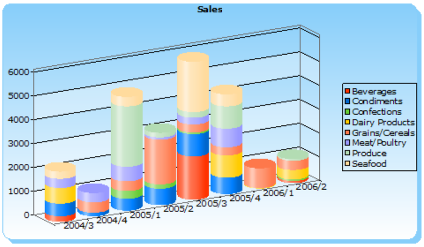
Simple 3D/Simple: Only two axes are available, showing the total amount of sales per month for example. The simple type is without the 3D effect.



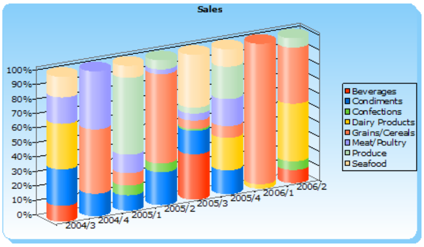
Multi row: A "normal" 3D chart, showing the sales per month and per employee, for example



Clustered 3D/Clustered: Values on the x axis are grouped allowing a direct comparison of values. The clustered type is without the 3D effect.

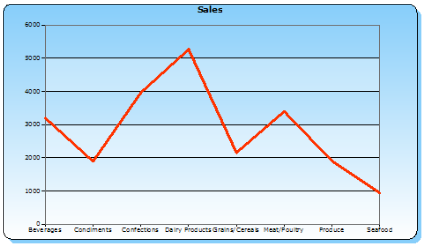


Stacked 3D/Stacked: This chart type is designed to show the share of each contribution, e.g. the share of sales each employee contributes to the total. This type is not available for pyramid or cone charts. The stacked type is without the 3D effect.

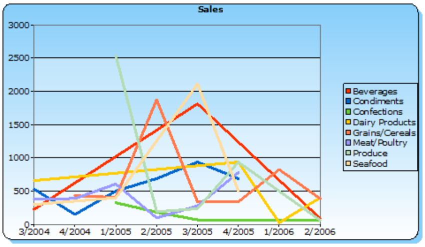


100% Stacked 3D/100% Stacked: Similar to the stacked chart, single contributions can be visualized with this chart. This chart type, however, shows percentile values. The value of a month's sales always equals 100% and the percentage of contribution for each employee can be extracted from the diagram. This type is not available for pyramid or cone charts. The stacked type is without the 3D effect.

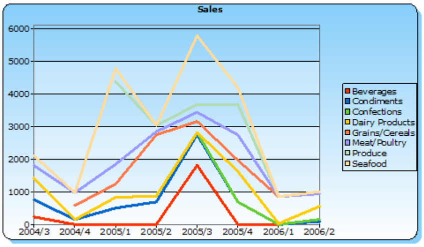
Sub types for line and symbol charts



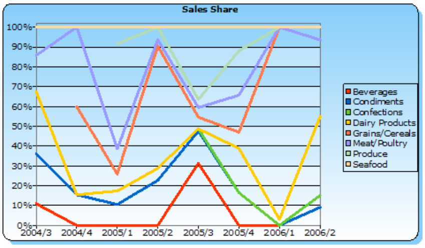
Simple: The values are entered as dots in a coordinate system. Depending on the type, the dots are connected by lines.



Multiple: The values are entered as dots in a coordinate system. Depending on the type, the dots are connected by lines. A multiple diagram contains more than one line.



Stacked: This chart type is designed to show the share of each contribution, e.g. the share of sales each employee contributes to the total.



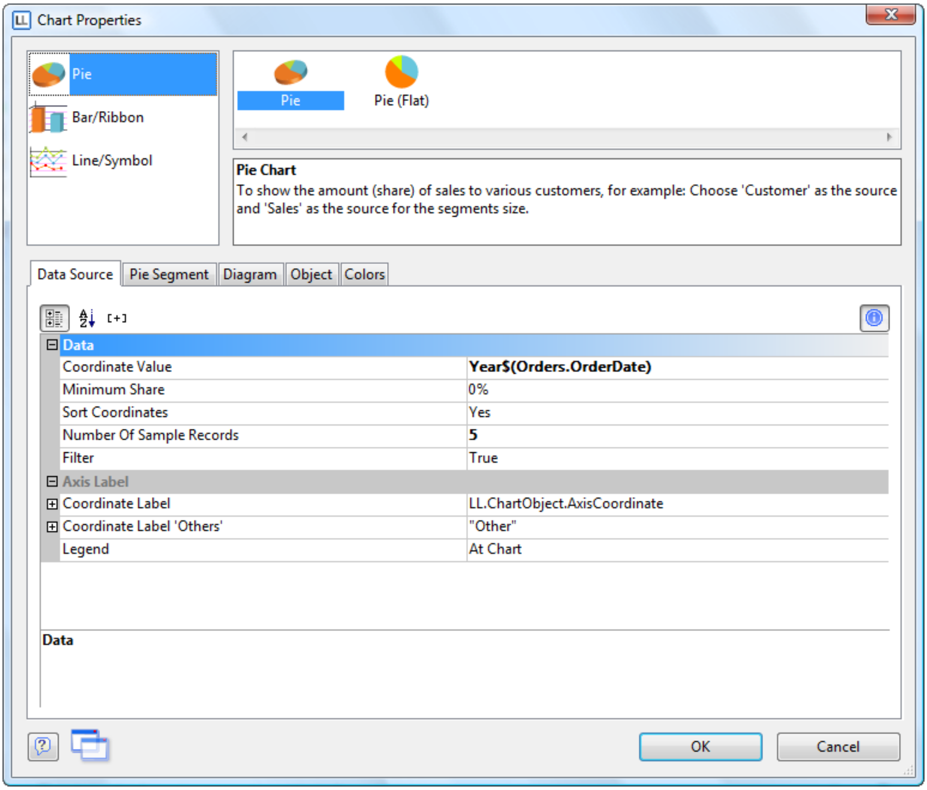
100% Stacked: Similar to the stacked chart, single contributions can be visualized with this chart. This chart type, however, shows percentile values. The value of a month's sales is always 100% and the percentage of contribution for each employee can be extracted from the diagram.

4.15.4 Chart Properties Pie Charts

Pie charts have a special position; there are no actual "axes" but segments. Thus, the configuration differs from that of the other chart types. The chart object provides a variety of options.

Options for the data source of the Pie chart

On the Data Source tab, you set the Data and Label of the pie chart.



Pie Chart
To show the amount (share) of sales to various customers, for example: Choose 'Customer' as the source and 'Sales' as the source for the segments size.

Data	
Coordinate Value	**Year$(Orders.OrderDate)**
Minimum Share	0%
Sort Coordinates	Yes
Number Of Sample Records	**5**
Filter	True
Axis Label	
Coordinate Label	LL.ChartObject.AxisCoordinate
Coordinate Label 'Others'	"Other"
Legend	At Chart

Data

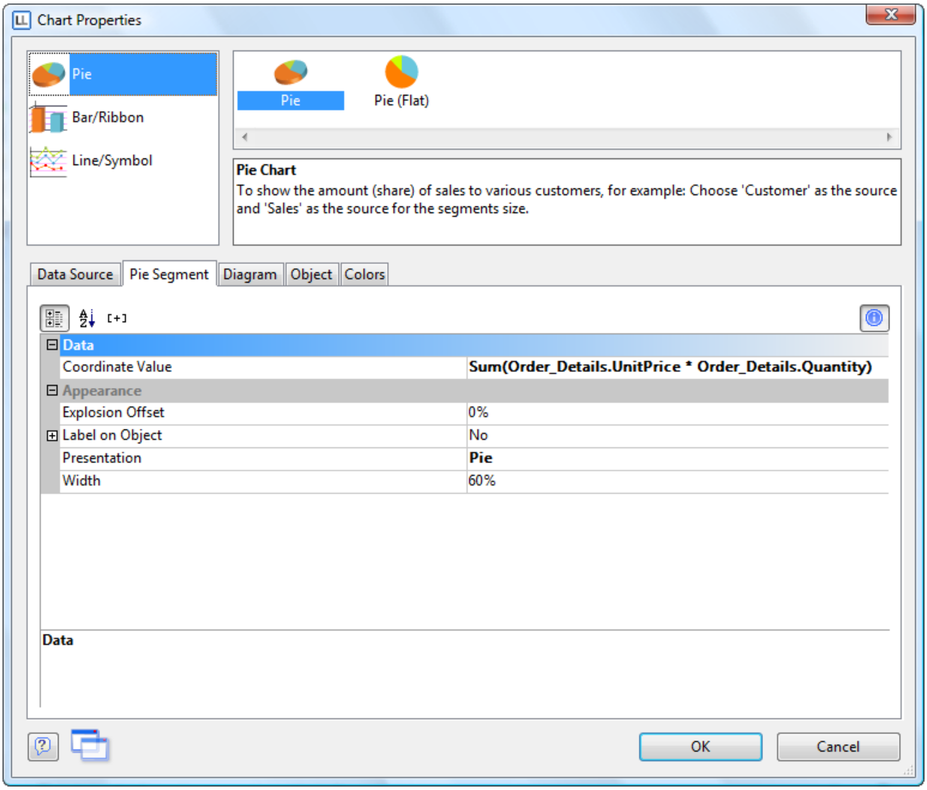
See also Chapter "Common Object Properties".

Property	Description	Value	Description
Coordinate Value	Select the data source for the segments, e.g. "Name" for people, "Month" for date values, and so on.	Formula	Formula dialog
Minimum share	Especially when you have numerous values that make up small contributions it may be desirable to summarize these in a common segment. Select the maximum value up to	Number	Formula dialog

	which the segments will be summarized.			
Sort Coordinates	Sets wether the data should be sorted (alpha numeric or numeric).		True False Formula	Yes No Formula dialog
Number of sample records	The Designer has no access to the "real" data which will appear in your chart when printing. In order to have a picture of how your chart is going to look like, you can set the number of segments to appear in the Designer here.		Number	Formula dialog
Filter	Allows to filter the diagram data. Only the records matching the condition will be used for the chart data. If you leave the default setting "True" the chart will display all values.		True False Formula	True False Formula dialog
Coordinate Label	Select the text which should be used to label the legend. Example for value and percent:age 'LL.ChartObject.AxisCoordinate + " - " + Str$(LL.ChartObject.ArcPerc,0,0) + "%"'		Formula	Formula dialog
	Fixed Fontsize	Sets wether the font size should be fixed. If not, the font size will be decreased in order to prevent text from overlapping.	True False Formula	Yes No Formula dialog
Coordinate Label "Others"	Defines the label for the summarized segment "Others". If you do not enter your own formula, the label will be adapted to the "normal" segment label layout on the data source card.		Formula	Formula dialog
Legend	Select the position of the legend. If you select "At Chart" the values are given directly at the segments.		None At Chart	

Options for the Pie Segment

On the **Pie Segment** tab, you set the Data and Appearance of the pie segment.



Pie Chart
To show the amount (share) of sales to various customers, for example: Choose 'Customer' as the source and 'Sales' as the source for the segments size.

Data	
Coordinate Value	**Sum(Order_Details.UnitPrice * Order_Details.Quantity)**
Appearance	
Explosion Offset	0%
Label on Object	No
Presentation	**Pie**
Width	60%

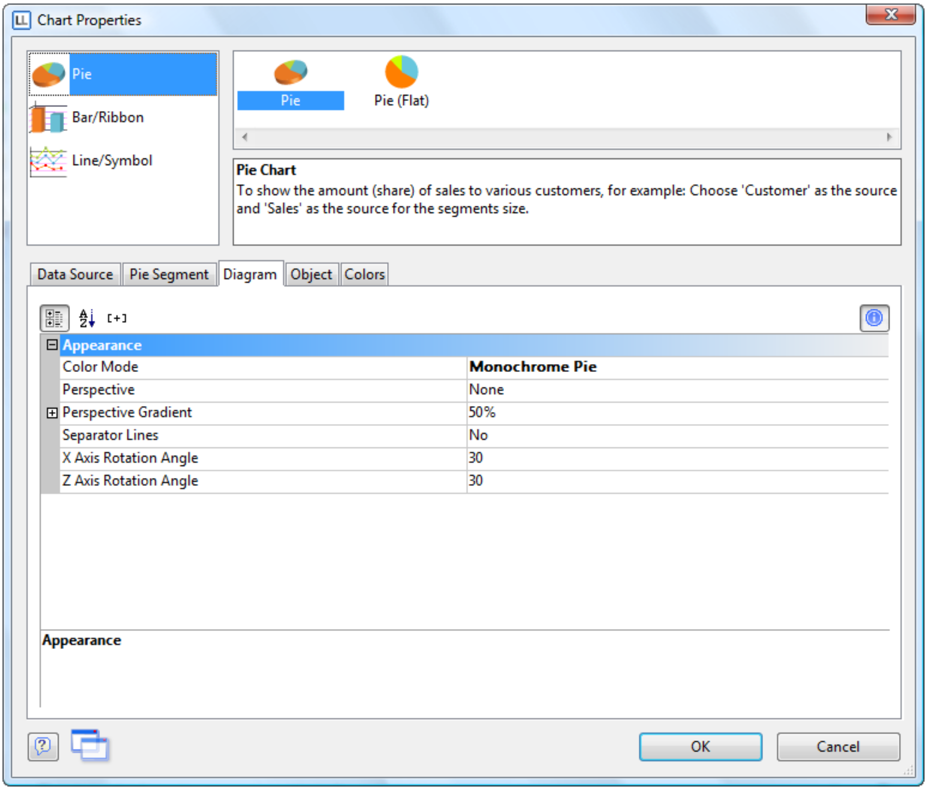
Data

See also Chapter "Common Object Properties".

Property	Description	Value	Description
Coordinate Value	Select the data source for the segment size, e.g. "Value", "Price" etc.	Formula	Formula dialog
Width	Pie width in percent.	Number	Formula dialog
Explosion Offset	Sets the distance of pie explosion in percent of pie radius.	Number	Formula dialog
Label on Object	Sets the text to display on the objects.	0	No
		1	Yes
		Formula	Formula dialog
	Content The label of the pie segment.	Formula	Formula dialog

Options for the pie chart diagram

On the **Diagram** tab, you set the Appearance of the pie chart diagram.



Pie Chart
To show the amount (share) of sales to various customers, for example: Choose 'Customer' as the source and 'Sales' as the source for the segments size.

Appearance	
Color Mode	**Monochrome Pie**
Perspective	None
Perspective Gradient	50%
Separator Lines	No
X Axis Rotation Angle	30
Z Axis Rotation Angle	30

Appearance

See also Chapter "Common Object Properties".

Property	Description	Value	Description
Color Mode.	The segments are colored with different colors in order to make the diagram easier to read.	0	Monochrome
		1	Colored
		Formula	Formula dialog
Perspective	Select the level of perspective for your pie chart here.	0	None
		1	light
		2	strong
		Formula	Formula dialog
Perspective Gradient	Sets wether a gradient fill should be applied to the pie chart surface (in percent).	Number	Formula dialog

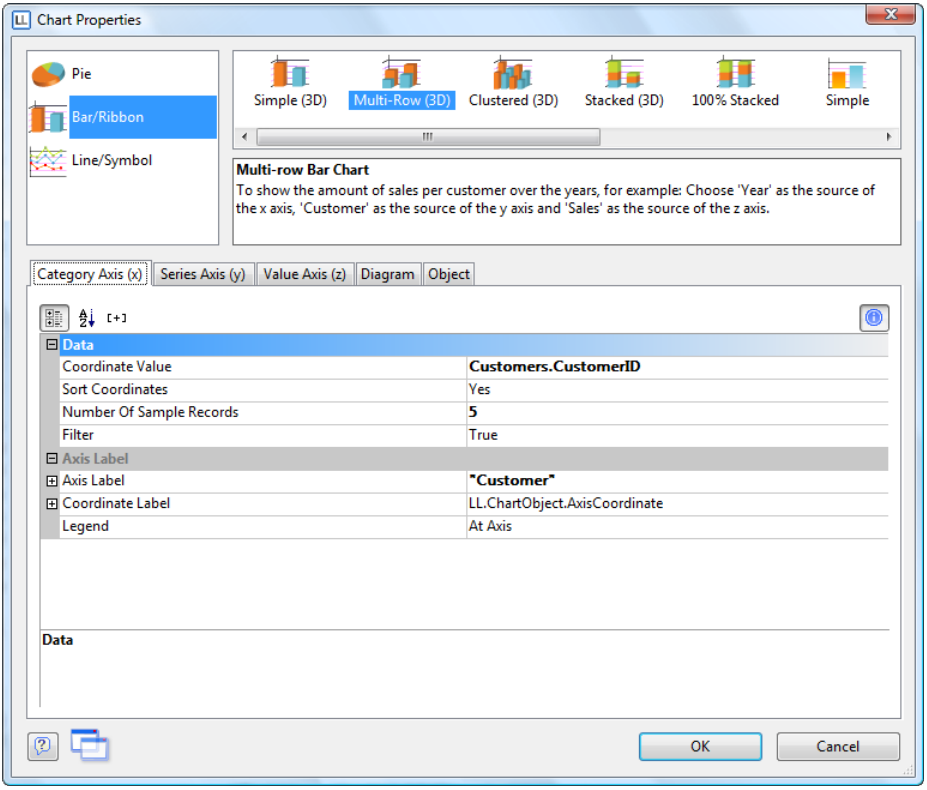
	Accentuate frame	Accentuates the frame of the pie.	True	Yes
			False	No
			Formula	Formula dialog
X Axis Rotation Angle	The rotation angle of the x axis upward in degrees, maximally 90° (perpendicular). You may also set these angle using the rotation buttons which appear when the chart is selected on the workspace.		Number	Formula dialog
Y Axis Rotation Angle	The rotation angle around the pie center against the clockwise direction in degrees. You may also set these angle using the rotation buttons which appear when the chart is selected on the workspace.		Number	Formula dialog
Separator Lines	Sets whether separator lines should be drawn between the pie segments.		True	Yes
			False	No
			Formula	Formula dialog

4.15.5 Chart Properties Bar/Line Charts

For the chart type "Bar/Ribbon" and "Line/Symbol" you have the following options. Depending on the chart type, there are two or three data axes. Using the tabs you can switch from one axis to another. Select the presentation (Cylinder, Bar, Pyramid, Ribbon, Line, Symbols, Line+Symbols, Octaeder, Cone) in the property "Presentation" on the tab "Value Axis".

Options for category and series axis

If you chose to have a three axis chart, both axes are available (as x and y-axis). Only the category axis is needed (as x axis) for two axis charts. You have the same options for both axes:



Multi-row Bar Chart
To show the amount of sales per customer over the years, for example: Choose 'Year' as the source of the x axis, 'Customer' as the source of the y axis and 'Sales' as the source of the z axis.

Data	
Coordinate Value	**Customers.CustomerID**
Sort Coordinates	Yes
Number Of Sample Records	**5**
Filter	True
Axis Label	
Axis Label	**"Customer"**
Coordinate Label	LL.ChartObject.AxisCoordinate
Legend	At Axis

Data

See also Chapter "Common Object Properties".

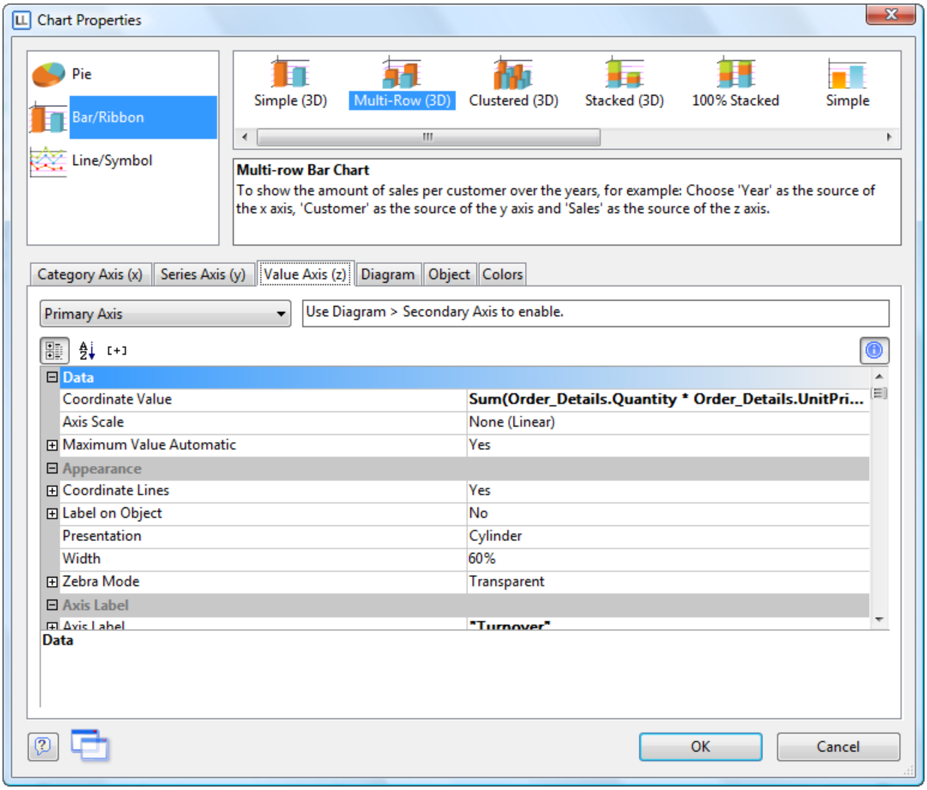
Property	Description	Value	Description
Coordinate Value	Select the data source for the row, e.g. "Name" for people, "Month" for date values, and so on.	Formula	Formula dialog

Sort Coordinates	Sets wether the data should be sorted (alpha numeric or numeric).		True False Formula	Yes No Formula dialog
Number of sample records	The designer has no access to the "real" data which will appear in your chart when printing. In order to have a picture of how your chart is going to look like, you can set the number of values to appear in the designer here.		Number	Formula dialog
Filter	Allows to filter the diagram data. Only the records matching the condition will be used for the chart data. If you leave the default setting "True" the chart will display all values.		True False Formula	True False Formula dialog
Axis label	Select a text for your axis label here.		Formula	Formula dialog
	Rotation	Rotation Angle in degrees.	Number	Formula dialog
Coordinate Label	Select the text which should be used to label the legend.		Formula	Formula dialog
	Rotation	Rotation Angle in degrees.	Number	Formula dialog
	Fixed Fontsize	Sets wether the font size should be fixed. If not, the font size will be decreased in order to prevent text from overlapping.	True False Formula	Yes No Formula dialog
Legend	Select the position of the legend. If you Select "at chart" the values are given directly at the axis. Otherwise, a legend will be added on the left, right, bottom or top of the chart.		None At Axis Top Left Right Bottom	

Options for the value axis

On the Value Axis tab, you set the Data, Appearance and Axis Label of the Bar Chart or Line Chart.

Primary Axis / Secondary Axis: A second value axis is supported. Enable the secondary axis on the tab "Diagram". Use the Combobox to switch between the properties of the axes.



Multi-row Bar Chart
To show the amount of sales per customer over the years, for example: Choose 'Year' as the source of the x axis, 'Customer' as the source of the y axis and 'Sales' as the source of the z axis.

Primary Axis

Use Diagram > Secondary Axis to enable.

Data	
Coordinate Value	Sum(Order_Details.Quantity * Order_Details.UnitPri...
Axis Scale	None (Linear)
Maximum Value Automatic	Yes
Appearance	
Coordinate Lines	Yes
Label on Object	No
Presentation	Cylinder
Width	60%
Zebra Mode	Transparent
Axis Label	
Axis Label	"Turnover"

Data

See also Chapter "Common Object Properties".

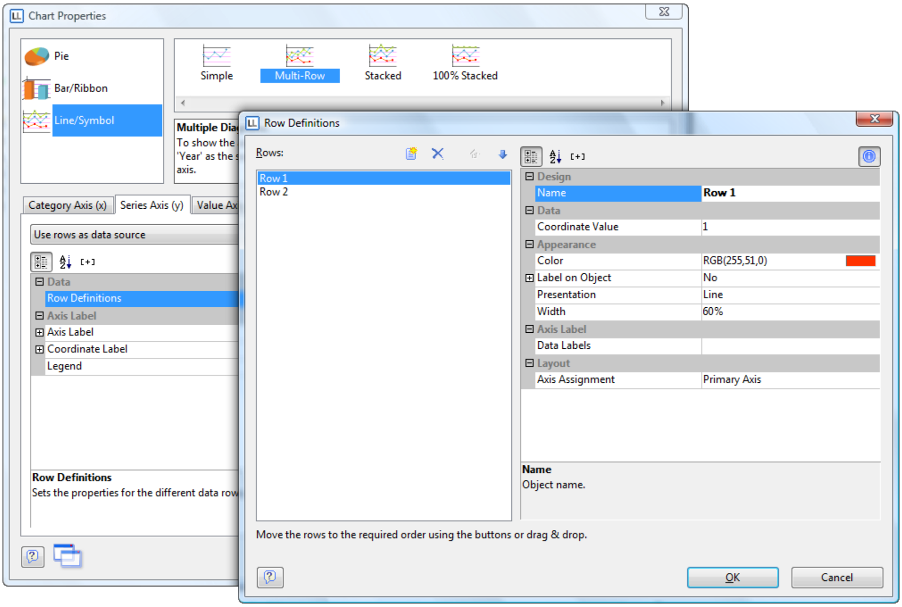
Property	Description	Value	Description
Coordinate Value	Select the data source for the value axis, e.g. "Value", "Price" etc.	Formula	Formula dialog
Axis scale	Axis scale type.	0	None (linear)
		1	Logarithmic, Base 10 (decadicly)
		2	Logarithmic,

				Base 2 (binary)
			Formula	Formula dialog
Maximum Value Automatic	You can limit the shown data range in order to e.g. regard huge divergences within the values. If your values have high peaks, you can cut them off by setting a max limit. If you leave the default setting "No" the chart will display all values.		True False Formula	Yes No Formula dialog
	Threshold	Maximal Axis Value.	Number	Formula dialog
Minimum Value Automatic	You can limit the shown data range in order to e.g. regard huge divergences within the values. If your values have low peaks, you can cut them off by setting a Minimum Value. If you leave the default setting "No" the chart will display all values.		True False Formula	Yes No Formula dialog
	Threshold	Minimal Axis Value.	Number	Formula dialog
Presentation	Visual presentation. Depending on selected type and subtype different types of representation are available: Pie: - Bar Simple, Clustered:1, 2, 3, 8, 9 Bar Multi Row:1, 2, 3, 4, 8, 9 Bar Stacked: 1, 2, 8 Line: 5, 6, 7		1 2 3 4 5 6 7 8 9 Formula	Cylinder Bar Pyramid Ribbon Line Symbols Line + Symbols Octaeder Cone Formula dialog
Width	Bar or Line width.		Number	Formula dialog
Label on objects	Sets the text to display on the objects.		0 1 Formula	No Yes Formula dialog
	Content	The formula of the elements label, e.g. LL.ChartObject.AxisCoordinate	Formula	Formula dialog
Coordinate Lines	Sets whether coordinate lines should be drawn in the chart background.		True False	Yes No

			Formula	Formula dialog
Zebra Mode	Sets whether a zebra pattern should be drawn in the chart background.			Transparent Pattern/Block Color Horiz. Gradient Vert. Gradient Horiz. Two-Part Gradient Vert. Two-Part Gradient Partly Transparent
Axis Label	Axis label text.		Formula	Formula dialog
	Rotation	Rotation of the axis label text in degrees.	Number	Formula dialog
Coordinate Label	Coordinate label resp. legend text, e.g. LL.ChartObject.AxisCoordinate.		Formula	Formula dialog
	Rotation	Rotation of the Coordinate Label text in degrees.	Number	Formula dialog
	Fixed Fontsize	Sets wether the font size should be fixed. If not, the font size will be decreased in order to prevent text from overlapping.	True False Formula	Yes No Formula dialog
Coordinate Tick Distance	Coordinate Tick Distance (Ticks).		True False Formula	Automatic Manual Formula dialog
Legend	Select the position of the legend. If you Select "at chart" the values are given directly at the axis. Otherwise, a legend will be added on the left, right, bottom or top of the chart.			Keine an Achse
Number of Ticks	The number of sub-ticks between two major coordinate ticks.		Number	Formula dialog

Define rows

The Series Axis (y-axis) for a three dimensional chart has the additional possibility to select the values not just by formula but also by rows. Modify via the Combobox located above the properties list for the entry "Use rows as data source". The property "Row Definitions" can be used to modify the properties of the axis. Open the dialog "Row definitions" and define the individual rows (e.g. Measured Value/Target/Actual values). It is possible to define properties for each row and then move it to the section orientation menu.



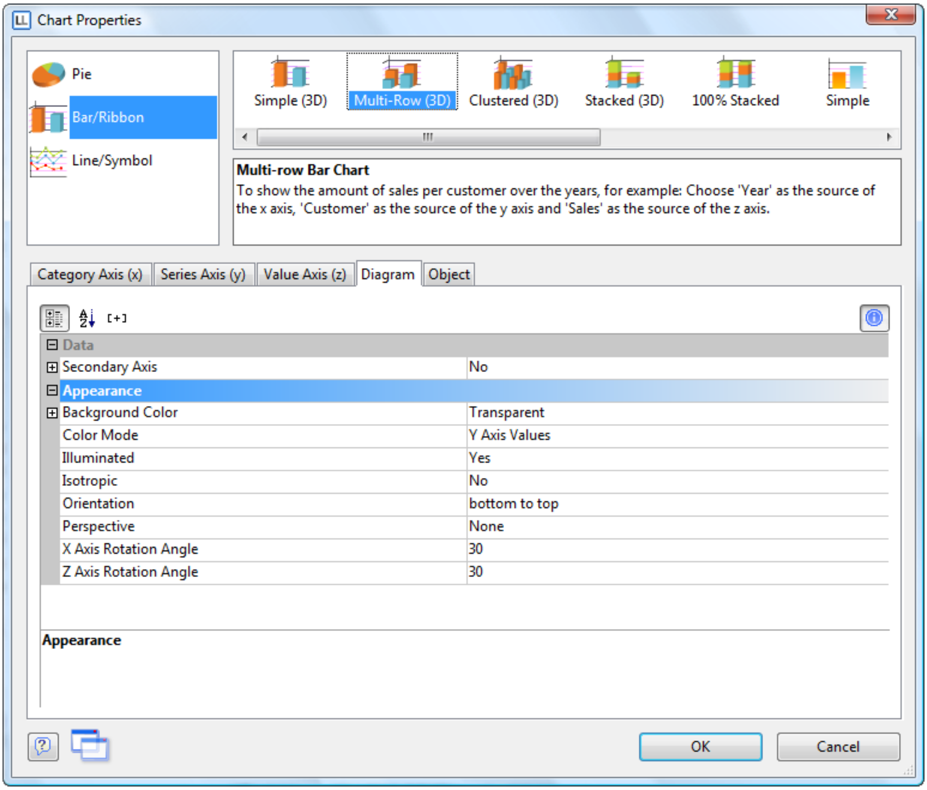
Design	
Name	**Row 1**
Data	
Coordinate Value	1
Appearance	
Color	RGB(255,51,0) 🟥
Label on Object	No
Presentation	Line
Width	60%
Axis Label	
Data Labels	
Layout	
Axis Assignment	Primary Axis

Name
Object name.

Move the rows to the required order using the buttons or drag & drop.

Diagram-Options for the chart object

On the Diagram tab, you set the optional secondary axis and Appearance of the Bar Chart or Line Chart..



Multi-row Bar Chart
To show the amount of sales per customer over the years, for example: Choose 'Year' as the source of the x axis, 'Customer' as the source of the y axis and 'Sales' as the source of the z axis.

⊟ Data	
⊞ Secondary Axis	No
⊟ Appearance	
⊞ Background Color	Transparent
Color Mode	Y Axis Values
Illuminated	Yes
Isotropic	No
Orientation	bottom to top
Perspective	None
X Axis Rotation Angle	30
Z Axis Rotation Angle	30

Appearance

See also Chapter "Common Object Properties".

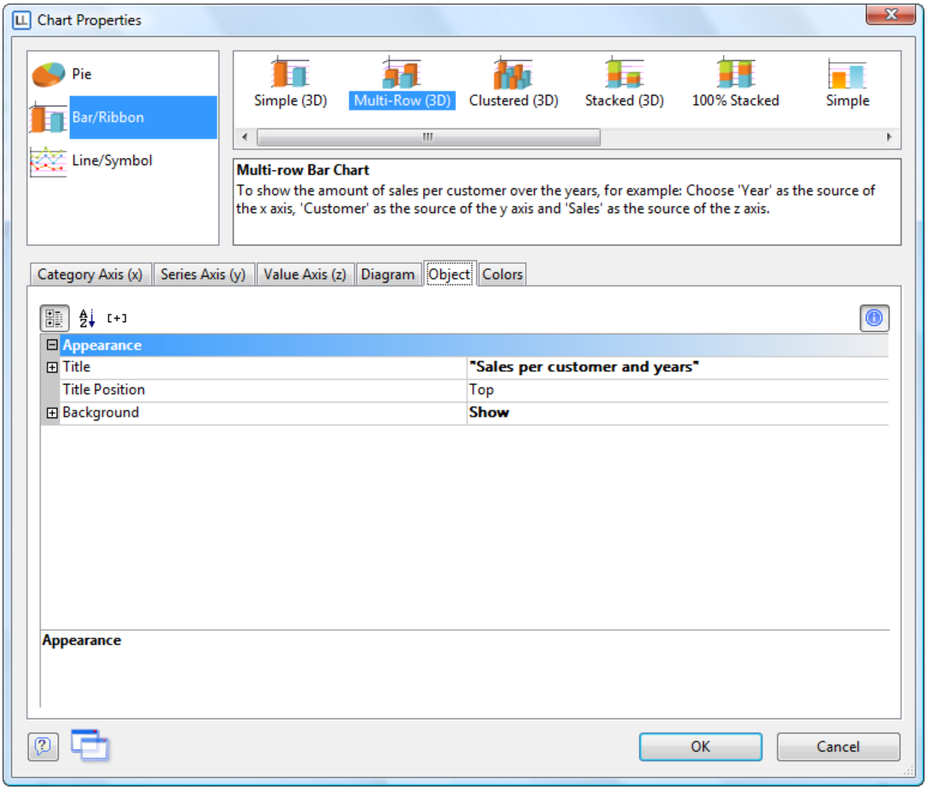
Property	Description		Value	Description
Secondary Axis	Enables a secondary axis. Use the Combobox on the tab "Value Axis" to switch between the properties of the axis.		True	Yes
			False	No
			Formula	Formula dialog
	Axis Assignment	Determines the value axis for the value.	0	Primary
			1	Secondary
			Formula	Formula dialog
Orientation	Orientation of the chart, e.g. a horizontal 2D-Bar-Diagram.			Left to right
				Bottom to top

Illuminated	Sets whether the chart should be illuminated.	True	Yes
		False	No
		Formula	Formula dialog
Color Mode	Sets the axis determining the color (if applicable).	0	Monochrome
		1	X Axis Value
		2	Y Axis Value
		Formula	Formula dialog
Background Color	Sets whether the background should be transparent or colored.		Transparent
			Pattern/Block Color
Isotropic	Sets if both axes use the same units (x- and y-axis).	True	Yes
		False	No
		Formula	Formula dialog
Perspective	Select the level of perspective for your object here.	0	None
		1	light
		2	strong
		Formula	Formula dialog
X Axis Rotation Angle	The rotation angle of the x axis upward in degrees, maximally 90° (perpendicular). You may also set these angle using the rotation buttons which appear when the chart is selected on the workspace.	Number	Formula dialog
Y Axis Rotation Angle	The rotation angle around the center against the clockwise direction in degrees. You may also set these angle using the rotation buttons which appear when the chart is selected on the workspace.	Number	Formula dialog

4.15.6 Chart Properties General

Options for the chart object

On the Object tab, you set the Appearance of the Chart-Object.



Multi-row Bar Chart

To show the amount of sales per customer over the years, for example: Choose 'Year' as the source of the x axis, 'Customer' as the source of the y axis and 'Sales' as the source of the z axis.

Appearance	
Title	**"Sales per customer and years"**
Title Position	Top
Background	**Show**

Appearance

See also Chapter "Common Object Properties".

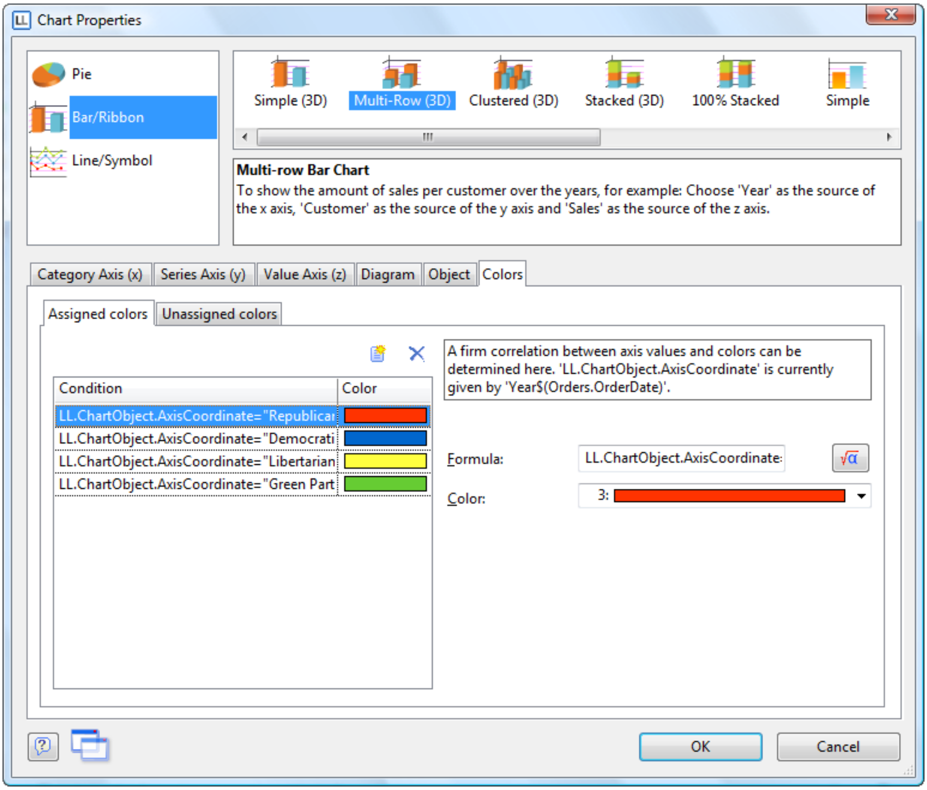
Property	Description	Wert	Description
Title	Select a title for your diagram. This title will be displayed above your chart.	Formula	Formula dialog
Title Position	Position of the chart title.	0	Top
		1	Bottom
		Formula	Formula dialog
Background	Select the color for the "background" of the chart. Alternatively the wall may also be transparent. To select a color, select it from the upper combo box. The "..." entry brings	True	Show
		False	Hide
		Formula	Formula dialog

	up a standard color selection dialog.			
	Filled	Filling of the chart-object.		Transparent Pattern/Block Color Horiz. Gradient Vert. Gradient Horiz. Two-Part Gradient Vert. Two-Part Gradient Partly Transparent Drawing
	Border	Object border.		Transparent Pattern/Block Color
	Shadow	Object shadow.		Transparent Pattern/Block Color
	Rounding	Roundingfactor for the edges (0=rectangular, 100=eliptical)	Number	Formula dialog

Colors for the chart object

On the **Colors** tab, you set the Colors and the order of colors of the Chart. For a bar chart, this is the color of the bars, for a pie chart the segment color..

Assigned colors	It is possible to assign colors to specific axis values. Select the button "New" to set a new assignment. To select a color, select it from the upper combo box. The "..." entry brings up a standard color selection dialog.
Unassigned colors	Select the colors and the order of colors for the chart, which are not assigned at the tab "Assigned colors". For a bar chart, this is the color of the bars, for a pie chart the segment color. To select a color, select it from the upper combo box. The "..." entry brings up a standard color selection dialog.



Multi-row Bar Chart

To show the amount of sales per customer over the years, for example: Choose 'Year' as the source of the x axis, 'Customer' as the source of the y axis and 'Sales' as the source of the z axis.

Condition	Color
LL.ChartObject.AxisCoordinate="Republicar	🟥
LL.ChartObject.AxisCoordinate="Democrati	🟦
LL.ChartObject.AxisCoordinate="Libertarian	🟨
LL.ChartObject.AxisCoordinate="Green Part	🟩

A firm correlation between axis values and colors can be determined here. 'LL.ChartObject.AxisCoordinate' is currently given by 'Year$(Orders.OrderDate)'.

Formula: LL.ChartObject.AxisCoordinate=

Color: 3: 🟥

4.15.7 Chart-Object Fields

There are several chart fields to access to the values of a chart.

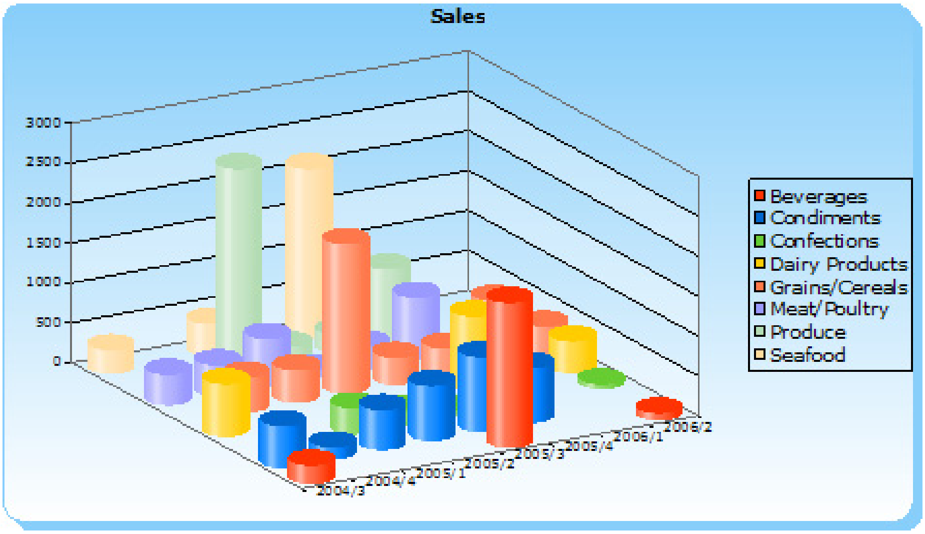
- LL.ChartObject.ArcIndex: returns the index of the actual pie segment (pie chart). The largest pie segment has index 1, the second largest has the index 2 etc.
- LL.ChartObject.ArcPerc: returns the percentage of the actual pie segment (pie chart).
- LL.ChartObject.ArcTotal: returns the value of the total dataset (pie chart).
- LL.ChartObject.ArcTypeIsOthers: returns True, if the actual pie segment is the "Others" pie segment (pie chart).
- LL.ChartObject.ArcValue: returns the value of the actual pie segment (pie chart).
- LL.ChartObject.AxisCoordinate: returns the value of the Axis Coordinate (usable for Coordinate label text).

4.15.8 Examples

Chart options depend on the available data. The following examples have therefore been kept general. An application may give you the fields "OrderDate", "Product Category", "Unit Price" and "Quantity", denoting the category of the product, the quarter in which a transaction took place and the total transaction volume (Unit Price*Quantity).

Multi-row bar chart

This would be the simplest way to analyze your data; you would have a diagram showing the total transaction volume for each category and quarter:

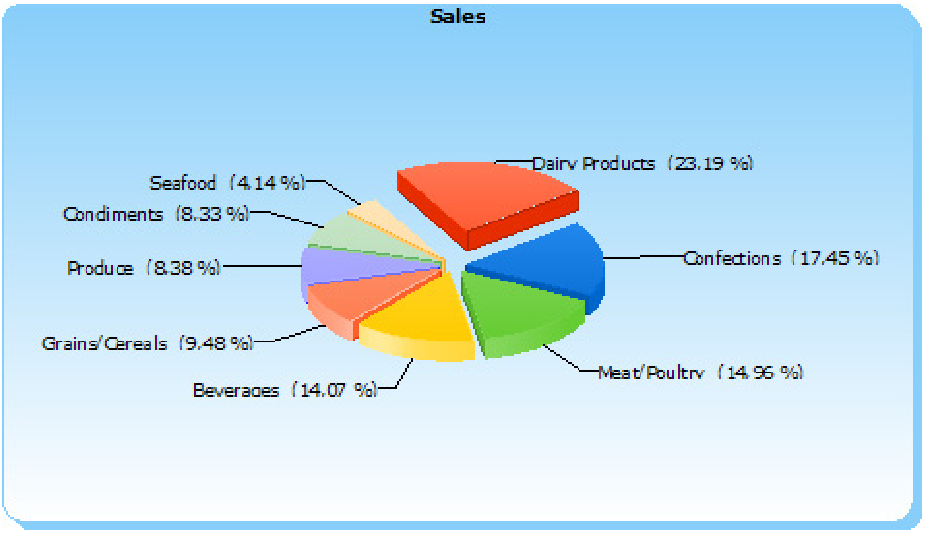


...this is how it's done:

1. Insert a new chart object; select "Bar" as type and "Multi-row" as sub-type.
2. Select "OrderDate" as Coordinate Value for the Category Axis and "Product Category" as Coordinate Value for the Series Axis.
3. For the Value Axis, calculate the total transaction volume in the filed "Coordinate Value", e.g. Sum(Unit Price*Quantity).
4. Set the "Presentation" to "1" (Cylinder), and the width of the cylinder to 80%.
5. Set "Partly Transparent" for the "Zebra mode", "LL.Color.White" for the zebra pattern and "80%" as Fading-In Factor.
6. Select a title for your diagram on the "Object" tab.

Pie chart

If you are interested in the share each product category contributes to the total sales over a period of time, you would select a pie chart. This chart directly displays the shares:

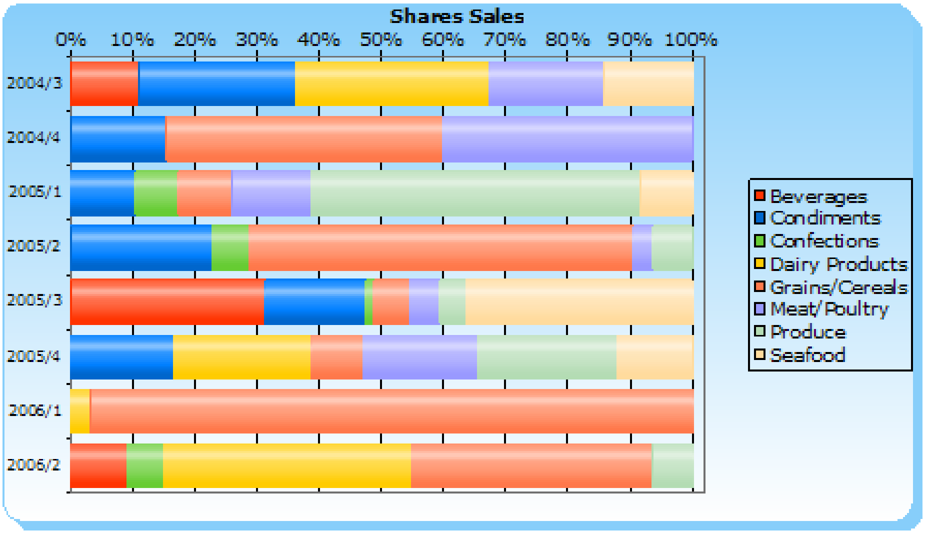


...this is how it's done:

1. Insert a new chart object, select "Pie" as type and sub-type.
2. Select "Product category" as Coordinate Value for the data source.
3. On the "Pie Segment" tab, calculate the total transaction volume in the field "Coordinate Value", e.g. Sum(Unit Price*Quantity).
4. If you set the Explosion Offset to "If(LL.ChartObject.ArcIndex=1,50,10)", the distance of the largest pie segment is set to 50% and all others are set to 10%.

100% stacked bar chart

The pie chart in the last example shows the share over a number of quarters. However, to quickly take a look at the shares for each quarter and extract trend lines, it would be interesting to see the change in the shares over a number of quarters. The 100% stacked bar chart is perfect for this.



...this is how it's done:

1. Insert a new chart object; select "Bar" as type and "100% stacked" as sub-type.
2. Select "OrderDate" as Coordinate Value for the category axis and "Product Category" as Coordinate Value for the series axis.
3. For the value axis, calculate the total transaction volume in the field "Coordinate Value", e.g. Sum(Unit Price*Quantity).
4. On the "Diagram" tab select "Left to right" for the "Orientation".

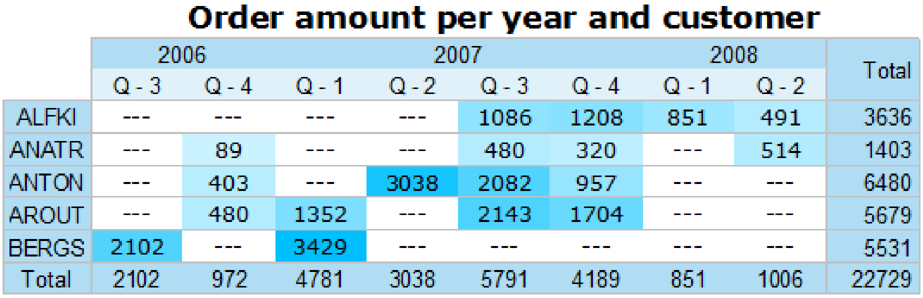
4.16 Inserting Crosstab Objects

A crosstab object can be inserted in the tool window "Report structure" or select Objects > Insert > Crosstab (depends of application).

This object is used to analyze and display different kinds of data in several dimensions. You can use it, for example, to investigate the changes in turnover per year and region and analyze sales by unit and customer, then show the totals by quarter and year. There are numerous layout options available to you.

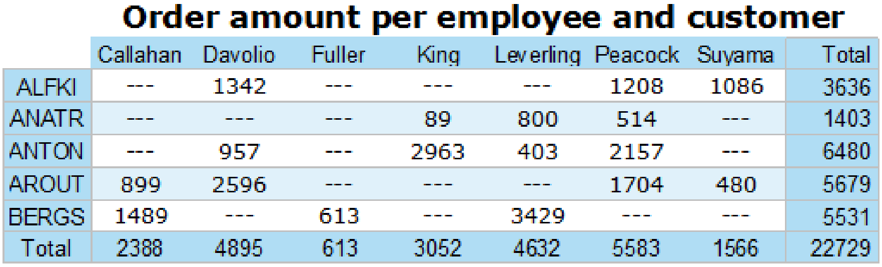
When you are creating a new crosstab, a wizard is available to you to guide you through the three tabs.

Example: Customer sales per sales period and employee



Order amount per year and customer

	2006		2007				2008		Total
	Q - 3	Q - 4	Q - 1	Q - 2	Q - 3	Q - 4	Q - 1	Q - 2	
ALFKI	---	---	---	---	1086	1208	851	491	3636
ANATR	---	89	---	---	480	320	---	514	1403
ANTON	---	403	---	3038	2082	957	---	---	6480
AROUT	---	480	1352	---	2143	1704	---	---	5679
BERGS	2102	---	3429	---	---	---	---	---	5531
Total	2102	972	4781	3038	5791	4189	851	1006	22729



Order amount per employee and customer

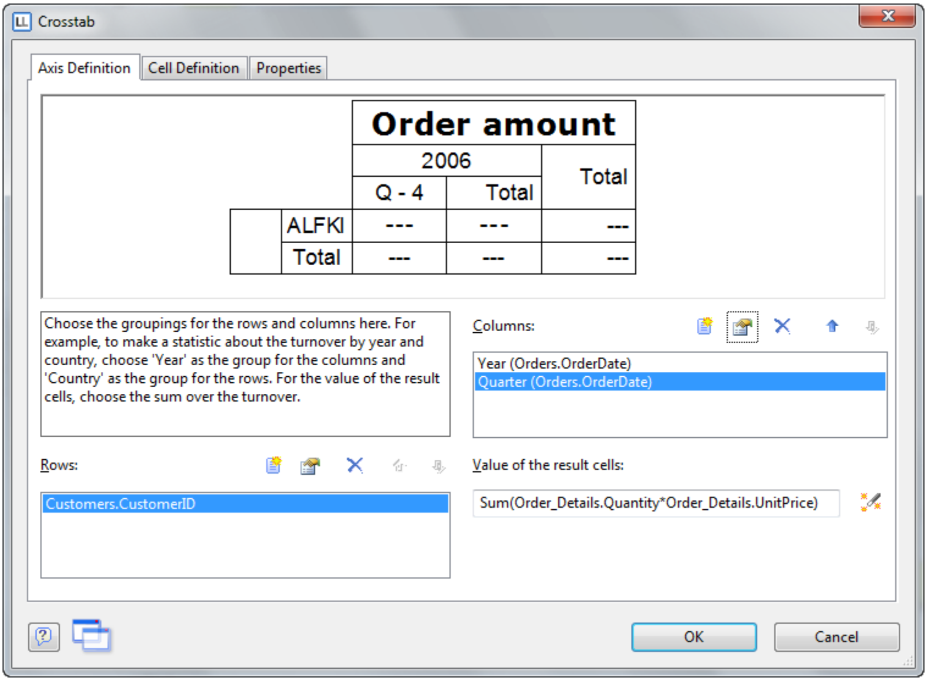
	Callahan	Davolio	Fuller	King	Leverling	Peacock	Suyama	Total
ALFKI	---	1342	---	---	---	1208	1086	3636
ANATR	---	---	---	89	800	514	---	1403
ANTON	---	957	---	2963	403	2157	---	6480
AROUT	899	2596	---	---	---	1704	480	5679
BERGS	1489	---	613	---	3429	---	---	5531
Total	2388	4895	613	3052	4632	5583	1566	22729

4.16.1 Properties

See chapter "4.13.3 Characteristics of Elements" in the report container.

4.16.2 Axis definition (Grouping)

In the "Axis Definition" tab, you can set the groupings for the rows and columns. For example, to create a statistics about the turnover by year and country, select "Country" as the group for the rows and "Year" as the group for the columns. For the contents of the result cells, select the sum over the turnover.



		Order amount		
		2006		Total
		Q - 4	Total	
	ALFKI	---	---	---
	Total	---	---	---

Choose the groupings for the rows and columns here. For example, to make a statistic about the turnover by year and country, choose 'Year' as the group for the columns and 'Country' as the group for the rows. For the value of the result cells, choose the sum over the turnover.

Columns:

Year (Orders.OrderDate)
Quarter (Orders.OrderDate)

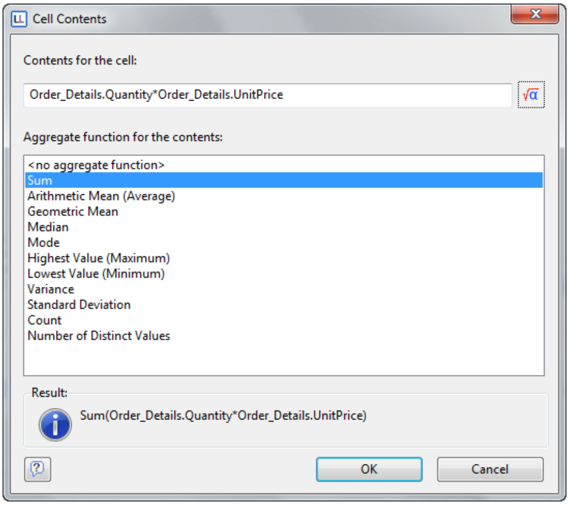
Rows:

Customers.CustomerID

Value of the result cells:

Sum(Order_Details.Quantity*Order_Details.UnitPrice)

- With the "New" button, you can add a group for rows or columns.
- With the "Properties" button you can edit the selected group, with the button "Delete" you can delete the selected group.
- You can add as many groups (layers) as you need, e.g. a group "Year" and another group "Quarter". With the arrow buttons, the order can be changed. The lowest row or column is the innermost group.
- In the field "Value of the result cells" the cell contents are defined. Here you can select with the button "Edit group result formula" in the dialog "Cell contents" an aggregate function for the content, e.g. sum or quantity. This is the formula that is set into the "value" property of each content cell. In the lower part of the dialog "Cell contents", the selected result function is displayed.



Contents for the cell:

Order_Details.Quantity*Order_Details.UnitPrice

Aggregate function for the contents:

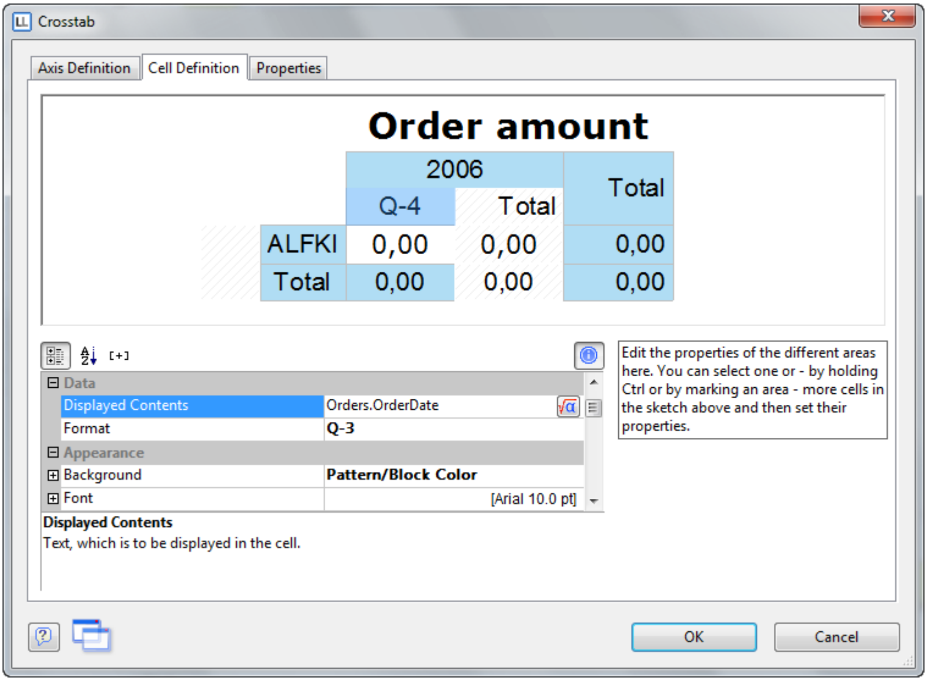
- <no aggregate function>
- Sum
- Arithmetic Mean (Average)
- Geometric Mean
- Median
- Mode
- Highest Value (Maximum)
- Lowest Value (Minimum)
- Variance
- Standard Deviation
- Count
- Number of Distinct Values

Result:

Sum(Order_Details.Quantity*Order_Details.UnitPrice)

4.16.3 Cell definition (Contents)

In the "Cell Definition" tab, the properties of the different cells can be edited.



Order amount

	2006		Total
	Q-4	Total	
ALFKI	0,00	0,00	0,00
Total	0,00	0,00	0,00

⊟ Data	
Displayed Contents	Orders.OrderDate
Format	Q-3
⊟ Appearance	
⊞ Background	Pattern/Block Color
⊞ Font	[Arial 10.0 pt]

Displayed Contents
Text, which is to be displayed in the cell.

Edit the properties of the different areas here. You can select one or - by holding Ctrl or by marking an area - more cells in the sketch above and then set their properties.

You can select the cells directly in the sketch in the top part of the dialog and can then edit their properties. To select more than one cell, hold the CTRL key and selector drag a frame around the cells with the cursor.

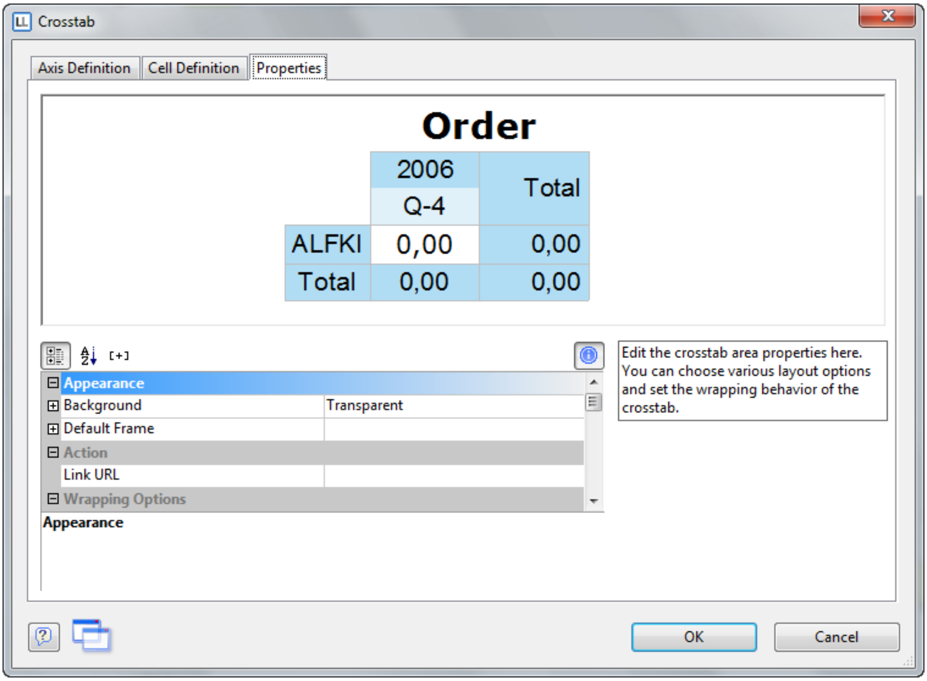
See also Chapter "Common Object Properties".

Property	Description	Value	Description
Value	Formula for the cell value. This is evaluated by the Crosstab.Cells... functions.	Formula	Formula dialog
Displayed contents	Text to be displayed in the cell, so it can be different from the property "Value".	Formula	Formula dialog
Link	Link that is opened when clicked (only in preview, PDF and HTML export).	Link	Link
			Formula dialog
Rotation	Rotates the object anti-clockwise. With this function you can, for example, turn column titles by 90°.	0	0°
		1	90°
		2	180°
		3	270°
		Formula	Formula dialog
Vertical alignment	Vertical alignment of the contents in the available space.	0	top
		1	centre
		2	bottom
		Formula	Formula dialog
Alignment (with Text)	The text alignment. Decimal means numbers are aligned by their decimal points.	0	Left
		1	Centre
		2	Right
		3	Decimal
		Formula	Formula dialog
	Decimal position Position of the decimal point (only applicable in decimal alignment, negative means from the right).	Number	
		Formula	Formula dialog
Maximum Width	Sets the maximum width for a cell. If the text is wider, the line is wrapped. With the function Join$ () you can then show the individual values in a cell.	Number	
		Formula	Formula dialog
Minimum Width	Sets the minimum width of the cell.	Number	
		Formula	Formula dialog

Minimum Height	Sets the minimum height of the cell.	Number	
		Formula	Formula dialog

4.16.4 Layout Options and Page Break Behavior

Edit the properties of the crosstab here, e.g. layout and page break behavior.



Appearance	
Background	Transparent
Default Frame	
Action	
Link URL	
Wrapping Options	

Edit the crosstab area properties here. You can choose various layout options and set the wrapping behavior of the crosstab.

See also Chapter "Common Object Properties".

Property	Description	Value	Description
PDF index text	Text for the PDF index.	Formula	Formula dialog
Link	Link that is opened when clicked (only in preview, PDF and HTML export).	Link	Formula dialog
Minimum Size	Sets by how much the crosstab can be shrunk in order to avoid a horizontal page break. 50=that it can be shrunk up to 50% to avoid a page break; 100=retain original size.	Number	Formula dialog
Minimum	Sets how much height should be available to an object. If less space is available, a	Number	Formula dialog

Height	page break is triggered.			
Columns	Controls the column properties at a page break.		Formula	Formula dialog
Columns	Repeat Labels	Sets whether the row headers are to be repeated at a column wrap.	True False Formula	Yes No Formula dialog
	Break Level	Sets the optimal page break level. "0" represents the innermost group (the bottom row in the "Columns" field of the "Axis definition" tab).	Number Formula	 Formula dialog
	Force	Forces page break after every corresponding group.	True False Formula	Yes No Formula dialog
	Page break on shadow pages	True: If the crosstab contents exceed the available space, the remaining contents will be printed to shadow pages. Shadow pages are additional, automatically inserted pages, which are not counted as pages (receives no page number). False: The remaining contents will be printed below the table.	True False Formula	Yes No Formula dialog
	Distance Before	Distance to the preceding table (void if table starts on page top).	Number Formula	 Formula dialog
Rows		Controls the row properties at a page break.	Formula	Formula dialog
	Repeat Labels	Sets whether the column headings are repeated after a row wrap.	Number Formula	 Formula dialog
	Break Level	Sets the optimal page break level. "0" represents the innermost group (the bottom row in the "Rows" field of the "Axis definition" tab).		
	Force	Forces page break after every corresponding group.		

4.16.5 Crosstab Functions

There are several crosstab functions to access to the values of each data record.

- Crosstab.Value() returns the cell´s content (as number).
- Crosstab.Cells.Avg() returns the average of the cell contents. Only available in crosstab objects.
- Crosstab.Cells.Max() returns the largest value of the cell contents. Only available in crosstab objects.
- Crosstab.Cells.Min() returns the smallest value of the cell contents. Only available in crosstab objects.
- Crosstab.Col$() returns the column header for the cell currently being output. Only available in crosstab objects.
- Crosstab.Col() returns the column index for the cell currently being output. Only available in crosstab objects.
- Crosstab.Row$() returns the row header for the cell currently being output. Only available in crosstab objects.
- Crosstab.Row() returns the row index for the cell currently being output. Only available in crosstab objects.
- Total() can be used for computations over all cells away. Otherwise calculations always run over all values, which concern the respective cell

Additional information can be found in Chapter "List of Available Functions".

4.17 Inserting Formatted Text

With the formatted text object you can, in contrast to normal text objects, change the format within a line. In addition you can also use variables in these objects. To create a formatted text object, select Objects > Insert > Formatted Text (CTRL+F).

When should you use the formatted text object and when the text object? Normal text objects should be given preference for every day jobs, since they contain less information and they are, therefore, printed faster. Consequently you should use a formatted text object in case you either can not realize a special format type with the normal text object or succeed only with great effort.

4.17.1 Properties

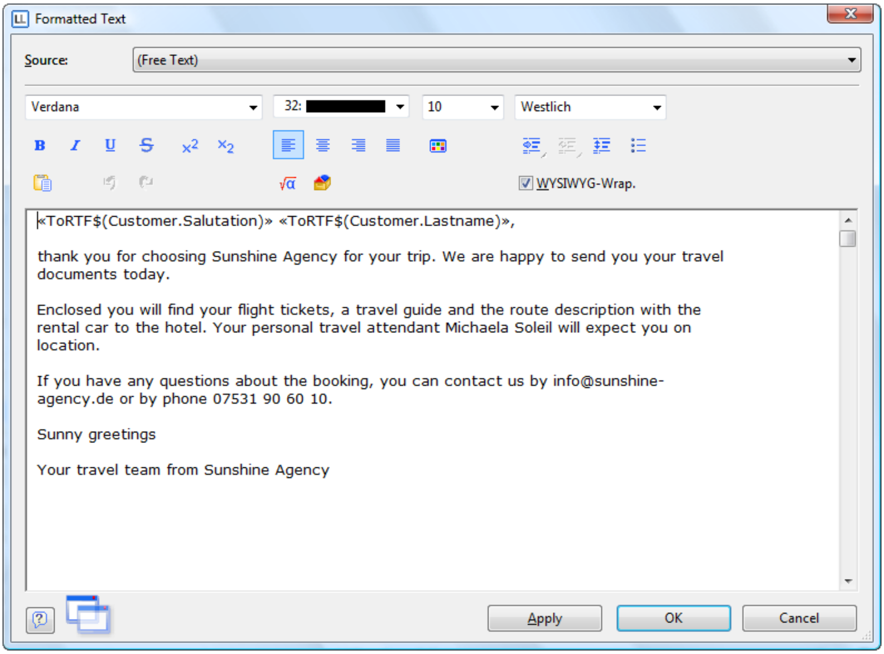
See also Chapter "Common Object Properties".

Property	Description	Value	Description
Rotation	Rotates the object counter clockwise.	0	0°

		1	90°
		2	180°
		3	270°
		Formula	Formula dialog
Page break	Defines if the object can cause a page break, or (in the case of an RTF object) enable another object to print the text exceeding the object's size. With labels, the next label will be started only when all objects with this option have been printed on the previous label. This option may not be available if page break ping is not supported by your application. The option is activated by default. If you wish to print an object on every page of a list project, this option must be deactivated or the object will otherwise only be printed once per project.	True	Yes
		False	No
		Formula	Formula dialog

4.17.2 Text Contents

The contents of formatted text objects can be edited in the Formatted Text Editor.



Source: (Free Text)

«ToRTF$(Customer.Salutation)» «ToRTF$(Customer.Lastname)»,

thank you for choosing Sunshine Agency for your trip. We are happy to send you your travel documents today.

Enclosed you will find your flight tickets, a travel guide and the route description with the rental car to the hotel. Your personal travel attendant Michaela Soleil will expect you on location.

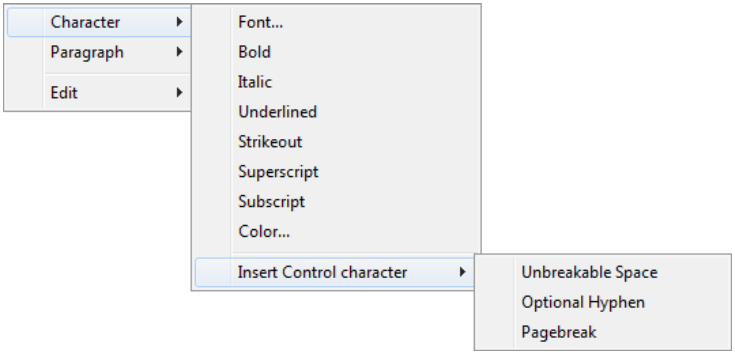
If you have any questions about the booking, you can contact us by info@sunshine-agency.de or by phone 07531 90 60 10.

Sunny greetings

Your travel team from Sunshine Agency

As a source, you may either select an RTF-Variable or "(Free Text)". If you select the latter, an input box and formatting aids are displayed in the dialog, which are explained below. Simply select the text to be formatted and select the respective format tool:

- Select the font, size, color and character set with the combo-boxes.
- Select Left justified, centered, right justified, block and text background color, Indent paragraph (2.5 mm, right Mouse click: indent following lines), Unindent paragraph (2.5 mm, right Mouse click: unindent following lines), Set line and paragraph spacing, Enumeration, Insert from clipboard, Undo action, Repeat action
- By clicking the button "Formula" you have access to the Formula Wizard. An important function of the Formula Wizard is ToRTF, which you can use to convert variables into RTF-text.
- The RTF-Object also supports so called "Embedded Objects", objects that are embedded into the text, for example graphics. These objects can be inserted using the clipboard from, for example, MSPaint.
- ☑ WYSIWYG What you see is what you get: Word wrapping according to object size. This is only an approximate display and minimal differences may occur.
- Tab: ctrl+tab
- A context menu is available in the editor, by which you can open various formatting possibilities for characters and paragraphs.



Exceeding Text Transfer

If the current RTF-object is linked to another RTF-object in which the option page break is activated, the option "exceeding text of ..." is available as data source. If this option is set, input is blocked within the RTF-object since the (rest) text is automatically integrated from the other RTF-object.

4.18 Inserting Form Controls

To use your project for form output and input, different form controls are available with the form control object. To insert a form control object into your project, select Objects > Insert > Form Control.

Form controls can directly be filled by the user in the List & Label preview and HTML or generate actions as e.g. sending an email.

The following determines the basic behavior of the element:

Property	Description	Value	Description
Type	Determines element type.	0	Edit
		1	Checkbox
		2	combo box
		3	Button
Tooltip	Tooltip which should appear.		

Corresponding to the type the properties of the control are changing.

In the following Chapter the different properties of the form control are described according to the chosen type-property:

4.18.1 Type Edit

Property	Description	Value	Description
Force Input	Determines whether input has to be entered by the user.	True	Yes
		False	No
		Formula	Formula dialog
Field Name	Determines the field name of a possibly data export via XML/XFDF.		Formula dialog
Validation Expression	Regular expression to validate input.		Formula dialog
Error Message	Message that is given if validation fails		Formula dialog
Value	Default value for input object.		Formula dialog
Multiline	Determines whether the input field is multilined or not. When multiline is not chosen, more characters can be entered and the input field scrolls automatically. But when printed these characters are cut off.	True	Yes
		False	No
		Formula	Formula dialog

4.18.2 Type Checkbox

Property	Description	Value	Description
Force Input	Determines whether input has to be entered by the user.	True	Yes
		False	No
		Formula	Formula dialog
Field Name	Determines the field name of a possibly data export via XML/XFDF.	Formula	Name
Value	Default value for input object.		Formula dialog

4.18.3 Type Combobox

Property	Description	Value	Description
Force Input	Determines whether input has to be entered by the user.	True	Yes
		False	No
		Formula	Formula dialog
Items	Available default values of the combobox	List	List of default values
Field Name	Determines the field name of a possibly data export via XML/XFDF.	Formula	Name
Validation Expression	Regular expression to validate input (only with variable text)		Formula dialog
Error Message	Message that is given if validation fails		Formula dialog
Editable	Determines if the user can enter different values than the default values.	True	Yes
		False	No
		Formula	Formula dialog
Value	Default value for input object.		Formula dialog

4.18.4 Type Button

Property	Description	Value	Description
Action	Determines the possible performance. Only available with type Button. 0: Send as email. The most relevant fields for sending an email can be preset. 1: Save. By clicking the button, the preview file, respectively the entered data will be	0 1 2	Send as email. Save. Sending via HTTPPost Link

	saved in the corresponding format. 2: Sending via HTTPPost 3: Defines a hyperlink area. The object is transparent and can therefore be displayed above other objects.	3	
Text	Button text	Formula	Formula dialog

4.19 Inserting HTML formatted text

To display HTML formatted text (e.g. web sites) within your projects, you can use the HTML text object. The object supports the HTML 3.2 specification. Partially supported are some extended tags as well as cascading style sheets.

To insert a HTML text object use the toolbar or Objects > Insert > HTML-Text.

The contents of the object are edited in the property list, where you have various options.

4.19.1 Properties

See also Chapter "Common Object Properties".

4.19.2 HTML-Contents

You can define the contents of the object in the HTML-Object Properties dialog.

- File: Select this option to display the contents of a previously saved HTML file. The "Browse..." button opens a file selection dialog. The file must be located on a local storage medium or network.
- URL: This option is used to display websites (e.g. www.combit.net). The contents are downloaded at runtime, thus you'll need a connection to the Internet.
- Field/Variable: If your application makes HTML contents available, these can be selected here. Refer to your application's documentation for details.
- Fit to Object: Select this option to fit the contents into the object. If the option is deactivated, the contents are fit to the object width and output may wrap over several pages.
- Proxy: If your access is via a proxy server, you must enter its address in the settings group. By default, your current proxy configuration will be set. Leave this setting (without editing the configuration) in order to be able to distribute your projects to other systems with a different proxy configuration. The current configuration will always be taken in this mode.

4.20 Inserting OLE-Server Documents

The OLE Container is available for the insertion of OLE-Server documents into your project. This gives you the capability to embed and print documents that were created with other applications, for example Word, Excel, Visio or MapPoint, into your project without changes.

You can create an OLE-Object using **Objects** > **Insert** > **OLE Container**. After defining the size and position of the object on the workspace, the standard "Insert Object" dialog will open. You can select the object type and select either "Create New" or "Create from File".

You can select, for example, an existing Excel file and insert it into the object frame.

Note: Only the first Page of an embedded document can be displayed, as there is no standard for multi-page OLE-Objects.

4.20.1 Properties

See Chapter "Common Object Properties".

4.20.2 Contents

If contents are available the host application will be opened, otherwise the standard "Insert Object" dialog will appear.

4.21 Inserting Form Templates

Form templates are drawings, e.g. scanned pictures, of actual forms that you can place in the background of your workspace to aid in the exact placement of objects. The templates are displayed on the workspace but will not be printed and cannot be edited. To insert a form template in the background of your workspace, use **Objects** > **Insert** > **Form Templates**. After creation, select the form template using the tool window Objects. The template can no longer be selected in the workspace.

> **Tip:** When using a template to create your own form or to fill out a form, size and position of the template are of critical importance. We recommend that the template be placed using the Property List, as this allows a more exact placement than the use of the mouse.

4.21.1 Properties

See also Chapter "Common Object Properties".

Property	Description	Value	Description
File name	Select the file that contains the required form template.	File name	Open file dialog
	Relative Path The path is relative to the	True	Yes

	project path.	False	No
Fade color	The selected color will be added to the template to fade the color.	… ▼	Color dialog Choice of predefined colors and the Formula dialog
Keep proportions	Using the option "Keep Proportions" you can define whether the graphic should be inserted in the correct relationship between height and width (True), or if the graphic should be resized to occupy the complete object (False).		
Visible in Preview	Sets whether the template should be visible in the preview.	True False Formula	Yes No Formula dialog

5. Appendix

5.1 List of List & Label Variables and Fields

List & Label automatically provides, dependent upon the application, several variables and fields (in list projects). You can find the variables and fields in the LL subfolder in the variable list.

5.1.1 List of Variables

Name	Description
LL.Color	Color value of the corresponding color.
LL.Device.Name	Name of the output device(printer).
LL.Device.Page.Name	Name of the output page type (example "A4").
LL.Device.Page.Size.cx	Physical page width of the output device in the project unit. Can be used in formulas to fit objects to larger output formats. Example: Set the property Position.Left in the Property List to the value 0, the property Position.Right to LL.Device.Page.Size.cx. The object will occupy the total page width.
LL.Device.Page.Size.cy	Physical page height of the output device in the project unit.
LL.Device.PrintableArea.Offset.cx	Width of the left non-printable border in the project unit. Example: In the Property List, set the property Position.Left to the value LL.Device.PrintableArea.Offset.cx. The object will always lie exactly on the left edge of the printable area of the printer.
LL.Device.PrintableArea.Offset.cy	Height of the top of the non-printable border in the project unit.
LL.Device.PrintableArea.Size.cx	Printable page width of the output device in the project unit.
LL.Device.PrintableArea.Size.cy	Printable page height of the output device in the project unit.
LL.CountData (not for Multitab projects)	Number of records transferred by the program so far during this printing process. This number also contains the records that were not printed due to filter conditions. This number will increase with every record.
LL.CountDataThisPage	Number of records transferred by the program so far

(not for Multitab projects)	during this printing process on the current page. This number also contains the records that were not printed due to filter conditions. This number will increase with every record.
LL.CountPrintedData (not for Multitab projects)	Number of the actually printed records.
LL.CountPrintedDataThisPage (not for Multitab projects)	Number of the actually printed records on the current page.
LL.CurrentContainerItem	Value of the property "name" for report container elements.For layout region.
LL.CurrentTableColumn	Index of the column in multi-column projects.
LL.IsForcedPage	Shows whether the page has been forced by the "Minimum Page count" project setting.
LL.FilterExpression	Selected project filter.
LL.OutputDevice	Output device. Can be used to assign object specific formats for certain output formats ("HTML", "RTF", "PDF",…)
LL.SortStrategy	Selected sorting can be set, dependent upon the application in the Project Menu.
@LLFAX.RecipName	Fax: Recipient name
@LLFAX.RecipNumber	Fax: Recipient fax number
@LLFAX.SenderBillingCode	Fax: Sender billing code
@LLFAX.SenderCompany	Fax: Sender company
@LLFAX.SenderDept	Fax: Sender department
@LLFAX.SenderName	Fax: Sender name

5.1.2 List of Fields

Name	Description
LL.ChartObject.ArcIndex (only in pie charts)	Index of the actual pie segment (pie chart). The largest pie segment has index 1.
LL.ChartObject.ArcPerc (only in pie charts)	Percentage of the actual pie segment (pie chart).
LL.ChartObject.ArcTotal (only in pie charts)	Value of the total dataset (pie chart).
LL.ChartObject.ArcTypeIsOthers (only in pie charts)	True, if the the actual pie segment is the "Others" pie segment (pie chart).

LL.ChartObject.ArcValue (only in pie charts)	Value of the actual pie segment (pie chart).
LL.ChartObject.AxisCoordinate (only in charts)	Value of the Axis Coordinate (usable for Coordinate label text).
LL.CurrentRelation (only in Multitab projects)	Description of the current relation to the parent table.
LL.CurrentSortOrder (only in Multitab projects)	Description of the current sort order in the table.
LL.CurrentTable (only in Multitab projects)	Name of the currently used table.
LL.CurrentTablePath (only in Multitab projects)	Name of the currently used table (hierarchical with parent tables), e.g. Customers.Orders.Order_Details
LL.FcountData	Number of transmitted records. This number also includes the records that were not printed due to filter conditions.
LL.FcountDataThisPage	Number of transmitted records on the current page. This number also includes the records that were not printed due to filter conditions.
LL.FcountPrintedData	Actual number of printed records.
LL.FcountPrintedDataThisPage	Actual number of printed records on the current page.
LL.Relations.* (only in Multitab projects)	Available relations to define a drill-down link.
LL.Tables.* (only in Multitab projects)	Available tables to define a drill-down link.

5.2 List of Available Functions

In the following table you will find all functions, listed alphabetically, available in List & Label. In the left column the **function names**, in the middle column the number and permitted **value types** of the **arguments** and in the right column, the value type of the **return value**.

Function	Arguments	Return values
Abs	Number	Number
AddDays	Date, Number	Date
AddHours	Date, Number	Date
AddMinutes	Date, Number	Date
AddMonths	Date, Number	Date
AddSeconds	Date, Number	Date
AddWeeks	Date, Number	Date
AddYears	Date, Number	Date
Alias$	String, String [,String]	String
ArcCos	Number [,Number]	Number
ArcSin	Number [,Number]	Number
ArcTan	Number [,Number]	Number
Asc	String	Number
AskString$	String, Boolean, String, Number	String
AskStringChoice$	String [,Boolean[,String[,Number]]]	String
Atrim$	String	String
Avg	Number [,Boolean]	Number
Barcode	String, String	Barcode
Barcode$	Barcode	String
BarcodeType$	Barcode	String
BasedStr$	Number, Number [,Number, [Boolean]]	String
BinaryAND	Number, Number	Number
BinaryNOT	Number	Number
BinaryOR	Number, Number	Number
BinarySHL	Number, Number	Number

BinarySHR	Number, Number	Number
BinaryXOR	Number, Number	Number
BMPMapToGray	Picture|String, Picture	Picture
BMPRotate	Picture|String, Number [,Number]	Picture
Case$	Number, String [,String]	String
Ceil	Number	Number
Century	Date [,Boolean]	Number
CheckMod10	String	Number
Chr$	Number [,Number]	String
ChrSubst$	String, String [,String]	String
Cond	Boolean, All, All	All
Constant.Pi	-	Number
Contains	String, String	Boolean
Continued	-	Boolean
Cos	Number	Number
Count	All [,Boolean]	Number
CountIf	Boolean [,Boolean]	Number
Crosstab.Cells.Avg	[Boolean [,Number[,Number]]]	Number
Crosstab.Cells.Max	[Boolean [,Number[,Number]]]	Number
Crosstab.Cells.Min	[Boolean [,Number[,Number]]]	Number
Crosstab.Col$	[Number]	String
Crosstab.Col	[Boolean]	Number
Crosstab.Row$	[Number]	String
Crosstab.Row	[Boolean]	Number
Crosstab.Value	-	Number
Cstr$	Number, String	String
Date	String	Date
Date$	Date, [String, [String]]	String
DateDiff	Date, Date	Number
DateDiff$	Date, Date	String
DateHMS	Number, Number, Number	Date

DateInLeapYear	Date	Boolean
DateInRange	Date, Date, Date	Boolean
DateToJulian	Date	Number
DateYMD	Number, Number, Number	Date
Day	Date	Number
Day$	Date	String
Decade	Date [,Boolean]	Number
Distinct	All	All
Dow	Date	Number
Dow$	Date	String
Picture	String	Picture
Drawing$	Picture	String
DrawingHeightSCM	Picture	Number
DrawingWidthSCM	Picture	Number
Empty	String	Boolean
Evaluate	String	All
Even	Number	Boolean
Exists	String	Boolean
Exp	Number	Number
Exp10	Number	Number
FirstHeaderThisTable	-	Boolean
Frac	Number	Number
FStr$	Number, String	String
GeometricAvg	Number [,Boolean]	Number
GetValue	String	All
Hour	[Boolean]	Number
HSL	Number, Number, Number	Number
Hyperlink$	String, String, Boolean	String
If	Boolean, All [,All]	All
Int	Number	Number
IsNULL	All	Boolean

IssueIndex$	-	String
Join$	String [,String, [Number, [Boolean]]]	String
JulianToDate	Number	Date
LastFooterThisTable	-	Boolean
LastPage	-	Boolean
Left$	String, Number [,Boolean]	String
Len	String	Number
LoadFile$	String [,String]	String
Locale$	Number[,String]	String
LocCurr$	Number[,String]	String
LocCurrL$	Number[,String]	String
LocDate$	Date, [,String, [Number]]	String
LocDateTime	String [,String]	Date
LocNumber$	Number[,String]	String
LocTime$	Date, [,String, [Number]]	String
LocVal	String [,String]	Number
Log	Number	Number
Log10	Number	Number
Lower$	String	String
Ltrim$	String	String
Max	Number | Date	Number | Date
Maximum	Number [,Boolean]	Number
Median	Number [,Boolean]	Number
Mid$	String, Number [,Number]	String
Min	Number | Date	Number | Date
Minimum	Number [,Boolean]	Number
Minute	[Boolean]	Number
Month	Date	Number
Month$	Date	String
Now	-	Date
NthLargest	Number, Number [,Boolean]	Number

NthLargestIndex	Number, Number [,Boolean]	Number
NthValue	All, Number [,Boolean]	All
NULL	-	All
NULLSafe	All [,All]	All
NumInRange	Number, Number, Number	Boolean
Odd	Number	Boolean
Ord	String	Number
Page	-	Number
Page$	-	String
Pow	Number, Number	Number
Previous	All	All
PreviousUsed	All	All
ProjectParameters$	String [,Boolean]	String
ProjectPath$	[Boolean]	String
Quarter	Date [,Boolean]	Number
RainbowColor	Number, Number, Number	Number
RegExMatch$	String, String, Number	String
RemainingTableSpace	[Boolean]	Number
Rep$	String, Number	String
RGB	Number, Number, Number	Number
Right$	String, Number [,Boolean]	String
Round	Number, Number	Number
Rtrim$	String	String
Second	[Boolean]	Number
Sin	Number	Number
Sqrt	Number	Number
StdDeviation	Number [,Boolean]	Number
Str$	Number | Date [,Number [,Number]]	String
StrPos	String, String [,Number]	Number
StrRPos	String, String [,Number]	Number
StrSubst$	String, String [,String]	String

Sum	Number [,Boolean]	Number
Tan	Number	Number
Time$	String	Number
Today	-	Date
Token$	String, Number, String [,String]	String
ToRTF$	String	String
Total	All	All
TotalPages$	-	String
UnitFromSCM	Number	Number
Upper$	String	String
Val	String	Number
Variance	Number [,Boolean]	Number
Woy	Date, [Number]	Number
Year	Date	Number
Year$	Date	String

Abs

Purpose:

Calculate the absolute value of a number. A negative value will be returned as positive and a positive value will remain unchanged.

Parameter:

Number

Return value:

Number

Example:

Abs(-3) = 3
Abs(3.12) = 3.12

AddDays

Purpose:

Adds the given number of days to the date, or subtracts the number of days when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddHours

Purpose:

Adds the given number of hours to the date, or subtracts the number of hours when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddMinutes

Purpose:

Adds the given number of minutes to the date, or subtracts the number of minutes when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddMonths

Purpose:

Adds the given number of months to the date, or subtracts the number of months when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddSeconds

Purpose:

Adds the given number of seconds to the date, or subtracts the number of seconds when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddWeeks

Purpose:

Adds the given number of weeks to the date, or subtracts the number of weeks when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddYears

Purpose:

Adds the given number of years to the date, or subtracts the number of years when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

Alias$

Purpose:

Returns the value that is specified for the key (first parameter) in the key/value-pairs (second parameter).

Parameter:

String	Expression for the value to be searched.
String	List of values (Form: <key=value>| [<key=value>]. To be able to use "|" or "=" in the value or key, place a "\" infront of it.

String (optional) Default if the value cannot be found.

Return value:

String

Example:

Alias$("USA", "DEU=Deutschland|USA=United States of America|GB=United Kingdom") Result: United States of America

ArcCos

Purpose:

Calculates the arccosine of the value.

Parameter:

Number	Value
Number	(optional) Mode (0=Degree, 1=Radian). Default: 0.

Return value:

Number

Example:

ArcCos (0) Result: 90

ArcSin

Purpose:

Calculates the arcsine of the value.

Parameter:

Number	Value
Number	(optional) Mode (0=Degree, 1=Radian). Default: 0.

Return value:

Number

Example:

ArcSin (0.5) Result: 30,00

ArcTan

Purpose:

Calculates the arccotangent of the value.

Parameter:

Number	Value
Number	(optional) Mode (0=Degree, 1=Radian). Default: 0.

Return value:

Number

Example:

ArcTan (1) Result: 45,00

Asc

Purpose:

Returns the ASCII-Code of the first character of the string.

Parameter:

String

Return value:

Number

Example:

Asc("A") Result: 65

AskString$

Purpose:

With this function, information can be requested from the user during printing. A typical example of use for this function would be in a project for a bank transfer form. Information that remains constant, such as name and bank details of the sender, can be integrated directly into the project as fixed text or variables. The transfer amount, however, will almost always be different. With the function AskString$(), this information can be requested from the user during printing.

At print time, a dialog will appear in which the needed information can be entered.

The dialog allows the entered value to be carried over. Abort with "Cancel".

With the button "All", the entered value will be automatically used for all future result for the AskString$ function during thus print job. This is useful when the value remains constant over all records.

Parameter:

String	The first parameter contains some descriptive text that will appear in the dialog. Since this is a formula, fixed text must be entered in quotation marks, for example "Transfer amount:”. This first parameter must be entered, all remaining parameters are optional. If no other parameter(s) is/are entered, the first string is also the default setting for the user input.

Boolean	(optional) The second parameter allows you to define whether the dialog should be shown once prior to printing (default, FALSE), or if the dialog should be shown for each record (TRUE).
String	(optional) The third parameter contains the string that appears as the recommended value for the user input. Since this is a formula, fixed text must be entered in quotation marks, for example "50.00 USD".
Number	(optional) The last parameter defines the number of characters that can be entered by the user. A value of 16, for example, allows the user to enter a maximum of 16 characters.

Return value:

String

Example:

AskString$("Transfer amount",True,"50.00 USD",16)

Opens a dialog with the title "Transfer amount”, a recommended value of "50.00 USD” and a maximum of 16 characters. Since the second parameter is TRUE, the dialog will be shown for each record to be printed.

AskStringChoice$

Purpose:

Prompts the user to choose a value for the specified variable from a combobox at print time.

Parameter:

String	Text, which is displayed and should specify what is to be entered.
Boolean	(optional) Sets whether the dialog should be shown once prior to printing (default, FALSE), or if the dialog should be shown for each record (TRUE).
String	(optional) The combobox entries. The single entries of the combo box are separated by "|". If one of the entries is '***' (three asterisks), the text is editable. So a new value which may be different from the list items can be entered.
Number	(optional) Maximum length (Default: 8192 characters).

Return value:

String

Example:

AskStringChoice$ ("Document type".F.,"Offer|Invoice|Delivery note|***)

ATrim$

Purpose:

Removes spaces from the beginning and end of a string.

Parameter:

String

Return value:

String

Example:

Atrim$("   combit GmbH   ")   Result: combit GmbH

Avg

Purpose:

Generates the mean of the set of values that is produced by the first argument.

Parameter:

Number	Expression of the value to be calculated.
Boolean	(optional) TRUE: After the output, the values which were stored for the calculation are deleted. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

Avg(Order_Details.Quantity*Order_Details.UnitPrice)

Barcode

Purpose:

This function converts a string to a barcode. This function can only be used in lists.

Parameter:

String	Barcode value (contents)
String	Barcode type. The possible barcode types will be listed by the auto-complete function of the wizard. If the barcode cannot be correctly interpreted it will not be printed. Some barcodes require special formats that must be used. Further information can be found in Chapter "List of Available Barcodes".

Return value:

Barcode

Example:

Barcode(Upper$(Name),"3of9")

Barcode$

Purpose:

Returns the text contents of a barcode.

Parameter:

Barcode

Return value:

String

Example:

Barcode$(BC_3OF9) Result: "Item 4711"

BarcodeType$

Purpose:

Returns the type of the barcode as a string.

Parameter:

Barcode

Return value:

String

BasedStr$

Purpose:

Returns the value to any radix.

Parameter:

Number	Value.
Number	Radix (2 to 36).
Number	(optional) Minimum length of the string (without optional prefix). 0 for the minimal length (Default).
Boolean	(optional) Defines if a prefix ('0b' for radix 2, '0o' for radix 8, '0x' for radix 16) is inserted before the string (Default: False).

Return value:

String

Example:

BasedStr$ (1,2,1,True) Result: 0b1

BinaryAND

Purpose:

Links the two (integer) parameters binary with 'and' and returns the result.

Parameter:

Number	Value.
Number	Value.

Return value:

String

Example:

BinaryAND (01,10)	Result: 0
BinaryAND (10,11)	Result: 10

BinaryNOT

Purpose:

Negates the value binary and returns the result.

Parameter:

Number	Value.

Return value:

String

Example:

BinaryNOT (10)	Result: 5	(ten equivalent 1010, five equivalent 0101)

BinaryOR

Purpose:

Links the two (integer) parameters binary with 'or' and returns the result.

Parameter:

Number	Value.
Number	Value.

Return value:

String

Example:

BinaryOR (01,10)	Result: 11
BinaryOR (10,11)	Result: 11

BinarySHL

Purpose:

Shifts the value binary to the left.

Parameter:

Number	Value.
Number	Number of bits, the value is shifted.

Return value:

String

Example:

BinarySHL (10,1)	Result: 20

BinarySHR

Purpose:

Shifts the value binary to the right.

Parameter:

Number	Value.
Number	Number of bits, the value is shifted.

Return value:

String

Example:

BinarySHR (10,1)	Result: 0,00

BinaryXOR

Purpose:

Links the two (integer) parameters binary with 'exclusive or' and returns the result.

Parameter:

Number	Value.
Number	Value.

Return value:

String

Example:

BinaryXOR (01,10)	Result: 11
BinaryXOR (10,11)	Result: 1

BMPMapToGray

Purpose:

Converts the picture to greyscales.

Parameter:

Picture or String

Return value:

Picture

Example:

BMPMapToGray ("sunshine.gif")

BMPRotate

Purpose:

Rotates a picture by the given degree.

Parameter:

Picture or String

Number	Rotation angle
Number	(optional) Mode (0=Degree, 1=Radian)

Return value:

Picture

Example:

BMPRotate(Article.Picture,90)

Case$

Purpose:

Converts a number, dependant upon the value, into a string. Assignment is made with a formatting string that contains the replacement string for the number values in ascending order.

Parameter:

Number	Number to be converted (n). The n-th value of the character string will be copied to the return value string. If enough values do not exist, the character string will remain empty.
String	Collection of strings separated by a particular character. If a third parameter does not exist, this is the "|" character, otherwise the first character of this parameter.
String	(optional) Separator for the formatting string (default: "|")

Return value:

String

Example:

Case$(Page(),"0|I|II|III|IV|V|VI|VII|VIII|IX|X")

Result: "III", if Page() = 3

Ceil

Purpose:

Calculates the next bigger integer based on the given value. See also function Floor().

Parameter:

Number	Value.

Return value:

Number

Example:

Ceil(5.6)	Result: 6

Century

Purpose:

Returns the century of the date.

Parameter:

Date	
Boolean	(optional) sets whether the calculation should be carried out 'simply' (century starts with year 0) or 'historically' (century starts with year 1). Default: False

Return value:

Number

Example:

Str$(Century(Today()),0,0)	Result: 21
Str$(Century(Date("01.01.2000")),0,0)	Result: 20
Str$(Century(Date("01.01.2001")),0,0)	Result: 21
Str$(Century(Date("01.01.2000"),.T.),0,0)	Result: 21

CheckMod10

Purpose:

Calculate the modulo 10 checksum digit of the string.

Parameter:

String A string of digits.

Return value:

Number

Chr$

Purpose:

Converts a number to a character. This character has the entered number as it's ASCII-Code. For multibyte character sets, the highword is the lead byte, for Unicode, the value is the Unicode code point.

Parameter:

Number	
Number	(optional) Defines the type of the parameter. 0=multibyte character sets, 1=Unicode. Default is dependent on the List & Label dll used.

Return value:

String

Example:

Chr$(64) Result: "@"

ChrSubst$

Purpose:

Searches a character string for a string that is contained in the second parameter. Every occurrence of this string will be replaced by the string defined in the third parameter. If no third parameter exists, the strings will be removed.

Parameter:

String	
String	
String	(optional)

Return value:

String

Example:

ChrSubst$("Otto", "Oo", "_")	Result: "_tt_"
ChrSubst$("Normalconsumer", "aeiou","??")	Result: "N??rm??lc??ns??m??r"
ChrSubst$("Normalconsumer", "aeiou")	Result: "Nrmlcnsmr"
ChrSubst$("3.1415926535",".",",")	Result: "3,1415926535"

Cond

Purpose:

Allows to define conditions. The first parameter is a logical expression that will be evaluated as "True" of "False". If the first expression is "True", the second expression will be returned as the result. If the expression is "False", the third expression will be returned as the result. If no third expression is entered, the return value will assume the following standard values, dependent upon its type:

2. Argument Type	Return value if expression = False
Boolean	False
String	"" (empty String)
Date	Julian Date value 0
Number	0
Picture	"" (empty String)
Barcode	"" (empty String)

Parameter:

Boolean

All

All (optional) The third parameter must be the same type as the second parameter.

Return value:

All

Example:

Cond(COUNTRY<>"USA",COUNTRY_LONG)

Cond(PRICE=0,"on request",Str$(PRICE,0,2))

Cond(empty(COMPANY),SALUTATION,COMPANY)

Constant.Pi

Purpose:

Returns the value of Pi.

Parameter: -

Return value:

Number

Example:

Constant.Pi() Result: 3,14159 (depending on the number of decimals)

Contains

Purpose:

Evaluates if a string contains another string (second parameter).

Parameter:

String

String

Return value:

Boolean

Example:

Contains("Itemnumber: 12345", "1234") Result: True

Continued

Purpose:

Indicates that a text or RTF-object had a page break. This means that the current page is a result of the page break.

Parameter:

-

Return value:

Boolean

Cos

Purpose:

Calculates the cosine of the value.

Parameter:

Number	Value
Number	(optional) Mode (0=Degree, 1=Radian). Default: 0.

Return value:

Number

Example:

Cos (90) Result: 0

Count

Purpose:

Counts the number of values of the first argument. With this function, all NULL values in the argument are included in the count. Use the CountIf() function when you want to disregard NULL values.

Hint: Sum variables (see "Working with Sum Variables") are an alternative way of creating counters. Sum variables are principally applicable to whole tables. Aggregate functions principally table specific.

Parameter:

All	Values to count (sets the value to count). Needed to define the table (subtable) for which the records shall be counted.
Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values will be already deleted within the table.

Return value:

Number

Example:

Count(Order_Details.ProductID)

NthLargest(Article.Price,Count(Distinct(Artikel.Stkpreis),True)-1, True)

Calculates the second-smallest value, only taking repeated values into account once.

CountIf

Purpose:

Counts the number of values that comply with the condition. Use the function Distinct() when repeated values are only to be counted once.

Parameter:

Boolean	Expression for the comparison.
Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values will be already deleted within the table.

Return value:

Number

Example:

CountIf(Customers.Region="EMEA")

CountIf(Distinct(Customers.Region="EMEA")) counts repeated values once only.

CountIf (IsNULL (Orders.OrderDate)) counts all values with empty fields.

Crosstab.Cells.Avg

Purpose:

Returns the average of the cell contents. Only available in crosstab objects.

Parameter:

Boolean	(optional) True: Only defined values are entered into the calculation (Default: False). Defined values: if you are, for example, analyzing customers and quarters, the quarters without turnover constitute an undefined value and can be treated separately in the calculation.
Number	(optional) Row layer (0= bottom layer or innermost group, 1= next lowest, ...). Default: 0.
Number	(optional) Column layer (0= bottom layer or innermost group, 1= next lowest, ...). Default: 0.

Return value:

Number

Crosstab.Cells.Max

Purpose:

Returns the largest value of the cell contents. Only available in crosstab objects. For the parameters and their meaning, see function Crosstab.Cells.Avg().

Crosstab.Cells.Min

Purpose:

Returns the smallest value of the cell contents. Only available in crosstab objects. For the parameters and their meaning, see function Crosstab.Cells.Avg().

Crosstab.Col$

Purpose:

Returns the column header for the cell currently being output. Only available in crosstab objects.

Parameter:

Number	(optional) Column layer (0= lowest layer or innermost group, 1= next lowest, ...). Default: 0.

Return value:

String

Crosstab.Col

Purpose:

Returns the column index for the cell currently being output. Only available in crosstab objects.

Parameter:

Boolean	(optional) True: layer (only cells in this layer count), Default: False.

Return value:

Number

Crosstab.Row$

Purpose:

Returns the row header for the cell currently being output. Only available in crosstab objects.

Parameter:

Number	(optional) Row layer (0= lowest layer or innermost group, 1= next lowest, ...). Default: 0.

Return value:

String

Crosstab.Row

Purpose:

Returns the row index for the cell currently being output. Only available in crosstab objects.

Parameter:

Boolean	(optional) True: layer (only cells in this layer count), Default: False.

Return value:

Number

Crosstab.Value

Purpose:

Returns the cell's content.

Parameter: -

Return value:

Number

CStr$

Purpose:

Formats a number according to a format character string. This is identical to the formatting information for the function printf() in the language C. The first parameter is a number of double precision, and the conversion operator can assume i.e. the following values: 'f', 'g', 'G', 'e', 'E'.

Parameter:

Number	
String	format string in C-notation, i.e. '%<format>f'.

Return value:

String

Example:

CStr(Pi,"%5.1f")	Result: " 3.1"
CStr(100*Pi,"nun: %g")	Result: "nun: 3.141593e+02"

Date

Purpose:

Converts a string to a date.

- If the string Contains a dot ".", it will be read in the "d.m.y" format (German).
- If the string contains a diagonal slash "/", it will be read in the "m/d/y" format (US English).
- If the string contains a dash "-", if will be read in the "y-m-d" format (ANSI).
- If the input cannot be correctly interpreted, then the date represents a value that is larger than all other values, (1e100). The return value can be evaluated for correctness using "JulianToDate(1e100)".
- When one or two digits represent the year, all values under 30 will be applied to the 21st century (20xx) and all values over 30 will be applied to the 20th century (19xx).

Parameter:

String

Return value:

Date

Example:

Date("17.10.2007")
Date("10/17/2007")
Date("2007-10-17")

Date$

Purpose:

Converts a date, using a format string, into an appropriately formatted string.

Composition of the format string: this is a normal string into which placeholders can be embedded.

Place holder	Description
%d	Day (1..31)
%<n>d	Day to <n> digits
%0<n>d	Day to <n> digits, filled on left with '0's
%w	Weekday (1..7)
%<n>w	Weekday to <n> digits
%0<n>w	Weekday to <n> digits, filled on left with '0's
%m	Month (1..12)
%<n>m	Month to <n> digits
%0<n>m	Month to <n> digits, filled on left with '0's
%y	Year
%<n>y	Year, to <n> digits
%0<n>y	Year, to <n> digits, filled on left with '0's
%D	Weekday, written out
%M	Month, written out
"%e", "%<n>e"	Year in the local calendar (Japan: Emperor's year)
"%g", "%<n>g"	Era of the local calendar (Japan: Emperor’s era)
"%g", "%1g"	Single letter, Latin letters
"%gg", "%2g"	Single letter, localized
"%ggg", "%3g"	Long name, localized
"%gggg", "%4g"	Long name, Latin letters
"%x"	Localized date, short form

As long as one of the above formats is used, the optional third parameter can be used to set the locale. If the second parameter contains a valid ISO-Country code, the third parameter can be used to set either the short "0" or long "1" format. See also Chapter "List of ISO 3166 Country Codes".

Parameter:

Date	Value to be formatted.
String	(optional) Format description or country code.
String	(optional) Country code or date format.

Return value:

String

Example:

Date$(Today(),"Date: %D, %d/%m/%y") Result: "Date: Thursday, 8/11/2007"

Date$(Today(),"%2wthWeek; %D, %2d/%2m/%4y") Result: "45th Week, Thursday, 8/11/2007"

Date$(Today(),"%D, %3d/%02m/%4y") Result: "Thursday, 8/11/2007"

DateDiff

Purpose:

Returns the difference between two dates in days.

Parameter:

Date	First date value
Date	Second date value

Return value:

Number

Example:

DateDiff(Date("01.01.2009"),Date("01.03.2009")) Result: 59

DateDiff$

Purpose:

Returns the difference between two dates in days as string.

Parameter:

Date	First date value
Date	Second date value
String	(optional) Format

Return value:

String

Example:

DateDiff$(Date("01/01/2009"),Date("03/01/2009")) Result: 2 Months

DateHMS

Purpose:

Converts three numbers for hour, minute and second into a date.

Parameter:

Number	Hour
Number	Minute
Number	Second

Return value:

Date

DateInLeapYear

Purpose:

Checks if the given date is in leap year or not. The calculation is made according to the proleptic grgorian calendar.

Parameter:

Date

Return value:

Boolean

Example:

DateInLeapYear("01.01.2012") Result: True

DateInRange

Purpose:

Evaluates if the date falls within the entered time interval:

Minimum Date: JulianToDate(0)

Maximum Date: JulianToDate(1e100)

Parameter:

Date	Date to be evaluated.
Date	Lower limit of the test interval.
Date	Upper limit of the test interval.

Return value:

Boolean

Example:

DateInRange(Date("2007.10.20"),Date("2007.2.29"),Today()) Result: True

DateToJulian

Purpose:

Calculates the Julian value of a date. Each day (even those in the past) are assigned a unique number.

Parameter:

Date

Return value:

Number

Example:

DateToJulian(Today()) Result: 2453992

DateYMD

Purpose:

Converts three numbers for day, month and year into a date.

Parameter:

Number	Year
Number	Month
Number	Day

Return value:

Date

Example:

DateYMD(2009, 11, 1) Result: 01.11.2009

Day

Purpose:

Determines the day (1...31) of the month and returns it as a number.

Parameter:

Date

Return value:

Number

Example:

Day(Date("17.10.2009"))	Result: 17

Day$

Purpose:

Determines the day (1…31) of the month of a date and returns it as a string.

Parameter:

Date

Return value:

String

Example:

Day$(Date("17.10.2009"))	Result: "17"

Decade

Purpose:

Returns the decade of the date. Hint: The value is always relative to the start of the century (1..10)!

Parameter:

Date	
Boolean	(optional) sets whether the calculation should be carried out 'simply' (decade starts with year 0) or 'historically' (decade starts with year 1). Default: False

Return value:

Number

Example:

Str$(Decade(Date("01.01.2009")),0,0)	Result: 1
Str$(Decade(Date("01.01.2000")),0,0)	Result: 10
Str$(Decade(Date("01.01.2000"),.T.),0,0)	Result: 1

Distinct

Purpose:

Affects the higher order aggregate function (e.g. Sum(), Avg(), Count()...) and causes equal values only to be used once in the calculation.

Parameter:

All

Return value:

All

Example:

CountIf(Distinct(Customers.Region="EMEA"))

Dow

Purpose:

Returns the day of the week to a number(1...7), 1=Sunday, 2=Monday, ...

Parameter:

Date

Return value:

Number

Example:

Dow(Date("01.01.2010")) Result: 4 (Wednesday).

Dow$

Purpose:

Returns the day of the week as a string in accordance with the country settings, "Sunday", "Monday", ...

Parameter:

Date

Return value:

String

Example:

Dow(Date("01.01.2010")) Result: "Friday"

Drawing

Purpose:

Converts a string type file path into a picture.

Parameter:

String

Return value:

Drawing

Drawing$

Purpose:

Converts a picture into a string type file path.

Parameter:

Picture

Return value:

String

DrawingHeightSCM

Purpose:

Returns the height of the Picture in SCM units (1/1000mm).

Parameter:

Picture

Return value:

Number

DrawingWidthSCM

Purpose:

Returns the width of the Picture in SCM units (1/1000mm).

Parameter:

Picture

Return value:

Number

Empty

Purpose:

Evaluates if a string is empty. If it is empty, "True” will be the return value, otherwise "False”. Useful, for example, to determine if the field "ADDRESS” is empty, and if it is, in combination with the IF-THEN-ELSE condition cond(), either print the contents of the field "ADDRESS” or "POBOX”.

The third parameter allows the removal of leading and trailing spaces. If this is evaluated as "True”, a string consisting only of multiple spaces will be recognized as empty.

Parameter:

String

Boolean (optional)

Return value:

Boolean

Example:

Empty("xyz") Result: False

Empty("") Result: True

Evaluate

Purpose:

Evaluates the expression passed as parameter.

Parameter:

String

Return value:

All

Example:

Str$(Evaluate("3*4"),0,0) Result: 12

Str$(Evaluate("4-3"),0,0) Result: 1

Even

Purpose:

Evaluates if a number is even. If the number is even, "True” will be returned, otherwise "False”.

Parameter:

Number

Return value:

Boolean

Example:

"Page number "+Cond(Even(Page()),"even","odd")

Exists

Purpose:

Checks if a variable or field is defined. Is often used in connection with GetValue().

Parameter:

String

Return value:

Boolean

Example:

Exists("CustomerID") Result: False

If(Exists("Customer.Status"),Evaluate("Customer.Status"),"no customer status")

Exp

Purpose:

Calculates the exponential (e^x).

Parameter:

Number

Return value:

Number

Example:

Exp(3)	Result: 20.08553692

Exp10

Purpose:

Calculates 10 raised to the power of number (10^x).

Parameter:

Number

Return value:

Number

Example:

Exp10(3)	Result: 1000

FirstHeaderThisTable

Purpose:

Returns whether the header of the table is being output for the first time. The function can be used as an appearance condition for the header to prevent it being printed more than once if the table continues onto the next page due to space limitations. The header is then only printed at the beginning of the table.

Parameter:

-

Return value:

Boolean

Floor

Purpose:

Calculates the next smaller integer based on the given value. See also function Ceil().

Parameter:

Number

Return value:

Number

Example:

Floor(5.6)   Result: 5

Frac

Purpose:

Calculates the fractional part of a number

Parameter:

Number

Return value:

Number

Example:

Frac(Pi)   Result: 0.1415926535

FStr$

Purpose:

Formats a number according to the format string.

These consist of the following characters ("if negative" refers to the value to be formatted):

*	Digit or '*'-Prefix
$	Local currency symbol
-	Digit or sign, if negative
+	Digit or sign
(	Digit or '('-Prefix if negative
)	')'-Postfix if negative
#	Digit or space prefix
&	Digit or '0'
.	Decimal point
,	Comma, or space prefix

A prefix is a sign that precedes a value, when needed. The expression `FStr$(1, "***")` results in "`**1`". The value "1" is preceded by the characters "**".

A Postfix is a character that, when needed, is placed after a number.

These formatting characters can be combined as needed. If the number is too large for the desired format, a "*" string will be returned.

With the third (optional) parameter, additional formatting can be accomplished.

Value	Description
1	Removal of leading spaces. The use is similar to the functions RTrim$() and LTrim$().
2	Empty string if value NULL.
3	Removal of leading spaces and empty strings when value is 0

Parameter:

Number	
String	Format string
Number	(optional) Additional formatting

Return value:

String

Example:

FStr(3.142, "#")	Result: "3"
FStr(5003.1,"#,###.&&")	Result: "5.003,10"
FStr$(3.142,"#.###")	Result: "3,142"
FStr$(3.142,".######")	Result: "*******"
FStr$(3.142,"(#.###)")	Result: " 3,142 "
FStr$(-3.142,"(#.###)")	Result: "(3,142)"
FStr$(3.142,"+#.###")	Result: "+3,142"
FStr$(3.142,"-#.###")	Result: " 3,142"
FStr$(-3.142,"-#.###")	Result: "-3,142"
FStr$(3.142,"&&&.&&&")	Result: "003,142"
FStr$(3.142,"***.***")	Result: "**3,142"
FStr$(3.142,"$$$.$$$")	Result: "$$3,142"
FStr$(3.142,"###.***")	Result: " 3,142"
FStr$(5003.1,"#,###.&&")	Result: "5.003,10"
FStr$(3.142,"#####")	Result: " 3"

GeometricAvg

Purpose:

Calculates the geometric average of the set of values that result from the first parameter / formula.

Parameter:

Number	Expression for the value to be averaged.
Boolean	(optional) True: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

GetValue

Purpose:

Returns the value of a variable or field. Is often used in connection with Exists().

Parameter:

String

Return value:

All

Example:

Str$(GetValue("Customers.CustomerID"),0,0)   Result: 1234

If(Exists("Customer.Status"),Evaluate("Customer.Status"),"no customer status")

Hour

Purpose:

Determines the hour of the date and returns it in number format. If the parameter is not used, the hour of the print time will be returned.

Parameter:

Date   (optional)

Return value:

Number

Example:

A condition can evaluate if the current hour has the value "10”. The value of the current hour must be determined and then compared to the value "10”.

Hour()=10

HSL

Purpose:

Calculates a color value in the HSL color space (Hue, Saturation, Lightness)

Parameter:

Number	Hue [0-360] (0°=red, 120°=green, 240°=blue)
Number	Saturation [0-1]
Number	Lightness [0-1] (0=no lightness, 1=full lightness)

Return value:

Number

Hyperlink$

Purpose:

The function Hyperlink$ creates a hyperlink text that can be inserted using an export module. When available, the hyperlink will only be embedded if the third parameter returns a result of "True” (=default).

To optionally use a hyperlink only on a HTML page, you need to use the List & Label variable LL.OutputDevice:

Hyperlink$("combit","http://www.combit.net",LL.OutputDevice="HTML")

If an object text contains the character string:

<!--begin:hyperlink="target"-->"Display text"<!--end:hyperlink-->

then a hyperlink will be automatically created in the exported HTML page. The hyperlink function automatically creates a string with the correct syntax.

Parameter:

String	Text
String	Hyperlink
Boolean	(optional) Embedded

Return value:

String

Example:

Hyperlink$("combit","http://www.combit.net")

If

see Cond

IssueIndex

Purpose:

Returns the Issue Index (1..) for display and layout region conditions, if multiple issues are selected in the project parameters

Parameter:

Number

Return value:

Number

Int

Purpose:

Calculates the integer value of a number. The value will be truncated.

Parameter:

Number

Return value:

Number

Example

Int(3,1)Result: 3

IsNULL

Purpose:

Checks whether the transferred value or the result of the expression is NULL, e.g. an empty date field.

Parameter:

All

Return value:

Boolean

Join$

Purpose:

Collection of strings separated by a particular character.

Parameter:

String	Collection of strings separated by a particular character.
String	(optional) Separator for the formatting string (default: ";")

Number	(optional) Maximum number of values ('...' will be appended). Default: all values.
Boolean	(optional) True: The values which were stored are deleted after output. (Default: True). Please note that the stored values are generally deleted for every (sub)table end. The second parameter only decides whether the values will be already deleted within the table.

Return value:

String

JulianToDate

Purpose:

Interprets a number as a Julian date (each day is assigned a unique number) and returns the appropriate date.

Parameter:

Number

Return value:

Date

Example:

JulianToDate(2454347)	Result: 2007.09.04

LastFooterThisTable

Purpose:

Returns whether the footer of the current table is being output for the last time. This function can be used as an appearance condition for the footer, in order to prevent the footer being printed if the table is continued on the next page due to space limitations. The footer is then only printed on the last page of the table.

Parameter:

-

Return value:

Boolean

Lastpage

Purpose:

Returns if the current page is also the last page. This function can only be used in the footer lines of tables, in objects linked with tables or in the layout regions condition! In all other cases, the result of Lastpage() is always False.

Parameter:

-

Return value:

Boolean

Example:

Cond(Lastpage(),"Total sum","Subtotal")

Left$

Purpose:

Reduces a string from the right so that only the number of characters set under Number remain. If the original string is already small enough, it is not affected.

Parameter:

String	The value to be shortened
Number	maximum number of positions of the result
Boolean	(optional) True: The cut off value is ended with "..." (Default: False). With numbers < 3 the setting is ignored.

Return value:

String

Examples:

If you had a customer database that contains, amongst other things, the field NAME for the surname. You now wish to search for all customers whose surname starts with "C". To do this, you must first identify the starting letters.

Left$(NAME, 1)	Result: the first letter of the NAME string.
Left$("combit", 2)	Result: "co"
Left$("combit", 4,.T.)	Result: "c…"
Left$("combit", 2,.T.)	Result: "co"

Len

Purpose:

Returns the number of characters in a string.

Parameter:

String

Return value:

Number

Example:

Len("1234"+"12") Result: 6

LoadFile$

Purpose:

Outputs the contents of the file as a character string.

Parameter:

String	
String	(optional) When the file is not available, the value set here is used.

Return value:

String

Example:

LoadFile$("C:\log.txt","File not found!")

Locale$

Purpose:

Returns information about the country settings, for example currency, decimals, separators, language and country code. The code for the appropriate country is entered in the second parameter, if no second parameter is used the default country settings will be used.

Parameter:

Number	Index of Locale Entry
String	(optional) Country code

Return value:

String

Example:

Locale$(42,"USA") Result: "Monday"

Possible constants for index entry:
http://msdn.microsoft.com/en-us/library/bb507201.aspx

See also Chapter "List of ISO 3166 Country Codes".

LocCurr$

Purpose:

Returns a string with the valid currency format without the currency symbol for the entered country.

Parameter:

Number	Value to be formatted

String (optional) Country code

Return value:

String

Example:

LocCurr$(123,"USA") Result: "123.00"

See also Chapter "List of ISO 3166 Country Codes".

LocCurrL$

Purpose:

Returns a string with the valid currency format and currency symbol for the entered country.

Parameter:

Number	Value to be formatted
String	(optional) Country code

Return value:

String

Example:

LocCurr$(123,"USA") Result: "$123.00 "

See also Chapter "List of ISO 3166 Country Codes".

LocDate$

Purpose:

Returns a string with the valid date format for the entered country.

Parameter:

Date	Value to be formatted
String	(optional) Country code
Number	(optional) Format

Return value:

String

Example

LocDate$(Date("17.11.2007"),"USA") Result: "11/17/2007"

See also Chapter "List of ISO 3166 Country Codes".

LocDateTime

Purpose

Converts the character string into a date (with time if required) in the relevant format for the country.

Parameter:

String	Date
String	(optional) Country code

Return value:

Date

Example

LocDateTime(Date("17.11.2007"),"DEU") Result: "17.11.2007"

See also Chapter "List of ISO 3166 Country Codes".

LocNumber$

Purpose:

Returns a string with the valid number format for the entered country.

Parameter:

Number	Value to be formatted
String	(optional) Country code

Return value:

String

Example:

LocNumber$(123,"USA") Result: "123.00"

See also Chapter List of ISO 3166 Country Codes".

LocTime$

Purpose:

Returns a string with the valid time format for the entered country.

Parameter:

Date	Value to be formatted
String	(optional) Country code
Number	(optional) Format

Return value:

String

Example:

LocTime$ (Now(),"USA") Result: 9:05:22 AM"

See also Chapter "List of ISO 3166 Country Codes".

LocVal

Purpose:

Interprets the string as a number and returns its value (while respecting any localized decimal or 1000 separators).

Parameter:

String	Number (as string)
String	(optional) Country code

Return value:

Number

Example:

LocVal ("12,00","USA") Result: 1200,00

See also Chapter "List of ISO 3166 Country Codes".

Log

Purpose:

Calculates the natural logarithm ln(x).

Parameter:

Number

Return value:

Number

Example:

Log(Exp(1)) Result: 1

Log10

Purpose:

Calculates the base-10 logarithm log(x).

Parameter:

Number

Return value:

Number

Example:

Log10(1000) Result: 3

Lower$

Purpose:

Converts the characters of a string into lower case letters.

Parameter:

String

Return value:

String

Example:

Lower$("George") Result: "george"

LTrim$

Purpose:

Removes the leading spaces of a string.

Parameter:

String

Return value:

String

Example:

LTrim$("   George") Result: "George"

Max

Purpose:

Returns the largest of the two values.

Parameter:

Number or Date

Number or Date

Return value:

Number or Date

Maximum

Purpose:

Calculates the maximum of the set of values that result from the first parameter / formula.

Parameter:

Number

Boolean (optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

Maximum(Order_Details.ProductID@Products.ProductID:UnitsInStock)

Median

Purpose:

Calculates the median of the set of values that result from the first parameter / formula.

Parameter:

Number	Expression for the value to be averaged.
Boolean	(optional) The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

Median(UnitsInStock)

Mid$

Purpose:

Returns a part of a string. The desired number of characters starting at the starting position will be returned.

If the third parameter is not used, the string will be returned from the starting position to the end.

The first character of the string has the Position 0.

Parameter:

String	
Number	Starting position
Number	(optional) Number of characters to be displayed.

Return value:

String

Example:

Mid$("Normalconsumer",6)	Result: "consumer"
Mid$("Normalconsumer",6,30)	Result: "consumer"
Mid$("Normalconsumer",6,3)	Result: "con"
Mid$(Name,0,6)	Result: "Normal"

Min

Purpose:

Returns the smallest of the two values.

Parameter:

Number or Date

Number or Date

Return value:

Number or Date

Minimum

Purpose:

Calculates the minimum of the set of values that result from the first parameter / formula.

Parameter:

Number	
Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

Minimum(Order_Details.ProductID@Products.ProductID:UnitsInStock)

Minute

Determines the minute of the entered date, and returns the result as a number. If the parameter is not used, the minute of the time of printing will be returned.

Parameter:

Date (optional)

Return value:

Number

Mode

Purpose:

Calculates the mode (most common value) of the set of values that result from the first parameter / formula.

Parameter:

Number	Expression for the value to be examined.
Boolean	(optional) The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Month

Purpose:

Determines and returns the month (1...12) as a number.

Parameter:

Date

Return value:

Number

Example:

Month(Date("2007.10.17")) Result: 10

Month$

Purpose:

Determines and returns the month (1...12) as a string.

Parameter:

Date

Return value:

String

Example:

Month$(Date("2007.10.17")) Result: "10"

Now

Purpose:

Returns the current date and time.

Parameter:

-

Return value:

Date

NthLargest

Purpose:

Calculates the nth-largest value of the set of values that result from the first parameter / formula.

Parameter:

Number	
Number	<n>, i.e. the index for the value which is to be returned (1-based).
Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

NthLargest(Order_Details.ProductID,2)	calculates the 2-largest number

NthLargestIndex

Purpose:

Calculates the index of the nth-largest value of the set of values that result from the first parameter / formula.

Parameter:

Number	
Number	<n>, i.e. the index for the value which is to be returned (1-based).
Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

NthLargestIndex(Order_Details.ProductID,2)

NthValue

Purpose:

Calculates the nth value of the set of values that result from the first parameter / formula.

Parameter:

All	
Number	<n>, i.e. the index for the value which is to be produced, alculated e.g. with NthLargestIndex().
Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

NthValue(NthLargestIndex(Order_Details.ProductID,2))

NULL

Purpose:

Returns a NULL value (value not available).

Parameter:

-

Return value:

All

NULLSafe

Purpose:

Checks if the parameter is NULL and returns a substitute valueif it is, otherwise it returns the value of the parameter.

Parameter:

All

Return value:

All

NumInRange

Purpose:

Evaluates if a number falls within the desired range.

Parameter:

Number	
Number	Upper limit
Number	Lower limit

Return value:

Boolean

Example:

NumInRange(Page(),1,10) Result: True, if page number is between 1 and 10.

Odd

Purpose:

Evaluates if a number is odd. If the number is odd "True" will be returned, otherwise "False".

Parameter:

Number

Return value:

Boolean

Example:

"Page number "+Cond(Odd(Page()),"odd","even")

Ord

Purpose:

Returns the ASCII value of the first character.

Parameter:

String

Return value:

Number

Example:

Ord("Combit") Result: 67

Page

Purpose:

Returns the current page number.

Parameter:

-

Return value:

Number

Example:

Case(Odd(Page()),"Even","Odd")+" page number"

Page$

Purpose:

Returns the page number of the printed page as a character string.

Parameter:

-

Return value:

String

Example:

"Page "+Page$()+"/"+TotalPages$() Result: Page 1/3

Pow

Purpose:

Corresponds to the function (Base) ^ (Exponent).

Parameter:

Number	Base
Number	Exponent

Return value:

Number

Example:

Pow(2,3) Result: 8

Previous

Purpose:

Returns the previous value of the variable, field or formula, i.e. the value it had for the last record.

Parameter:

All	Variable, field or formula

Return value:

All

Example:

Previous(NAME) Result: "consumer"

PreviousUsed

Purpose:

Returns the value the given variable or expression had when it was last evaluated.

Parameter:

All	Variable, field or formula

Return value:

All

Example:

PreviousUsed(NAME) Result: "Mustermann"

ProjectParameter$

Purpose:

Returns the value of a project parameter. Available parameters:

LL.FAX.Queue	Print queue
LL.FAX.RecipName	Recipient name
LL.FAX.RecipNumber	Recipient fax number
LL.FAX.SenderBillingCode	Sender billing code
LL.FAX.SenderCompany	Sender company
LL.FAX.SenderDept	Sender department
LL.FAX.SenderName	Sender name
LL.MAIL.To	Mail address
LL.MAIL.CC	Mail address for carbon copy
LL.MAIL.BCC	Mail address for blind carbon copy
LL.MAIL.Subject	Subject line
LL.MAIL.From	Sender mail address

LL.MAIL.ReplyTo	Reply To mail address
LL.MinPageCount	Minimum page count.
LL.ProjectDescription	Project Description
LL.SlideShow.TransformationID	Default transition effect for the preview´s slideshow mode.
LL.MAIL.ShowDialog	Show mail dialog before sending

Parameter:

String	Name of the project parameter
Boolean	(optional) sets whether the return value (possibly a formula) should be returned directly (True), or should be evaluated (False). Default: False

Return value:

String

Example:

ProjectParameter$("LL.ProjectDescription") Result:"Article list"

ProjectPath$

Purpose:

Returns the path of the project file, optionally including the file name (otherwise with "\" at the end)

Parameter:

Boolean	True: Sets that the path is returned including the file name (Default: False).

Return value:

String

Example:

ProjectPath$(.T.) Result: C:\Program Files\LL\crosstab.lsr

Quarter

Purpose:

Returns the quarter of the year (1..4)

Parameter:

Date

Boolean	(optional) sets whether the quarter calculation should be returned relative to the year (1..4) or in absolute terms since 1.1.0001 (1..). Default: False (relative).

Return value:

Number

Example:

Str$(Quarter(Date("01.01.2008")),0,0)	Result: 1
Str$(Quarter(Date("01.05.2008")),0,0)	Result: 2
Str$(Quarter(Date("01.05.2008"),.T.),0,0)	Result: 8030

RainbowColor

Purpose:

Calculates a color value between violet and red corresponding to the value of the first parameter e.g. for rainbow colors in crosstabs.

Parameter:

Number	Value to be displayed.
Number	Limiting value for violet.
Number	Limiting value for red.

Return value:

Number

RegExMatch$

Purpose:

Returns the part of the string that corresponds to the regular expression or the group passed in the third parameter.

The regular expression corresponds to Pearl 5 Syntax, which in most details equals the regular expression syntax of the Visual Basic Scripting engine.

Parameter:

String

String

Number

Return value:

String

Example:

Division of the "STREET" field to street and number:

"Street: " + RegExMatch$(STREET,"((?:\w*)+)(\d+[\w]*$)",1) "Number: " + RegExMatch$(STREET,"((?:\w*)+)(\d+[\w]*$)",2)

RegExMatch$("test1234xyz0815", "[0-9]+") Result: "1234"

RemainingTableSpace

Purpose:

Returns the space available to data and group lines in a table object. The parameter defines the unit of the return value. The function can be used to carry out conditional page breaks before group lines, e.g. "Page break before only 5% space is left”.

Parameter:

Boolean	(optional) TRUE: the value is in units which are independent of the system (SCM-units), FALSE: the value is a percentage of the entire table size (default: FALSE).

Return value:

Number

Rep$

Purpose:

Returns a string that contains the appropriate number of strings defined in the first parameter.

Parameter:

String

Number

Return value:

String

Example:

Rep$("-",10) Result: "----------"

Rep$("+-",5) Result: "+-+-+-+-+-"

RGB

Purpose:

Calculates the color value using the relative red, green and blue saturation values (between 0 and 255). No saturation has the value 0, full saturation the value 255. This function can be used to set the font color using a formula.

Parameter:

Number	red saturation
Number	green saturation
Number	blue saturation

Return value:

Number

Example:

Cond(Amount<0, RGB(255,0,0), RGB(0,0,0) Result: red for negative amounts

Right$

Purpose:

Reduces a string from the left so that only the number of characters set under Number remain. If the original string is already small enough, it is not affected.

Parameter:

String

Number

Boolean (optional) True: The cut off value starts with "..." (Default: False). If Number < 3 the setting is ignored.

Return value:

String

Example:

Right$("normalconsumer", 8)	Result: "consumer"
Right$("normalconsumer", 11,.T.)	Result: "...consumer"

Round

Purpose:

Rounds a value to the entered number of decimal places. Default is 0.

Parameter:

Number

Number (optional)

Return value:

Number

Example:

Round(3.1454,2)	Result: 3,15
Round(3.1454)	Result: 3

RTrim$

Purpose:

Removes spaces from the end of a string.

Parameter:

String

Return value:

String

Example:

RTrim$("John   ")   Result: "John"

Second

Determines the second of the entered date and returns the result as a number. If the parameter is not used, the second of the print time will be returned.

Parameter:

Date	(optional)

Return value:

Number

Sign

Purpose:

Returns the sign of the value (+1 for a positive value, -1 for a negative value or 0 if the value is 0).

Parameter:

Number	Value

Return value:

Number

Example:

Sign (-3)   Result: -1

Sin

Purpose:

Calculates the sine of the value.

Parameter:

Number	Value
Number	(optional) Mode (0=Degree, 1=Radian). Default: 0.

Return value:

Number

Example:

Sin (90)	Result: 1

Sqrt

Purpose:

Calculates the square root of a number.

Parameter:

Number

Return value:

Number

Example:

Sqrt(4)	Result: 2

StdDeviation

Purpose:

Calculates the standard deviation of the set of values that result from the first parameter / formula.

Parameter:

Number	
Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

StdDeviation(Order_Details.Quantity*Order_Details.UnitPrice)

Str$

Purpose:

Converts a number into a string. The number will be formatted with 6 decimal places that may be rounded. The length is variable.

Parameter:

Number	
Number	(optional) Defines the length of the string (default:6). If the number is too large for this format, the resulting string may then be longer

	than desired. If the number is too small, spaces will be attached dependent upon the prefix, right (negative) or left (positive).
Number	(optional) Defines the precision (number of decimal places). If the number is positive, the number will be displayed as a floating-point number, if negative in scientific format.

Return value:

String

Example:

Str$(Pi)	Result: "3.141593"
Str$(Pi,3)	Result: " 3"
Str$(Pi,3,0)	Result: " 3"
Str$(-Pi,12,-3)	Result: "-3.141e+00"
Str$(Page())	Result: "5.000000"
Str$(Page(),10)	Result: " 5"
Str$(Page(),-10)	Result: "5 "

StrPos

Purpose:

Returns the position of the nth appearance of a search string. The third parameter determines which appearance will be returned. Default is 1.

The first character in the string has the position 0.

-1 as return value signifies the search string no longer appears.

Parameter:

String	
String	Search string
Number	(optional)

Return value:

Number

Example:

StrPos("Normalconsumer","or")	Result: 1
StrPos("Normalconsumer","r")	Result: 2
StrPos("Normalconsumer","r",1)	Result: 2
StrPos("Normalconsumer","r",2)	Result: 13

StrRPos

Purpose:

Returns the position of a search string within a string. This is a backwards search. The third parameter, determines which appearance will be returned. Default is 1.

The first character in the string has the position 0.

-1 as return value signifies the search string no longer appears.

Parameter:

String	
String	Search string
Number	(optional)

Return value:

Number

Examples:

StrRPos("Normalconsumer","or")	Result: 1
StrRPos("Normalconsumer","r")	Result: 13
StrRPos("Normalconsumer","r",1)	Result: 13
StrRPos("Normalconsumer","r",2)	Result: 2

StrSubst$

Purpose:

Searches a string for the appearance of a search string and replaces it with the string contained in the third parameter (replacement string). If no third parameter is used, the string located using the search string will be deleted.

Parameter:

String	
String	Search string
String	(optional) Replacement string

Return value:

String

Example:

Assume that you want to print address labels that contain the company name. You do not have much space available on the label and cannot afford to completely print long company names, for example, "Forrer Construction, Incorporated”.

With the expression `StrSubst$(COMPANY,"Incorporated","Inc.")` every appearance of "Incorporated" in the COMPANY field will be replaced with "Inc."

Sum

Purpose:

Calculates the sum of the parameter / formula in the parameter.

Hint: Sum variables (see "Working with Sum Variables") are an alternative way of creating sums and counters. Sum variables are principally applicable to whole tables. Aggregate functions principally table specific.

Parameter:

Number	
Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

Sum (Order_Details.UnitPrice)

Tan

Purpose:

Calculates the tangent of the value.

Parameter:

Number	Value
Number	(optional) Mode (0=Degree, 1=Radian). Default: 0.

Return value:

Number

Example:

Tan (45)   Result: 1,00

Time$

Purpose:

Returns the current time in string format.

The following formats are available:

Placeholder	Description
%h	Hours in 24 hour format
%H	Hours in 12 hour format

%m	Minutes
%s	Seconds
%P	Display the part of day (A.M. / P.M.)
%p	Display the part of day (a.m. / p.m.)

Parameter:

String

Return value:

String

Example:

Time$("%02h:%02m:%02s") Result: "18:30:45"

Today

Purpose:

Returns the current date.

Parameter:

-

Return value:

Date

Example:

Date$(Today(),"%D, %m.%d.%4y") Result: "Friday, 11/8/2007"

Token$

See Case$

ToRTF$

Purpose:

Returns a string in RTF-format. This is necessary because some strings may contain one of the specially defined RTF-format symbols. ('\', '{' or '}').

Parameter:

String

Return value:

String

Example:

If, for example, the field PRODCODE could contain one of the characters, then the text should be inserted in the following way:

"<<SALUTATION>> <<NAME>>, You have received our product <<PRODUCT>>, Code <<ToRTF$(PRODCODE)>>..."

Total$

Purpose:

Sets that the expression in the argument is calculated for the whole object, e.g. in a crosstable.

Parameter:

All

Return value:

All

Example:

Sum(sales)/Total(Sum(sales))*100

TotalPages$

Purpose:

Returns the total number of pages. The returned character string is replaced by the total number of pages when printing.

Please note when using this function that the timing behavior of the print process can be affected. A progress bar may reach 100% faster, but because of further processing of the output, there may be a delay before the actual printout is produced. No calculations may be performed with the result of this function.

Parameter:

-

Return value:

String

Example:

"Page "+Page$()+"/"+TotalPages$()   Result: Page 1/3

UnitFromSCM

Purpose:

Converts a SCM-Unit (1/1000 mm) to the print unit (inch/mm). Important for the definitions of property values independent of the selected print unit.

Parameter:

Number

Return value:

Number

Example:

Cond(Page()=1,UnitFromSCM(100000),UnitFromSCM(20000))

Result: 10cm for Page 1, 20 cm for the other pages.

Upper$

Purpose:

Converts the characters of a string to capital letters.

Parameter:

String

Return value:

String

Example:

Upper$("Otto")	Result: "OTTO"

Val

Purpose:

The string is interpreted and returned as a number. If an error occurs, the return value is 0. The decimal sign must always be entered as ".".

Parameter:

String

Return value:

Number

Example:

Val("3.141")	Result: 3.141
Val("3,141")	Result: 3
Val("3.141e2")	Result: 314.2
Val(ChrSubst$("3,141", ",", "."))	Result: 3.141

Variance

Purpose:

Calculates the variance of the set of values that result from the first parameter / formula.

Parameter:

Number

Boolean	(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

Variance(Order_Details.Quantity*Order_Details.UnitPrice)

Woy

Purpose:

Returns the week number of a given date.

The optional second parameter determines the setting for the first week of the year.

0	Week with the first working day
1	Week of January, 1
2	First week with at least 4 days
3	First week with 7 days
4	Week with the first Monday

Parameter:

Date

Number (optional)

Return value:

Number

Year

Purpose:

Determines the year of a date and returns it as a number.

Parameter:

Date

Return value:

Number

Example:

Year(Today())	Result: 2008
Year$(Date("1.1.2008"))	Result: 2008

Year$

Purpose:

Determines the year of a date and returns it as a string.

Parameter:

Date

Return value:

String

Example:

Year$(Today())	Result: "2008"
Year$(Date("1.1.2008"))	Result: "2008"

5.3 List of ISO 3166 Country Codes

The following country codes can be used with, among others, the Loc.. functions. (CountryCode: Prefix):

ALB	355	COL	57	GTM	502	NIC	505	SVK	42
ARG	54	CRI	506	HND	504	NLD	31	SVN	386
AUS	61	CZE	42	HRV	385	NOR	47	SWE	46
AUT	43	DEU	49	HUN	36	NZL	64	TTO	1
BEL	32	DNK	45	IDN	62	PAN	507	TUR	90
BGR	359	DOM	1	IRL	353	PER	51	UKR	380
BLR	375	ECU	593	ISL	354	POL	48	URY	598
BLZ	501	ESP	34	ITA	39	PRI	1	USA	1
BOL	591	EST	372	JAM	1	PRT	351	VEN	58
BRA	55	FIN	358	LIE	41	PRY	595	ZAF	27
CAN	2	FRAU	33	LTU	370	ROM	40		
CAR	1	FRO	298	LUX	352	RUS	7		
CHE	41	GBR	44	LVA	371	SLV	503		
CHL	56	GRC	30	MEX	52	SPB	381		

5.4 List of Available Barcodes

List & Label supports a wide range of barcode formats. Normally, no special printers, fonts, etc. are necessary, the barcodes will be printed directly from List & Label.

5.4.1 List of 1-D-Codes

GTIN-13, EAN-13, UCC-13, JAN-13

The EAN13-(International Article Numbering)-Code is a very common barcode. The number string used with EAN13 must conform to one of the following formats:

"cc|nnnnn|aaaaa" (normal EAN13)
"cc|nnnnn|aaaaa|xx" (EAN13 for periodicals, "ISSN")
"cc|nnnnn|aaaaa|xxxxx" (EAN13 for books, "Bookland")
"ppp|nnnn|aaaaa" (normal EAN13)
"ppp|nnnn|aaaaa|xx" (EAN13 for periodicals, "ISSN")
"ppp|nnnn|aaaaa|xxxxx" (EAN13 for books, "Bookland")

with

	Description	Value Range
Cc	Country code	c='0'..'9'
Ppp	Product code	
Nnnnn	Company code	n='0'..'9'
aaaaa	Article code	a='0'..'9'
'|'	= Character code chr(124)	
xx, xxxxx	Supplemental code	x=´0´..´9´

The check digit will be calculated and attached automatically. Each character is 7 bar-widths wide, a code should have a minimum width of (12*7+11)*0.3 mm = 2.85 cm.

Ideal size (bar symbol) nominal size SC2:

Width = 31.4 mm
Height = 24.5 mm

Minimum offset that should be kept free around the symbol:

left: 3.6 mm
top: 0.3 mm
right: 2.3 mm
bottom: 0.0 mm

The text may partially exceed this area.

EAN-14, UCC-14

These number codes require a high print quality.

Allowed characters: '0'..'9'. The code must be 14 characters long.

GTIN-8, EAN-8, JAN-8, UCC-8

The number string in EAN8 must conform to the following format:

"nn|nnnnn"

with

- n = '0'..'9'
- | = Character code chr(124)

Each character is also 7 bar widths wide, a code should then have a minimum width of (8*7+11)*0.3 mm=2.01 cm.

Ideal size (bar symbol)	nominal size SC2:
dx :	22.1 mm
dy:	19.9 mm

Minimum offset that should be kept free around the symbol:

left:	2.3 mm
top:	0.3 mm
right:	2.3 mm
bottom:	0.0 mm (if printed, otherwise 0.3 mm)

The text may partially exceed this area.

UPC-A

The UPC-A-Code (Universal Product Code) is common in the USA. The code must conform to the following format:

"c|nnnnn|aaaaa"

with

- c = Number system
- nnnnn = Company code
- aaaaa = Article code
- | = Character code chr(124)

The check digit will be calculated and attached automatically. Each character is also 7 bar widths wide, a code should then have a minimum width of (13*7+6)*0.3 mm=2.88 cm.

UPC-E

The UPC-E-Code (Universal Product Code, short version) is common in the USA. The code must conform to the following format:

"c|nnnnn"

with

- c = Number system
- | = Character code chr(124)
- nnnnn = Code, interpretation dependent upon the last digit

The check digit will be calculated and attached automatically. Each character is also 7 bar widths wide, a code should then have a minimum width of (13*7+6)*0.3 mm=2.88 cm

2-of-5 Industrial

A number code of low information density.

Allowed characters: '0'..'9'

A Code is (14*number of characters+18) bar widths wide.

2-of-5 Interleaved (ITF)

A number code of high information density, requires high print quality.

Allowed characters: '0'..'9'. The number of characters must be even.

A code is (9*number of characters+9) bar widths wide.

2-of-5 Matrix

A number code of high information density, requires high print quality.

Allowed characters: '0'..'9'.

A Code is (10* number of characters +18) bar widths wide.

2-of-5 Datalogic

A number code of high information density, requires high print quality.

Allowed characters: '0'..'9'.

A code is (10* number of characters +11) bar widths wide.

Codabar

The Codabar-Code is a numerical code with 6 special characters. The information density is low. The code must conform to the following format:

"fnnnnf"

with

- f = Frame code ('A', 'B', 'C', or 'D')
- nnnnn = arbitrary quantity of numbers or special characters ('0'..'9', '-', '$', ':', '/', '.', '+')

Every character is either 2*3+6*1 (characters '0'..'9', '-', '$') or 3*3+5*1 (characters ':', '/', '.', '+', 'A'..'D') bar widths wide. The characters for the frame code will not be printed with the text.

CODE11

Allowed characters: '0'..'9','-'.

Code 11 has, depending on the length, 1 or 2 check digits. List & Label calculates only 1 instead of 2 check digits if the length of the text is a maximum of 10 characters.

Code39, 3-of-9, Alpha39

One of the few codes that can also display letters. All characters must be entered; the customary bracketing with ´*´ should also be entered ('*TEST*').

Allowed characters: ' ','$','/','%','*','+','-','0'..'9','A'..'Z'

The expanded code can be activated by a combination of the standard code: for example: '+A' -> 'a'. Every character is 16 bar widths wide, a text has (16*number of characters –1) bars.

Code 39 with CRC

Allowed characters: ' ','$','/','%','*','+','-','0'..'9','A'..'Z'

Code 93 (simple and extended)

Code 93 is an extension of Code 39, but has the advantage that it is somewhat smaller. It covers the complete 128 bytes of the ASCII Allowed characters, including zero. This must be transferred as chr$(255).

It contains two check digits that are automatically generated by List & Label.

The characters consist of 9 bar widths, that each have 3 bars and 3 spaces. There are two options for the extended code:

- transfer of the shift character from the host program as

$	chr$(254)
%	chr$(253)
/	chr$(252)
+	chr$(251)

- transfer of the desired character, L&L adds the appropriate shift character.

Code 128

This code is basically identical to EAN128, with the exception that the first character (FNC1 ... FNC4) is defined by the user.

The explanations with EAN128 apply here as well.

Determine the code set that should be used:

Use one of the following codes as a start character:

chr$(135) – start with code A

chr$(136) – start with code B

chr$(137) – start with code C

In order to switch between different code sets within the barcode, you can use the usual control characters:

Starting from code A to
B: chr$(132)
C: chr$(131)
Starting from code B to
A: chr$(133)
C: chr$(131)
Starting from code C
A: chr$(133)
B: chr$(132)

Example: <Subset B> "RL" <Subset C> "04432476" <Subset B> "0DE110"

Barcode(Chr$(136)+"RL"+chr$(131)+"04432476"+chr$(132)+"0DE110", "Code 128")

GS1 128, EAN128

This code can display all printable ASCII-characters, as well as non-explicit characters, umlauts and 'ß'. The text can have any appearance. The width is not easy to display because certain characters have different widths.

Special characters must be replaced with:

NUL	chr$(255)
FNC1	chr$(254)
FNC2	chr$(253)
FNC3	chr$(252)
FNC4	chr$(251)

GS1 DataBar

It supports the codes GS1 DataBar, GS1 DataBar Limited, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Stacked Truncated

Allowed characters: [0-9]

Format: nnn|nnnnnnnnnn (13-digits)

GS1 DataBar Expanded

Allowed characters: Data begins with AI. Max 74 numerical/41 alphanumrical characters

Format: nnn|nnnnnnnnnn (12-digits, no check digit)

ISBN

Allowed characters: [0-9]

Format: nnn|nnnnnnnnnn (12-digits, no check digit)

The International Standard Book Number, abbreviated, ISBN, is a unique 13-digit number for the identification of books and other independent publications, such as multimedia products and software.

ISBN-13 and EAN-13 are identical.

Example: 979-3-86680-192-9

ISBN comprises five components.

1. A prefix depending on the book (978 or 979).
2. The country number, e.g. 0 and 1 for English-speaking countries, 2 for French-speaking countries and 3 for German-speaking countries.
3. The publisher code, which is the identifying number of the publishing house.
4. The title number given by the publisher.
5. The check digit.

MSI

MSI is a binary barcode in which every character consists of 8 bars.

The character set is limited to '0'..'9' and is suitable only for the display of numbers. Commonly used in libraries.

Pharma-Zentral-Nummer

Allowed characters: '0'..'9'

The PZN code is used for identifying pharmaceuticals. The pharmaceutical central numbers (=PZN) are distributed by the "Informationsstelle für Arzneispezialitäen IfA GmbH". The represented string of numbers has to correspond to the following format:

"nnnnnn"

QR Code

Allowed characters: all characters

Format: any

The QR code (QR=quick response) is a two dimensional barcode. It has an exceptionally high fault tolerance at a high density of data and can be scanned quickly. The QR code is widespread in Japan and is meanwhile also used for digital cameras and mobile telephones. There, the picture that has been taken is decoded so that the context information or a web address (e.g. via a UMTS network) can be accessed. Free reading programs are available for many mobile telephones.

In order to enter non-printable characters (binary data) in barcode text, they have to be packed in a special character string. The data is entered via the character string ~dNNN where NNN represents the ASCII code.

Example: ~d065 stands for the letter "A".

SSCC/NVE

The SSCC (Serial Shipping Container Code) barcode is being increasingly used in the logistics industry.

Allowed characters: [0-9]

Format: nnnnnnnnnnnnnnnnn (17 figures)

5.4.2 List of 2-D-Codes

Aztec

2D barcode system with high information density and excellent scanability, which is highly optimized and has high error tolerance. The Aztec barcode is mostly used in medicine technology, transport sector and administration. Character set: All available and unprintable characters. Minimum length 12 characters up to approx. 3000.

Datamatrix

(Application dependent)

This barcode is used in many different industrial areas. Most printable characters can be displayed. In addition, very dense information is provided and can still be scanned correctly with extensive damage. You can freely choose symbol size or choose a symbol size which is automatically adapted to object size. These properties can be found in the option dialog of the barcode.

In order to enter non-printable characters (binary data) in barcode text, they have to be packed in a special character string. The data is entered via the character string ~dNNN where NNN represents the ASCII code.

Example: ~d065 steht für den Buchstaben "A".

PDF417

Allowed characters: Most available and also non-printable characters. To use non-printable characters in barcode text, tags in the form {binary:xx} can be used in the barcode text, with xx standing for any sequence of two digit hexadecimal numbers. This is especially important when Maxicodes are to be created to UPS specifications. The required special characters can be entered in this way.

Example 1: To encode a NULL and a backspace (BS) symbol in the data, use "{binary:0008}" (corresponds to "{binary:00}{binary:08}").

Example 2: To pass on a page break, use "Hello{binary:0d0a}World".

5.4.3 List of Postcodes (1-D und 2-D-Codes)

DP-Identcode

A number code of high information density, requires high print quality.

Allowed characters: '0'..'9'.

The code must conform to the following format:

"nn.nnnnn.nnnn"
"nn.nnnn.nnnnn"
"nn.nnn.nnnnnn"

A Code is (9*Number+9) bar widths wide.

Width: 32,0mm - 58,5mm (Clear zone right and left: minimum 5mm). height: 25mm.

The check digit is calculated automatically; Relation: 4:9; Special code of 2 of 5 IL.

DP-Leitcode

A number code of high information density, requires high print quality.

Allowed characters: '0'..'9'.

The code must conform to the following format:

"nnnnn.nnn.nnn.nn"

A Code is (9*Number+9) bar widths wide.

Width: 37,25 mm - 67,5 mm (Clear zone right and left: minimum 5mm). Height: 25mm.

The check digit is calculated automatically; Relation: 4:9; Special code of 2 of 5 IL.

FIM Barcode

Minimum size: 1/2 * 5/8 = 12.7 mm * 15.87 mm.

Transfer values: "A", "B" or "C".

The FIM-Barcode is always printed to the size required by the United States Postal Service. It can expand beyond the object frame.

German Parcel

A number code of low information density.

Allowed characters: '0'..'9'.

A Code is (14*Number+18) bar widths wide.

Relation: 1:2

Japanese Postal Code

Japanese postcode.

Allows characters: n=[0-9], Address=[A-Z], [0-9], [-]

Formats: nnn-nnnn, then max. 13 character address

Maxicode

Allowed characters: Most available and also non-printable characters. To use non-printable characters in barcode text, tags in the form {binary:xx} can be used in the barcode text, with xx standing for any sequence of two digit hexadecimal numbers. This is especially important when Maxicodes are to be created to UPS specifications. The required special characters can be entered in this way.

Example 1: To encode a NULL and a backspace (BS) symbol in the data, use "{binary:0008}" (corresponds to "{binary:00}{binary:08}").

Example 2: To pass on a page break, use "Hello{binary:0d0a}World".

Postnet

This is a barcode used by the United States Postal Service. It is available in three variants. In reference to placement and offset from other objects, please refer to the specifications in the appropriate literature.

5-digit:	"nnnnn"
10-digit:	"nnnnn-nnnn"
12-digit:	"nnnnn-nnnnnn"

Minimum size: 1.245 * 4/16 = 31.6 mm * 6.35 mm (10-digit).

Minimum bar spacing 1/24 = 1.058 mm.

The error-correction digit will be automatically amended.

This barcode will be automatically printed in the correct size.

RM4SCC, KIX®

Allowed characters: '0'..'9', 'A'..'Z', 'a'...'z'.

This barcode is used by the Dutch post in distribution. Make sure to pass on a content according to the specification. For further information please contact the Dutch post.

Royal Mail

Allowed characters: '0'..'9', 'A'..'Z', 'a'...'z'.

This code is used with the mail merge procedures "Cleanmail" and "Mailsort" by the British "Royal Mail" to encode postal codes. Royal Mail postal codes contain a combination of numbers and letters. The Allowed characters therefore includes the numbers 0..9 and capital letters A..Z. The space character is not included.

Either the postal code (e.g. LU17 8XE) or the postal codes with an additional "Delivery Point" (e.g. LU17 8XE 2B) are encoded. The maximum number of usable spaces is therefore limited to 9 digits.

6. Index