

List & Label 12

The Report Designer

Copyright © combit GmbH 1991-2006

<http://en.combit.net>

All rights reserved.

Information in this document is without guarantee and subject to change without notice. The software described in this documentation is provided on the basis of the combit GmbH license agreement.

Neither this manual nor any part of it may be hard copied or stored in any other way (e.g. digitally) without the express written consent of combit GmbH. All references to personal or company names in this handbook are, as far as could be determined, purely fictitious. Any similarities are unintentional. Avery and all Avery brands, product names and codes are trademarks of Avery Dennison Corporation.

Copyright © combit GmbH 1991-2006; Rev. 12.001

<http://en.combit.net>

All rights reserved.

1. Introduction	7
1.1. General	7
1.2. Help Functions	9
1.2.1. Context sensitive help	9
1.2.2. Online Help	9
1.2.3. Tooltips for dialog controls	9
2. General Methods and Procedures	10
2.1. The Designer Interface	10
2.1.1. Dockable and Floating Tool Windows and Toolbars	11
2.1.2. Object Bar	13
2.1.3. Toolbar	13
2.1.4. Context menu	13
2.1.5. Status Bar	13
2.1.6. Workspace	14
2.1.7. Variable List	16
2.1.8. Layers Window	16
2.1.9. Objects Window	17
2.1.10. Table Structure	18
2.1.11. Preview	18
2.1.12. Properties Window	19
2.1.13. Using the Keyboard	20
2.1.14. Drag & Drop	21
2.2. Exiting the Designer	21
2.3. Variables, Formulas and Expressions	21
2.3.1. The Elements of an Expression	22
2.3.2. Insert variables	25
2.3.3. Insert fixed text	26
2.4. Working with Functions	27
2.4.1. Function Syntax	29
2.4.2. Value Types	29
2.4.3. Examples for the Use of Functions	30
2.4.4. Insert logical condition	32
2.4.5. Formatting Numbers, Dates and Currencies	33
2.5. Operators	35
2.5.1. Relational Operators	37
2.5.2. Arithmetic Operators	37
2.5.3. Logical Operators	38
2.6. Working with Sum Variables	38
2.6.1. Defining Sum Variables	39
2.6.2. Defining a Counter	40
2.7. Working with User Variables	40
2.8. Using Filters	40
2.8.1. Record Filters	41
2.8.2. Project Specific Filters for Records	41

2.8.3. Layer Specific Filters for Objects	41
2.8.4. Object Specific Filters for Objects	41
2.8.5. Define Filters	41
3. Projects	42
3.1. Creating or Opening a Project	42
3.1.1. Create a new project	42
3.1.2. The Project Wizard	42
3.1.3. Opening an Existing Project	43
3.1.4. Project Import	44
3.1.5. Project Types	44
3.1.6. Project Properties	44
3.2. Saving projects	45
3.3. Defining Page Setup	45
3.3.1. Printer Selection	45
3.3.2. Export Media	46
3.3.3. Templates and Page Layout for Labels	47
3.4. Default Values and Options	49
3.4.1. Options for the Project	49
3.4.2. Options for Objects	50
3.4.3. Options for the Preview	51
3.4.4. Options for the Workspace	52
3.4.5. Compatibility Options	53
3.4.6. Options for the Formula Wizard	55
3.5. Including Projects	55
3.6. Printing Projects	56
3.6.1. Print Sample from the Designer	56
3.6.2. Printing to Real Data Preview	58
4. Objects	62
4.1. Inserting Objects	62
4.2. Editing Objects	63
4.2.1. Selecting Objects	64
4.2.2. Arrange	65
4.2.3. Alignment	66
4.2.4. Moving and Sizing Objects	67
4.2.5. Grouping Objects	69
4.3. Working with Layers	69
4.3.1. Defining Layers	69
4.3.2. Assigning Layers	71
4.3.3. Copying into Layers	71
4.3.4. Switching Layers On/Off	71
4.4. Object List	72
4.5. Interlinking objects	73
4.5.1. Detaching Interlinks	75
4.5.2. Individual Interlinking (Size and Position Adaption)	75

4.5.3. The "At End, Keep Size" Interlink	77
4.5.4. The "sequential" interlink	77
4.5.5. Examples for Individual Interlinks	79
4.5.6. Advanced Interlinking	89
4.6. Common Object Properties	89
4.6.1. Locked	90
4.6.2. Name	90
4.6.3. Appearance Condition	91
4.6.4. Pagewrap Before Object Output	91
4.6.5. Position	91
4.6.6. Frame / Background	92
4.6.7. Colors	93
4.6.8. Pattern	94
4.6.9. Filled / Background	95
4.6.10. Formatting	95
4.6.11. Font	96
4.6.12. "Contents" of Objects	98
4.7. Report Structure	98
4.7.1. Characteristics of Elements	99
4.7.2. Working with Report Structure	99
4.8. Inserting Text Objects	102
4.8.1. Properties	102
4.8.2. Text Contents	103
4.8.3. Editing Paragraphs	103
4.8.4. The Card "Tab"	105
4.8.5. Paragraph Properties	106
4.9. Inserting Lines	109
4.9.1. Properties	109
4.10. Inserting Rectangles	109
4.10.1. Properties	109
4.11. Inserting Ellipses	110
4.11.1. Properties	110
4.12. Inserting Picture Objects	111
4.12.1. Properties	111
4.13. Inserting Barcode Objects	112
4.13.1. Properties	112
4.13.2. Barcode Contents	113
4.14. Inserting Table objects	114
4.14.1. Properties	115
4.14.2. Table Contents	117
4.14.3. Definition of Table Lines	118
4.14.4. Defining Line Layouts	119
4.14.5. Editing the Line Definitions List	120
4.14.6. Define Column Contents	120
4.14.7. Format Lines	122
4.14.8. Appearance Conditions for Table Lines	123

4.14.9. Column Properties	123
4.14.10. Defining Footer Lines	126
4.14.11. Defining Group Lines	126
4.14.12. Define Group Footers	129
4.15. Inserting Formatted Text	129
4.15.1. Properties	129
4.15.2. Text Contents	130
4.16. Inserting Chart Objects	132
4.16.1. Properties	132
4.16.2. Selecting the Chart Type	133
4.16.3. Selecting the Sub Type	134
4.16.4. Chart Properties	137
4.16.5. Examples	142
4.17. Inserting a Crosstab	144
4.17.1. Properties	145
4.17.2. Axis definition (Grouping)	146
4.17.3. Cell definition (Contents)	148
4.17.4. Layout Options and Page Break Behavior	150
4.17.5. Crosstab Creation Wizard	151
4.17.6. Crosstab Functions	152
4.18. Inserting Form Controls	152
4.18.1. Type Edit	153
4.18.2. Type Checkbox	153
4.18.3. Type Combobox	153
4.18.4. Type Button	154
4.19. Inserting HTML formatted text	154
4.19.1. Properties	155
4.19.2. HTML-Contents	155
4.20. Inserting OLE-Server Documents	156
4.20.1. Properties	156
4.20.2. Contents	156
4.21. Inserting Form Templates	156
4.21.1. Properties	157
5. Appendix	158
5.1. List of List & Label Variables	158
5.1.1. List of Variables	158
5.1.2. List of Fields	159
5.2. List of Available Functions	160
5.3. List of Available Barcodes	219
5.4. List of ISO 3166 Country Codes	227
6. Index	228

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.

	Texts: These objects can contain fixed text as well as the variable content of individual records. You can define the fonts, object orientation, and alignment of the contents.
	Rectangles: These objects are rectangular frames for which layout options such as frame width, colors, filling pattern and shading can be defined.
	Circles and Ellipses: Define Circles and Ellipses. Layout options such as frame width, colors and filling pattern can be defined.
	Lines: With this object type you can define lines with layout options such as line width and color.
	Picture Files: These objects allow you to integrate graphics into projects. These can be either fixed graphics (ex. a company logo) or variable graphics (i.e. different for each record).

	Barcodes: These objects allow you to display fixed text or variable contents as barcodes. You can define characteristics such as the type of barcode, color, text, and orientation.
	Tables (Lists): These objects allow you to display fixed text and variable contents from various records. You can define layout characteristics such as appearance, fonts, column widths and alignments, word wraps, footers, etc. Tables appear in list projects only.
	Formatted Texts: With this object type you can change the format within a line.
	Form Controls: Permits input and changes within the Preview.
	Charts: With this object you can create diagrams.
	Crosstab (Pivot Table): With this object you can insert a crosstab object and analyze and display different kinds of data in several dimensions.
	HTML Texts: With this object you can display the contents of web sites and other HTML formatted text. (see Chapter "5.3. Text Objects").
	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. See Chapter "5.13. Insert OLE-Server Documents" for more details.
	Form Templates: These graphic objects 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 "2.1.2. 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:

- Define the page layout
- Set preferences and options (only necessary once)
- Insert objects
- Edit objects
- Save the project

1.2. Help Functions

1.2.1. 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.

1.2.2. 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.

1.2.3. 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

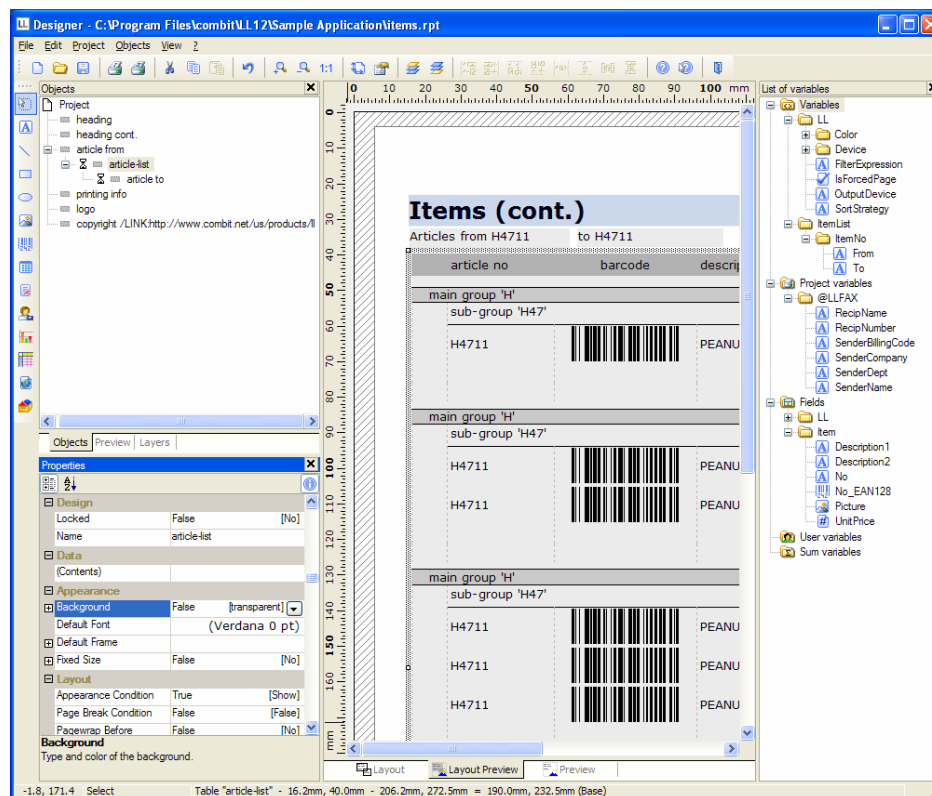
This manual is divided into three sections. The first section is an introduction to the List & Label Designer and the functions which are available to you.

The second section describes in turn the commands which are available via the menu.

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.

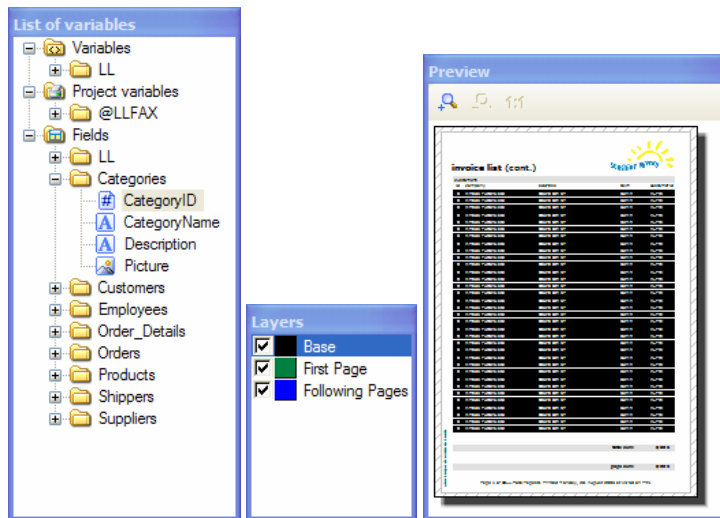
2.1. The Designer Interface



2.1.1. Dockable and Floating Tool Windows and Toolbars

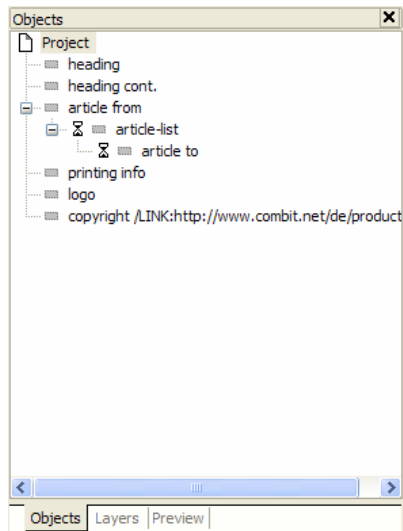
Tool Windows

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.

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).

2.1.2. 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'**.

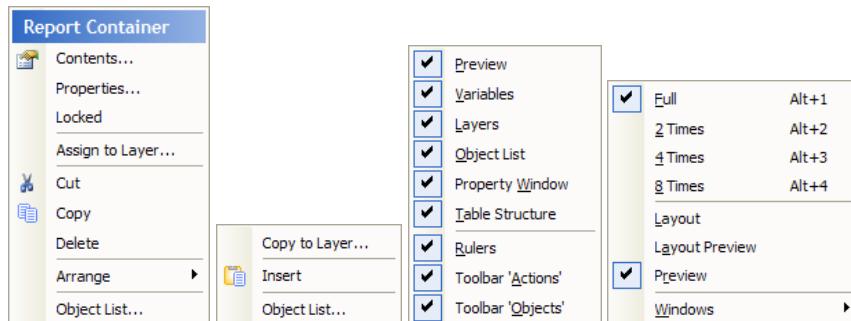
2.1.3. Toolbar

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



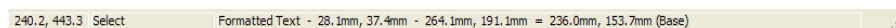
2.1.4. Context menu

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.

2.1.5. Status Bar



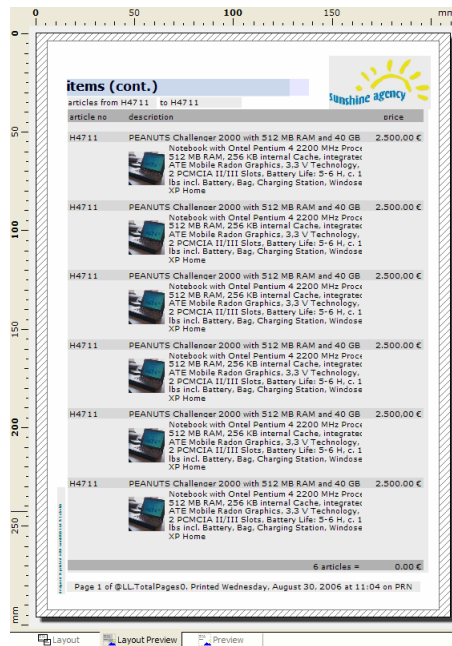
The status bar is divided into three sections:

- 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

2.1.6. 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**).



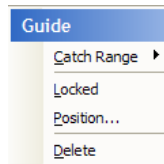
The entity of all objects on the workspace and its corresponding layout are called a project. List & Label distinguishes between list projects (list/table creation), card and label projects (label creation).

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.

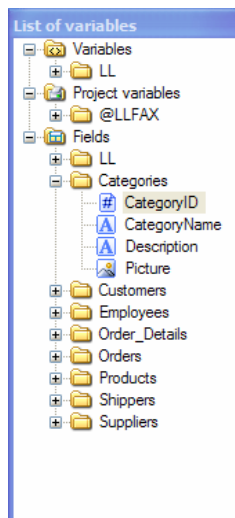
2.1.7. Variable List

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 area in the project workspace, a new text object is automatically generated.

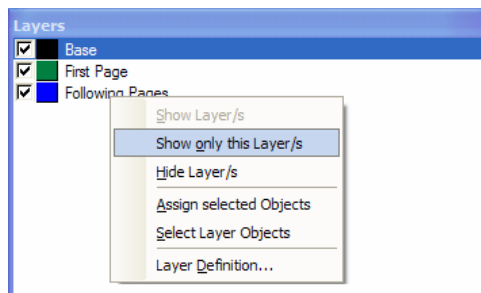


2.1.8. Layers Window

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

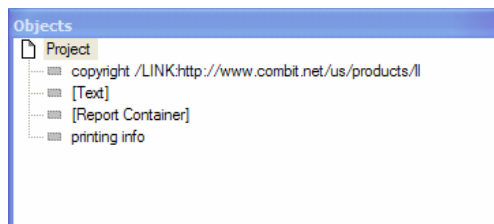
You can improve the overview in complex projects by assigning objects to an individual layer. You can switch layers on and off at any time by using the Layers Window. You can give these layers their own appearance conditions. For example, one layer may only be valid for the first page of a multipage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

More detailed information on working with layers can be found in Chapter "4.3. Working with Layers".



2.1.9. 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.

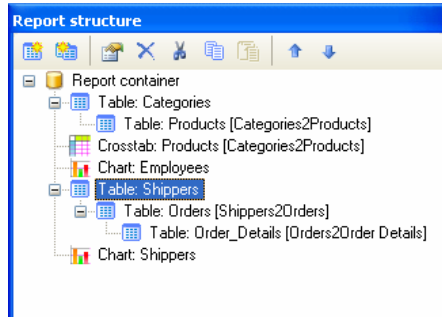


The sequence of the objects can be changed by pulling an object with the mouse into another position in the hierarchical list. If the Shift-key is depressed, the object will be attached to the object at that position, changing the hierarchy of the objects.

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 "4.5. ".

2.1.10. Table Structure

The fifth window is the hierarchical table structure, which is only available in some project types. All tables and their respective sub-tables of a table object are displayed with their table name, sort order and relation information 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.

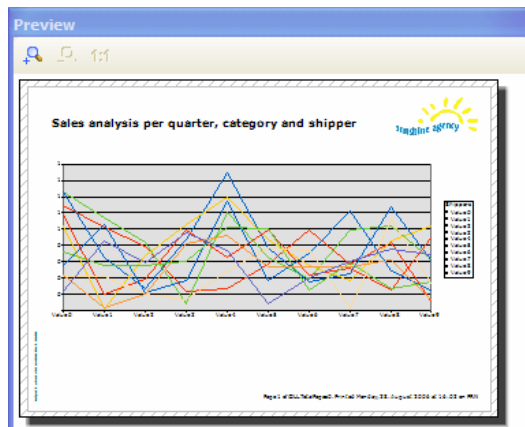


Using the buttons in the tool window, you can

- Attach a new table or sub-table
- Delete, cut, copy and paste tables
- Change the order of the tables

2.1.11. Preview

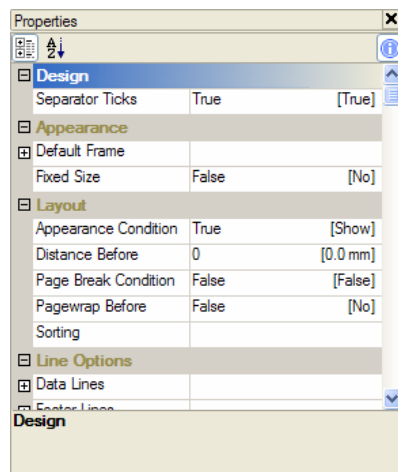
The Preview Window is available for you in order to check the layout of your workspace.



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.

2.1.12. 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".

A good example for a property is the position and size of an object. This can be defined in the "Layout" section of the Properties Window or in the workspace. The position of an object is defined by the properties of the x- and y- coordinates of the upper left corner, the object width and object height. You can define the size of an object in the Properties Window if you do not wish to do so by using the mouse

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.

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 button  and a formula dialog will open. You will find a detailed description of this function in Chapter "2.3. 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  button.

2.1.13. 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
Change to previous property or header	Cursor up
Open category	+
Close category	-
Open all property groups of a category or sub-groups	*

of a group	
Edit property	Tab
End editing	Shift + Tab
Open combo box	Ctrl + F4 (in Edit mode)
ENTER	Depends on value: open content dialog, invert Yes/No, open combo box

2.1.14. 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

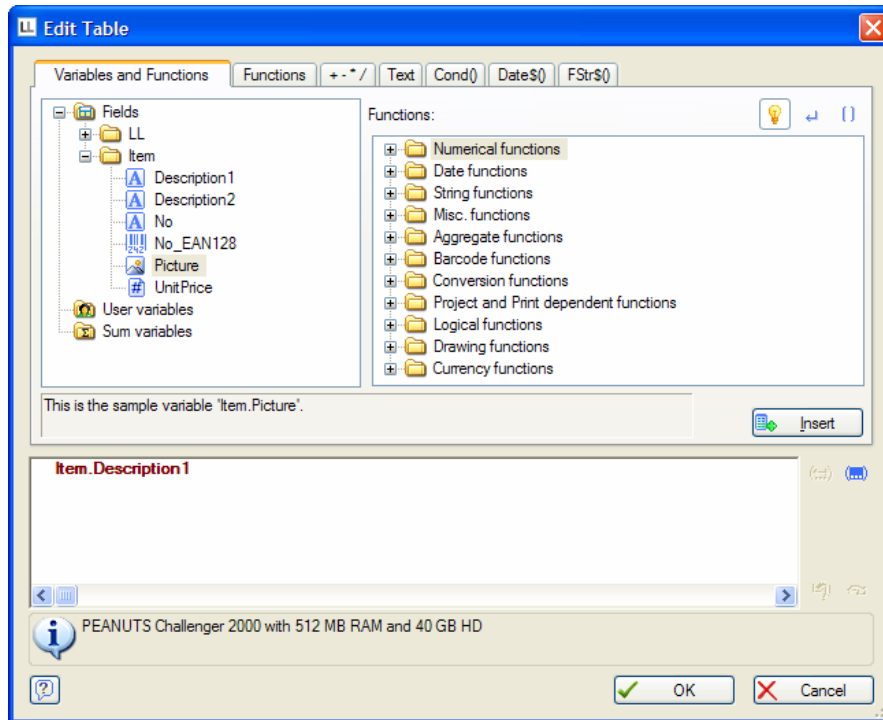
text can be linked and even edited. To help you with this there are **"formulas"** (for calculations with numbers) and **"expressions"** (for the combination of text and numeric values and for logical conditions). In formulas and expressions you can insert fixed text and variables in **"functions"** 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.

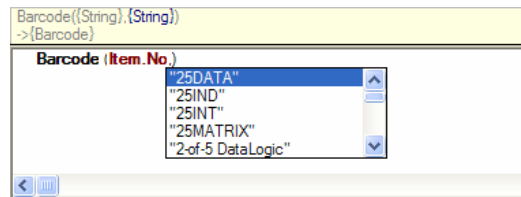
2.3.1. The Elements of an Expression

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.



Input Enhancements of the Formula Wizard

The formula wizard supports you with various input enhancements that can be (de)activated under **Options > Formula Wizard**:



- **Show Function Syntax:** if this option is selected, a tool tip appears in the formula assistant that describes the selected function, lists the necessary parameters and displays the result types.
- **Autocomplete for Functions:** if this option is activated, after entering 2 characters, all available functions that begin with those 2 characters will be shown and can be selected. This expedites the entry of the function.
- **Autocomplete for Parameters:** the assistant recommends values based upon the available variables and fields.
- **Syntax Coloring:** functions, parameters and operators are displayed in different colors. This will improve the transparency in complex formulas.
- **Automatic Type Conversion:** if this option is activated, variables and fields will be automatically converted at the time of entry of the expression so the type of data conforms to expectations.

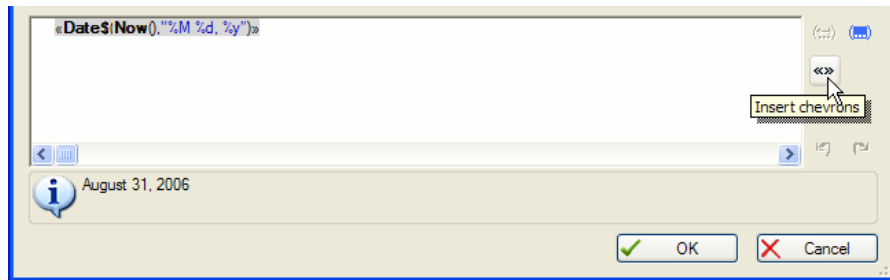
Expression Modes

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.



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
Function	a list of the functions available
+ -*/	a list of the available link operators
Text	a dialog for the entry of fixed text
Cond()	a special dialog for the definition of IF-THEN-ELSE conditions
Date\$()	a list of the available date formats
Fstr\$()	a list of the available number formats
Tab	the setting possibilities for tabs (only available in text objects)

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

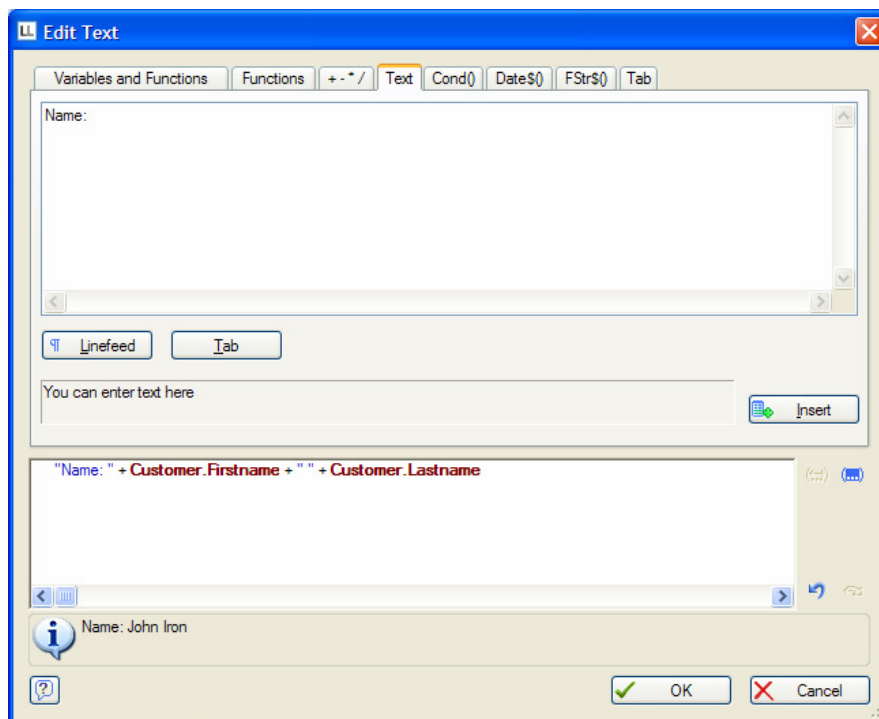
You may also insert variables by dragging the required variable to the object on the workspace where it should be inserted. The variable is then added automatically to the object as a new line.

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.



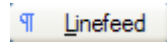
Enter the text and click on the "**Insert**" button to insert the text into the expression below. The text will be inserted automatically, depending upon the requirements,

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.

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

Inserting Linefeed



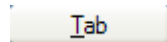
To insert a linefeed, click the button **"Linefeed"**. The linefeed is represented by a special character.

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



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 card "Tab". For more information please refer to Chapter "4.8. Inserting Text Objects".

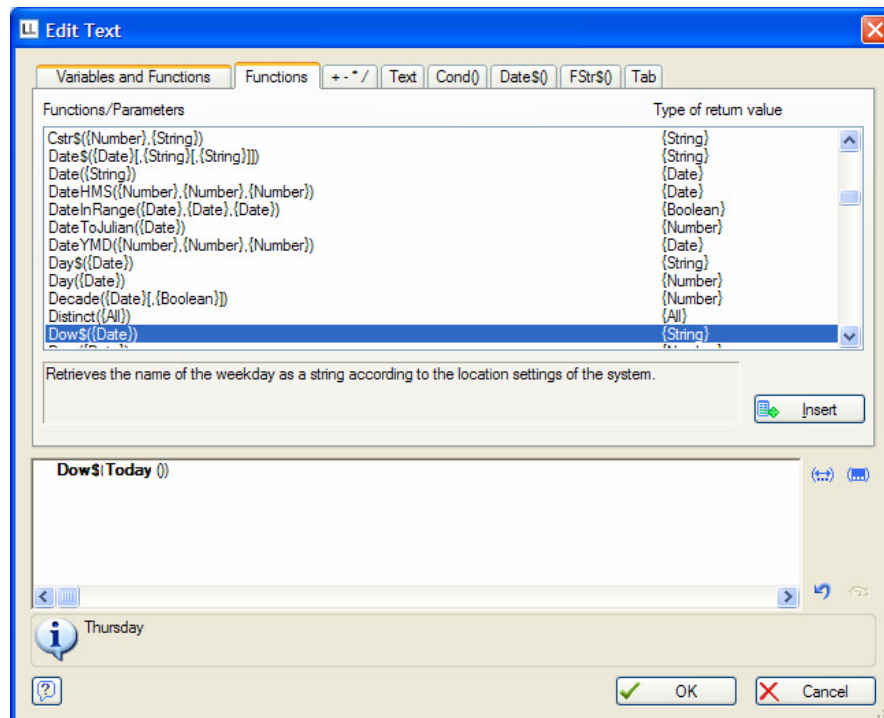
2.4. Working with Functions

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 displayed in the selected order. You may select to sort them by

- function group (e.g. "Numerical Functions", ...)
- type of return value (e.g. "Numerical Value",...) or
- type of the first parameter (e.g. "String",...)



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.

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.

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
0	" 0"
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, "#####&.#")

Value	Result
999.55	" 999.55"
1100	" 1100.00"
1099.5	" 1099.50"

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"

Some additional examples:

Formula	Result
FStr\$(3.142, "#####")	"3.142"
FStr\$(3.142, "#####")	"*****" (overflow!)
FStr\$(3.142, "(&####)")	" 3.142 "
FStr\$(3.142, "(&####)")	"(3.142)"
FStr\$(3.142, "+++&####")	" +3.142"
FStr\$(3.142, "---&####")	" 3.142"
FStr\$(-3.142, "---&####")	" -3.142"
FStr\$(3.142, "###&###")	"003.142"
FStr\$(3.142, "###.###")	"**3.142"
FStr\$(3.142, "\$\$\$.\$")	" \$3.142"
FStr\$(5003.1, "#,###&")	"5,003.10"
FStr\$(3.142, "##&###")	" 3.142"

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.

Let us assume that your data base contains both house and post office box numbers, and that you would like to print address labels using this data. Naturally, only the street address or the post office box number should be printed, but not both. You decide to use primarily the post office box number and only to use the street address when the post office box number is not available. With the help of an IF-THEN-ELSE condition, this will be no problem:

The screenshot shows a software window titled "Edit Text" with a blue title bar and standard window controls. Inside, there's a tabbed interface with tabs for "Variables and Functions", "Functions", "+ - * /", "Text", "Cond()", "Date\$", "FStr\$", and "Tab". The "Cond()" tab is selected. It contains three main sections: "Condition" with a text field containing "Empty (POBOX)" and an "Edit..." button; "Expression if condition is True" with a text field containing "STREET" and an "Edit..." button; and "Expression if condition is False" with a text field containing "P.O.: " + POBOX" and an "Edit..." button. At the bottom right, there is an "Insert" button with a small icon.

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

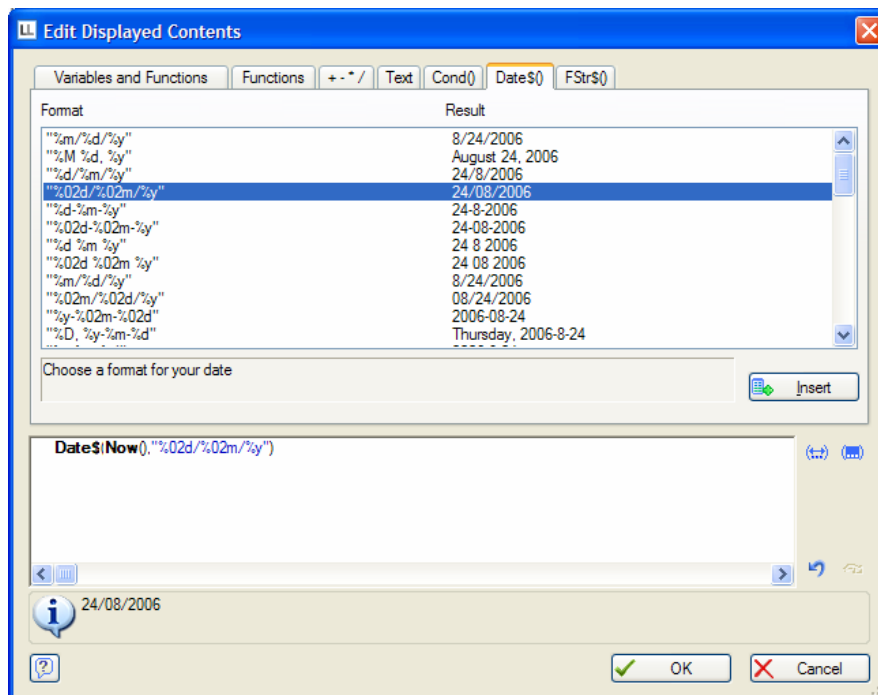
the "Edit" button starts an additional assistant to help you edit the appropriate part. In our example, we used the function Empty() to evaluate if the POBOX field is empty. When you press "Insert", the parts will be wrapped into the "Cond(...)" function and inserted into the expression line below.

In the field "True", enter the expression to be used if the entered condition is TRUE. If nothing should be done, enter two consecutive quotation marks (""). In our example, the condition is true if the POBOX field is empty, and in this case the STREET should be printed. In the field "FALSE", enter the expression to be used if the entered condition is FALSE. If nothing should be done, enter two consecutive quotation marks (""). In our example, the condition is false if the POBOX field is not empty, and in this case the POBOX should be printed. 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 Numbers, Dates and Currencies

Formatting Dates with Date\$()

Using the dialog with the card "Date\$()" 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 Today() is selected as date parameter which will display the current date. Of course you can replace it with any date value you like:

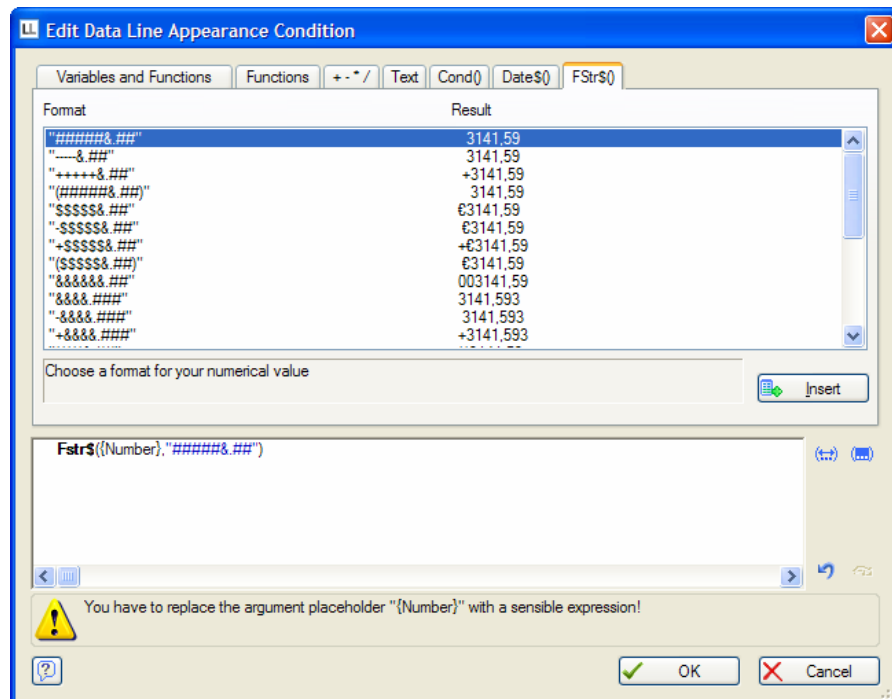
```
Date$(Today() + 7, "%d.%m.%y")
```

This would display the date one week from today.

In the above example, the date function "Today()" 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 "FStr\$()" 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.



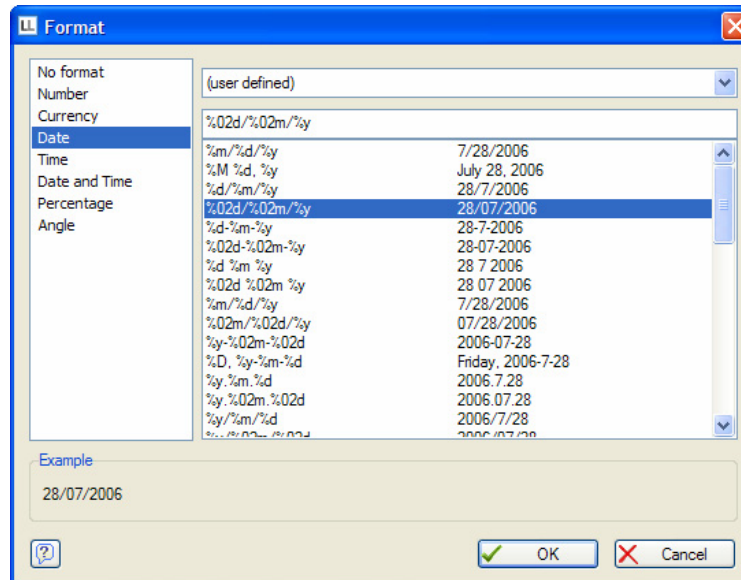
In this example, the price will be displayed with 6 digits including 2 decimals. If the number just before the decimal point is 0, only one '0' will be displayed.

Additional information about the function "FStr\$()" can be found in Chapter "2.4.3. Examples for the Use of Functions".

Formatting with the format editor

The format editor 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\$() and FStr\$() in the formula dialog.

With the format editor you can set the format for numbers, currency, date, time, percentage and angle. By default, the system settings are used.



2.5. Operators

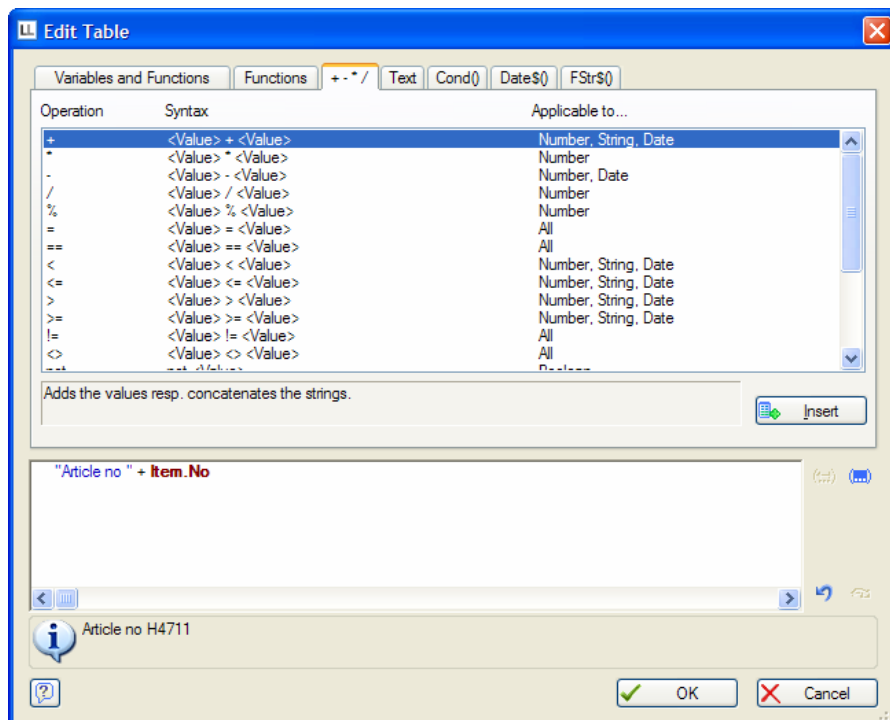
The card "+ - * /" displays a list of all available operators. These can be used to make calculations, combine variables and/or fixed text, as well as to perform logical comparisons.

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.



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*0.16)
```

With this, the value of the PRICENETTO field would have 16% 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. We recommend using the functions Sum() and Count() respectively. Further information can be found under Sum() and Count() in the chapter "5.2. List of Available Functions". Statistical analysis can be done directly using the aggregate functions Sum, Avg, Min, Max, Variance, StdDeviation and Count.

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).

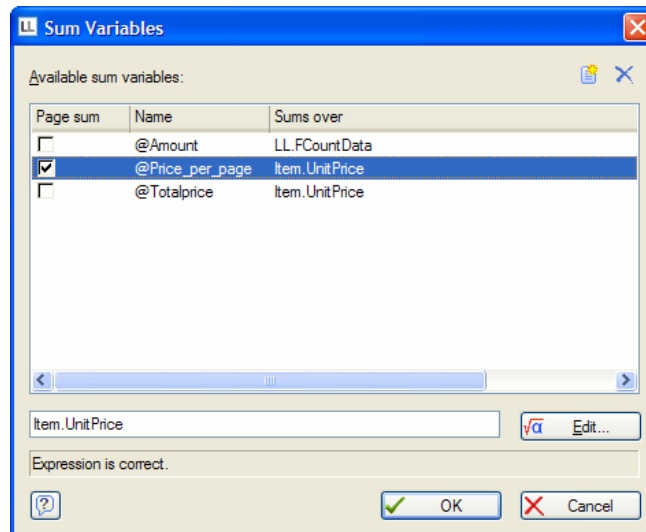
2.6.1. Defining Sum Variables

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".



To use sum variables of this kind ("@SumXX") in projects they must be assigned 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.

The easiest way is to simply select a numeric variable, e.g. PRICE. The formula for the result would be:

$@SumXX = @SumXX + <PRICE>$

ex. the sum of all previously printed prices is saved in the sum variable "@SumXX".

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

2.6.2. 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.

`@SumXX = @SumXX + 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.

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 "2.3. Variables, Formulas and Expressions"). In the card "Variables and Functions", you will also find the Sum Variables `@SumXX` 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**.

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 "2.3. 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** or 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 "2.3. Variables, Formulas and Expressions").

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

3.1. Creating or Opening a Project

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

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:

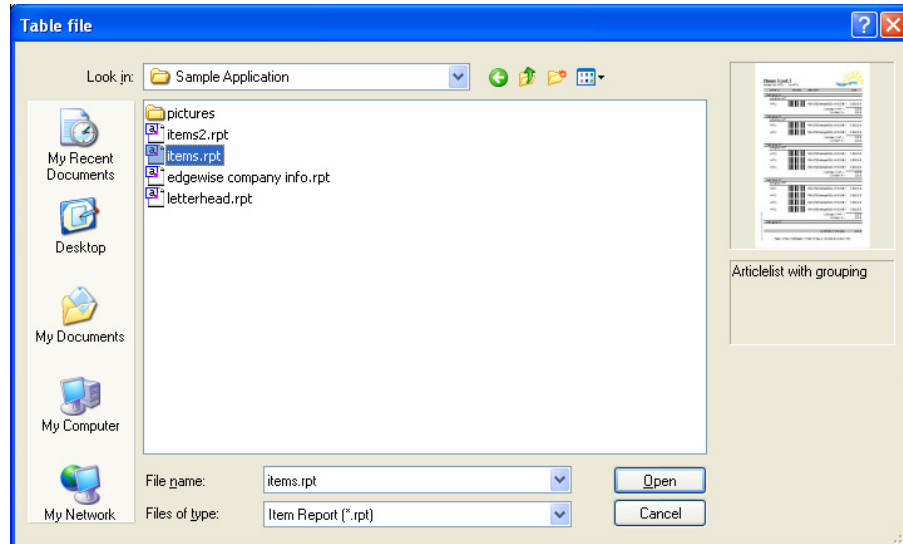
- page numbering
- page title (first page only or all pages)
- zebra pattern for the table
- create a summary
- select the data for the table object

3.1.3. Opening an Existing Project

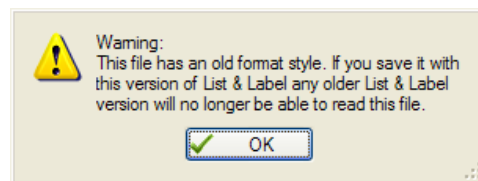


To open an existing project select **File > Open**. If your current project has unsaved changes, you will be asked whether you would like to save them.

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



If a description has been given to a project (see **File > Save As**) 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.

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.

Minimum page count

In the "Minimum page count" field you can set that a certain number of pages must be printed.

Fax and mail variables

You can send List & Label documents directly as a fax with the Windows 2000 / XP fax machine. The corresponding printer driver is set up automatically (in Windows 2000 /XP) when you have a fax-capable modem installed on your computer. When sending a fax, additional information is needed to be able to address the fax, i.e. at least the fax number must be present as information in the document. These fax variables are defined directly in the properties window of the project. The properties window is displayed when no objects are selected in the workspace.

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 Settings".

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. Printer Selection

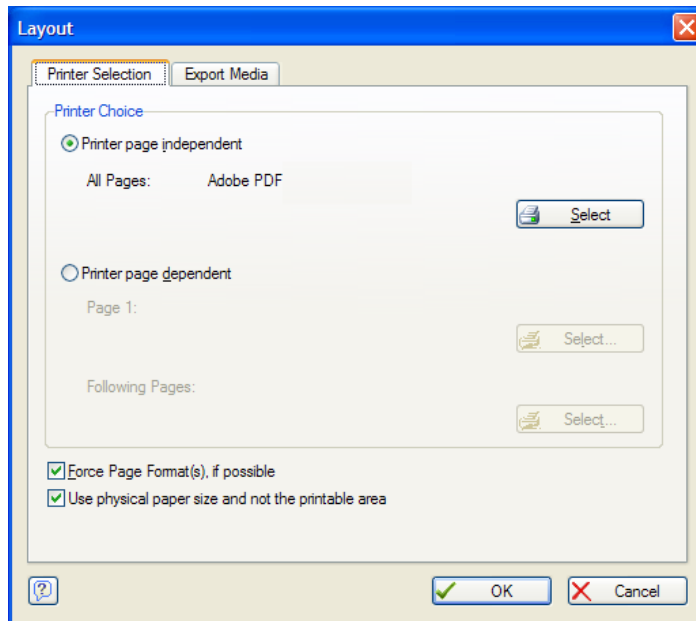
Selecting Printer and Paper Format

Click the "Select" button to define the paper size and orientation and select the printer for the project.

You can also either open a predefined label format, or freely create your own. The paper orientation is saved with the project and does not change the Windows default printer settings.

If the project consists of only one page or requires one printer specification for all pages select "**Printer page independent**".

For projects with more than one page it can be useful to select a different printer specification or different printer. If you check the "**Printer page dependent**" option, you can select different printers (-options) for the first and the following pages. In this way you can use corporate letter-head paper for the first page and normal paper for the following pages.



Force Page Format(s), if possible

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.

Use physical paper size 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.

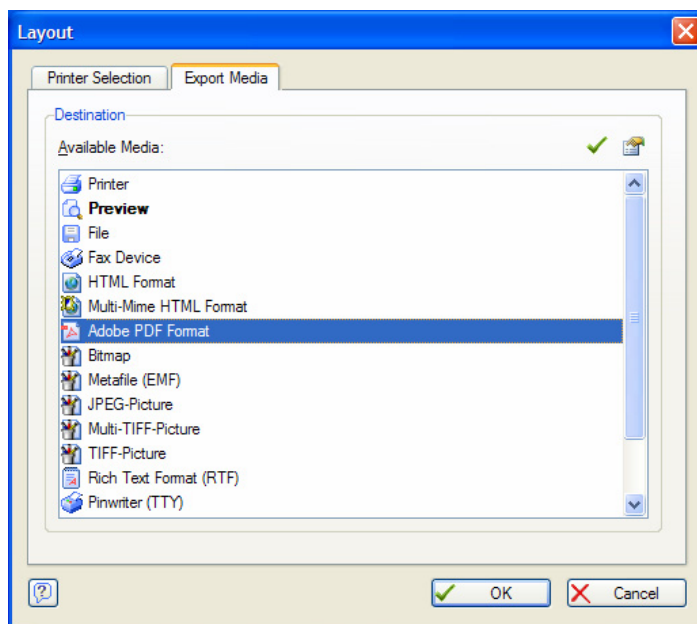
3.3.2. Export Media

Depending on the application, different export possibilities of List & Label are listed. You can configure the output into different export formats by the corresponding option dialogs.



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

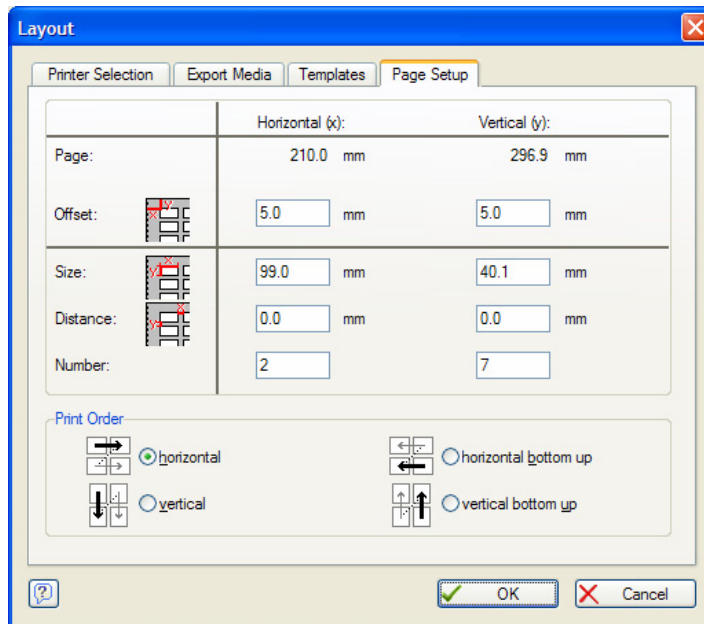
A previously or manually defined label format can be loaded in the page layout for labels.

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.

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:



Setting the 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.

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

In addition to the printer and paper format, in a label project you can also define the print order of the labels on a label sheet. 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.

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.

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.

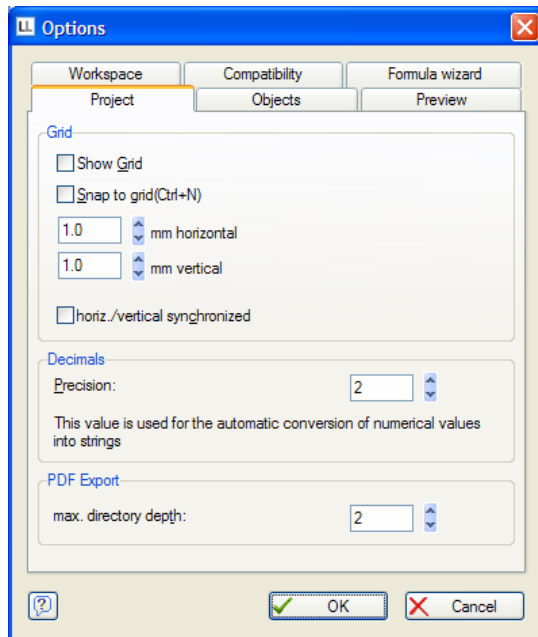
With the option "**Snap to Grid**" you ensure that objects are not freely inserted or moved on the workspace but only along the (sometimes invisible) grid lines.

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\$).

Exporting to PDF

Here you can set the maximum directory depth of group headers, which are turned into PDF bookmarks when exporting to PDF.



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.

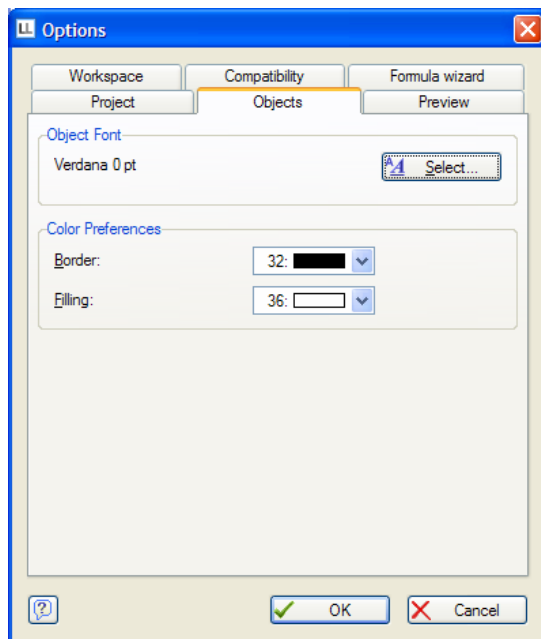
Object Font

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.



3.4.3. Options for the Preview

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

Setting Colors for the Preview

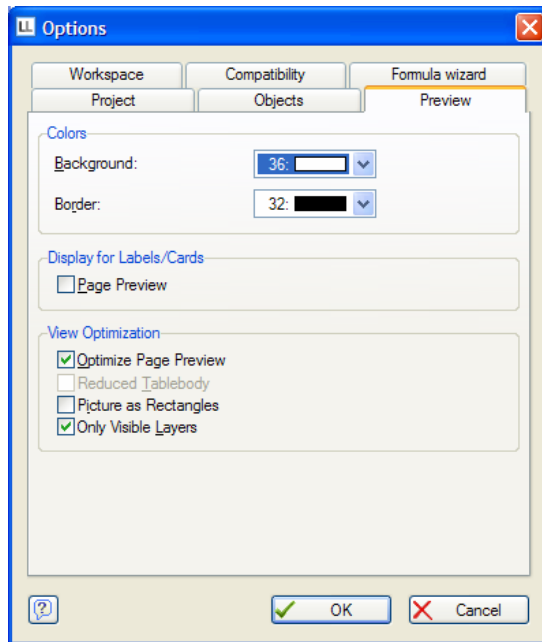
In the "Colors" group you can define the color for the background of the preview window using the combo box "Background". With the combo box "Border" you can select the color of the simulated paper border in the preview.

View of Label/Card Projects

Here you can select if the preview should contain only one label or the complete page (only available for label or card projects).

Optimized View

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



3.4.4. Options for the Workspace

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

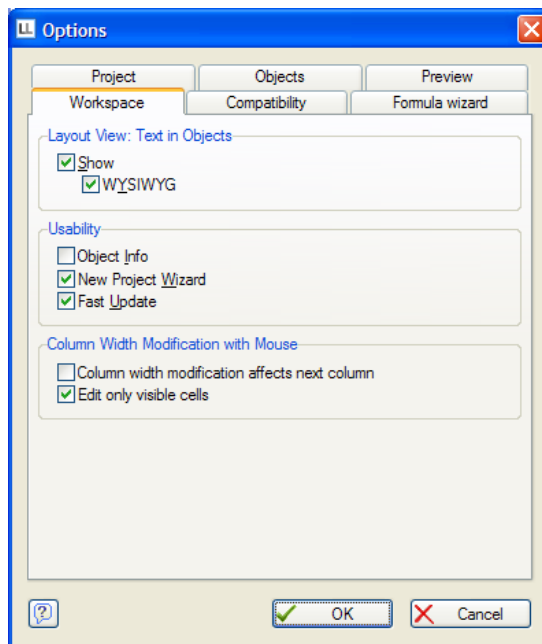
Displaying Objects

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.



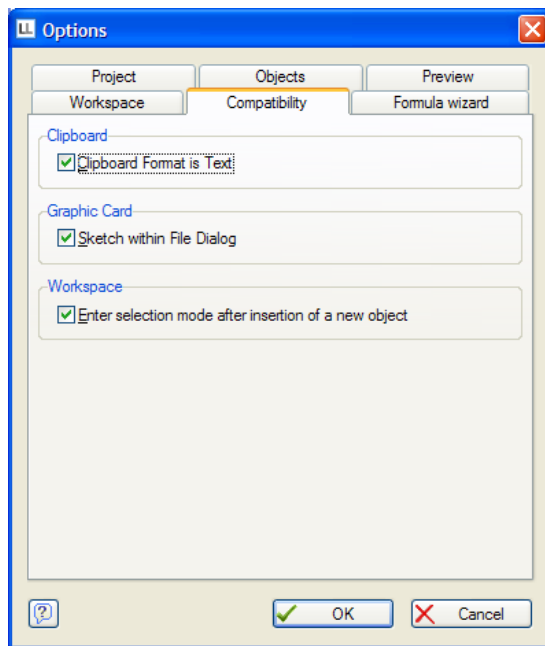
Changing Column Widths Using the Cursor

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 ± 2 mm 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.

3.4.5. Compatibility Options

Here you can set various options which concern the compatibility with other components and previous versions of List & Label.



Clipboard Format is Text

This option defines the format in which objects will be copied to the clipboard in the List & Label Designer. This option only takes effect when pasting Designer objects into other applications.

When the option is set, the internal definition of the object can be inserted via the clipboard into other programs.

When the option is not set, List & Label uses its own clipboard format.

Sketch within File Dialog

This option defines whether a sketch of the currently selected project is shown in the file selection dialog. When the option is set, a sketch will be created each time you save the project.

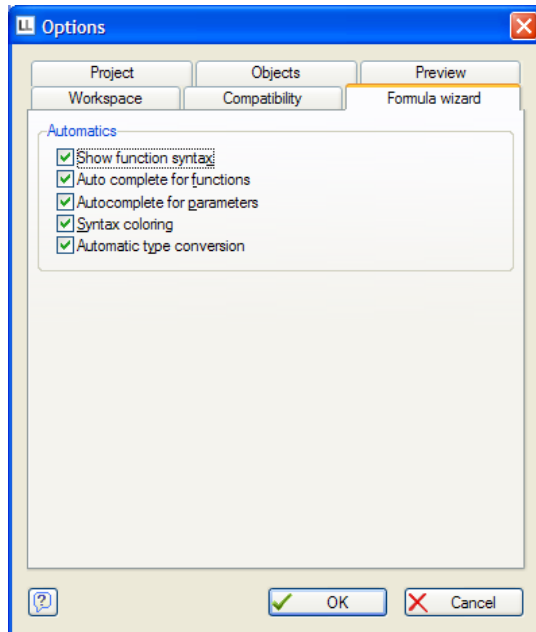
Some very old graphic card drivers unfortunately have problems in the support of this function. In this case it is helpful to switch off the option.

Selection mode after object insertion

- 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.

3.4.6. Options for the Formula Wizard

The Formula Wizard offers many ways of simplifying the creation of formulas and functions. For further details regarding working with formulas, look at “2.3. Variables, Formulas and Expressions”. In this interface, you can deactivate the functions if desired. See “Input Enhancements of the Formula Wizard” for specific details on the options.

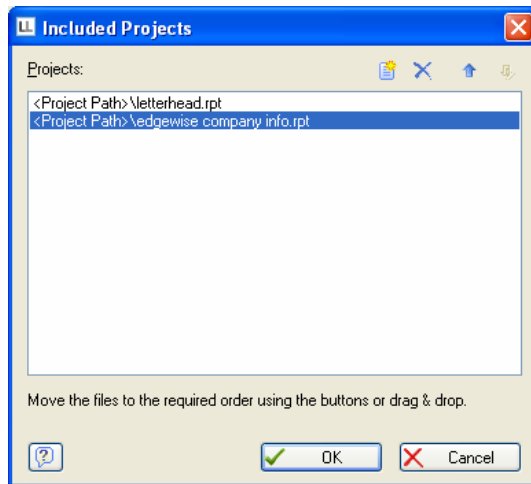


3.5. Including Projects

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. 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



List & Label offers two ways of printing: using the test print in the designer which uses some sample data values, and the final print with the "real" data, initiated by the application.

3.6.1. 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 with Frames



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.

Print Sample without Frames



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.

Print Sample 1st Page



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.

Print Sample Following Page



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.

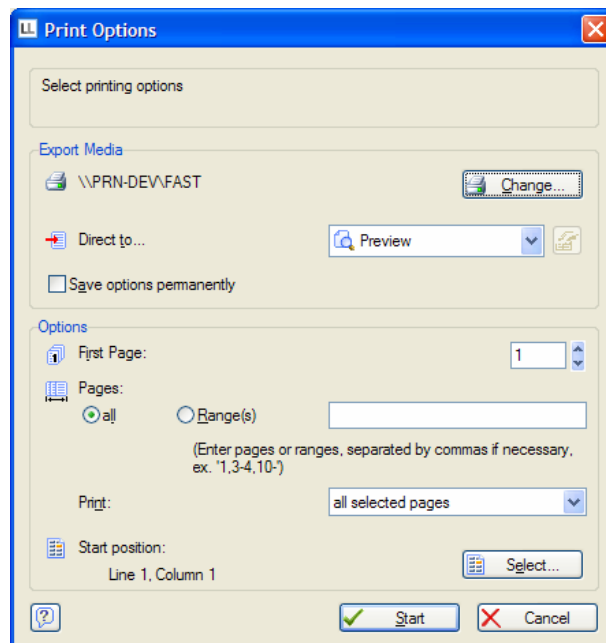
This distinction is irrelevant if no objects have been created with the appearance condition "Lastpage()".

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 "4.5. ".

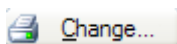
3.6.2. Printing to Real Data Preview

In addition to the Sample Print using sample data, List & Label offers a real data preview. But in contrast to the sample print, this can't be used in the List & Label Designer, but 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.

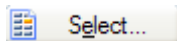


The dialog shown on the left will be presented which allows you to select the destination printer and - in case of label projects - the starting label.

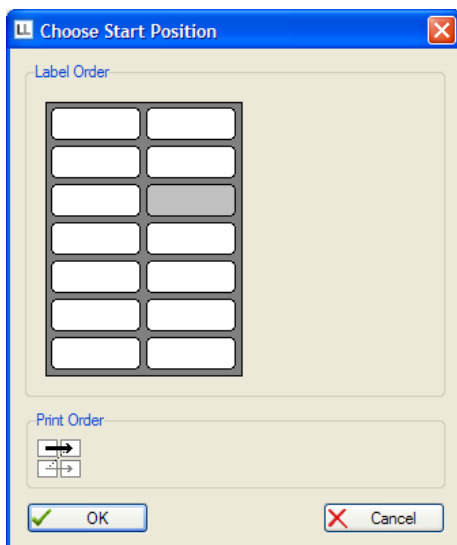


In the category **"Export Media"** you can, by using the **"Change..."** button select a different printer or different printer options than the defaults defined in your project in the Designer. This setting can be either one time (option **"Save Options Permanently"** not selected), or permanently saved to your project as the default (option **"Save Options Permanently"** selected). Depending on your application, various formats are available in **"Direct to"**.

In the category **"Options"**, you can enter the number of copies, the starting page number and the page range to be printed.

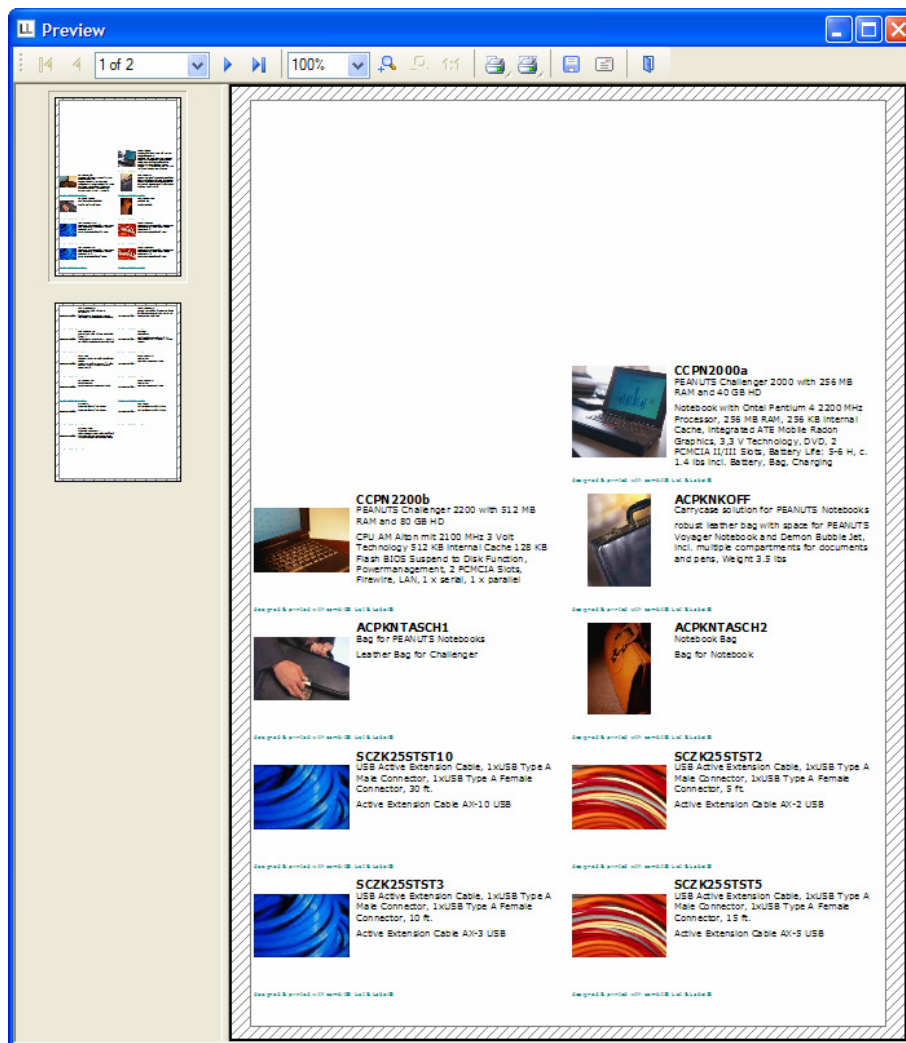


For label projects, you can additionally select the position of the first label on the first page. This is useful when printing on pages where labels have already been used.



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.

The entire print data can be viewed and printed in a preview window. The non-printable margins will be shaded. On the left side of the Preview window, you will find the individual pages of your project in reduced size as "thumbnails". This allows a very fast, direct navigation through the document. When you click on a thumbnail, the appropriate page will be displayed in the preview. The number of copies set in the printing options window is ignored in the preview and only becomes relevant when printing begins.



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

The toolbar can, as with the List & Label Designer, be docked to all four sides of the work space or left floating as an individual window.

The 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.

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 (*.LL) e.g. through your MAPI-Mail-Client to an email recipient. In order to view the preview, 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"**.

4. Objects

This menu contains various functions for the editing of objects. A description of the properties that all objects have in common can be found in Chapter "4.6. Common Object Properties".

When inserting objects, always proceed in the following way:

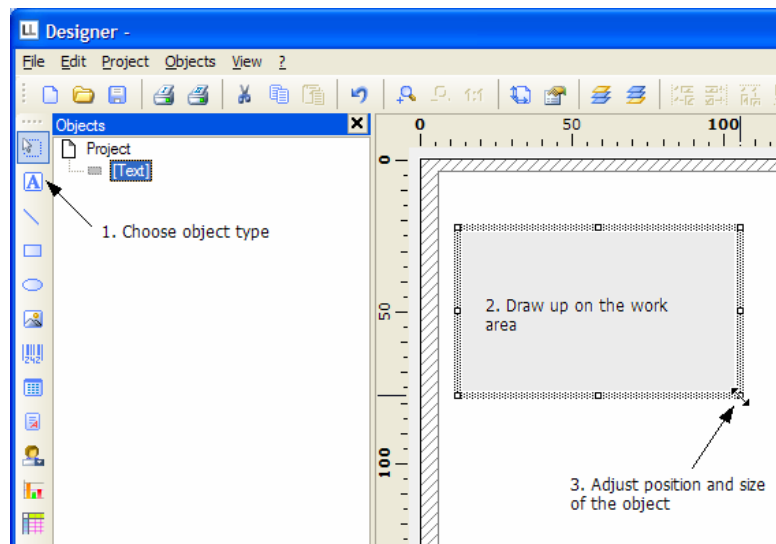
- Create an object as described in "4.1. Inserting Objects". The object will be selected automatically.
- Edit the properties of the object with the appropriate tool window and "Properties".
- Hit ENTER in the work space or choose Contents from the Objects menu in the context menu or the properties list. A window appears in which you can define the internal setup of the objects.

4.1. Inserting 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.

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.

- Select the desired object type. The mouse pointer will change to a crosshair.
- 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.
- 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 of Chapter "4. Objects"
- 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.

Selecting Multiple Objects

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

Selection Mode

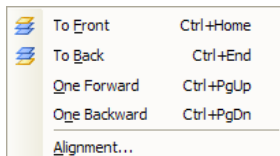
When the command **Objects > Select** is selected, a sub-menu opens with which you can select from various select modes.

- 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

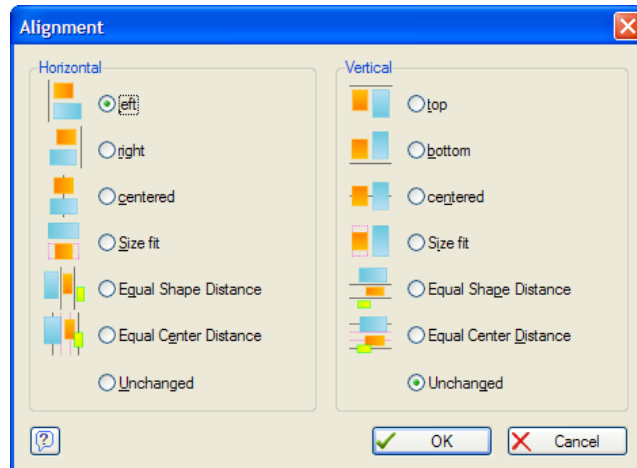
Using the command **Objects > Arrange > One Forward** or the shortcut CTRL+PAGEUP, the selected objects will be moved one position up in the stack of sheets.

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** you can align multiple objects in relation to each other. A minimum of two objects must be selected to use this command.



Each direction (horizontal and vertical) has 6 alignment methods. Select the ones you like, even horizontal and vertical simultaneously. If the objects should not change in one direction, leave the method of this direction to "Unchanged".

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.

Unchanged

Select this if you do not wish to make any changes in the corresponding direction.

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

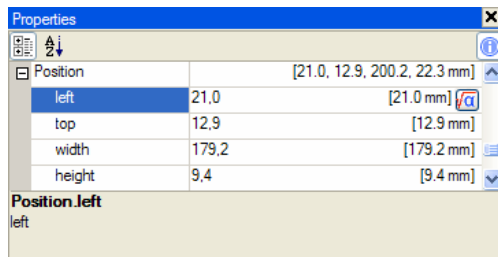
- Select the desired object.
- When the mouse pointer is placed over the frame, it changes into a bi-directional arrow. The size can be changed by depressing the left mouse button and pulling in one of the displayed directions. To simultaneously change the objects size in both the horizontal and vertical,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

Repositioning an Object

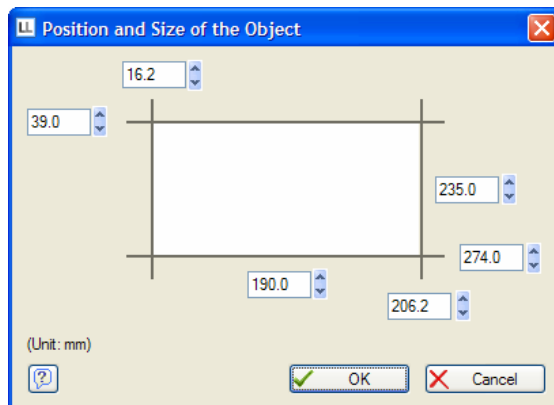
- Select the object you require.
- Click into the object to reposition it. If the mouse pointer is positioned over the selection frame, a four directional arrow will be displayed.
- Release the mouse button when the object has reached the position you require.

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.



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.

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

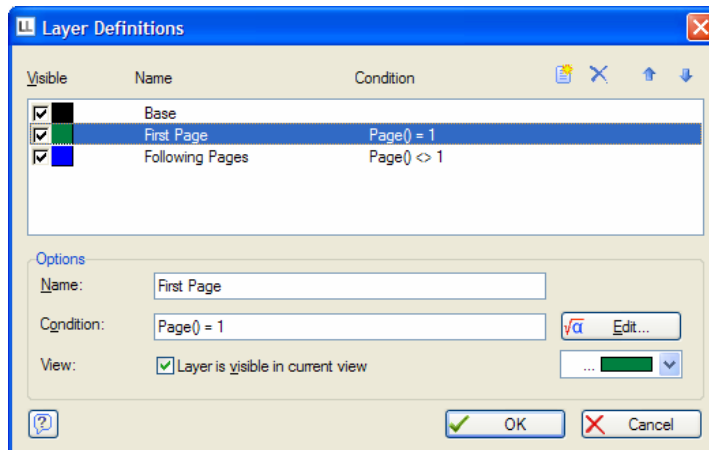
Layers can be defined using the command **Project > Layer Definitions** or by using the command **Layer Definitions** in the context menu of Layers tool window. The defined layers apply to the current project.

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.

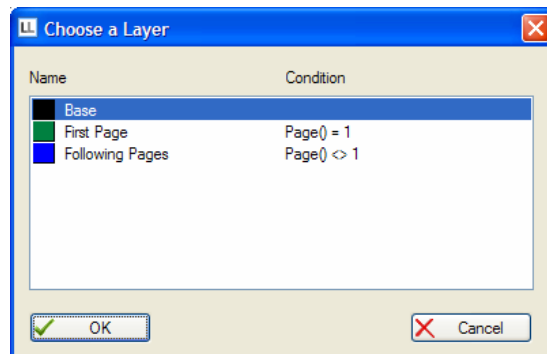
Additionally, logical expressions can be used as appearance conditions.

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".

4.3.2. Assigning Layers

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

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.

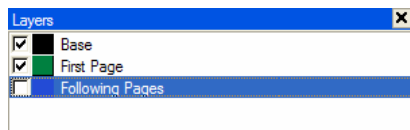
The original object stays on its layer. A copy of the object is made on the target layer.

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".

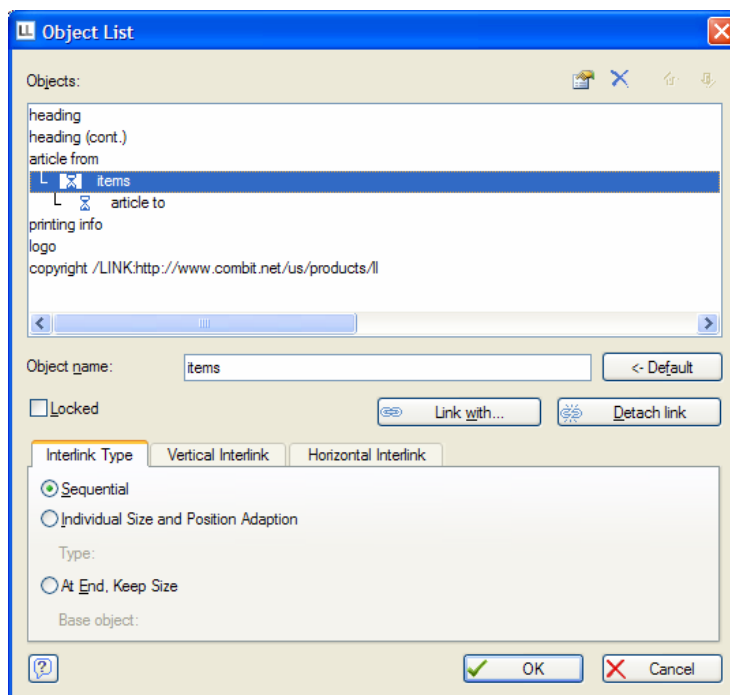
You can turn layers on and off by double-clicking on the corresponding layer in the layers window or by clicking directly on the checkbox corresponding to the layer.



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

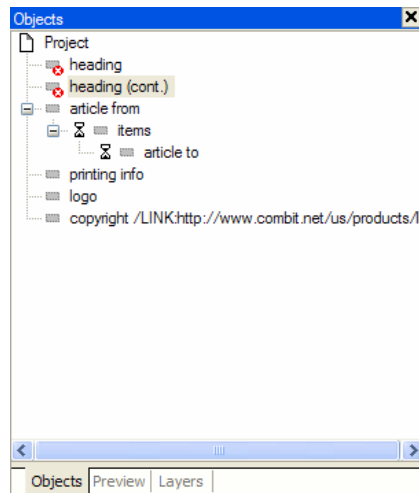


Editing or Deleting Objects

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").

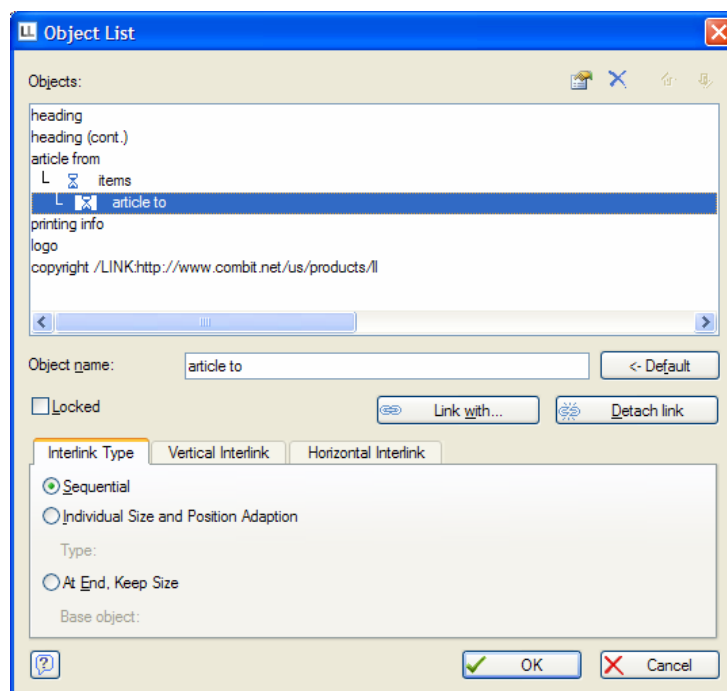
There are three types of interlinking:

- 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 an 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)

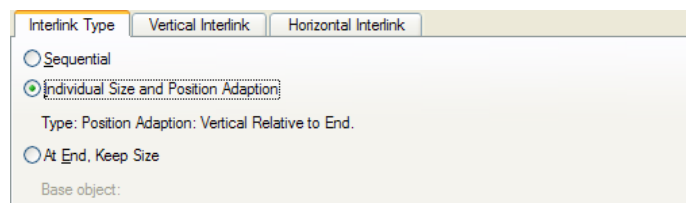
Both types of interlinking are relating to the used space of the parent object, i.e. the linked object will be subordinated to the parent object spatially, which means it will be subordinated in position and size. This automatically creates a sequential link too.

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() $\geq$ 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.

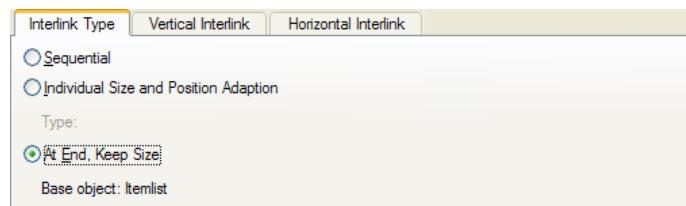
- Option "inverse": The size of the child object changes inversely proportional to the size of the parent object. This is an especially useful option: if the parent object becomes 1cm shorter, the child object becomes 1cm longer.

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.

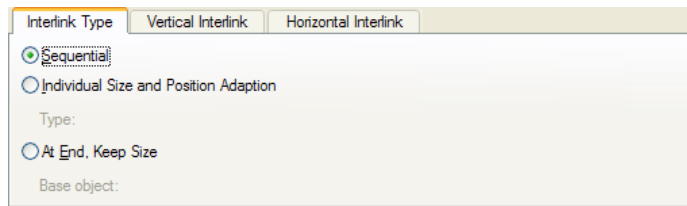
Example 1: You are writing text and want to have the number of letters and words written on every page counted. The linked object only knows this when the text (text object) has been printed. The sum can then be printed at the end of the page.

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.

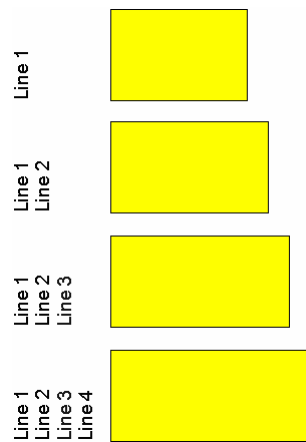
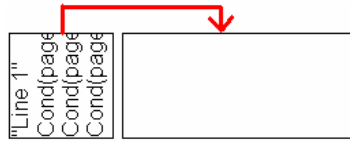
When the sequential interlink is selected, none of the options for position and size adaption are available in the interlink field.

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



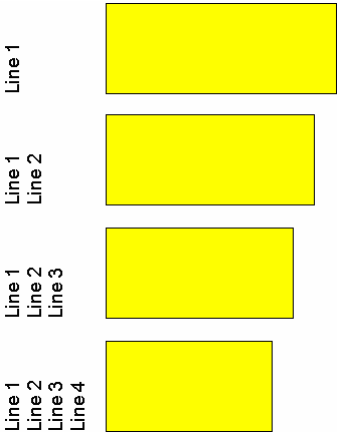
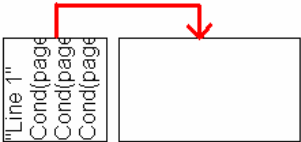
4.5.5. Examples for Individual Interlinks

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



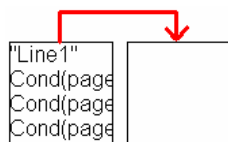
Interlink Type		Vertical Interlink		Horizontal Interlink	
Position Adaption:				Size Adaption::	
☒ Horizontal	☐ Relative to Begin	☐ Relative to End	☐ To End	☒ Horizontal	☒ Proportional
					☐ Inverse

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



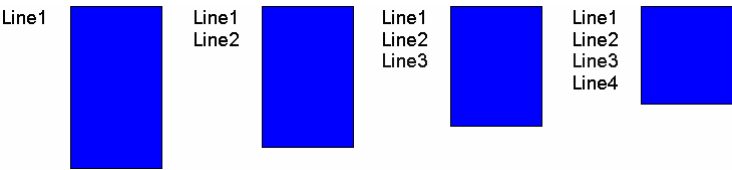
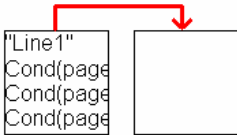
Interlink Type	Vertical Interlink	Horizontal Interlink
Position Adaption:		
☒ Horizontal	☐ Relative to <u>B</u> egin	☒ Horizontal
	☐ Relative to <u>E</u> nd	☐ Proportional
	☐ To End	☒ <u>I</u> nverse

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



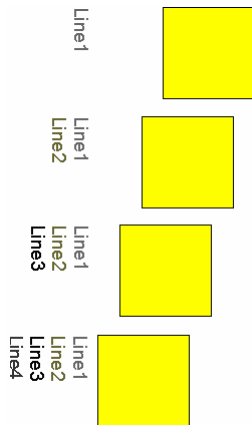
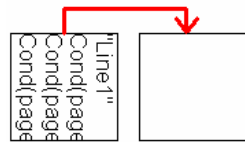
A screenshot of the "Interlink Type" dialog box. The "Vertical Interlink" tab is selected. Under "Position Adaption:", the "Vertical" checkbox is checked. Under "Size Adaption::", the "Vertical" checkbox is checked, and the "Proportional" radio button is selected. The "Horizontal Interlink" tab is also visible.

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



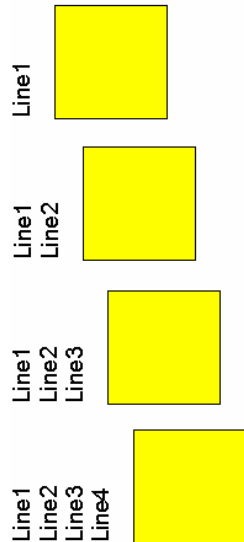
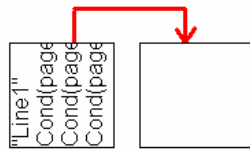
Interlink Type	
Vertical Interlink	Horizontal Interlink
Position Adaption:	
☒ Vertical	☐ Relative to Begin
☐ Relative to End	☐ To End
Size Adaption::	
☒ Vertical	☐ Proportional
☒ Inverse	

Example interlink "Position Adaption: Horizontal, Relative to Begin": The linked object (rectangle) changes in position depending on the top left corner of the parent object (text object). You can see that the linked object (rectangle) adjusts to the position of the parent object (text object). Because both objects are linked together "relative to begin", they behave in the opposite way to the "relative to end" interlink. Here, the parent object has been turned by 180° compared with "relative to end".



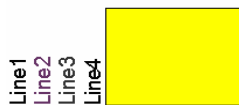
Interlink Type	Vertical Interlink	Horizontal Interlink
Position Adaption:		
☒ Horizontal		
☒ Relative to Begin		
☐ Relative to End		
☐ To End		
Size Adaption::		
☐ Horizontal		
☐ Proportional		
☐ Inverse		

Example interlink "Position Adaption: Horizontal Relative to End": The linked object (rectangle) changes in position depending on the bottom right corner of the parent object (text object).



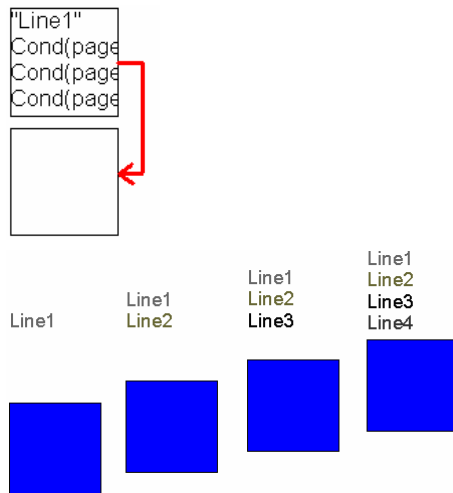
Interlink Type		Vertical Interlink	Horizontal Interlink
Position Adaption:			
☒ Horizontal		☐ Relative to Begin ☒ Relative to End ☐ To End	
		Size Adaption:: ☐ Horizontal ☐ Proportional ☐ Inverse	

Example interlink "Position Adaption: Horizontal To End": The linked object (rectangle) changes its position depending on the parent object (text 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.



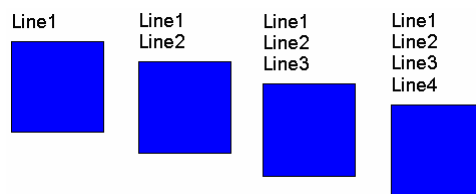
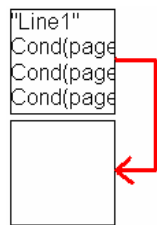
Interlink Type		Vertical Interlink	Horizontal Interlink
Position Adaption:			
☒ Horizontal			
☐ Relative to Begin			
☐ Relative to End			
☒ To End			
Size Adaption::			
☐ Horizontal			
☐ Proportional			
☐ Inverse			

Example interlink "Position Adaption: Vertical Relative to Begin": The linked object (rectangle) changes its position depending on the top left corner of the parent object (text 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.



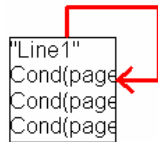
Interlink Type	
Vertical Interlink	Horizontal Interlink
Position Adaption: ☒ Vertical ☐ Relative to Begin ☐ Relative to End ☐ To End Size Adaption: ☐ Vertical ☐ Proportional ☐ Inverse	

Example interlink "Position Adaption: Vertical Relative to End": The linked object (rectangle) changes its position depending on the top left corner of the parent object (text object). If the text object moves down a row, the dark rectangle moves downward proportionally too.



Interlink Type	Vertical Interlink	Horizontal Interlink
Position Adaption:		
☒ Vertical		
☐ Relative to Begin		
☒ Relative to End		
☐ To End		
Size Adaption::		
☐ Vertical		
☐ Proportional		
☐ Inverse		

Example interlink "Position Adaption: Vertical To End": The linked object (rectangle) changes its position depending on the parent object (text object). This causes a change in size, since the top edge of the linked object will be changed because of the parent object but the position of the linked object does not change. The parent object moves down and the linked object only changes its position in that it shrinks. The linked object must overlap (in the example, the text is in the foreground), otherwise this type of interlink has no effect.



Interlink Type	
Vertical Interlink	Horizontal Interlink

Position Adaption:
☒ Vertical
☐ Relative to Begin
☐ Relative to End
☒ To End

Size Adaption::
☐ Vertical
☐ Proportional
☐ Inverse

4.5.6. Advanced Interlinking

The object list

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.

Objects with word wrap

List & Label objects on the base layer are normally printed on every page of a multi-page document. There are 4 exceptions to this: tables, text, formatted text and HTML.

These objects can cause a page break and are then printed over several pages. (The OLE container is not one of these, even though it can contain multi-page documents. Only the first page of the document is ever printed.)

Report Container / Table: is strictly only printed once per project.

Text, formatted text, HTML: These can be printed on several pages or on every page of the project. The selection is made with the object property "Page Break".

LastPage()

List & Label works through its print data sequentially. The application passes one "Data row" after the other and it is unknown during printing how much more data is still to come.

Therefore, it is difficult to find out if the last print page has been reached. The function LastPage() can for this reason only return "True" in very particular circumstances:

- in the footer of a table
- with an object linked with a table, as long as a page break has not occurred while printing the attached objects.

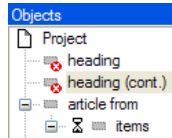
4.6. Common Object Properties

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 objects 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.

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.



Hint: 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 "2.3. 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. Pagewrap Before Object Output

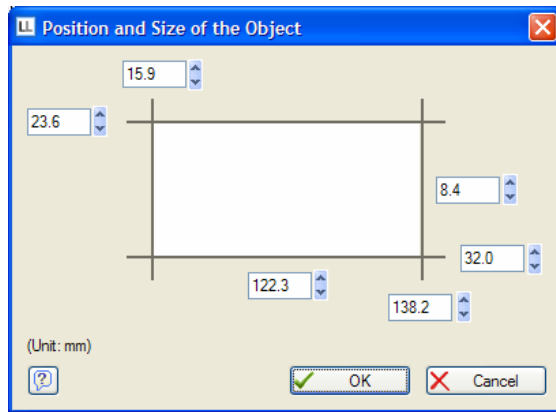
Every object may initiate a page wrap prior to being printed; this means that the object will be printed on a new page.

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

4.6.5. 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

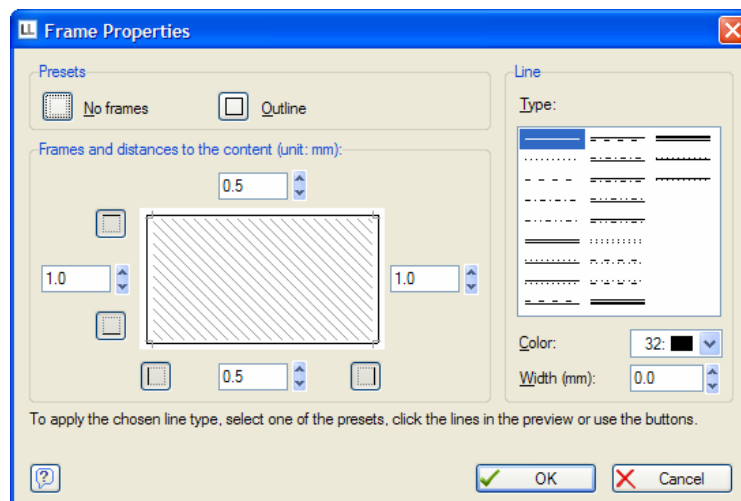


4.6.6. Frame / Background

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:		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 (see "4.6.7. Colors")		
Line Type	Choice of the line type.	Choice of pre-	Choice of pre-defined

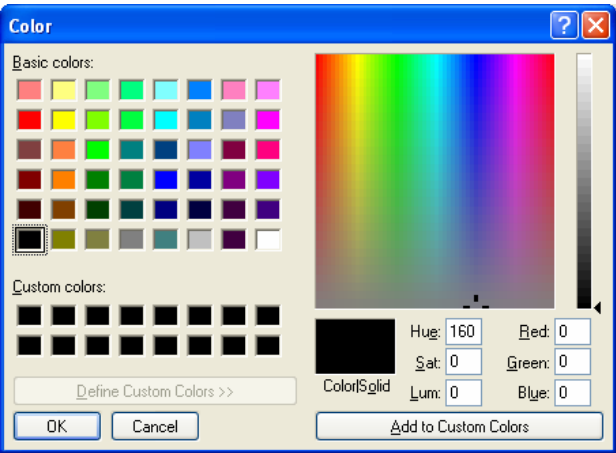
		defined lines (20)	lines (20)
		Formula	Formula Dialog
Width	Line Width	Number	
		Formula	Formula Dialog



4.6.7. Colors

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

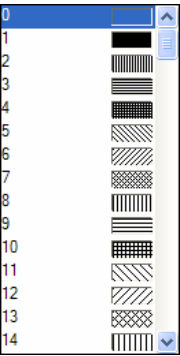
Property	Description	Value	Description
Color	The color can be selected from either the list of predefined colors or by using a formula. (1) The color consists of a Hue, Saturation and Lightness portion and is selected using the HSL function. (2) The color consists of a red, green and blue portion and 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. The three portions are mixed additively, with RGB (0, 0, 0) denoting black, RGB (255,255,255) white. (3) A color dialog is also available.	 	Choice of predefined colors and the Formula dialog Color dialog



4.6.8. 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.

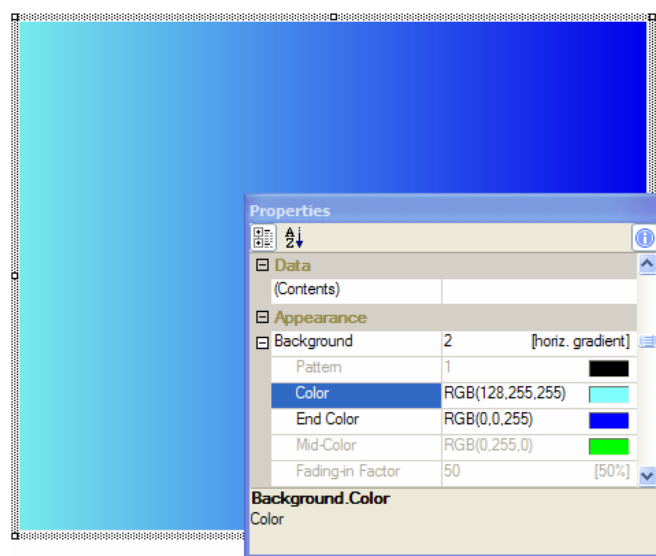
Property	Description	Value	Description
Pattern	Choose a pattern from many predefined patterns. A value is always defining a pattern. This property is only analyzed if "Filled" is set to "True".	Number	Predefined sample formula dialog
		Formula	



4.6.9. Filled / 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
		Formula	Formula Editor

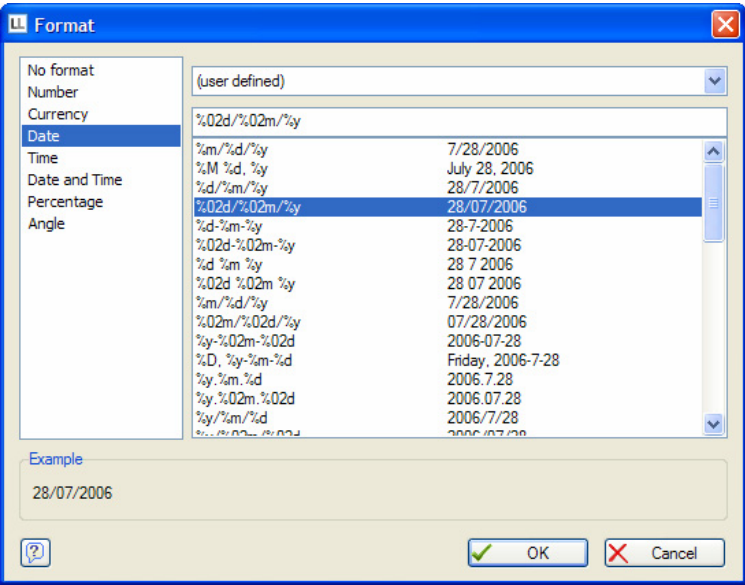


4.6.10. Formatting

The format editor is an alternative to formatting with the functions Date\$() and FStr\$() in the formula dialog. With the format editor you can set the format for numbers, currency, date, time, percentage and angle. By default, the system settings are used.

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\$() and FStr\$() in the formula dialog.

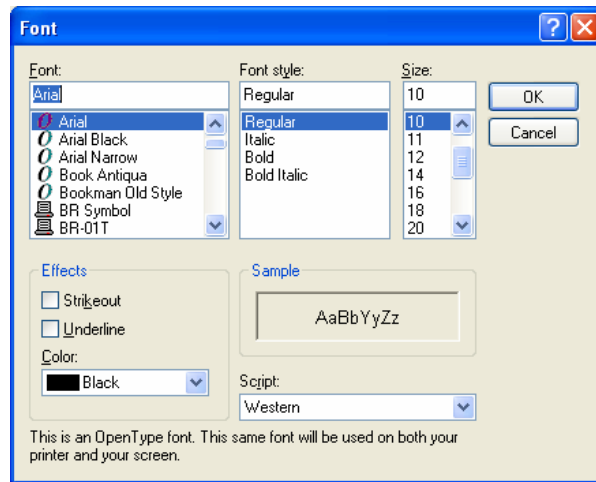


4.6.11. Font

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.		Font dialog
Default	The default font is used instead of the one set	True False Formula	Default font No Formula dialog
Name	Selection of font. All installed fonts are	List	Font

	displayed.	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 Formula	Default size Formula dialog
Width	Sets the width of the font 0 means standard width, otherwise the average character width will be used.	Number Formula	Width Formula dialog
Bold	Turns on or off the text setting "Bold"	True False Formula	Yes No Formula dialog
Italic	Turns on or off the text setting "Italic"	True False Formula	Yes No Formula dialog
Underline	Turns on or off the text setting "Underlined"	True False Formula	Yes No Formula dialog
Strikeout	Turns on or off the text setting "Strikeout"	True False Formula	Yes No Formula dialog
Color	Font color		Color dialog Selection of predefined colors



4.6.12. "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.

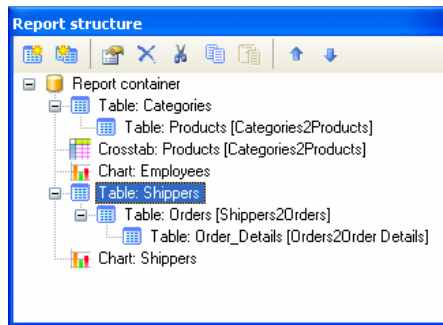
If available, a dialog window will open that may have a different appearance depending on the type of object.

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

4.7. Report Structure

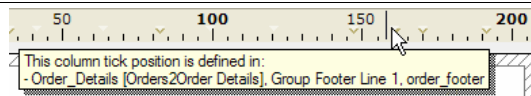
Multiple (sub)tables in the table object are only available when they are being used by an application that defines appropriate relationships.

A table object can contain various sub-tables. This means it is possible to define (sub)reports with almost any inter-table relationship.



4.7.1. Characteristics of Elements

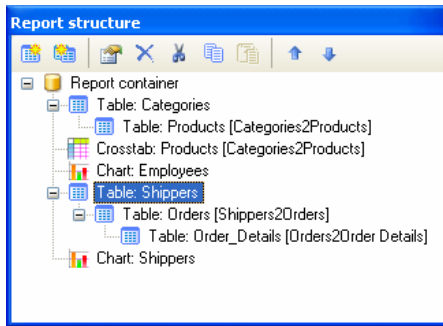
The base properties of objects will change when using multi tables. Some settings (e.g. "Locked") are only available with base objects. Others (e.g. "Fixed Size") are only available in sub tables. To view the properties of sub tables, select the table in the tool window "Table Structure". In addition to the table object properties, some other settings become available when using multi tables.

Property	Description	Value	Description
Separators	To easier align (sub)tables and columns accurately, you can show separator tick marks to the ruler.	True	Yes
		False	No
			
Distance before	Vertical distance from the previous table (does not apply when the table is at the beginning of a page).	Number	Formula dialog
		Formula	
Sorting	The order in which the table is presented. All available sorting choices are displayed in a list.	List of available sorting options.	
Pagewrap before	When the value "true" is returned, a page break will be initiated before the sub table is printed.	True	Wrap
		False	None
		Formula	Formula dialog

4.7.2. Working with Report Structure

Further tables and sub-tables and the desired hierarchical structure can be defined in the tool window "Table Structure". Here you can see the layout of the entire project,

e.g. on the highest level is the shipper, below this is the order, and on the lowest level the items of the order.



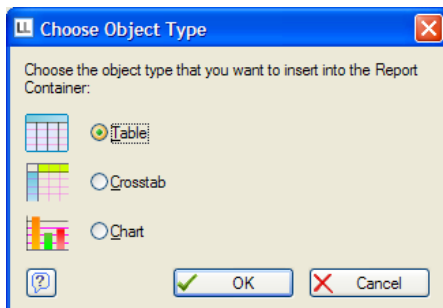
This can be continued as required. You can insert further tables, crosstabs and charts to fulfill almost any requirement. The table description is composed of the table name [relation name, sort name]. The most important commands for the various tables are available through context menus in the workspace, and a toolbar.

Using the buttons in the tool window, you can

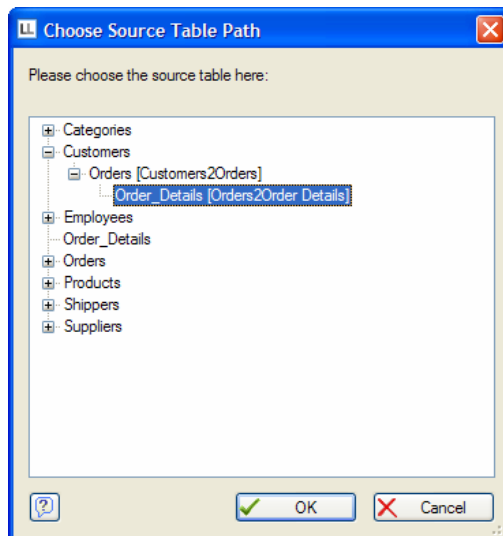
- Attach a new table
- Attach a new sub table
- Delete, cut, copy and paste tables
- Change the sequence of the sub tables of the same hierarchical level

Inserting multi-table objects

If there is more than one table available to use in your project, the options will be presented in a window when you try to create a table object. Simply choose the table type you wish to create.



When defining the table rows you will be assisted by the field selection assistant, 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. Further information regarding the field selection assistant and the definition of table contents can be found in "4.14.6. Define Column Contents". The representation of the fields in the variables window is automatically hierarchical, i.e. all fields of a database table can be found in its own subfolder.

Headers and Footers in sub tables

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.

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. In this situation, there are two 1:n relationships to consider: 1 customer has n orders, 1 order has n items

Customers				
Pos	Company	Adress	City	CustomerID
1	Alfred's Futterkiste	Obere Str. 57	Berlin	ALFKI
Orders				
Pos	CustomerID	OrderDate	ShippedDate	Freight
1	ALFKI	08.01.1900	09.01.1900	29,46
OrderDetails				
Pos	OrderID	ProductID	Quantity	ProductName
1	10643	28	15	Rössle Sauerkraut
2	10643	39	21	Chartreuse verte
3	10643	46	2	Spegesild
				total: 75,60 €
Orders				
Pos	CustomerID	OrderDate	ShippedDate	Freight
1	ALFKI	10.01.1900	10.01.1900	61,02
OrderDetails				
Pos	OrderID	ProductID	Quantity	ProductName
1	10692	63	20	Vegie-spread
				total: 43,90 €

4.8. 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**.

4.8.1. Properties

Property	Description	Value	Description
(Contents)	Opens the Contents dialog		Opens dialog
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	Bottom Aligned
		False	Top Aligned
		Formula	Opens Formula dialog
Rotation	Rotates the object counter-clockwise. Please note that only TrueType fonts can be rotated.	0	0°
		1	90°

		2	180°
		3	270°
		Formula	Formula dialog
Background	This option allows text objects to be assigned a background color. When the value is "True", a background color can be selected using the "Color" option.	True	With background color
		False	without background color
		Formula	Formula dialog
Frame	Through this option it is possible to provide the text object with a frame. When the value is "True", frame properties can be defined by the options color, width and distance.	True	with frame
		False	no frame
		Formula	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 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.	True	Page break
		False	No Page break
		Formula	Formula dialog

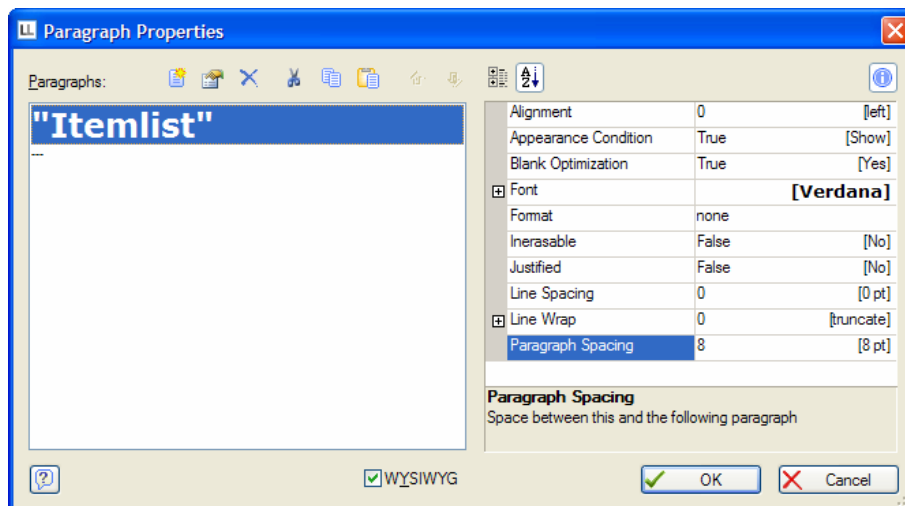
4.8.2. Text Contents

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.

4.8.3. 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.



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".

4.8.4. The Card "Tab"

When a text contains a tab, e.g. by inserting it in the "Text" card, the "Tab" card 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.

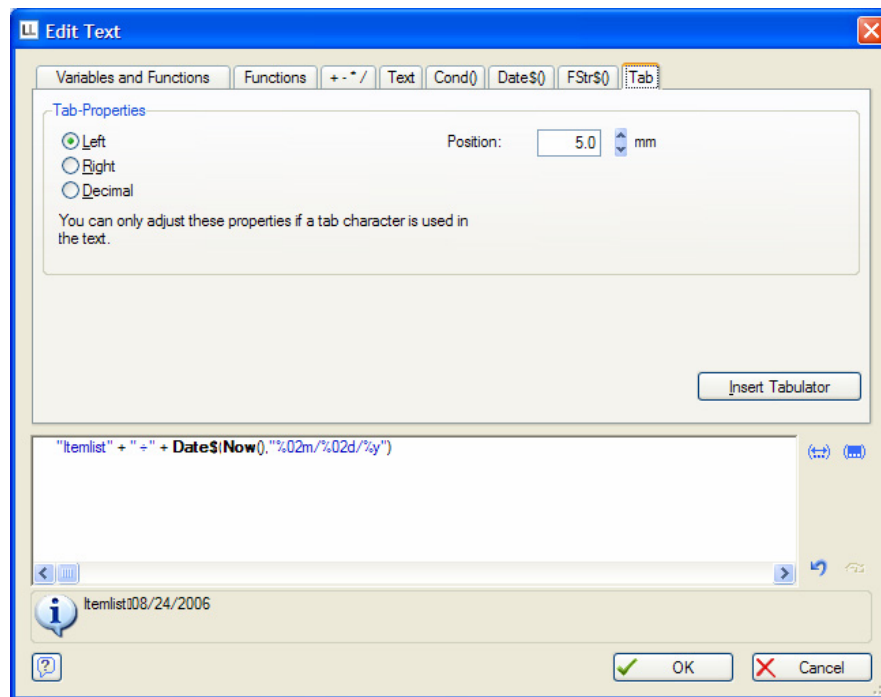
The text after the tabulator will be wrapped, dependent on the type of tab stop, if the paragraph property "Line Wrap" is selected:

- Left justified: the text is left justified in the area between the tabulator and the right edge of the object.
- Right justified: the text is right justified in the area between the tabulator and the right edge of the object.
- Decimal: 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.



4.8.5. Paragraph Properties

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

Property	Description	Value	Description
Paragraph spacing	Space between paragraphs ("paragraph end spacing"), vertical blank space	Number	
	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.	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.	True	Yes
		False	No
		Formula	Formula dialog

	The last line will be displayed in accordance with the property " Alignment ".		
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
Font	If default is set, the default font will be used. The Font Definition dialog can be opened using the font button. All properties in this properties group can be set in one dialog.		Font dialog
Default	Default font will be used instead of the set values. You can define the default font by selecting the command Project > Options. Default values set	True False Formula	Default font No Formula dialog

on the card " Objects " are valid for all new objects, until they are changed.			
Name	The name of the font.	Fonts list	
Character Set	Defines country interpretation of the character set.	Number	Character set
Size	Font size in points. A list of available sizes for the selected font will be shown.	Number	Predefined size
		Formula	Formula Dialog
Font Width	Font width. 0 signifies standard width; otherwise the average character width will be used. Other values define a "run width" which narrows or enlarge the character.	Number	Width
		Formula	Formula Dialog
Bold	Sets / Resets the text property " Bold "	True	Yes
		False	No
		Formula	Formula dialog
Italic	Sets / Resets the text property " Italic "	True	Yes
		False	No
		Formula	Formula dialog
Underlined	Sets / Resets the text property " Underlined "	True	Yes
		False	No
		Formula	Formula dialog
Strike Out	Sets / Resets the text property " Strike Out "	True	Yes
		False	No
		Formula	Formula dialog
Uneraseable	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 " Uneraseable " the empty line will remain if it would be empty after use of the variable.	True	Yes
		False	No
		Formula	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. If the text consists not of several words, but only of a long word, it is cut off likewise at value "1".	0	Truncate
		1	Wrap
		2	Shrink
		3	Compress
		Formula	Formula dialog
Keep	With this property you define whether the	True	Wrap

Together	paragraph will be kept together during a page break.	False	Keep together
		Formula	

4.9. Inserting Lines



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

4.9.1. Properties

Property	Description	Value	Description
Width	Line width in the measuring system used in workspace	Number Formula	Formula dialog
Color, Appearance Condition, Frame, Position, Pagewrap Before	See Chapter "4.6. Common Object Properties"		

4.10. Inserting Rectangles



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

4.10.1. Properties

Property	Description	Value	Description
Filled	See Chapter "4.6. Common Object Properties"		
Border	Define whether the rectangle should have a border.	0	transparent
		1	pattern / block color
		Formula	Formula dialog
	Color	Color of the frame. See Chapter "4.6. Common Object Properties"	
	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	Number	

	in %. 0% signifies angular 100% signifies: The corners of the rectangle are completely rounded.	Formula	Formula dialog
Shadow	Define whether the rectangle should have a shadow.	0 1 Formula	transparent pattern / block color Formula dialog
Pattern	Shadowpattern. See Chapter "4.6. Common Object Properties"		
Color	Shadowcolor. See Chapter "4.6. Common Object Properties"		
Width	Width of the shadow in the measuring system used in workspace	Number Formula	Formula dialog
Appearance Condition, Position, Pagewrap Before	See Chapter "4.6. Common Object Properties"		

4.11. 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).

4.11.1. 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. See Chapter "4.6. Common Object Properties"		
Circle	Define if the ellipse should always be displayed as a circle centered in the.	True False Formula	Yes No Formula dialog
Border	Define if the ellipse should have a border.	0 1 Formula	transparent pattern / block color Formula dialog

	Color	Color of the frame. See Chapter "4.6. Common Object Properties"		
	Width	Width of the frame in the measuring system used in workspace	Number Formula	Formula dialog
Appearance Condition, Position, Pagewrap Before	See Chapter "4.6. Common Object Properties"			

4.12. 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.

4.12.1. Properties

Property	Description	Value	Description
Data source	Select how the data source for the graphic to be displayed is defined.	File name Formula Variable	Formula dialog
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. In this case, embedded will be displayed instead of the file name.	File name 	File > Open dialog
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	Formula 	Formula dialog

		type "Drawing" with the function Drawing().		
	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
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
Appearance Condition, Frame, Position, Page wrap before	See Chapter "4.6. Common Object Properties"			

4.13. 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 "5.3. List of Available Barcodes". To create a barcode object select Objects > Insert > Barcode (CTRL+B).

4.13.1. Properties

Property	Description	Value	Description
(Contents)	Opens the "Contents" dialog		Opens dialog
Bar Color	Color for the selected Barcode See Chapter 4.6.7. Colors		
Bar Width	The narrowest bar width in SCM units (1/1000 mm). 0=automatic adjustment. Not supported by all barcode types.	Formula	Formula dialog

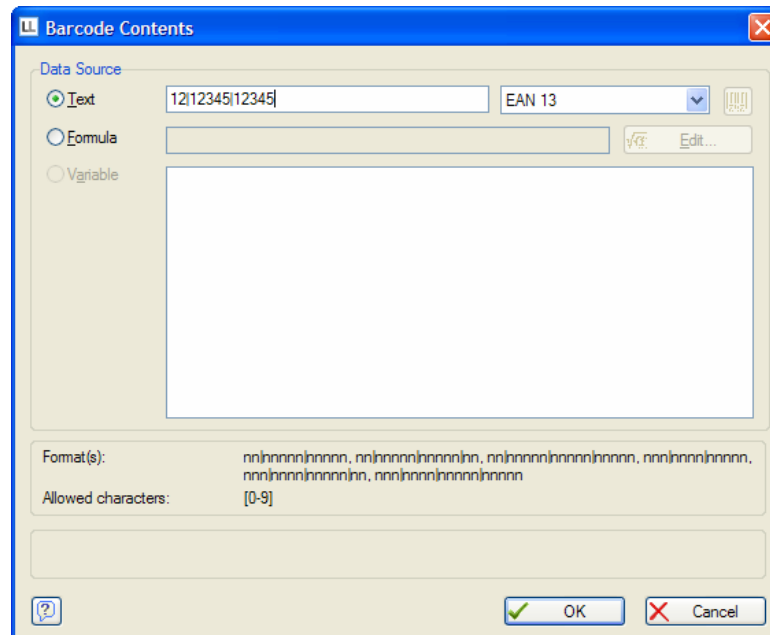
Orientation	Orientation of the barcode within the object frame.	0	left
		1	centered
		2	right
		Formula	Formula dialog
Bar Width Ratio	The ratio between the bars or spaces. Not supported by all barcode types.	Formula	Formula dialog
Rotation	Rotates the object counter clockwise	0	0°
		1	90°
		2	180°
		3	270°
		Formula	Formula dialog
Optimum Size	Sets the optimum size for the Barcode. This property could affect size changes with the following formats: EAN 13 (all), DP-Leitcode, DP-Identcode, German Parcel, Postnet and FIM.	True	Yes
		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. See Chapter 4.6. Common Object Properties.		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
Appearance Condition, Frame, Position, Page wrap before	See Chapter 4.6. Common Object Properties.		

4.13.2. Barcode Contents

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

- 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.



4.14. Inserting Table objects

 In order to create lists, tables, reports or similar forms, you need a table object. To create a table object, select **Objects > Insert > Table** (CTRL+E).

In multi-tables you add a table object over the tool window report structure (See Chapter "4.7. Report Structure").

4.14.1.Properties

Note: If the tool window "report structure" is available, other properties apply (See Chapter "4.7. Report Structure").

Property	Description		Value	Description
(Contents)	Opens the “Contents” dialog			Opens dialog
Data lines				
	Force Sums	Sums will be calculated even with the suppression of the data lines.	True	Yes
			False	No
			Formula	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	Yes
			False	No
			Formula	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	Yes
			False	No
			Formula	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	Yes
			False	No
			Formula	Formula dialog
	Color	Color of the selected pattern. See Chapter 4.6.7. Colors		
	Pattern	Pattern, with which every other data line is assigned. Select one of the predefined patterns. Each number represents a different pattern.	Number	Predefined 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	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. When using multi-tables, this setting is only available for sub-tables and not for the Report Container.		True False Formula	Yes No Formula dialog
	Separator s 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 last data line. This option is only available for tables of "Fixed Size".	True False Formula	Yes No Formula dialog
Locked	See Chapter 4.6.1. Locked. When using multi-tables, this setting is only available for the Report Container and not for sub-tables.			
Background	This property allows you to assign a background		True	Background

	to your table object. See Chapter 4.6. Common Object Properties. When using multi-tables, this setting is only available for the Report Container and not for sub-tables.	False Formula	color no background color Formula dialog
Default Font	With this property you can assign the default font for the table. The Font dialog can be opened using the appropriate button. See Chapter 4.6. Common Object Properties. When using multi-tables, this setting is only available for the Report Container and not for sub-tables.		Font dialog
Default Frame	With this property you can assign the default frame for the table. See Chapter 4.6. Common Object Properties.		Font dialog
Page wrap Conditions	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
Appearance Condition, Position, Page wrap before	See Chapter 4.6. Common Object Properties.		

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

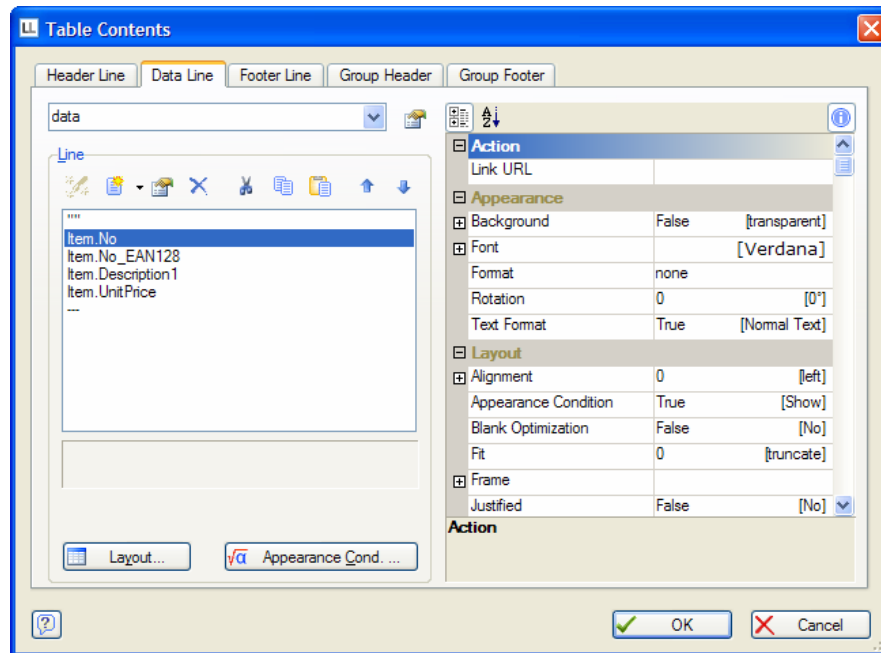
lines can have a completely different appearance than the following data or footer lines.

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.

In this way, "sub reports" (tables that contain sub-tables) can be created with a maximum of flexibility.

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".



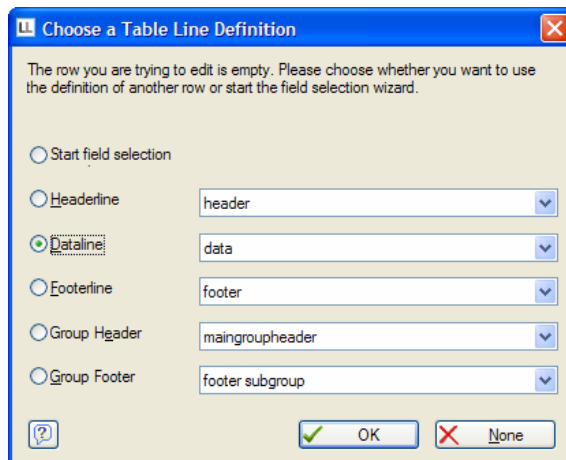
A card is provided for the definition of every line type and the columns.

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.

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



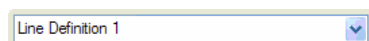
In the example pictured above, the definition for line definition 1 was selected for use in the data line.

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.

4.14.4. Defining Line Layouts

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.



For every line type, there are 100 – 150 different line definitions available.

To create multiple line layouts for a line, select the line definition that you would like to edit. 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.

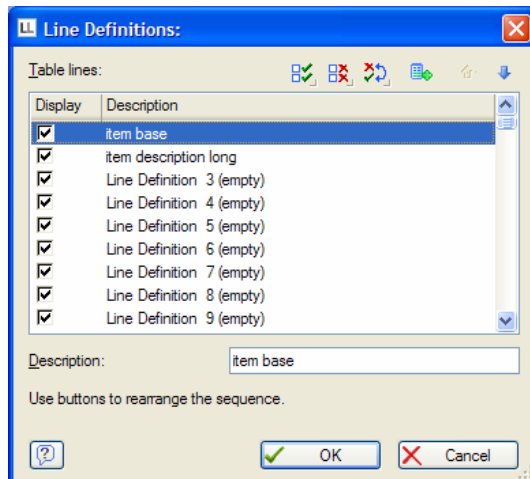
As described in Chapter "4.14.6. Define Column Contents", you can now define the columns, their contents and properties for the new layout. Using the button "**Layout**" you can define the complete appearance for the new line layout. You can define, for example, special frames.

4.14.5. Editing the Line Definitions List



The Line Definitions list can be edited using the button "Properties".

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. Line definitions can be compromised, that means that empty line definitions will be moved to the bottom of the list, by using the  button.

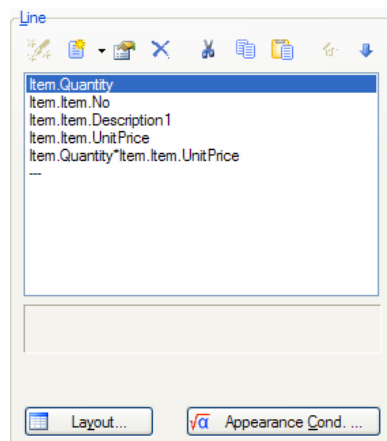
With the "Display" option you can hide single lines in the workspace. This is very useful when you have a lot of line definitions. The controls at the top of the dialog can be used to show/hide several highlighted lines in one step. By right-clicking the button, all the lines can be shown or hidden.

4.14.6. Define Column Contents

Table lines are column oriented, that means 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 line in the list. Multiple expressions or variables can appear within a column.



With the buttons you can:

- start up the field selection assistant
- insert a text column
- insert a column of a certain type, the ▼ button opens an appropriate menu
- edit the selected column(s)
- delete the selected column(s)
- copy the selected columns to the clipboard and then delete them
- copy the selected columns to the clipboard
- insert columns from the clipboard
- move the selected columns to the left
- move the selected columns to the right

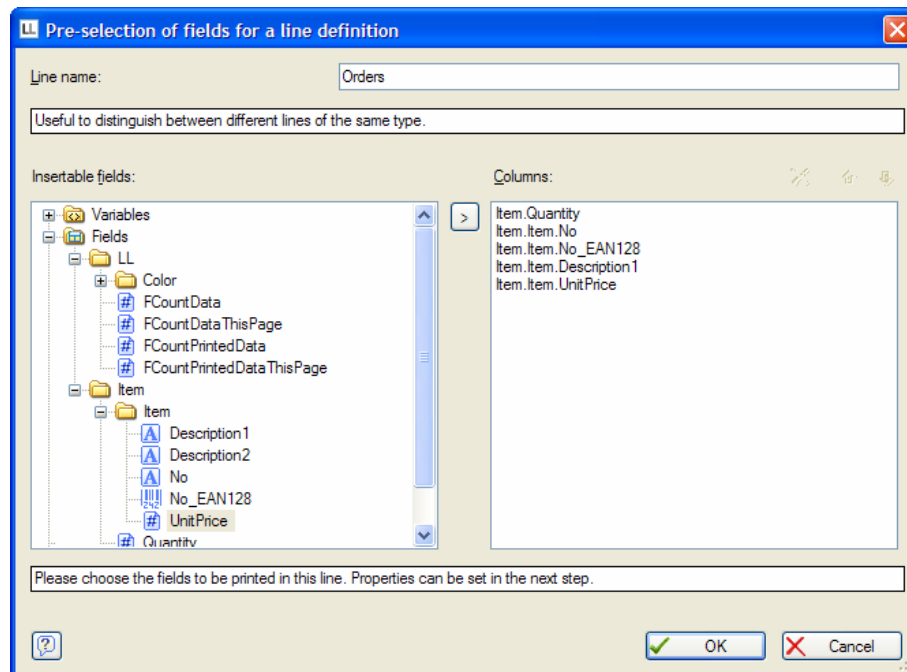
Editing of the column is done in the familiar Formula Editor, in which expressions can be defined as column contents, see Chapter "2.3. Variables, Formulas and Expressions for more information".

The Field Selection Assistant

The field selection assistant makes it easier to add new columns. With table objects, the assistant is able to:

- add several columns in a single step.

- assemble the columns using the available fields.



4.14.7. Format Lines

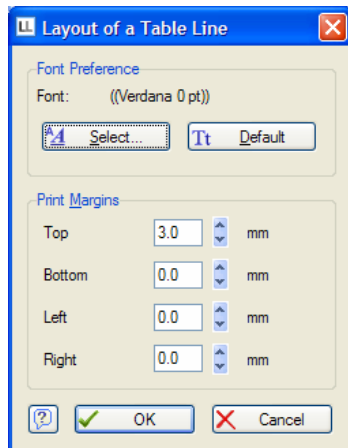
 **Layout...** With this button you can define the appearance of the current table line.

You can set the font preference for the complete line using the Select button. Newly inserted columns appear at the beginning in this font. Using the appropriate option in the Column Properties, you can select a different font for each column.

In the category “**Print Margins**”, 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.



4.14.8. Appearance Conditions for Table Lines

 Appearance Cond.... Appearance conditions for the table lines can be assigned using the button “**Appearance Cond.**” Using this button opens the familiar dialog for the definition of logical expressions (see Chapter “2.3. 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.

4.14.9. 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.

Property	Description	Value	Description
Text-Format	Defines the text column (normal text or RTF text).	True	Normal text
		False	RTF Text

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	With this option you define the vertical alignment of the selected column.	0	Top
		1	Centered
		2	Bottom
		Formula	Formula dialog
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
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 dialog
		Formula	
Bar color (only with barcodes)	With this option, you define the color of barcodes. See Chapter 4.6.7.		
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 False Formula	Yes No Formula dialog
Rotation	Rotates the object counter-clockwise. E.g. turn column titles by 90°	0 1 2 3 Formula	0° 90° 180° 270° Formula dialog
Fit	Determines the behavior when the content is too long for the line.	0 1 2 3 Formula	Truncate Wrap Shrink Compress 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
Background	The background color for the column can be defined. See Chapter 4.6. Common Object Properties.		
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

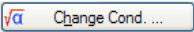
Frame	This sets the frame properties and the spacing between the frame and the individual cells of a table. Along with the font size, the cell edges "top" and "bottom" affect the height of the table row in text objects. Further information regarding the frame dialog can be found in "4.6.6. Frame".		Opens dialog
Appearance Condition, Font	See Chapter 4.6. Common Object Properties.		

4.14.10. Defining Footer Lines

It is often desirable to print footers along with your table. These are considered to be normal table lines with the exception that footers always appear at the end of the table on each page. If you want the footers to be printed on the last page only, assign the appearance condition `LastPage()`. Additional information about this function can be found in Chapter 5.2. List of Available Functions.

4.14.11. 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.



Conditions are also important here. Any string, for example `"Left$(NAME,1)"`, can be used as a condition. Every time that the result of the expression changes from line to line, the result will be used as an intermediate header.

The assigned column definition for the group line could be: `"Letter: «Left$(NAME,1)»"` or `"Letter: "+Left$(NAME)`, depending on which *modus* you are in. The syntax assistant will automatically display the proper syntax. For every new beginning letter in the NAME, the text "Letter: " followed by the letter 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 NAME. Additionally, you can define a second line layout which generates an intermediate header using the first two characters of the variable NAME. The expression for this would be `"Left$(NAME,2)"`, a suitable column definition would be `"Letters: «Left$(NAME,2)»"`. For this second layout, special margins can be set using the button "Layout", for example `"left" = 10.0 mm`. The result could look like:

.....

Letter: A

Letters: AA

...

data line

...

Letters: AB

...

data line

...

...

Letter: B

Letters: BA

...

data line

...

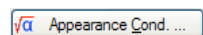
Letters: BB

...

data line

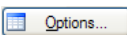
...

Appearance Condition



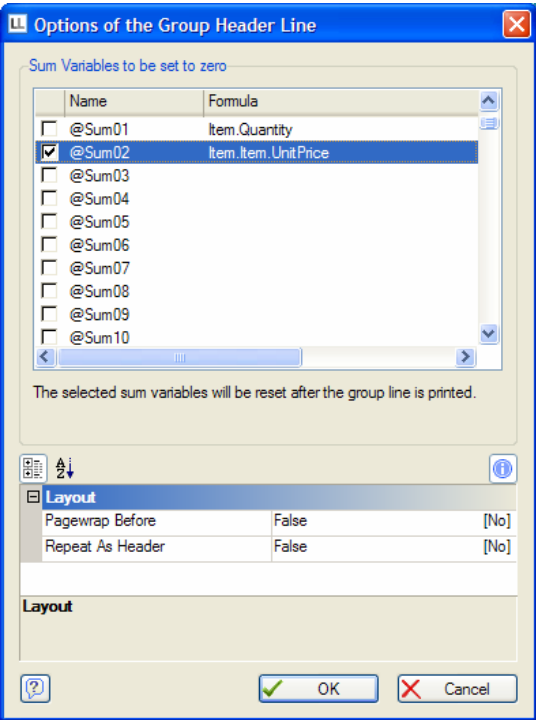
It is also possible to use appearance conditions for group header lines. Further information can be found in the section “Appearance Conditions for Table Lines” in this chapter.

Options for Group Header Lines

By clicking on the button  you open the options dialog for group headers:

Select the Sum Variable(s) that should be returned by the group header line. This setting is useful to create group sums, for example to add the price of all items of a certain item group.

Hint: We recommend to use the function Sum() for the accumulation. Further information finds you under the description of function "Sum ()" in Chapter 5.2. List of Available Functions.



Additional options of the group header are defined in a property list:

Property	Description	Value	Description
Repeat as Header	Prints the group header again after a page break.	True	Yes
		False	No
		Formula	Formula dialog
Pagewrap Before	Before a group header is printed, a page break is triggered. 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	True	Yes
		False	No
		Formula	Formula dialog

appendix under the description of the function "RemainingTableSpace".

4.14.12. 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.

Property	Description	Value	Description
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	Yes
		False	No
		Formula	Formula dialog

4.15. 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.

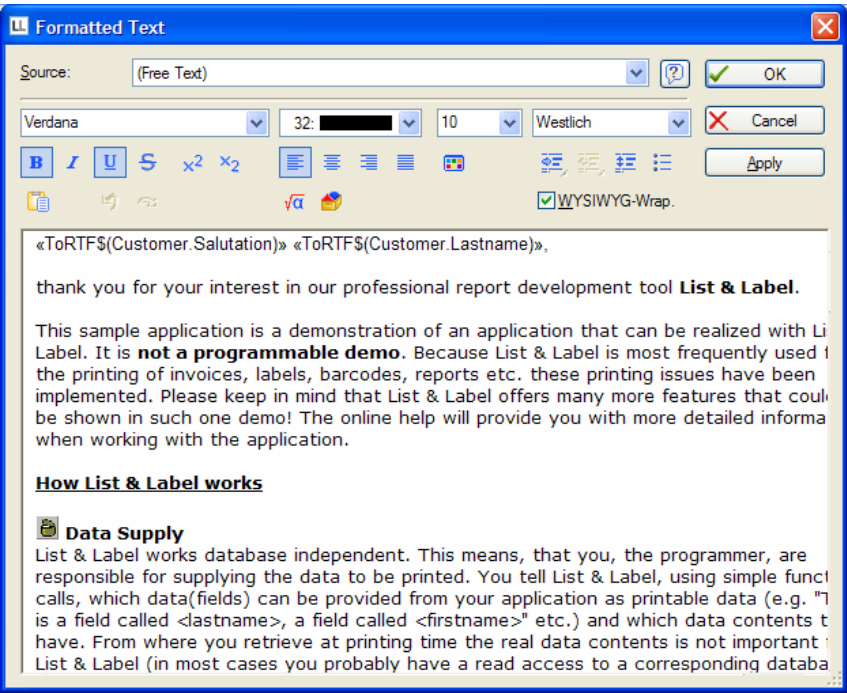
4.15.1. Properties

Property	Description	Value	Description
(Contents)	Opens the Contents		Opens dialog
Rotation	Rotates the object counter clockwise. This function is only available on Windows NT based systems (Windows NT, Windows 2000, and Windows XP).	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

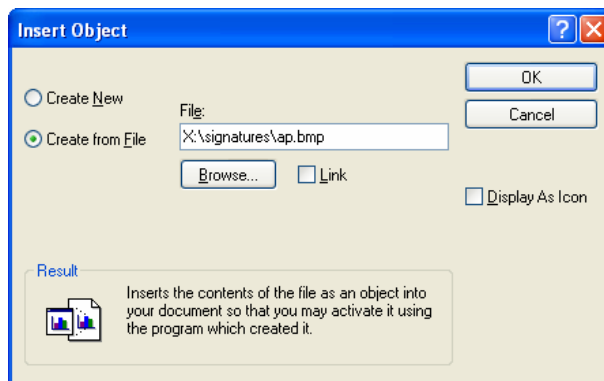
4.15.2. Text Contents

The contents of formatted text objects can be edited in the Formatted Text Editor.

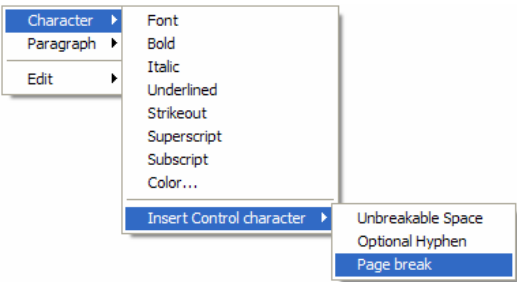


As a source, you may either select an RTF-Variable or "(Free Text)". If you select the latter, an input box and formatting aids are displayed in the dialog, which are explained below. Simply select the text to be formatted and select the respective format tool:

- 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.

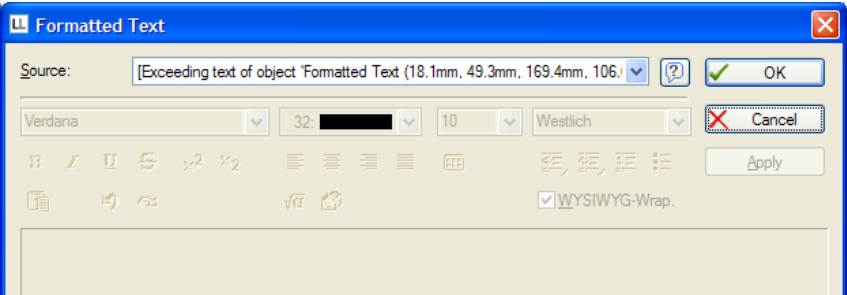


-  **WYSIWYG** What you see is what you get: Word wrapping according to object size. This is only an approximate display and minimal differences may occur.
- Tab: CTRL+TAB
- A context menu is available in the editor, by which you can open various formatting possibilities for characters and paragraphs.



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.



4.16. Inserting Chart Objects

By selecting **Objects > Insert > Chart** or clicking the respective icon you can insert a chart object. 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.

In general, three and two axis charts are differed. 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 of the whole company over a number of months.

4.16.1. Properties

Property	Description	Value	Description
----------	-------------	-------	-------------

(Content)

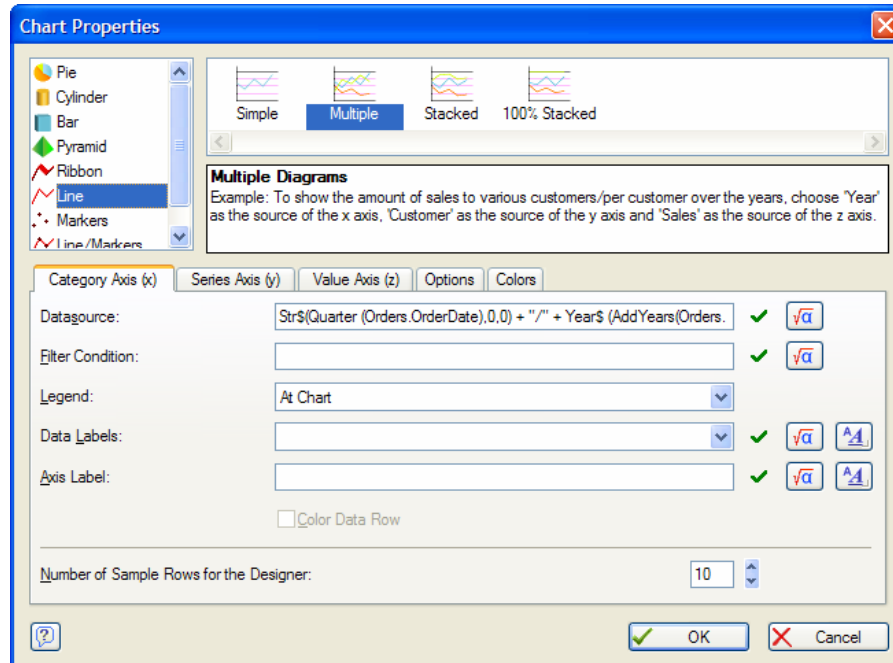
Opens the contents dialog



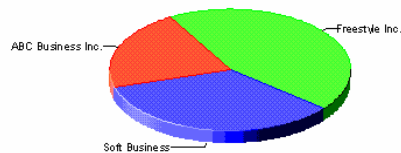
Opens dialog

4.16.2. Selecting the Chart Type

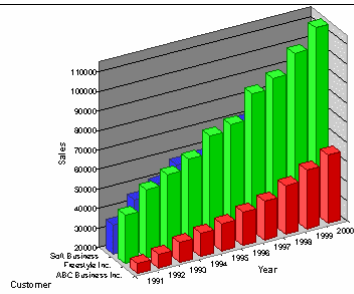
Double click a chart object to reach to the property dialog.



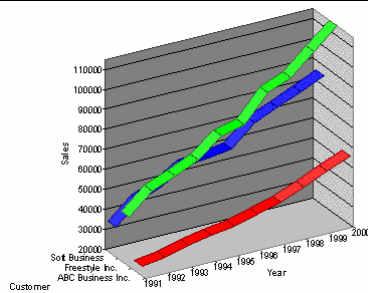
Select the chart type from the list on the left hand side. The following types are available:



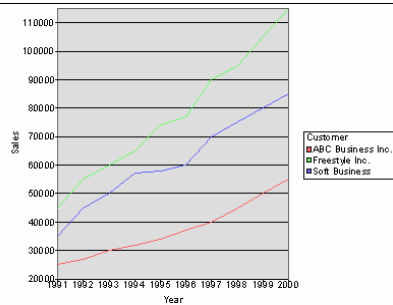
Pie Chart: a typical pie chart.



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



Ribbon: a three dimensional ribbon chart.

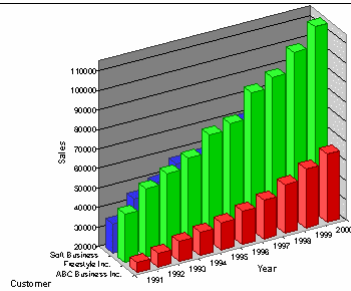


Line/Symbols/Line & Symbols: two dimensional charts.

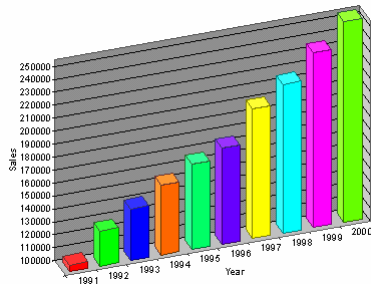
4.16.3. Selecting the Sub Type

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

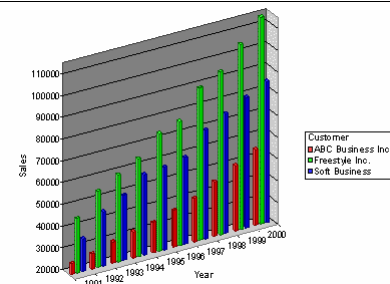
Sub types for cylinder, bar and pyramid charts:



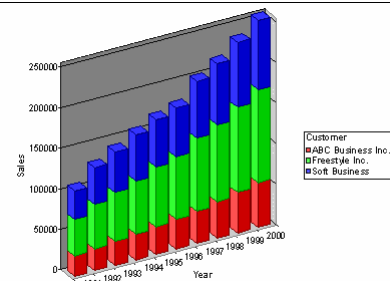
Multi row: A "normal" 3D chart, showing the sales per month and per employee, for example



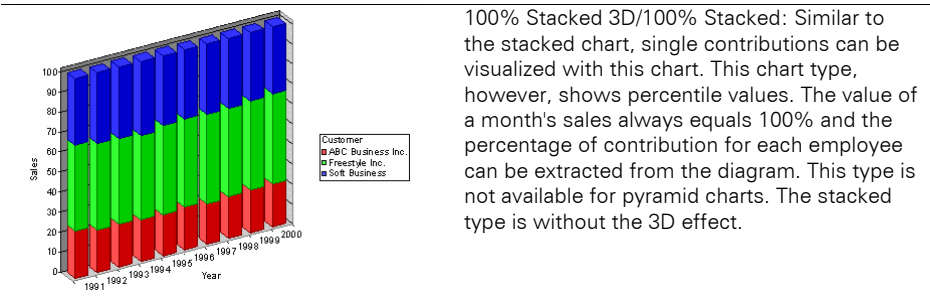
Simple 3D/Simple: Only two axes are available, showing the total amount of sales per month for example. The simple type is without the 3D effect.



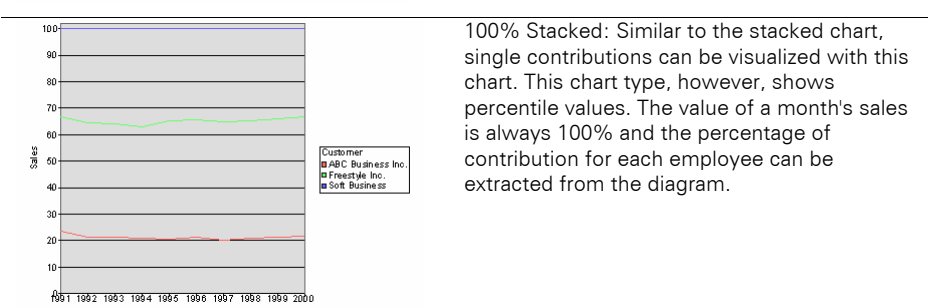
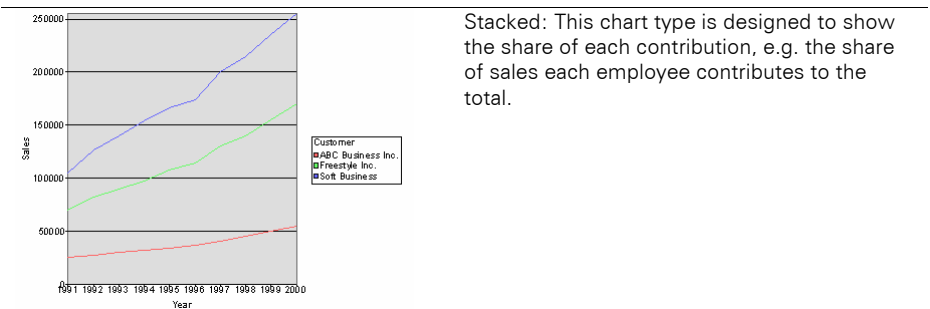
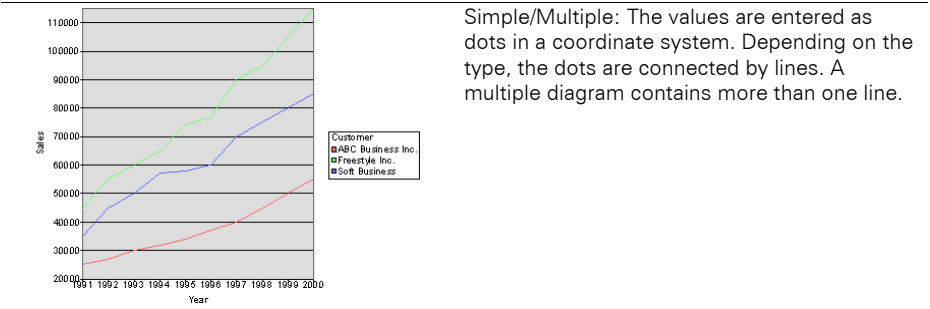
Clustered 3D/Clustered: Values on the x axis are grouped allowing a direct comparison of values. The clustered type is without the 3D effect.



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 charts. The stacked type is without the 3D effect.



Sub types for line and symbol charts



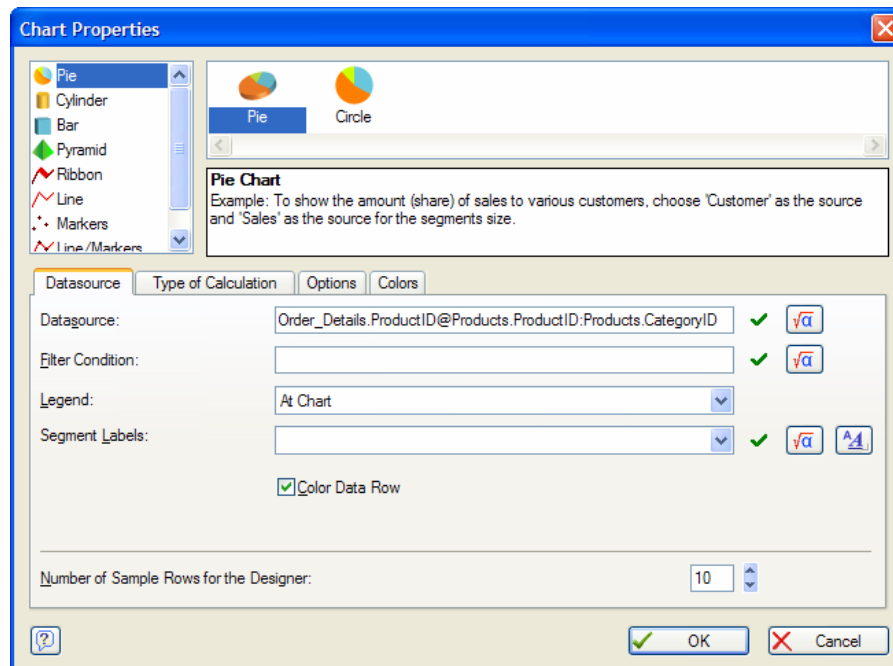
4.16.4. Chart Properties

The chart object provides a variety of options. Depending on the chart type, there are two or three data axes. Using the cards you can switch from one axis to another. You have the following options:

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.

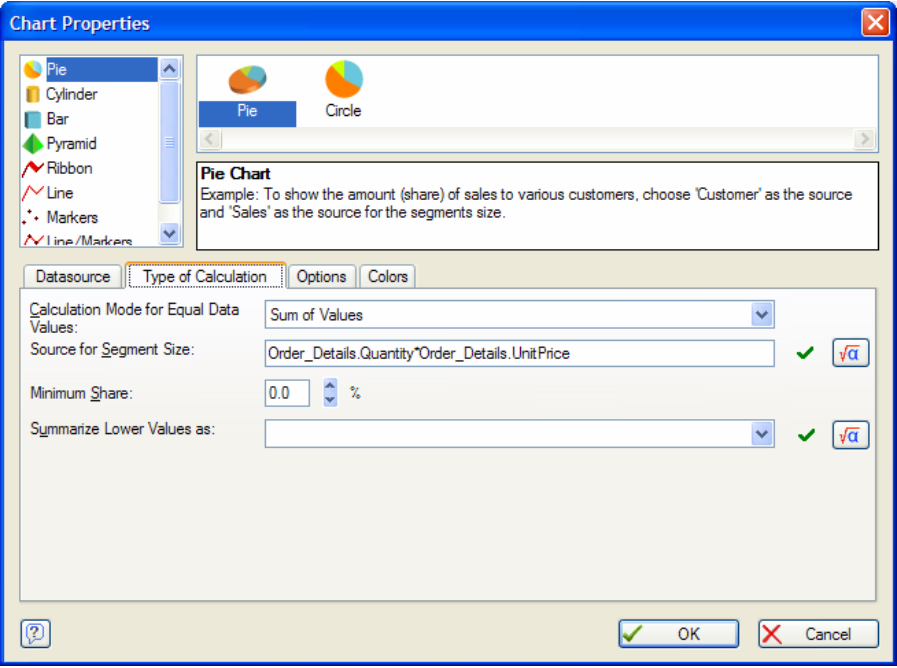
Options for the data source (Pie charts)



Data source	Select the data source for the segments, e.g. "Name" for people, "Month" for date values, and so on.
Filter condition	If you like you can define a filter condition. Only the records matching the condition will be used for the chart data (e.g. <code>Left\$(Name, 1) = 'A'</code>).
Legend	Select the position of the legend. If you select "at chart" the values are given directly at the segments.
Segment labels	Select the text which should be used to label the legend. Some preconfigured values are available, e.g. "Value and Percentage without decimals". Alternatively you may enter your own formula, select "Formula..." from the list in order to do so. By clicking on the button "formula" you may also enter a formula.

	 The  allows you to set the font for the labels. A left click opens a font dialog; a right click resets the font to the default object font.
Color data row	The segments are colored with different colors in order to make the diagram easier to read.
Number of sample rows for the Designer	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.

Options for the type of calculation (Pie charts)

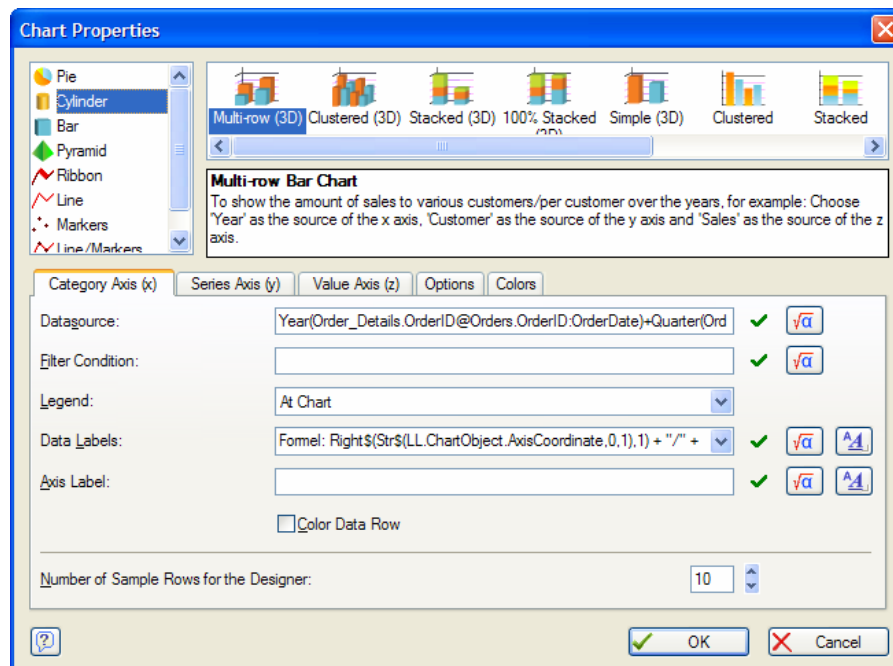


Calculation mode for equal values	Specifies the handling of equal source values. An example: you want to have the sum of sales to your customers. In this case, select "sum of values". If you're interested in the average price of sales to a customer, select "mean value". If only the number of sales to a customer is of interest, select "number of values". As the actual value is not of interest in this case, the source field will be deactivated.
Source for segment size	Select the data source for the segment size, e.g. "Value", "Price" etc.
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 which the segments will be summarized.

Summarize lower values as	Defines the label for the summarized segment. If you do not enter your own formula, the label will be adapted to the "normal" segment label layout on the data source card.
---------------------------	---

Options for category and series axis (except pie charts)

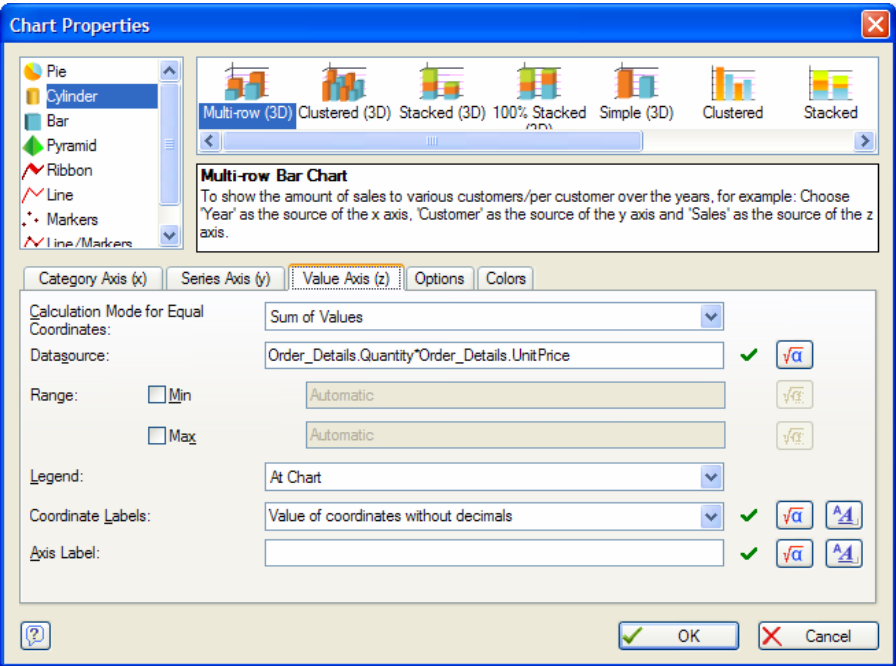
If you chose to have a three axis chart, both axes are available (as x and y-axis). Only the category axis is needed (as x axis) for two axis charts. You have the same options for both axes:



Data source	Select the data source for the row, e.g. "Name" for people, "Month" for date values, and so on.
Filter condition	If you like you can define a filter condition. Only the records matching the condition will be used for the chart data (e.g. <code>Left\$(Name, 1) = 'A'</code>).
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.
Data labels	Select the text which should be used to label the legend. Some preconfigured values are available, e.g. "Value of source". This setting will display the source value on the axis, i.e. the name of a customer, the month,... Alternatively you may enter your own formula. Select "Formula..." from the list in order to do so. By clicking on the button "formula" you may also enter

	a formula. The button "font" allows you to set the font for the labels. A left click opens a file selection dialog; a right click resets the font to the default object font.
Axis label	Select a text for your axis label here. By clicking on the button "formula" you may also enter a formula. The button "font" allows you to set the font for the labels. A left click opens a file selection dialog; a right click resets the font to the default object font.
Color data row	This attribute may only be selected for either the series or the category axis. The selected data row will then be colored in altering colors.
Number of sample rows for the designer	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.

Options for the value axis (except pie charts)

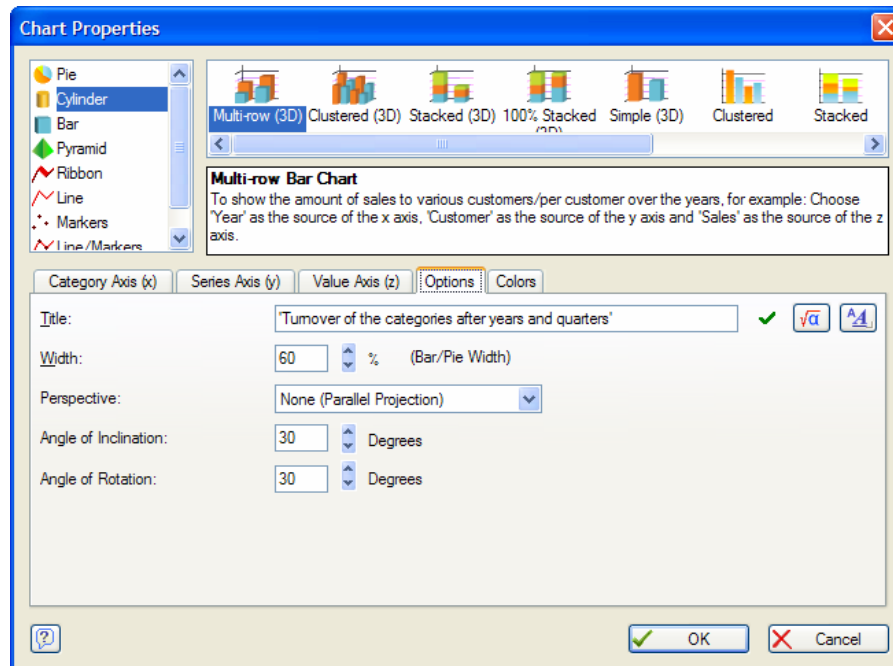


Calculation mode for equal coordinates	Specifies the handling of equal source coordinates. An example: you want to sum the sales to customer Smith in January. In this case, select "sum of values". If you're interested in the average price of your sales to this customer, select "mean value". If only the number of sales to the customer is of interest, select "number of values". As the actual value is not of interest in this case, the source field will be deactivated.
--	--

Data source	Select the data source for the value axis, e.g. "Value", "Price" etc.
Range	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 "automatic" the chart will display all values.
Legend, Coordinate labels, Axis label	These options are the same as for the series and category axis. The legend can only be displayed at the chart or not at all, however.

Options for the chart object

On the options card you may set different layout options for the chart object.

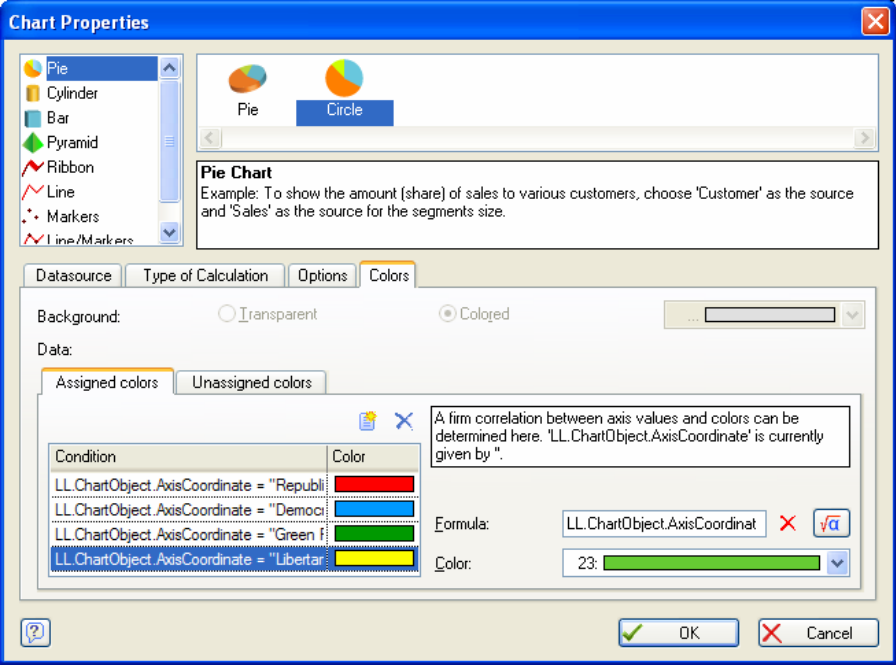


Title	Select a title for your diagram. This title will be displayed above your chart. By clicking on the formula-button you may also enter a formula. The font-button allows you to set the font for the labels. A left click opens a font dialog; a right click resets the font to the default object font.
Width	Sets – depending on the chart type – the bar width, the line width or the width of the segments.
Perspective	Select the level of perspective for your object here.
Angle of	Sets the rotation of the diagram. You may also set these angles using the

inclination, Angle of rotation	rotation buttons which appear when the chart is selected on the workspace.
--------------------------------------	--

Colors for the chart object

On the colors card you can set the colors for your chart.



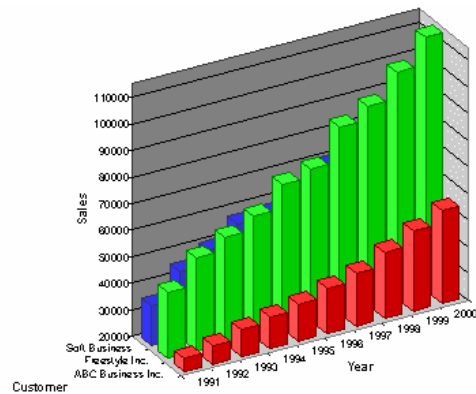
Background	Select the color for the "back wall" of the chart. Alternatively the wall may also be transparent. To select a color, select it from the upper combo box. The "..." entry brings up a standard color selection dialog.
Color map	Select the colors and the order of colors for the chart. For a bar 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.

4.16.5. Examples

Chart options depend on the available data. Thus, the following examples are kept general. An application may give you the fields "Name", "Month" and "Sales", denoting the name of your customer, the month in which a transaction took place and the total transaction volume.

Multi row bar chart

This would be the simplest way to analyze your data; you'd have a diagram showing the total transaction volume for each month and customer:

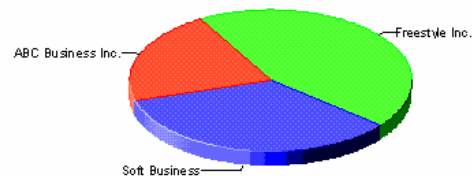


...this is how it's done:

Insert a new chart object; select "Bar" as type and "Multi row" as sub type. Select "Month" as the data source for the category axis and "Customer" as the data source for the series axis. For the value axis, select "Sales". Edit the axis labels and title as required – you're finished.

Pie chart

If you're interested in the share each customer contributes to the total sales over a number of months, you'd select a pie chart. This chart directly displays the shares:

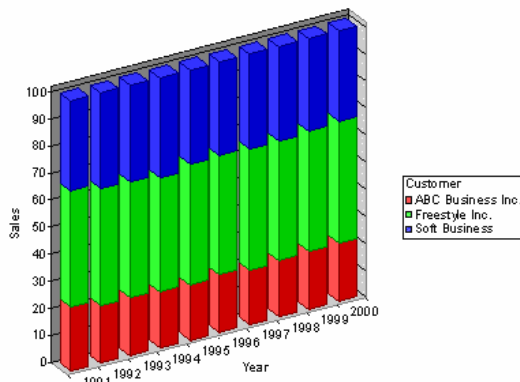


...this is how it's done:

Insert a new chart object, select "Pie" as type and sub type. Select "Customer" as the data source and "Sales" as the data source for the segment size. Edit the axis labels and title as required – you're finished.

100% stacked bar chart

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



...this is how it's done:

Insert a new chart object; select "Bar" as type and "100% stacked" as sub type. Select "Month" as the data source for the category axis and "Customer" as the data source for the series axis. For the value axis, select "Sales". Edit the axis labels and title as required – the procedure is exactly the same as for the multi-row bar chart. This is why you can simply switch between the two sub types without having to reconfigure your data sources.

4.17. Inserting a Crosstab



You have two ways to insert a Crosstab:

- In multi-tables you can insert a crosstab object through the report structure tool window (see chapter "4.7. Report Structure").
- Through **Objects > Insert > Crosstab (Pivot Table)**.

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.

Example 1: Customer sales per sales period and employee

Order amount per year and customer

	2004				2005				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

Example 2: Turnover per sales period and country

	2004		2005				2006
	Q 3	Q 4	Q 1	Q 2	Q 3	Q 4	Q 1
Canada	0,00 €	0,00 €	6.366,30 €	896,00 €	0,00 €	3.118,00 €	0,00 €
France	1.176,00 €	13.012,70 €	4.049,00 €	5.666,80 €	3.662,80 €	4.857,75 €	0,00 €
Germany	0,00 €	0,00 €	0,00 €	285,80 €	1.880,00 €	1.208,00 €	1.476,00 €
Mexico	0,00 €	492,00 €	0,00 €	3.037,75 €	2.561,75 €	1.652,40 €	660,00 €
Spain	0,00 €	982,00 €	0,00 €	0,00 €	0,00 €	0,00 €	4.035,80 €
Sweden	2.102,00 €	0,00 €	3.429,00 €	3.192,65 €	3.069,25 €	4.879,20 €	0,00 €
UK	0,00 €	480,00 €	1.352,00 €	0,00 €	2.142,90 €	2.325,00 €	1.950,10 €
Total	3.278,00 €	14.966,70 €	15.196,30 €	13.079,00 €	13.316,70 €	18.040,35 €	8.121,90 €

4.17.1. Properties

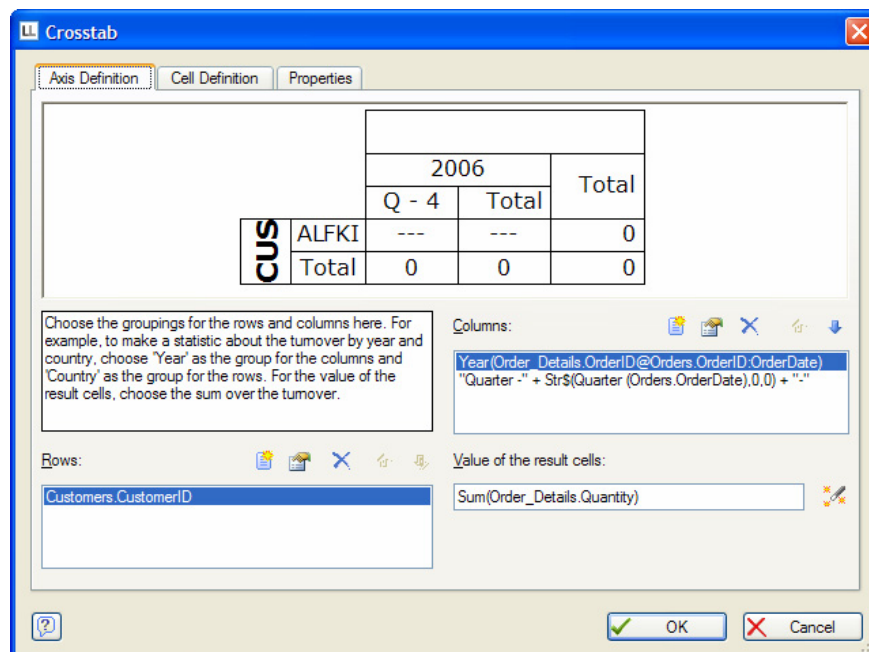
Please note: As long as the tool window "Report Structure" is available, other properties are applicable (see P.98).

Property	Description	Value	Description
(Contents)	Opens the "Contents"-dialog		Opens dialog
Background	Background of the object. See chapter "4.6. Common Object Properties".		
Frame	Frame properties and clearances of the object		

	frame. See chapter "4.6. Common Object Properties".
Frame defaults, minimum size, columns, rows, link, PDF index text	Frame default settings for sub-objects. See chapter 4.17.4. Layout Options and Page Break Behavior.
Display properties, position, page break before	See chapter "4.6. Common Object Properties".

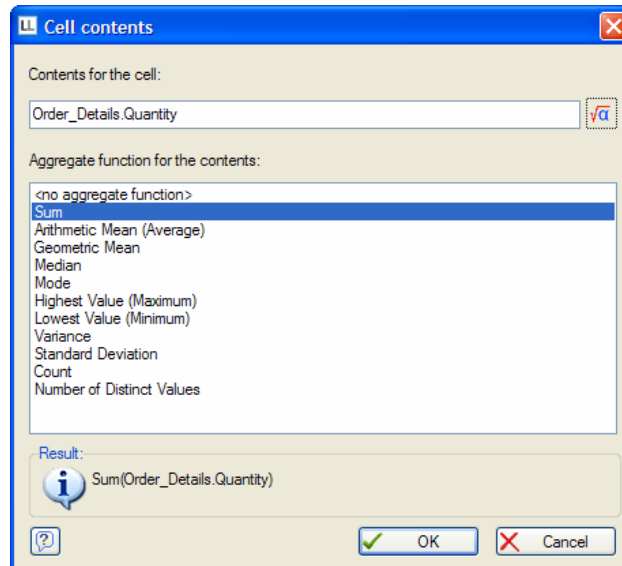
4.17.2. Axis definition (Grouping)

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.



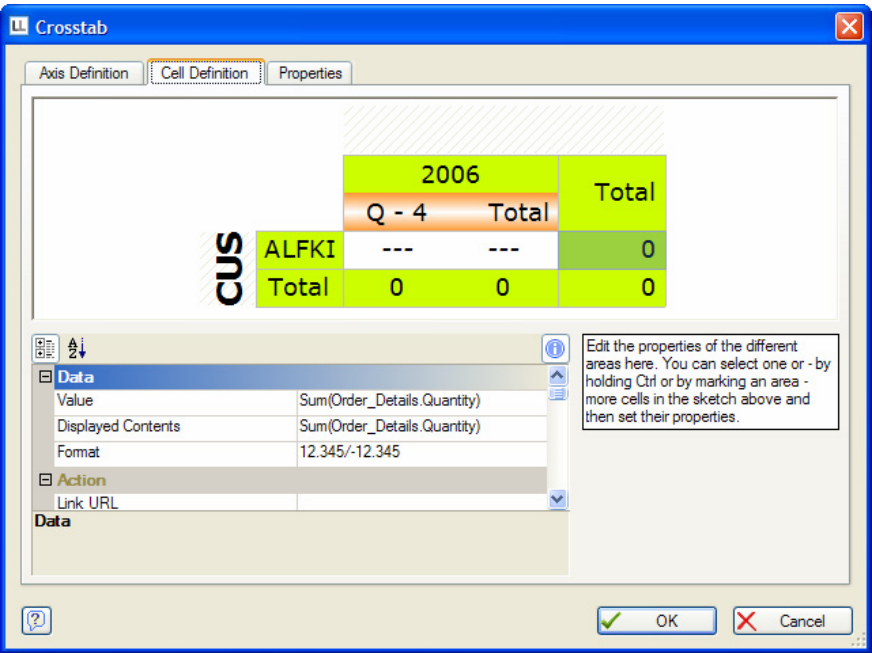
- With the "New" button, you can add a group for rows or columns.
- With the "Properties" button you can edit the selected group.

- 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.



4.17.3. Cell definition (Contents)

In the "Cell Definition" tab, the properties of the different cells can be edited.



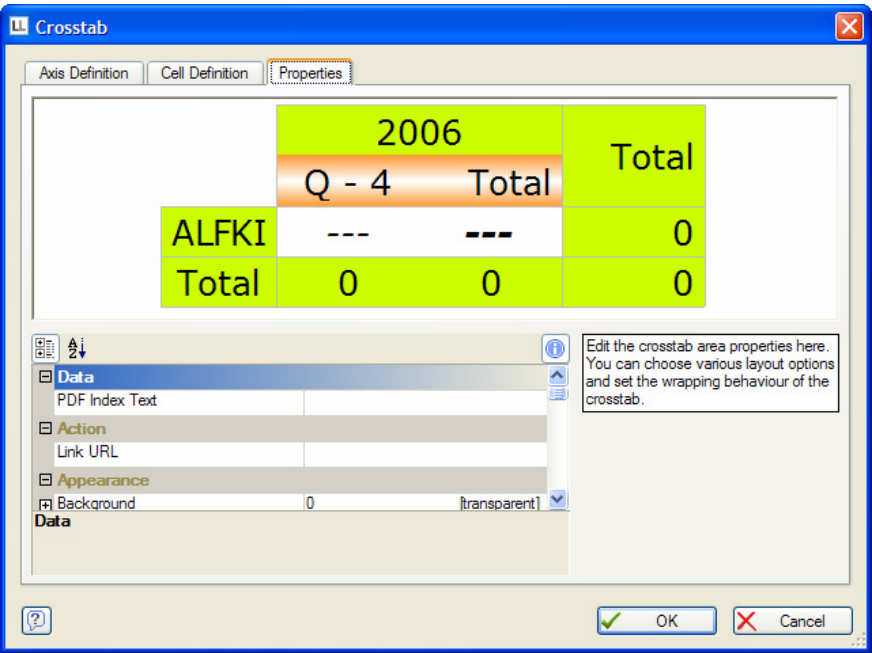
You can select the cells directly in the sketch in the top part of the dialog and can then edit their properties. To select more than one cell, hold the CTRL key and selector drag a frame around the cells with the cursor.

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

Background, font, format, frame	See chapter "4.6. Common Object Properties".		
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
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

4.17.4. Layout Options and Page Break Behavior

Edit the properties of the crosstab here, e.g. layout and page break behavior.



PDF index text	Text for the PDF index.	
Link	Link that is opened when clicked (only in preview, PDF and HTML export).	
Background, Default Frame	See chapter "4.6. Common Object Properties".	
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.	
Minimum height	Sets how much height should be available to an object. If less space is available, a page break is triggered.	
Columns	Repeat headings	Sets whether the column headers are repeated after a page break.
	Page break layer	Sets the ideal page break layer. 0=innermost group, i.e. the lowest row in the "Columns" field of the "Axis definition" tab.
	force	Forces page break after every corresponding group.

Rows	Repeat headings	Sets whether the row headings are repeated after a page break.
	Page break layer	Sets the ideal page break layer. 0=innermost group, i.e. the lowest row in the "Rows" field of the "Axis definition" tab.
	force	Forces page break after every corresponding group.

4.17.5. Crosstab Creation Wizard

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

Add new crosstab

		Heading	
		2006	Total
Heading	Venezuela	0,00	0,00
	Total	0,00	0,00

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)

Rows: Order_Details.OrderID@Orders.OrderID:ShipCountry

Value of the result cells: Sum(Sum(Order_Details.Quantity))

< Previous Next > Cancel

- In the first, you set the groups for the rows. For example, to make a statistic about the turnover per year and country, select "Year" as the group for the rows and "Country" as the group for the columns. For the contents of the result cells, select the sum over the turnover.
- With the "Next" button, you come to the "Cell Definition" dialog. The properties of the different cells are edited here. You can select the cells directly in the upper area of the dialog and can then edit their properties. To select several cells, hold the Ctrl key or drag a frame around the cells with the cursor.

- With the "Next" button, you come to the dialog for editing the properties of the crosstab, e.g. the layout and page break behavior.

4.17.6. Crosstab Functions

There are several crosstab functions to access to the values of each data record.

- `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 about this function can be found in Chapter 5.2. List of Available Functions.

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

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.	Name	
Validation Expression	Regular expression to validate input.		
Error Message	Message that is given if validation fails		
Value	Default value for input object.		
Tooltip	Tooltip which should appear		
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.	Name	
Value	Default value for input object.		
Tooltip	Tooltip which should appear		

4.18.3. Type Combobox

Property	Description	Value	Description
Force Input	Determines whether input has to be entered	True	Yes

	by the user.	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.	Name	
Validation Expression	Regular expression to validate input (only with variable text)		
Error Message	Message that is given if validation fails		
Editable	Determines if the user can enter different values than the default values.	True False Formula	Yes No Formula dialog
Value	Default value for input object.		
Tooltip	Tooltip which should appear		

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 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.	0 1 2 3	Send as email. Save. Sending via HTTPPost Link
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. Please note that due to license restrictions GIF files can not be displayed.

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**.

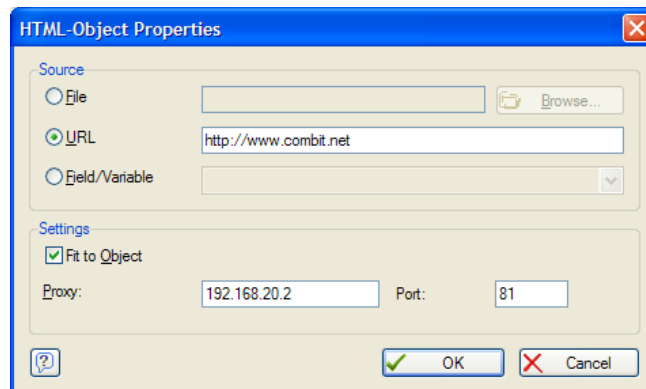
The contents of the object are edited in the property list, where you have various options.

4.19.1. Properties

Property	Description	Value	Description
(Contents)	Opens the "Properties" dialog		Opens dialog

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

Property	Description	Value	Description
(Contents)	Opens "Contents" dialog		Opens dialog or host application

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.

4.21.1.Properties

Property	Description	Value	Description
File name	Select the file that contains the required form template. The same formats are supported as in graphic files. (See Chapter 4.12. Inserting Picture Objects).	File name 	Open file dialog
Fade color	The selected color will be added to the template to fade the color. See Chapter 4.6.7. Colors		
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).	True False Formula	Yes No Formula dialog

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.

5. Appendix

5.1. List of List & Label Variables

List & Label automatically provides, dependent upon the application, several variables and fields (in list projects). These contain general information about the print project, printing etc. You can find the variables and fields in the LL subfolder in the variable list.

5.1.1. List of Variables

Name	Description
LL.SortStrategy	Selected sorting can be set, dependent upon the application in the Project Menu.
LL.FilterExpression	Selected project filter, can be set in Project -Menu.
LL.OutputDevice	Output device. Can be used to assign object specific formats for certain output formats ("HTML", "RTF", "PDF",...)
LL.CountData (not for List and 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 (not for List and Multitab projects)	Number of records transferred by the program so far 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 List and Multitab projects)	Number of the actually printed records.
LL.CountPrintedDataThisPage (not for List and Multitab projects)	Number of the actually printed records on the current page.
LL.Device.Page.Size.cx	Physical page width of the output device in project coordinates in SCM units (1/1000 mm). 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 SCM units (1/1000 mm).
LL.Device.Page.Name	Name of the output page type (example "A4").
LL.Device.PrintableArea.Size.cx	Printable page width of the output device in SCM units (1/1000 mm).

LL.Device.PrintableArea.Size.cy	Printable page height of the output device in SCM units (1/1000 mm).
LL.Device.PrintableArea.Offset.cx	Width of the left non-printable border in project coordinates in SCM units (1/1000 mm). 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 project coordinates in SCM units (1/1000 mm).
LL.Device.Name	Name of the output device(printer).

5.1.2. List of Fields

Name	Description
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.
Additionally, depending on project type	
LL.CurrentRelation	Name of the current relation
LL.CurrentSortOrder	Order of the current table
LL.CurrentTable	Name of the current table
LL.CurrentTablePath	Indicates the current table hierarchy, e.g. Customers.Orders.Order_Details

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
Asc	String	Number
AskString\$	String, Boolean, String, Number	String
AskStringChoice\$	String [,Boolean[,String[,Zahl]]]	String
Atrim\$	String	String
Avg	Number [,Boolean]	Number
Barcode	String, String	Barcode
Barcode\$	Barcode	String
BarcodeType\$	Barcode	String
Case\$	Number, String [,String]	String
CheckMod10	String	Number
Chr\$	Number [,Number]	String
ChrSubst\$	String, String [,String]	String
Cond	Boolean, All, All	All
Contains	String, String	Boolean
Continued	-	Boolean
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
Cstr\$	Number, String	String
Date	String	Date
Date\$	Date, [String, [String]]	String
DateHMS	Number, Number, Number	Date
DateInRange	Date, Date, Date	Boolean
DateToJulian	Date	Number
DateYMD	Number, Number, Number	Date
Day	Date	Number
Day\$	Date	String
Distinct	All	All
Dow\$	Date	String
Dow	Date	Number
Drawing	String	Drawing
Drawing\$	Drawing	String
Empty	String	Boolean
Even	Number	Boolean
Exp	Number	Number
Exp10	Number	Number
FirstHeaderThisTable	-	Boolean
Frac	Number	Number
FStr\$	Number, String	String
GeometricAvg	Number [,Boolean]	Number
Hour	[Boolean]	Number
HSL	Number, Number, Number	Number
Hyperlink\$	String, String, Boolean	String
If	Boolean, All [,All]	All
Int	Number	Number
IsNull	All	Boolean
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
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
NumInRange	Number, Number, Number	Boolean
Odd	Number	Boolean
Ord	String	Number
Page	-	Number
Page\$	-	String
Pow	Number, Number	Number
Previous	All	All
ProjectPath\$	[Boolean]	String
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
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
Time\$	String	Number
Today	-	Date
Token\$	String, Number, String [,String]	String
ToRTF\$	String	String
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 "\" in front 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
```

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.

ATrim\$

Purpose:

Removes spaces from the beginning and end of a string.

Parameter:

String

Return value:

String

Example:

Assume that you have an address database which contains, among others, the field COMPANY in which a company's name has been entered. You want to print a label that contains the company name. So as not to waste space or unintentional word wrap, any spaces before or after the company name should not be printed.

The function `ATrim$(COMPANY)` removes all spaces that appear in the field COMPANY before or after the company's name.

From " combit GmbH ", the return value would be "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 "5.3. 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

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

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" or "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
Drawing	"" (empty String)
Barcode	"" (empty String)

Parameter:

Boolean

All

All (optional)

Return value:

All

Example:

Imagine that your database contains items that are free samples and the items that you normally sell.

Cond(PRICE=0,"Free Sample","Price:" + PRICE + " USD")

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

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.

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

Example:

```
Crosstab.Cells.Avg(.T.,2,0)
```

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

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 "-", it 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)".

Parameter:

String

Return value:

Date

Example:

```
Date("17.10.2006")
Date("10/17/2006")
Date("2006-10-17")
```

Tip:

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).

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 "5.4. 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, 7/11/2006"

Date$(Today(),"%2wthWeek; %D, %2d/%2m/%4y")
Result: "45th Week, Thursday, 7/11/2006"

Date$(Today(),"%D, %3d/%02m/%4y")
Result: "Thursday, 7/11/2006"
```

DateHMS

Purpose:

Converts three numbers for hour, minute and second into a date.

Parameter:

Number Hour
Number Minute
Number Second

Return value:

Date

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 ("2006.10.20"), Date ("2006.2.29"), 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:

```
Date (2006, 11, 1)
```

Day

Purpose:

Determines the day (1...31) of the month and returns it as a number.

Parameter:

Date

Return value:

Number

Example:

```
Day(Date("17.10.2006"))      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("17.10.2006"))    Result: "17"
```

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(Today())` Result: 4

If the day for the current date is a 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(Today())` Result: "Wednesday"

If the day for the current date is a Wednesday.

Drawing

Purpose:

Converts a string type file path into a drawing.

Parameter:

String

Return value:

Drawing

Drawing\$

Purpose:

Converts a drawing into a string type file path.

Parameter:

Drawing

Return value:

String

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

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")
```

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

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 <code>RTrim\$()</code> and <code>LTrim\$()</code> .
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:

<code>FStr(3.142, "#")</code>	Result: <code>"3"</code>
<code>FStr(5003.1, "#,###.&&")</code>	Result: <code>"5.003,10"</code>
<code>FStr\$(3.142, "#.###")</code>	Result: <code>"3,142"</code>
<code>FStr\$(3.142, ".#####")</code>	Result: <code>"*****"</code>
<code>FStr\$(3.142, "(#.###)")</code>	Result: <code>" 3,142 "</code>
<code>FStr\$(-3.142, "(#.###)")</code>	Result: <code>" (3,142) "</code>
<code>FStr\$(3.142, "+#.###")</code>	Result: <code>" +3,142 "</code>
<code>FStr\$(3.142, "-#.###")</code>	Result: <code>" 3,142 "</code>
<code>FStr\$(-3.142, "-#.###")</code>	Result: <code>" -3,142 "</code>
<code>FStr\$(3.142, "&&&&&&")</code>	Result: <code>"003,142"</code>
<code>FStr\$(3.142, "***.***")</code>	Result: <code>"**3,142"</code>
<code>FStr\$(3.142, "\$\$\$.\$\$\$")</code>	Result: <code>"\$\$3,142"</code>
<code>FStr\$(3.142, "###.***")</code>	Result: <code>" 3,142"</code>
<code>FStr\$(5003.1, "#,###.&&")</code>	Result: <code>"5.003,10"</code>
<code>FStr\$(3.142, "#####")</code>	Result: <code>" 3"</code>

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

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]

Number Saturation [0-1]

Number Lightness [0-1]

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).

Parameter:

String Text

String Hyperlink

Boolean (optional) Embedded

Return value:

String

Example:

```
Hyperlink$("combit", "http://www.combit.net")
```

Tip:

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.

If

see Cond

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

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(2453992)      Result: 2006.09.14`

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.

Note: This function can only be used in the footer lines of tables or in objects linked with tables! 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/library/default.asp?url=/library/en-us/intl/nls_8rse.asp

See also Chapter "5.4. 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 "5.4. 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 "5.4. 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

Zahl (optional) Format

Return value:

String

Example

```
LocDate$(Date("17.11.2006"),"USA") Result: "11/17/2006"
```

See also Chapter "5.4. 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("17.11.2006"), "DEU") Result: "17.11.2006"
```

See also Chapter "5.4. 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

Example:

```
LocNumber$(123, "USA") Result: "123.00"
```

See also Chapter "5.4. 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

Example:

```
LocTime$ (Now(),"USA")          Result: 9:05:22 AM"
```

See also Chapter "5.4. 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

Example:

Whether the current minute has the value "10" should be determined by a condition. The value of the current minute will be determined and compared with the value "10":

```
Minute()=10
```

Mode

Purpose:

Calculates the mode (most common value) of the set of values that result from the first parameter / formula.

Parameter:

Zahl 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

Example:

`Mode(UnitsInStock)`

Month

Purpose:

Determines and returns the month (1...12) as a number.

Parameter:

Date

Return value:

Number

Example:

```
Month(Date("2006.10.17")) Result: 10
```

Month\$

Purpose:

Determines and returns the month (1...12) as a string.

Parameter:

Date

Return value:

String

Example:

```
Month$(Date("2006.10.17")) 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 second-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

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 the 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 $(\text{Base}) ^ (\text{Exponent})$.

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"
```

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\LL12\crosstab.lsr
```

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

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:

<code>StrRPos("Normalconsumer","or")</code>	Result: 1
<code>StrRPos("Normalconsumer","r")</code>	Result: 13
<code>StrRPos("Normalconsumer","r",1)</code>	Result: 13
<code>StrRPos("Normalconsumer","r",2)</code>	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.

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)
```

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, 11/8/2006"

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)>>..."
```

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:

<code>Year(Today())</code>	Result: 2006
<code>Year\$(Date("1.1.2006"))</code>	Result: 2006

Year\$

Purpose:

Determines the year of a date and returns it as a string.

Parameter:

Date

Return value:

String

Example:

<code>Year\$(Today())</code>	<code>Result: "2006"</code>
<code>Year\$(Date("1.1.2006"))</code>	<code>Result: "2006"</code>

5.3. 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.

The following bar codes are available in List & Label:

- 2-of-5 DataLogic
- 2-of-5 Industrial
- 2-of-5 Interleaved
- 2-of-5 Matrix
- 3-of-9
- Aztec
- Codabar
- Code 128
- Code 39
- Code 39 with CRC
- Code 93
- CODE11
- Datamatrix
- DP-Identcode
- DP-Leitcode
- EAN 128
- EAN 13
- EAN 14
- EAN 8
- extended Code 39
- FIM
- German Parcel
- Japanese Postal Code
- Maxicode
- Maxicode/UPS
- MSI
- MSI PLAIN
- MSI+10+10
- MSI+10+CD
- MSI+11+10
- PDF417
- Pharma-Zentral-Nummer
- Postnet
- RM4SCC, KIX(®)
- Royal Mail with CRC
- SSCC/NVE
- UCC-14
- UPC-A
- UPC-E

2-of-5 Industrial

A number code of low information density.

Permitted characters: '0'..'9'

A Code is $(14 \times \text{number of characters} + 18)$ bar widths wide.

2-of-5 Interleaved (ITF)

A number code of high information density, requires high print quality.

Permitted characters: '0'..'9'. The number of characters must be even.

A code is $(9 \times \text{number of characters} + 9)$ bar widths wide.

2-of-5 Matrix

A number code of high information density, requires high print quality.

Permitted characters: '0'..'9'.

A Code is $(10 \times \text{number of characters} + 18)$ bar widths wide.

2-of-5 Datalogic

A number code of high information density, requires high print quality.

Permitted characters: '0'..'9'.

A code is $(10 \times \text{number of characters} + 11)$ bar widths wide.

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.

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 $2 \times 3 + 6 \times 1$ (characters '0'..'9', '/', '\$') or $3 \times 3 + 5 \times 1$ (characters '+', ':', '+', 'A'..'D') bar widths wide. The characters for the frame code will not be printed with the text.

Alpha39 (or also 3-of-9 or Code39)

One of the few codes that can also display letters. All characters must be entered; the customary bracketing with '*' should also be entered (*TEST*).

Permitted 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.

Code39 with CRC

Character set: ' ','\$','/', '%','*', '+','-', '0'..'9','A'..'Z'

Code93 (simple and extended)

Code93 is an extension of Code39, but has the advantage that it is somewhat smaller. It covers the complete 128 bytes of the ASCII character set, 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.

Code128

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.

CODE11

Character set: '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.

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.

DP-Identcode

A number code of high information density, requires high print quality.

Permitted characters: '0'..'9'.

The code must conform to the following format:

"nn.nnnnn.nnnn"

"nn.nnnnn.nnnnn"

"nn.nnn.nnnnnnn"

A Code is $(9 * \text{Number} + 9)$ 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.

Permitted characters: '0'..'9'.

The code must conform to the following format:

"nnnnnn.nnn.nnn.nn"

A Code is $(9 * \text{Number} + 9)$ 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.

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:
Character		Replacement [using chr(...), chr\$(...)]
NUL		chr\$(255)
FNC1		chr\$(254)
FNC2		chr\$(253)
FNC3		chr\$(252)
FNC4		chr\$(251)

EAN13, JAN13

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|nnnn|aaaa"      (normal EAN13)
"cc|nnnn|aaaa|xx"   (EAN13 for periodicals, "ISSN")
"cc|nnnn|aaaa|xxxx" (EAN13 for books, "Bookland")
"ppp|nnnn|aaaa"     (normal EAN13)
"ppp|nnnn|aaaa|xx"  (EAN13 for periodicals, "ISSN")
"ppp|nnnn|aaaa|xxxx" (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 $(12 \times 7 + 11) \times 0.3 \text{ mm} = 2.85 \text{ cm}$.

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.

EAN14, UCC14

These number codes require a high print quality.

Permitted characters: '0'..'9'. The code must be 14 characters long.

EAN8, JAN8

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 $(8*7+11)*0.3\text{ mm}=2.01\text{ cm}$.

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.

FIM Barcode

Minimum size: $1/2" * 5/8" = 12.7\text{ mm} * 15.87\text{ mm}$.

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.

Permitted characters: '0'..'9'.

A Code is $(14*\text{Number}+18)$ bar widths wide.

Relation: 1:2

Japanese Postcode

Japanese postcode.

Allows characters: n=[0-9], Address=[A-Z], [0-9], [-]

Formats: nnn-nnnn, then max. 13 character address

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.

Maxicode and PDF417

(Application dependent)

Character set: 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".

Pharma-Zentral-Nummer

Character set: '0'..'9'

The PZN code is used for identifying pharmaceuticals. The pharmaceutical central numbers (=PZN) are distributed by the "Informationsstelle für Arzneispezialitäten IfA GmbH". The represented string of numbers has to correspond to the following format:

"nnnnnn"

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: $1.245'' \times 4/16'' = 31.6 \text{ mm} \times 6.35 \text{ mm}$ (10-digit).

Minimum bar spacing $1/24'' = 1.058 \text{ mm}$.

The error-correction digit will be automatically amended.

This barcode will be automatically printed in the correct size.

RM4SCC, KIX®

Character set: '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

Character set: '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 character set 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.

SSCC/NVE

The SSCC (Serial Shipping Container Code) barcode is being increasingly used in the logistics industry.

Character set: [0-9]

Format: nnnnnnnnnnnnnnnnn (17 figures)

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 $(13 \times 7 + 6) \times 0.3$ mm = 2.88 cm.

5.4. 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		

6. Index

@

@User 40

2

2-of-5 Datalogic 220
2-of-5 Industrial 220
2-of-5 Interleaved 220
2-of-5 Matrix 220

A

Abs 163
AddDays 163
AddHours 164
AddMinutes 164
AddMonths 164
AddSeconds 165
AddWeeks 165
AddYears 165
Alias\$ 166
Alignment 66
Alpha39 221
Appearance conditions 91
Appendix 158, 228
Arrange 65
 Alignment 66
 Back One 66
 Forward One 65
 To Back 65
 To Front 65
Asc 166
AskString\$ 167
ATrim\$ 168
Avg 168
Aztec 220

B

Back One 66
Background103, 113, 116, 117, 126
Barcode 169

Barcode contents 113

Barcode objects

 Barcode contents 113

Barcode Objects 112

Barcode\$ 169

Barcodes 8, 112, 219

 2-of-5 Datalogic 220

 2-of-5 Industrial 220

 2-of-5 Interleaved 220

 2-of-5 Matrix 220

 Alpha39 221

 Aztec 220

 Codabar 220

 CODE11 221

 Code128 221

 Code93 221

 Datamatrix 221

 DP-Idencode 222

 DP-Leitcode 222

 EAN128 222

 EAN13 223

 EAN14 223

 EAN8 223

 FIM Barcodes 224

 German Parcel 224

 JAN13 223

 JAN8 223

 Japanese Postcode 224

 KIX 225

 Maxicode 225

 MSI 225

 NVE 226

 PDF417 225

 Postnet 225

 PZN 225

 Royal Mail 226

 SSCC 226

 UCC14 223

 UPC-A 226

 UPC-E 226

BarcodeType\$ 169

C

Case\$ 170

Chart Objects 132

Charts			
100% Stacked	136		
Axis Label	140		
Category Axis	139		
Clustered	135		
Colors	142		
Data Labels	139		
Datasource	137		
Examples	142		
Filter	137		
Legend	137		
Line	134		
Minimum Share	138		
Multi Row	135		
Options	141		
Properties	137		
Range	141		
Ribbon	134		
Sample Rows for Designer	138		
Segment Size	138		
Series Axis	139		
Stacked	135		
Sub Types	134		
Type of Calculation	138		
Types	133, 155, 156		
Value Axis	140		
CheckMod10	170		
Chr\$	171		
ChrSubst\$	171		
Circles	7		
Codabar	220		
CODE11	221		
Code128	221		
Code93	221		
Color	93		
Column Properties	123		
column width	53		
Cond	172		
Contains	172		
Content	98		
Context menu	13		
Continued	173		
Count	173		
CountIf	174		
Crosstab Objects	144		
Crosstab.Cells.Avg	174		
Crosstab.Cells.Max	175		
Crosstab.Cells.Min	175		
Crosstab.Col	175		
Crosstab.Col\$	175		
Crosstab.Row	176		
Crosstab.Row\$	176		
CStr\$	176		
D			
Datamatrix	221		
Date	177		
Date			
calculating with	163		
Formats	33		
Date			
Formats	178		
Date\$	178		
DateHMS	179		
DateInRange	179		
DateToJulian	180		
DateYMD	180		
Day	180		
Day\$	181		
Diagrams			
Data filters	148, 150		
Data Filters	148		
key	149		
Key	150		
Distinct	181		
DoW	181		
DoW\$	182		
DP-Identcode	222		
DP-Leitcode	222		
Drag & Drop	21		
Drawing	182		
Drawing\$	183		
E			
EAN128	222		
EAN13	223		
EAN14	223		
EAN8	223		
Editing Objects	63		
Ellipse	7		
Ellipses			
Insert	110		
Properties	110		
Ellipses	110		

Empty	183	Formatting Numbers	34
Even	183	Formulas	22, 40
Exceeding text transfer	132	Forward One	65
Exit	21	Frac	185
Exp	184	Frames	92
Exp10	184	FStr\$	185
Export media	46	Functions	30
Expression	22	Abs	163
Date Formats	33	AddDays	163
Fixed Text	26	AddHours	164
Numeric Formats	34	AddMinutes	164
Operators	35	AddMonths	164
Syntax	29	AddSeconds	165
Value Types	29	AddWeeks	165
Expression Modes	23	AddYears	165
F		Alias\$	166
Fax Variables	44	Asc	166
File		AskString\$	167
Exit	21	ATrim\$	168
New	42	Avg	168
Open	43	Barcode	169
Open	43	Barcode\$	169
Print Sample	56	BarcodeType\$	169
Save	45	Case\$	170
Save As	45	CheckMod10	170
Filter	40	Chr\$	171
Record filters	41	ChrSubst\$	171
Records	41	Cond	172
FIM Barcodes	224	Contains	172
FirstHeaderThisTable	184	Continued	173
Fixed Text	26	Count	173
Footer Lines	126	CountIf	174
Form Controls		Crosstab.Cells.Avg	174
Button	154	Crosstab.Cells.Max	175
Checkbox	153	Crosstab.Cells.Min	175
Combobox	153	Crosstab.Col	175
Edit	153	Crosstab.Col\$	175
Form Controls	152	Crosstab.Row	176
Form Templates		Crosstab.Row\$	176
Create	156	CStr\$	176
Properties	156	Date	177
Formatted Text Objects		Date\$	178
Contents	130	DateHMS	179
Formatting	95	DateInRange	179
Date\$	33	DateToJulian	180
Format Editor	107, 125	DateYMD	180
FStr\$	33	Day	180
		Day\$	181
		Distinct	181
		DoW	181
		DoW\$	182

Drawing	182	Odd	203
Drawing\$	183	Ord	203
Empty	183	Page	204
Even	183	Page\$	204
Exp	184	Pow	204
Exp10	184	Previous	205
FirstHeaderThisTable	184	ProjectPath\$	205
Frac	185	RainbowColor	205
FStr\$	185	RegexMatch\$	206
FStr\$()	30	RemainingTableSpace\$	206
GeometricAvg	187	Rep\$	207
Hour	187	RGB	207
HSL	187	Right\$	208
Hyperlink\$	188	Round	208
If	188	RTrim\$	208
Int	189	Second	209
IsNull	189	Sqrt	209
JulianToDate	189	StdDeviation	209
LastFooterThisTable	190	Str\$	210
LastPage	190	StrPos	211
Lastpage()	31	StrRPos	211
Left\$	190	StrSubst\$	191, 212
Left\$()	32	Sum	212
Len	191	Time	213
List	160	Today	213
Locale\$	192	Token\$	214
LocCurr\$	192	ToRTF\$	214
LocCurrL\$	193	TotalPages\$	214
LocDate\$	193	UnitFromSCM	215
LocDateTime	194	Upper\$	215
LocNumber\$	194	Val	216
LocTime\$	194	Variance	216
Log	195	Woy	216
Log10	195	Year	217
Lower	195	Year\$	217
LTrim\$	196		
Max	196	G	
Maximum	196	GeometricAvg	187
Median	197	German Parcel	224
Mid\$	197	Group Footers	129
Min	198	Group Lines	126
Minimum	198	Grouping Objects	69
Minute	199		
Mode	199	H	
Month	200	Help	9
Month\$	200	Hour	187
Now	200	HSL	187
NthLargest	201	HTML-Objects	154
NthLargestIndex	201	Hyperlink\$	188
NthValue	202		
NULL	202		
NumInRange	202		

I

If	188
Including projects	55
Insert	
Barcode Objects	112
Chart Objects	132
Crosstab Objects	144
Ellipses	110
Form Controls	152
Form Templates	156
HTML-Objects	154
Line objects	109
OLE Container	156
Picture objects	111
Rectangles	109
RTF Text	129
Table Objects	114
Text Objects	102
Int	189
Interface	10
Interlinking	
At End, Keep Size	77
sequential	77
Size adaption	76
spatial	75
Inverse interlinking	79
Invert Selection	64
IsNull	189
ISO 3166	227

J

JAN13	223
JAN8	223
Japanese Postcode	224
JulianToDate	189

K

Keyboard	68
Use	20
KIX	225

L

Labels	
Define Format	47
Paper Format	45
Predefined Formats	47

Print Order	49
Templates	47
Last Page	71
LastFooterThisTable	190
LastPage	190
LastPage()	31
Layer Window	16
Layers	69
Appearance Condition	70
Assigning	71
Copying Objects into	71
Define	69
Switch Visibility	71
Left\$	190
Len	191
Line Chart	134
Line definitions	
editing	120
Line layouts	119
Line objects	
Insert	109
Properties	109
Line objects	109
Line objects	109
Linefeed	27
Lines	7
Linking Objects	73
Lists	8
Locale\$	192
LocCurr\$	192
LocCurrL\$	193
LocDate\$	193
LocDateTime	194
LocNumber\$	194
LocTime\$	194
Log	195
Log10	195
Lower	195
LTrim\$	196

M

Mail Variables	44
Master-Detail Report	118, 119
Max	196
Maxicode	225
Maximum	196

Median	197
Mid\$	197
Min	198
Minimum	198
Minute	199
Mode	199
Month	200
Month\$	200
Moving Objects	67
MSI	225

N

Name	90
New	42
Next object	65
Now	200
NthLargest	201
NthLargestIndex	201
NthValue	202
NULL	202
NumInRange	202
NVE	226

O

Object Bar	13
Object List	72
Object Properties	91
Objects	17, 62
Appearance conditions	91
Arrange	65
Barcode-Objects	112
Chart Objects	132
Color	93
Content	98
Copy to Layer	71
Crosstab Objects	144
Edit	63
Ellipses Objects	110
Form Controls	152
Form Templates	156
Formatting	95
Frames	92
Grouping	69
HTML-Objects	154
Insert	62
Layer assign to	71
Line objects	109

Lines	7
Links	73
Moving	67
Name	73, 90
Object List	72
OLE Container	156
Picture objects	111
Position	91
Properties	89
Rectangles Objects	109
RTF Text	129
Selecting	63
Sizing	67
Table Objects	114
Text Objects	102

Odd	203
OLE Container	156
Open	43
Operators	35
Arithmetic	38
Logical	38
Relation	37
Options	
Alignment Grid	49
Compatibility	53, 55
for Objects	50
for Projects	49
for Workspace	52
Precision	49
Preview	51
Ord	203

P

Page	204
Page Layout	
Labels	45, 47
Page\$	204
Paragraph properties	106
PDF Export	49
PDF417	225
Picture file	7
Keep proportions	157
Picture objects	111
Insert	111
Properties	111
Position	91
Position and Size Dialog	68
Postnet	225
Pow	204

Preview	18
Previous	205
Previous object	65
Print	56
Printer	58
Print Sample	
Lists	57
Print Sample	56
Procedure	
Project open	43
Project	14
Fax Variables	44
Import	44
Include	55
Mail Variables	44
Open	42, 43
Print	56
Save	45
Save As	45
Types	44
ProjectPath\$	205
Properties of Objects	89
Property List	19
PZN	225

R

RainbowColor	205
Record filters	41
Record Filters	41
Rectangles	7, 109
Insert	109
Properties	109
RegExMatch\$	206
RemainingTableSpace\$	206
Rep\$	207
Report Structure	98
Resizing Objects	67
RGB	207
Ribbon Chart	134
Right\$	208
Round	208
Royal Mail	226
RTF Text	129
RTrim\$	208

S

Save	45
Save As	45
Second	209
Select All	64
Selecting	63
Selection	
Invert	64
Next object	65
of Multiple Objects	64
of Objects	64
Previous object	65
Selection Mode	64
Shortcuts	
Properties	20
Tool window	20
Sizing Objects	67
Sketch in File Dialog	44
Sqrt	209
SSCC	226
Status Bar	13
StdDeviation	209
Str\$	210
StrPos	211
StrRPos	211
StrSubst\$	191, 212
Subreports	118, 119
Sum	212

T

Table contents	117
Table objects	
Rotate headers	148
Table Objects	114
Appearance Conditions	123
Column definitions	120
Column Properties	123
column width	53
Columns edit	120
Define Footer Lines	126
defining lines	118
Format lines	122
Formatting	33
Frames	92
Group Footers	129
Group Lines	126
Line layouts	119

Rotate Header	125	ToRTF\$	214
Table contents	117	TotalPages\$	214
Tables	8	U	
Tabs	27	UCC14	223
Tabulator	105	UnitFromSCM	215
Text and Variables	7	UPC-A	226
Text contents	103	UPC-E	226
Text Objects	102	Upper\$	215
Background color	103, 113, 116, 117, 126	User Variables	40
Format	107, 125	User-Variables	40
Formatting	33	V	
Frame	103	Val	216
Insert	102	Variable List	16
Linefeed	27	Variables	25
Lines edit	103	Variance	216
Paragraph properties	106	W	
Properties	102	Workspace	14
Tabs	27	Woy	216
Text contents	103	Y	
Variables	25	Year	217
Time	213	Year\$	217
To Back	65	Z	
To Front	65	Zoom	15
Today	213		
Token\$	214		
Tool Bar	13		
Tool Window			
Layers	16		
Objects	17		
Preview	18		
Property List	19		
Table Structure	18		
Variables List	16		