

combit List & Label

Designer Help



List & Label® 16



No responsibility is taken for the correctness of the information contained in this manual. The information is subject to alteration without previous notice. combit GmbH accepts no liabilities in connection with this document. The availability of many of the functions described in this manual (e.g. the procedure for accessing the functions) is dependent on your system's version and release, the installed service packs (e.g. operating system, text processing software, mail program etc.) and the configuration of the system. If in doubt, please ask the person responsible for IT.

This manual or excerpts from this manual may not be copied or replicated in any other form (e.g. digital) without the written approval of combit GmbH. PDF creation utilizes wPDF3 (c) wpCubed GmbH - <u>www.pdfcontrol.com</u>

Copyright © combit GmbH 1992-2011; Rev. 16.002 www.combit.net

All rights reserved.



Introduction

With the List & Label Designer, you create or edit different print templates for printing information that originates either from a database or another data source. In the Designer, you then have all the data at your disposal and can prepare it for printing in different ways.

The Designer always uses the data that is passed from the program, in other words, the application handles the process of selecting and compiling the data.

Thus, "the Designer" is not a self-contained application, but rather is incorporated in List & Label as an inherent part of an application.

The Designer \diamondsuit the print, preview and export dialog \diamondsuit is normally started via a menu item (e.g. Configuration > Print or Output > Print).

See also:

- General
- Screen



General

The different print templates are referred to as "Projects". Along with the actual print information, a project also contains, above all, layout specifications such as page size and orientation, fonts, colors, frames, circles, lines, graphics etc. The List & Label Designer can handle three different types of projects: Lists, labels and cards.

The individual elements of such a project are called "objects". For example, a project can contain text objects, picture objects and a report container.

The required objects are normally selected in the object list, created on the workspace with the mouse and then given the respective contents and layout properties.

The List & Label Designer provides different types of objects which you can position freely in the workspace and change the size as required.

A	Text and contents of variables
1	Lines
	Rectangles
0	Circles and ellipses
<u>~</u>	Picture object
 	Barcode
	Report container or Table object (depending on the application): The Report container can hold tables, charts and crosstabs.
la.	Charts . Schematic view of data.
	Crosstabs. For presenting data in multiple dimensions.
	Formatted text. For changing the formatting within a line.
2	Form elements. For entering data in the preview and PDF format.
	Gauge

	HTML content
1	OLE container. For linking OLE documents (e.g. Word, Excel).
	Form templates are placed in the background of the workspace as a template so that other objects can be aligned to them. Form templates are not included when printing.



Screen

The Designer interface consists of a workspace and different tool windows.

- § You add objects by means of the 'Objects' toolbar.
- § The objects are placed in the workspace. You change the shape and size of the workspace via **Project > Page Setup**.
- § You can select the viewing mode by means of the tabs in the bottom margin of the workspace: Layout, Layout Preview, Preview.
- § The status bar with the current mouse position, the current operation and the position/size.
- § The variable list displays all variables which are available in the current project; for list projects, all available fields are also displayed.
- § The layer list shows the different display layers.
- § The hierarchical object list shows all objects in the order in which they will later be printed.
- § The "Report Structure" shows all elements belonging to a report container in the order in which they will later be printed.
- § The properties of the selected objects or project are displayed in the properties tool window.



Getting Started

This chapter describes how to create a simple list based on the combit List & Label Sample Application.

If you use List & Label as part of an application, you can usually start the Designer via a menu item or similar. The functionality can differ slightly from the description below.

See also:

- The Sample Application
- The Sample Database
- Produce a Simple Report
- Producing a Simple Invoice
- Creating a Simple Label



The Sample Application

The Designer is not available as a self-contained application, but rather is incorporated in List & Label as an inherent part of an application. For this reason, the Sample Application is used to create the examples.

You will find the Sample Application in the program menu of the List & Label installation. Alternatively, you can download it as part of the combit List & Label program installation at <u>www.combit.net/en/reporting/</u>.

You can then start the Designer via the "Design" menu item. Start the print, preview and export dialog via the "Print" menu item. Under options, you will find various configuration settings such as the language of the program interface or the number of sample files.

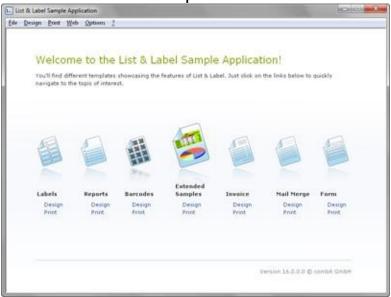


Figure 2.1: Sample Application



The Sample Database

The List & Label Sample Application contains a Microsoft Access Database for a fictitious company called "Northwind" with sample data for a food wholesaler.

It is a relational database meaning that the data is distributed across several tables. Each table contains data elements that are related to one another. For instance, one table contains the product data. Another contains the product categories. A table's columns are referred to as fields (product name), and the rows of a table as data records.

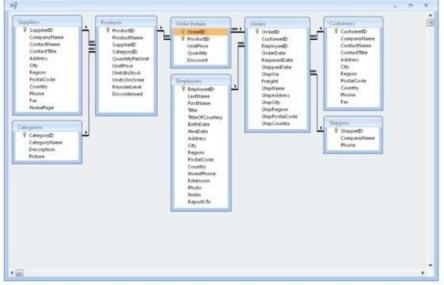


Figure 2.2: Structure of the Microsoft Northwind sample database



Produce a Simple Report

For a first simple report, you will produce a product list in alphabetical order with a report title and a company logo.

- 1. Start the combit List & Label Sample Application
- 2. Choose Design > Extended Samples to start the Designer.
- 3. A file selection dialog will appear. To start a new project, choose the New button.

Note: This button is not available in Windows XP. Enter a name for the new print template in the "File name" field and then choose "Open".

To open or edit an existing project, choose Open.

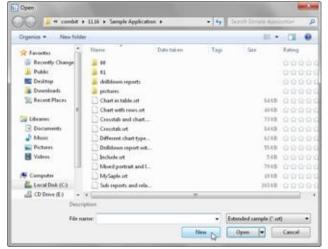


Figure 2.3: Open dialog

4. The standard project for the respective project type is displayed. This standard project is normally an empty workspace with a certain paper size and alignment.

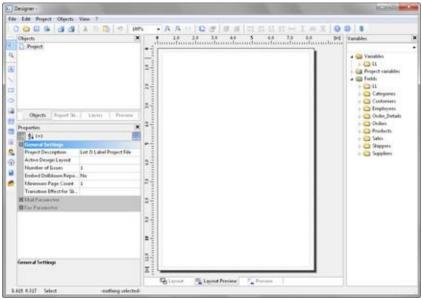


Figure 2.4: Empty workspace

With label projects, a certain label format (size and arrangement of the individual labels on the sheet) is already specified as a default value.

Note: A standard project is a standard template for creating projects. However, you can change it to suit your needs and save it again under the name "Default".

5. If you have chosen "New project wizard" under Project > Options > Workspace the project wizard will start. The project wizard simplifies the job of creating new projects by leading you through different page setup options. But you will not be using the wizard for this example.

See also:

- Inserting a Company Logo Into the Report
- Adding a Title to the Report
- Insert the Table for the Product List
- Formatting Table Fields
- Displaying a Preview of the Report
- Adding a Page Number
- Print or Export the Report



Inserting a Company Logo Into the Report

For the company logo, please use the "sunshine.gif" file, which you will find in the directory for the Sample Application. We also need a "Picture" object. Proceed as follows:

1. In the "Objects" toolbar, choose the "Picture" object type.

Figure 2.5: "Objects" toolbar

Note: Objects are your project's building blocks. They are generated in the workspace where they are also given a border with which their size and position can be changed. This border defines the space that the object takes up and thus also the maximum size to which the contents of the respective object can be expanded. Objects may overlap fully or partly.

2. In the workspace, point the mouse to the position where the upper left corner of the object is to begin. The mouse cursor changes to a crosshair. Hold down the left mouse button and drag the crosshair to the lower right corner of the planned object. Release the mouse button when the object (the dashed border) is the right size.

Note: Objects can be added to the workspace in different ways: via the menu **Objects > Insert**, via the toolbar or via keyboard shortcuts, or with Drag & Drop from the list of variables.

3. A file selection dialog will appear. Select the image file that you want.

Note: The following formats are available: WMF, EMF, BMP, DIB, PCX, SCR, TIFF, GIF, JPEG, PCD, PNG and ICO. As a general rule, you should use the RGB color space (not CYMK). List & Label itself supports transparency in PNG files by using the corresponding Windows functions. In our experience the majority of printer drivers do not support transparency so that reports with e.g. partly transparent PNG files should thoroughly be tested on the actual hard-software combination. If that is not possible we recommend doing without the alpha channel. In addition, you can insert the picture into the project by enabling the ("**Embed image in project file**" checkbox option. This option copies the image to the project thus making it available even without the external file.

4. After you select the file, the logo will be inserted.

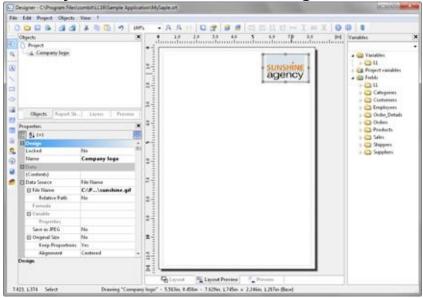


Figure 2.6: Report with picture object



Adding a Title to the Report

Use a text object to add a title to the report. Text objects let you place text in the workspace. As well as fixed text, you can also insert the contents of fields (variables) from the database (e.g. company name), or you can use functions (page number, date etc.).

In the "Objects" toolbar, choose the "Text" object type.
 In the "Objects" toolbar, choose the "Text" object type.

```
Figure 2.7: "Objects" toolbar
```

- 2. In the workspace, hold down the left mouse button and pull the object to the required size.
- 3. The formula wizard will now appear which you can use to define the contents of the text object. This dialog consists of a series of tabs each containing different elements to be edited. The following chapters explain the meanings of these elements in more detail. § Variables and Functions: the available variables, fields
 - and functions.
 - § Condition: for defining IF-THEN-ELSE conditions.
 - § Text: for entering fixed text and tabs.
 - § Date Format: different date formats.
 - § Number Format: different number formats.
 - § Operators: available conjunction operators.

You can also enter the expression that you want directly in the edit box or modify the text that is there (e.g. put something in brackets).

Therefore, enter our title "List of products" directly. Fixed text must be enclosed in quotation marks. Enter names of variables and functions without brackets.

	 Functions: 	[
e 🤐 Vanables Cal Project vanables	Barrencal functions Date functions Source functions Source functions Date functions Date functions Date functions Barcode functions Barcode functions	
Lat of products"		post 2g

Figure 2.8: Text in the formula wizard

Note: Please note that there are two ways of writing expressions, depending on their use. You will find more information about this in the chapter <u>Variables, Formulae</u> and <u>Expressions</u>.

- 4. The title will be displayed when you close the dialog with OK.
- 5. Enlarge the letters by changing the font size in the mini toolbar that appears automatically.

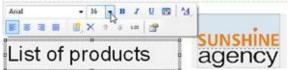


Figure 2.9: Formatting menu in mini toolbar

6. You have now added the title:

No. 10 No. 10<	2
Name X I I II III III IIII IIIIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	
Name X I I II III III IIII IIIIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	
4. Deeper tape And + 10 + 10 + 10 + 10 + 10 + 10 + 10 + 10	
SUNSHINE ABAR	and down
List of products agency	
List of products agency	
	and a second
- C 194	
C Com	
日 注引 (3.4)	
C Mark Land Insteil (and) Annual 1	100 C
Agente Agente	
a m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
Lated No. 9-	
Etai	
B References	
Reference Languese	
Farst is Police Inc. 2-	
There are a second se	
Nature F	
Witness 1	
Appropriet Catalities Martin Trave Ro-	
19 Outline Invet 8	
Papers No.	
Pageboot Referer No. 10	
Profession (Part 115 Auto 176 m)	
8-	
and the second se	
Drap II	
Giner Standbarn Street	
234,129 Selver Teer - Kritter 1234e - 4140e 1355e - 4140e Boot	

Figure 2.10: Report with text object



Insert the Table for the Product List

Use the "Report Container" object to add a table to the report. As the name says, a report container can hold several objects: tables, charts and crosstabs can be added in any order.

Note: The report container is not available in all applications. In applications that don't have the report container, use the "Table" object.

1. In the "Objects" toolbar, choose the "Report Container" object type.

Figure 2.11: "Objects" toolbar

- 2. In the workspace, hold down the left mouse button and pull the object to the required size.
- 3. A selection dialog will appear for the chosen object type. <u>Choose the "Table" object type.</u>

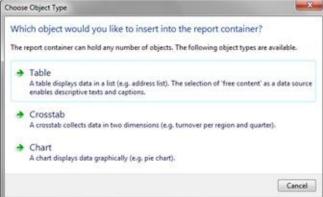


Figure 2.12: Choosing the object type

4. Now supply the data source in the following dialog. All available tables are shown hierarchically; in other words, under the tables, you will find the relational tables in each

case.

Please choose the source	table here:	_		
Free content Categories Customers Customers Customers Order Details Orders Products Sales Shippers Suppliers				
0		OK	Canc	el

Figure 2.13: Choosing the data source

Select the "Products" table because it contains the fields that we want for our product list.

5. A selection dialog will appear with all the fields in the "Products" table. In addition, underneath the "Products" table, you will also find the fields in the tables "Categories" and "Suppliers" which have a 1:1.relationship with "Products".

Field Selection Wizard			
ine name:			
As multiple lines of the same type can be used, you should choose an appr	opriate description to be able to disting	ush the line better from other	5.
insertable fields:	<u>⊆ohumns:</u>	×	4.4
•	Products Productflame		
CaregoryB CaregoryB	Products Category/D@Categori Products UnitPrice	es.Categorytiame	
Please choose the fields to be printed in this line. Properties can be set after	wardı.		
0			Cancel

Figure 2.14: Field selection dialog

In this dialog, now choose the columns for the table. Double-click a field, e.g. "ProductName". The field will be added to the "Columns" area.

- Repeat this step for all fields that are to be shown in the table; i.e. also the "CategoryName" field from the linked table "Categories" and the "UnitPrice" field for the unit price. Confirm your selection with OK.
- The table will now be displayed in the workspace.
 § The selected fields are displayed in the data line, in other words, the data line contains the data.
 - § In addition, a header line is automatically produced. Header lines are used mostly as column titles, i.e. the selected field names are now shown here as text.
 - § If you pull the report container widthwise to make it wider or narrower while holding down the CTRL key, the columns will be adjusted proportionally to fit.
 - § List & Label adjusts the width of the columns automatically. You can adjust the width of a column manually by moving the separating line to the right or the left with the mouse.

Note: This changes all table columns, whose separators are within +/-2 mm from the mouse position. If you hold down the CTRL key, the action will only be carried out for the line on which the mouse is positioned. If the "Column width modification affects next column" option is enabled (Project > Options > Workspace), you can alter the column width while making the next column smaller.

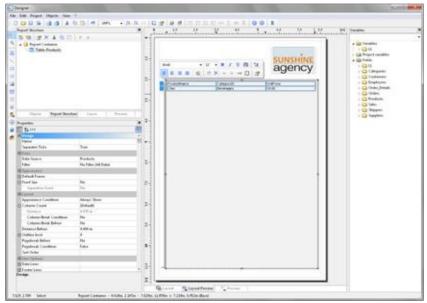


Figure 2.15: Report with table in the report container object



Formatting Table Fields

Chapter <u>Creating Professional Tables</u> describes how you can format and modify a table. Only a few basic formatting options are explained here.

 In the product category column title, the field name "CategoryName" is shown. Click the respective field in order to change this text to "Category". The formula wizard will now appear, which you can use to change the contents of the field. Please consider that text must be enclosed in quotation marks.

idit Table		
anables and Functions	Test Date Format Number Format Operators	
	Functions	10
> ∰ Yanables > ∰ Fields	Numerical functions Mathematical functions Date functions String functions Min: functions Min: functions Bagespate functions Barcode functions	•
"Category"		Just 1
9		Σ _Σ β
Category		
21 C	OK	Cancel

Figure 2.16: Formula wizard with text input

2. In addition, we want to format the unit price as currency and align the entire column to the right. You will find both formatting options in the mini toolbar that appears automatically. Click in the upper left corner to select the field containing the unit price.



Figure 2.17: Selecting a single field

3. Click the button for the formatting dialog

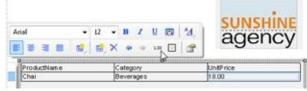


Figure 2.18: Formatting a field directly

4. A selection dialog will now appear in which you can select the formatting that you want for the numeric field. To do this, choose the type on the left hand side i.e. "Currency". In the right pane, you can specify the currency formatting in detail. The settings for the relevant application will be used in each case as standard. Alternatively, you can choose the system setting or a user defined setting.

No format Number	Dgcimal places:	Application setting	
Currency	Decimal point:	Application setting	
Date Time	Thousands separator:	Application setting	
Date and Time Percentage	Currency symbol	Application setting	
Angle Date-/Time Difference	Format for positive values:	Application setting	
	Format for <u>pegative</u> values:	Application setting	•
	Exponential format:	No	
	No output if <u>v</u> alue is 0 V Always 0 before decimal point		
Preview			
	(\$12,345.67)		

Figure 2.19: Formatting dialog

5. Finally, select both fields (the header and the data line) to align them to the right. To do this, hold down the CTRL key and select both fields by clicking in the upper left corner in each case. Then click the button for right alignment.



Figure 2.20: Multiple field selection



Displaying a Preview of the Report

Until now, you have only seen the report in layout view as a structure with a record. To get an impression of the result, you can display a preview of the report. Use the tabs in the bottom margin of the workspace to change to the preview.

Note: Real data preview mode is not supported by all applications.

🖶 Layout 🗮 Layout Preview 📃 Preview

Figure 2.21: "Preview Options" toolbar

The report will now be displayed with the data from the "Products" table. You can also change the number of "products" or data records in the Sample Application.

- 1. To do this, save the report with File > Save.
- 2. End the Designer with File > Close.
- 3. In the Sample Application, choose Options > Settings and increase the maximum number of root records to 50.
- Open the print template that you created again via Design > Extended Samples, and switch to preview mode. Your report will look roughly like this:

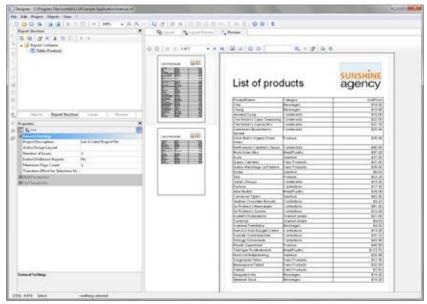


Figure 2.22: Real data preview



Adding a Page Number

It's a good idea to add a page number in the lower area of the page. To do this, add a new text object. You will be using functions (such as the page number function) as well as fixed text with this object.

- In the "Objects" toolbar, choose the "Text" object type.
 Figure 2.23: "Objects" toolbar
- 2. In the workspace, hold down the left mouse button and pull the object to the required size.
- 3. The formula wizard will now appear which you can use to define the contents of the text object. The available functions are shown in the right pane. You can use an auto filter with this list. Type "page" in the filter field. This will cause all functions containing the expression "page" to be displayed.

§ The "Page\$ ()" function returns the page number.

§ The "TotalPage\$ ()" function returns the total number of pages.

Add the "Page\$ ()" function to the result area by doubleclicking.

- 4. If you now want to output a footer in the form of "Page 1 of 2", you can enter the text "of" directly in the usual way. Please consider that individual parts must be joined with a "+" and that text must be enclosed in quotation marks.
- 5. Now add the "TotalPages\$ ()" function by double-clicking to get the total number pages. The formula will look like this:

	Functions:	100
⊳ 🤬 Project variables ⊳ 🏜 Project variables	Description Descripti	
Page\$() + " of " + TetalPages\$()		inert DE 1 DE 1

Figure 2.24: Formula wizard with functions and text

6. You can center this line in the report in the usual way:

Anal 12 - B I U E 4

Figure 2.25: Quick format "centered" alignment

7. Switch to preview mode to view the result:

In Drive Dave De					
	DD HIMS + B.B.Y				
Report Nuclear	×	Giand Risenstreen	To Parameter		
10 J X X 11	1.1.1.1		10000	B. + # 0	(4))
a Ga Report Container			C. (19) C. MR. (10) (2 MAL 10) (2	AL 1 (21)	
Table Products					
			List of proc	ducts	sunshine agency
		Total Ba	Public	Largery	Lindfrom
		KOP 11 0	120	Perenager	814.00
		10 E	Day	Enclosed .	810.00
		10 10 10 10 10 10 10 10 10 10 10 10 10 1	Designations	and and a second	811.00
			Destinate Cash Searching		411.00
there. Separate	then is in the second		Electricitan's Datas Min	Conditionality (8.0.00
Rearing	×		Conceptuary Brown Marry		10.00
		Landonthom Balls	Datast		1000
1 A 112		and the party of the local division of the	Lindia Battin (Ingena 2000) Prime	F10848	8.8.00
and the second se		ED: 27 5	Forth-south millionly fands	Commonia	441-00
Locied.	244	Bern Bar 1	March Street Television	Mant off	80.00
Adapted 1		10 HT 11	max.	Evented	111 (R)
#Drive		BC - BB	Strates Calvates	Erem Producto	821.00
Contraction			Shares Marchage Laif where	Crany Products	838.00
Biggs marrie			Party.	Greeker.	81.05
(i) to 1 ground	Temperat		200	Frend	8.0.24
Ballion Magned	The second se		State-Means	Code and	
Louis or Particip	ha		Patrice	Contractions	917.61
Frank .	177		2004-06-000	Read States	414.00
			Canavio 1981	Contraction of	393.20
Pointer .			Die Router of a Manual State	C-amplications	841.00
Bises .			The Randword's Docement	Contractions	911.00
Approximent Conditions	Alloan Store		Eugen constants	instant anali	821.00
Daties heat			Durantee	Erwis Causio	20
Papinesi	No.		Tomata Faitados	Downani .	
Faploral Bries	100		Particle for the get Creme	Contractions	814.00
The Printeen	19-448 12-214 7-528 12-528 wd		Standa Sumhkärchen.	Contentions	814.21
CANCER .			Schuge Louissian	- aritaking	94.00
			Place Laurines Thange Exceloses	Product Name	96.00 F0120
			Pharinger Rocking Source	Factoria Control	1010
			Diagonas Vite	East Protein	11.8
			Manager and Value	El ani Pitalachi	11.2
			interest interest	Party Products	8192
			Downald Ave	Rentages	814.00
U			Disease Lines	Frankel.	814.00
Deinge					
				1.40	
				199	

Figure 2.26: Real data preview in the Designer



Print or Export the Report

You can start the print function directly from the higher-level program, from the real data preview or via **File > Export**.

- 1. Choose File > Export.
- 2. The Print Options dialog will appear.
 - § Under "**Print target**", you can change the printer or the printer configuration.
 - § Select the output format (e.g. preview, printer, Excel) under "Direct to".

int Options			
Select printing options			
Print target			
付 HP LasetJet 5		{	Qhange
🚽 Direct to	🍯 Printer	•	Qptions
Save options permanen	бу		
Options			
First page:	1		
Pages:			
🤨 AJI			
Page gange(s)			
(Enter pages or ra	nges, separated by comm	a if necessary, er	c 'L3-4,10-")
Prigt	All selected pages		•
(2)	[Start	Cancel

Figure 2.27: Output options

3. If you have selected an export format (such as PDF), choose the storage location in the following "Save As" dialog and enter a name for the file to be created in the "File name" field.

Organize • Ne	w folde	er							0
Favorites Facently Char Char Char Char Char Char Char Char			ne 60 01 dnildown reports pictures	Date taken		Tags	Sae	Rating QQ1 QQ1 QQ1 QQ1	20
📲 Computer	-	•			m				
File name:	espo	t.pdf							
Save as type:	PDF F	iles (*.p	df)						
		regist after t	the file in the ered application he output. exported files by						

Figure 2.28: Save dialog when exporting to a different format



Producing a Simple Invoice

In this chapter, you will now meet functions and possibilities for designing reports.

For the next example, you will create a multi-page invoice with covering letter, invoice header, totals, footers, enclosure and printed copy.

See also:

- Create a New Print Template
- Adding a Company Logo
- Add the Address Field
- Create Invoice Header With Number and Date
- Adding the Item List
- Alignment and Formatting
- Background Color and Frame
- Invoice Footer With Totals
- Create Additional Footer Lines



Create a New Print Template

- 1. Start the combit List & Label Sample Application (see Chapter <u>The Sample Application</u>).
- 2. Choose Design > Invoice to start the Designer.
- 3. A file selection dialog will appear. To start a new project, choose the New button.

Note: This button is not available in Windows XP. Enter a name for the new print template in the "File name" field and then choose "Open".

To open or edit an existing project, choose **Open**.



Adding a Company Logo

Use the picture object to add a logo.

- In the "Objects" toolbar, choose the "Picture" object type.
 Image: Image:
- 2. Pull the object to the right size and select an image file (see Chapter Inserting a Company Logo Into the Report).



Add the Address Field

To add an address, use the text object. Text objects let you place text or the contents of fields in the workspace.



Figure 2.30: Address field: it should look like this.

In the "Objects" toolbar, choose the "Text" object type.
 In the "Objects" toolbar, choose the "Text" object type.

Figure 2.31: "Objects" toolbar

- 2. Pull the object to the required size.
- 3. The formula wizard will now appear which you can use to define the contents of the text object.

This dialog consists of a series of tabs. On the "Variables and Functions" tab, select the variable for the company address (company) from the list of available variables and fields.

You will see an auto filter field above the list of variables. This means that you can enter "Company" to display all fields and variables containing this expression. Select the variable that you want by double-clicking and confirm your selection with OK. You have now defined the first line of the address field.

	Fext Date Format Number Format Operators	_
Customer Address Sample variable Customer.Company Tale B Popert variable Customer.Company This text can be defined by your applie		
Customer.Company		T ₂

Figure 2.32: Formula wizard with variable

4. A text object can hold as many paragraphs as you want and they can all have completely different display properties. You can add more paragraphs by means of the mini toolbar which appears automatically. Choose "Append paragraph" to define an additional line.

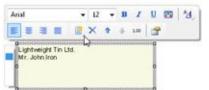


Figure 2.33: Adding another paragraph via the mini toolbar

- 5. The formula wizard will now appear, which you can use to define the contents of the paragraph. Now enter the first name and the last name of the recipient. First choose the "Firstname" variable.
- 6. You should insert a space before choosing the "Lastname" variable to prevent the contents of the two variables from being placed end to end. A space is simply "Text". Fixed text must be enclosed in quotation marks. So now enter "+" as a joining operator followed by " " for the space.
- 7. Now select the "Lastname" variable. You must naturally also join this with "+". The whole line looks like this:

Customer.Firstname +** + Customer.Lastname	111 (33
	ZZ 🖸
()	
🚺 John Iron	
0 🔁	OK Cancel

Figure 2.34: Linking variables and text

8. Now continue with the other variables: street and city. You have now completed the address field.

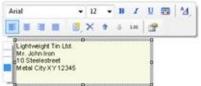


Figure 2.35: Text field with 4 paragraphs



Create Invoice Header With Number and Date

Use the text object again to add the text "Invoice", the invoice date and the invoice number to an invoice header.



Figure 2.36: Invoice header: it should look like this.

- 1. In the "Objects" toolbar, choose the "Text" object type.
- 2. Pull the object to the required size.
- 3. The formula wizard will now appear, which you can use to define the contents of the text object. Type "Invoice".
- 4. Add another paragraph by means of the mini toolbar that appears automatically and the "Append paragraph" (insert) button.
- 5. The formula wizard will appear. Type "Invoice date".
- 6. Before you enter the invoice date, insert a tab stop so that the invoice date is aligned to the right. You can insert a tab stop by means of the "Text" tab. You create a tab stop with the "Tab" button and you define the position and alignment with the "Properties" button.

Note: Only one tab stop can be inserted on each line. A tab stop causes the preceding text to run only as far as the tab stop. A tab stop that is right aligned will cause the text that follows it to be justified to the right. The distance from the left margin determines the position of the tab stop.

Since a tab stop is also a character, it must also be

enclosed in quotation marks. Alternatively, in this example, you can insert the tab stop with "Insert" to the existing "Invoice date" text.

Edit Text	L Tab Properties			×
Variables and Functions Condition Text Date Format	 Jeft aligned gopht aligned gecimal tab 	Position	0.000 🚡 mm	
e Linefeed Inth Properties				ert .
"Date+"				E C
· 🗆 Date -				5

Figure 2.37: Adding a tab stop to a text expression

7. Finally, position the cursor outside of the text field and insert the "Invoice_Date" variable with the "Variables and Functions" tab.

The format will be converted automatically.

Conversion with the "Date\$()" function is essential since the invoice date has the "date" type and the value that is returned is already defined as a "string" by the "Invoice date" text expression entered earlier. The return value must always be unique. The date value must therefore be converted to a string by means of a function. The "Date\$()" function is one way of doing this.

Variables and Functions Condition Text Date Format	Number Format Operators	
Format	Result	
"50, 59,500,504" "50, 59,5020,5024" "50,5020,5024-59" "50,5020,5024-59" "50,5020,5024-59"	Friday, 2010-7-2 Friday, 2010-07-02 Friday, 7-2-2010 Friday, 07-02-2010 July 2 2010	
5000 "5002d55020".5049; "502945025" "50550".505; 504550255025 "50550".506 "5050".506 "5050".506 "10550".506".	7/2/2211 42.47.2010.90.92.40 2.7.2010.902.40 2010.772.8942.40 AM 2010-Q3 Q3 2010	J
Choose a format for your date		Insert
"Date+" + date\$(Invoice_Date, "%u")		rea a D _L g
Date=7/2/2010		, 10 (

Figure 2.38: Format conversion with Date\$()

The automatic format conversion uses "%x" as the formatting parameter. If you want a different output format, switch to the "Date Format" tab. You can choose a format here and then select the "Invoice_Date" variable.

8. Now add the invoice number in the same way. The format will be converted automatically here as well.

Figure 2.39: Format conversion with Str\$()

Conversion with the "Str\$()" function is essential since the invoice number is a field with the "number" type and the value that is returned is already defined as a "string" by the "Invoice number" text expression entered earlier. The return value must always be unique. The number must therefore be converted to a string by means of a function. The "Str\$()" function is one way of doing this. This function has 3 parameters: The first parameter is the number that is to be converted to a string; the second parameter specifies the minimum length and the third parameter the number of decimal places.

- 9. Now increase the font size for "Invoice". To do this, select the relevant line.
- 10. Set the font size to 28 by means of the "font size" button in the mini toolbar that appears automatically.

11. Now increase the distance to the next line, i.e. open the paragraph properties dialog via the "Object properties" button in the mini toolbar. In this dialog, you can define the properties for each line. Set the value for the "Paragraph Spacing" property for the first line to 12 pt.

Qaragraphi: 📑 🛣 🗙 🖄 🔯 🔹 🔹	(E) 24 (+)		
"Invoice"	Appearance		
	E Font		[Arial 24.0 pt]
"Date+" + date\$(Invoice_Date, "%x")	Format	None	
"Number.+" + str\$(Invoice_Number.0,0)	E Layout		
	Alignment	Left	
	Appearance Condition	Always Show	
	Blank Optimization	Yes	
	Inerasable	No	
	Justified	No	
	Line Spacing	0.0 pt	
	🖽 Line Wrap	Wrap	
	Paragraph Spacing	12	10
	Paragraph Spacing Space between this and the fi	sllowing paragraph	

Figure 2.40: Paragraph properties dialog

12. And you now have an attractive invoice header:



Figure 2.41: Invoice header



Adding the Item List

Use the "Report Container" object to add the item list to the report. As the name says, a report container can hold several objects: tables and freely defined content, charts and cross tabulations. You need the "Table" element for the item list, and "Free content" for the covering letter. Please see chapter Insert the Table for the Product List for an introduction about how to create tables.

1. In the "Objects" toolbar, choose the "Report Container" object type.

```
19 9 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
Figure 2.42: "Objects" toolbar
```

- 2. In the workspace, hold down the left mouse button and pull the object to the required size.
- 3. A selection dialog will appear for the chosen object type. Choose the "Table" object type.

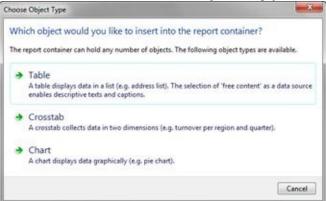


Figure 2.43: Object type selection

4. Now select the data source in the following dialog. All available tables are shown hierarchically; in other words, under the tables, you will find the related tables in each case.

Select the "Items" table because it contains the fields that we want for our item list.

- 5. A selection dialog will appear with all the fields in the "Items" table.
- In this dialog, now choose the columns for the item list. Double-click one after another on the fields: "Quantity", "No", "Description1", "UnitPrice". This will add the fields to the "Columns" area. Confirm your selection with OK.

noestable fields:	<u>C</u> olumns:	×	-
Vanubles	Bern Quantity Bern Na Bern Decopion I Bern UndPrice		
Tease choose the fields to be printed in this line. Pro			

Figure 2.44: Field selection dialog

7. The table will now be displayed in the workspace. A header line with strings as the column titles and a data line with the contents of the fields will be created automatically.

List & Label adjusts the width of the columns automatically. You can adjust the width of a column manually by moving the separating line to the right or the left with the mouse.

Note: This changes all table columns, whose separators are within +/-2 mm from the mouse position. If you hold down the CTRL key, the action will only be carried out for the line on which the mouse is positioned. The "Column width modification affects next column" option (Project > Options > Workspace), lets you alter the column width while making the next column smaller.

Lightweight Dear Mr. Jo 10 Steelest Metal City >	hn Iron rest		sunshin agency	CALLER .
krial		B Z U ⊠ ⁴ 4, ● ● 1= □ 27	Invoice Date 7/2/2 Number 47110	
0 Couerbity		9		
1.00	EXPSA01	Description1 Southern Atrica Explorer: 20-day to from Cape Town to Victoria Falls excluding fight	UnitPrice 1500.00	0
	0) 010 01 01 01 01 01 01 01 01 01 01 01 01		******	

Figure 2.45: Changing column widths with the mouse

8. In the next step, you add a new column: "Total". Select the "UnitPrice" field by clicking in the upper left corner of the field. Then choose the "Append column" button in the mini toolbar.



Figure 2.46: Select column

- 9. A menu will appear for the selected field type. Choose "Text" here.
- 10. The formula wizard will now appear, which you can use to define the contents of the field. Type "Total" and confirm with OK.

A message will now appear giving you the option of

reducing the width of the columns to allow the new column to be inserted in the visible area. Confirm this dialog. You have now added an additional column to the header line. You now need this column in the data line as well.

11. Now select the data line field containing the item price and add a new column to it.



Figure 2.47: Add a new column

The total price is now to be calculated here. The total price is calculated by entering "Quantity*ItemPrice". To do this, select the relevant variables by double-clicking them.

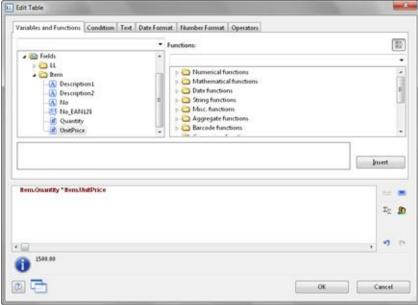


Figure 2.48: Multiplying two values in the formula wizard

Note: Open the list of available operators by clicking the "Operators" tab. Operators join two or more values or variables to give a new value. In this way, you can formulate arithmetic expressions (basic arithmetic operations) or logical expression. You can combine multiple operations in one expression. Please consider the calculation hierarchy of the operators used and place them in brackets if necessary. The "+" operator has a special meaning. It is not only suitable for additions ("Number" and "Date" value types) but is also used for joining strings.



Alignment and Formatting

Make use of the numerous formatting options to present the invoice in the form that you want. For example, change the column title and format the amounts as currency.

- Click the column title that you want to change. The formula wizard will now appear, which you can use to change the contents of the field. Please consider that text must be enclosed in quotation marks.
- 2. In addition, you should also align the columns for the unit price and the total price with the column titles. To do this, hold down the SHIFT key and select all 4 fields by clicking in the upper left corner of the "Item Price" data line. Then choose the button for right alignment in the mini toolbar.
- 3. To format both amounts as currency, hold down the CTRL key and select both fields by clicking in the upper left corner of each one. Then choose the button for the formatting dialog.

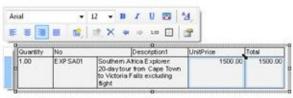


Figure 2.49: Multiple field selection

4. A selection dialog will now appear in which you can select the formatting that you want for the numeric field. To do this, choose the type on the left hand side i.e. "Currency". In the right pane, you can specify the currency formatting in detail. The settings for the relevant application will be used in each case as standard. Alternatively, you can choose the system setting or a user defined setting.

No format Number	Dgcimal places:	Application setting	+
Currency Date	Decimal pgint:	Application setting	•
Time Date and Time	Thousands separator:	Application setting	-
Percentage	Currency symbol	Application setting	-
Angle Date-/Time Difference	Format for gositive values:	Application setting	-
	Format for negative values:	Application setting	-
	Egponential format:	No	•
Preview	(\$12,345.67)		

Figure 2.50: Formatting a field

- 5. Format the field for the quantity as a number without decimal points.
- 6. Our item list will now look roughly like this:

Querety	No	Cescription1	UnitPrice	Total
1.00	EXPSAD	Southern Africa Explorer 20-day four from Cape Town to Victoria Falls excluding fight	\$1,500.00	\$1,500.00
2.00	EXPMALO	Maldives diving trip: 1.4 days, southern Male Atoli, Planador Desch **** excluding 6ight	\$1,000.00	\$3,600.00
2.00	EXPINICAL	Yucatan, On the Trail of the Mays, 2-week round trip, excluding fight	\$1,200.00	\$2,400.00
1.00	TRPPRAD	From Gothic to Baroque to Art Nouveau. City trip to Prague Including accommodation	\$355.00	\$355.00
1.00	TRVBRADS	Flight only: Brazil	\$783.00	\$783.00
1.00	EXCRAROS	Paris, admission to the latest cabaret showst the Moulin Rouge including three-course meal	\$178.00	\$178.00

Figure 2.51: Preview



Background Color and Frame

In order to structure the invoice and make it more attractive, it's a good idea to give at least the column headers a colored background. You can also modify the background color.

- 1. To do this, select a column title.
- 2. Now open the paragraph properties dialog via the "Object properties" button in the mini toolbar. In this dialog, you can define the properties for each column and row. Hold down the CTRL key and select all columns.
- 3. Click the "+" sign to expand the "Background" property group.
- 4. For the background, choose the "Pattern/block color" property and the color, e.g. LL.Color.LightSkyBlue. You reach the LL.Colors via the arrow button and the standard color dialog via the ... button.

Header Line	✓ Data Leve	Footer Line	Group Header	Group Footer	
X = + #	XXBD	happen.	출출 E+3		
a 😂 Line Definiti	on 1		artise.		
A Quantit	8	57.A	operatation of		
(A) "P4o"			schground	Pattern/Block Color	
- Descrip		100	Pattern		1
A UnitPro			Celar	LL Color LightSkyBlue	
A "Total"			Adult Color	LL Color Milety	
			End Color	3.L. Color White	
			Fading-in-Factor	58%	
		8	Diawaraj	File Diame	
		6	port as Picture	No	
		(日)	tre		Minal 12.0 pt
		E.	semat	None	
			station	<i>8</i> °	
		T	ext Format	Normal Test	
			tion		
		100	esk URL		
			gent		
<u></u>			grownd.Color 1		
-					

Figure 2.52: Object properties dialog for the table: background

5. In addition, you want to increase the spacing between the header and the data line. Since this is a line property (and not a column property), select the line.

Here you have the "Spacing" property. Choose e.g. 0.1 inch as "Bottom" to increase the spacing below the column title.

*	Header Line	٠.	Data Lase	-	Foote	Line	Geoup Header	Graup Footer	
	1 8 - 2	×	¥ 8	ED	t	10 24	1+2		(6)
	-					BDeniy	A		
	a 📥 Lete Defind					Dept	ey in Designei	Stew	
	- A "Quant	4				Name			
	-A 'No'					BAppe			
	- WestPa					E Defai	ik Fant		(Avial 12.0 pt
	Total"					BLayer			
	101 100						arance Condition	Always Show	
							e Empty Line	Yet	
						E Outle			
						E Space			000, 8.000, 8.000, 0.200 ws
						Le		0.000 in	
						To	p.	8.898 in	
							yle	0.000 im	
						Bo	thom	0.100 in.	
					_	Spacing. Left.	Left		
-	6								

Figure 2.53: Object properties dialog for the table: margins

6. Now it only remains to make the settings for the frame. You can define different settings independently for each cell. However, it's normally a good idea to make the settings for the entire table. For this reason, the settings for the frame are included in the table properties. You will find the table properties, as standard, on the lower left when you select the table in the "Report Container" tool window. When you select the property, there is a button which you can use to open the dialog for the frame properties.

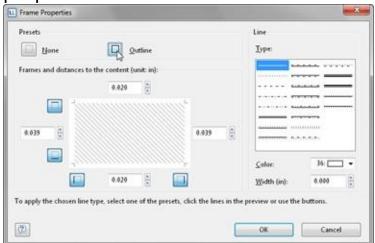


Figure 2.54: Frame properties

7. Choose "white" as the color and click "Outline" to apply the color.

Please note: If you don't apply the setting but simply confirm the dialog with OK, your settings will be lost!

8. The invoice will now look roughly like this in the preview:

ARC thus Imay Mr LiARC Di Gampilar			age	ency
			Invoid See	
Duartite		Description	-	Table
1.08	ROPENOV	South-en Alliss Explorer 25-last sout tion Cade Town to Notion Patienticate Bate	\$1,000.00	#1,550.00
2.00	Estavelia	Madows Birtshyrt: 18 days, middweit Klain Allet, Parades Reach. ^{16,16} mittading Rytt	\$1,000.00	\$1,500.00
2.08	Enhorical	Vacatari, On the Teal of the Marks 2 weat round the, micholog flight	\$r,200.00	\$2.400
1.00	TREFFICACE	From Gothic'to Baroque to All Insurema, Cito tracto Preque Richaling accommodation	\$105.00	\$265.0
100	EXCENERAL	Flight seld: (basis) Faint, advances to the basis object should the Moule Things tobuling three objects made	#795.66 #779.00	1753

Figure 2.55: Preview (frame and background)



Invoice Footer With Totals

You now need to output the net totals, the VAT and the total amount in the invoice footer. You can simply insert these values if they are held as fields in the database. In our case, however, you must first calculate the values. To do this, you use the "Sum()" aggregate function. This function totals the values that are passed to it. Since the totals are to be output at the end of the table, you define a footer for this purpose.

1. Click the "Append Line" button in the mini toolbar and then click "Footer".



Figure 2.56: Add new line

2. You have not yet defined a line for this line type so you will be asked whether you want to use an existing line definition from one of the other line types. This wouldn't save any time at this point. Therefore, choose the "Single field or free content" option.



Figure 2.57: Applying a line definition

- 3. The formula wizard will open to allow you to define the contents of the first column. Enter "Subtotal", not forgetting to enclose it in quotation marks.
- 4. Once you have created the first line, add another column to it for the aggregation. Use the "Sum()" function for aggregating values. You can either type it in directly in the result field in the formula wizard or you can select the function from the list of aggregate functions by double-clicking. As a parameter, enter the value to be aggregated for this case Quantity*ItemPrice.

Edit Table	the second s		
Variables and Functions Condition Test	Date Format Number Format Operators		
	Functions:		
> ∰ Yanables > ∰ Project variables > ∰ Fields	Markenatical functions Mathematical functions Mathematical functions Mathematical functions Minc functions Marc functions Agregate functions Baccode functions		
Calculates the sum of the parameter / form Example: Sum()flumber 'Unit price) -> < Sum()flumber)()Ecole av()) ->(flumber) Econesion for the value to be calcu	nula in the parameter. Total =		Junt
Sum (ternsthutPrice)			αs Σ _Σ
- 0'			. •
		OK	Cancel

Figure 2.58: Totaling with Sum()

- 5. The entire line should now be moved a little to the right. Hold down the CTRL key and drag the column separator on the extreme left at the outer edge of the table to the right. An empty column will be inserted automatically as a spacer on the extreme left.
- 6. You must also format the total field as currency, give both columns the same background color as the header line and set the "Top" margin to 0.1 inch to correspond with the "Bottom" margin of the header line.
- 7. As these footers are only to be output on the last page, even in the case of multiple page invoices, you must now specify the "Lastpage()" or "LastFooterThisTable()" function for the "Appearance Condition" line property.

e.,	Header L	ine	~	Data	Line	4	Footer Line		Group Header	Group Footer	
	1 1	- #	×	×			4		ĝ. c-1		8
-	~							80	enign		
	- C Li		tion 1					N	ame		
	-(A	"Total:	3					80	ata		Ξ.
						entity, Fals	30	13	(ontents)	Sum(Item.UnitPrice*Item	h
	1.00	Shumite	em.unit	Price	item.yu	entry, Pars	90	. 0	ptions		
									рриагалсе		
								E 8.	eckground	Pattern/Block Color	
								Ð	port as Picture	No	
								E Fe	set	[Arial 10.0	pQ.
								Fe	emat	\$123,456.12/(\$123,456.12)
								R	otation	0*	
								Te	ext Format	Normal Text	
								BA	ction		
								- Li	nk UFL		
								81	yout.		
								1 A	ignment	Decimal	
									nosseance Condition	Always Shrive	
Ì								Displ	eys the properties dialog	ŀ	
L	-			_				1			
2	-									OK Can	

Figure 2.59: Object dialog with 3 columns

8. You have now defined the first footer and the invoice looks like this:



Figure 2.60: Preview (footer)



Create Additional Footer Lines

You can create as many line definitions as required for each line type (data line, footer). The different line definitions are shown in the object dialog as a tree structure.

For our example, now create two more footers for the VAT and the grand total.

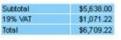


Figure 2.61: Preview 3 footers

1. To add more lines in the object dialog, first select an existing line definition. You can then create a new line definition with the "Append line definition" button.

	Table Contents - Barn		-
Edit columns and line definitions — e definitions — Column content —	Header Line Data Line P	Factor Line	General Auster Second Second
			Cancel Cancel

Figure 2.62: Object dialog with 3 columns

Alternatively, you can add a new line in the workspace via the context menu or with the mini toolbar.

2. A dialog will appear asking you whether you want to use

an existing line definition. It saves some time if you use the footer that you have already created as you will not have to configure the left margin, alignment or background colors. Therefore, choose the "Line definition 1" entry from the "Footer" area.

- 3. The line will now be inserted. Click the "Subtotal" field so that you can then change the content to "VAT 19%" with the formula wizard. Don't forget to enclose the text in quotation marks.
- 4. Once you have created the column, click in the column to enter the aggregation. Instead of the total net amount, the 19% VAT is to be output here. Therefore, you simply multiply the amount by 0.19. Sum(Item.Quantity * Item.UnitPrice * 0.19)
- Proceed in the same way with the third footer. Copy the existing footer, change the content and, in the formula, add the VAT to the total, like this: Sum(Item.Quantity * Item.UnitPrice * 1.19)
- 6. You have now defined the first footer and the invoice looks like this:

Quantity No Description1 UnitPrice Total 1.00 EXPSA01 Southern Akics Explorer: 20-day Falls excluding flight \$1,500.00 \$1,500.00 \$1,500.00 2.00 EXPMALD1 Maidwes duing the: 14 days, southern Main ASU, Paradise Beach *** excluding flight \$1,800.00 \$3,600.00 \$3,600.00 2.00 EXPVLC01 Vuoden, On the Trail of the Mays, 2-week round tip, excluding flight \$1,200.00 \$2,400.00 1.00 TRPPRAD1 From Odhic to Baroque to Art Nouveau, City to to Prague including accommodation \$355.00 \$355.00	ABC Busin Dear Mr. J 5 ABC Blv Springfield	ohn Smith d.			
1.00 EXPSA01 Southern Atrice Explorer: 20-day four from Gee Town to Victoria \$1,500.00 \$1,500.00 2.00 EXPMAL01 Meldvess dwng tht: 14 dwn, southern Mele Acti, P endse Beach "** excluding tight \$1,800.00 \$3,800.00 2.00 EXPMAL01 Meldvess dwng tht: 14 dwn, southern Mele Acti, P endse Beach "** excluding tight \$1,200.00 \$3,800.00 2.00 EXPVUC01 Vuotan, On the Trail of the Marya, 2-week round \$10, excluding tight \$1,200.00 \$2,400.00 1.00 TRPPRA01 From Gotricto Baroque to Art Nouveau, City thp to Prague including accommodation \$355.00 \$355.00				Date	7/2/2010 47110015
1.00 EXPSA01 Southern Atica Explorer. 20-day four from Gee Town to Victoria \$1,500.00 \$1,500.00 2.00 EXPMAL01 Meldves dwng thr. 14 dwn, southern Mile Abol, P endse Beach "** excluding tight \$1,800.00 \$3,600.00 2.00 EXPMAL01 Meldves dwng thr. 14 dwn, southern Mile Abol, P endse Beach "** excluding tight \$1,200.00 \$3,600.00 2.00 EXPVUC01 Vuotan, On the Trail of the Mays, 2-week cound the, excluding tight \$1,200.00 \$2,400.00 1.00 TRPPRA01 From Gotricto Baroque to Art Nouveau, City the to Prague including accommodation \$355.00 \$355.00	Guantity	No	Description	LinitPrice	Total
2.00 EXPMAL01 Maldhess diving trip: 14 days, pouthern Niele AOI, P andste Beach*** excluding fight \$1,800.00 \$3,800.00 2.00 EXPVUC01 Vuostan, On the Trail of the Mays, 2-week round trip, excluding fight \$1,200.00 \$2,400.00 1.00 TRPPRA01 From Gottic to Baroque to Art Nouveau. City trip to Prague including flagin \$355.00 \$355.00			Southern Atrica Explorer: 20-day tour from Cape Town to Victoria		
Mayn, 2-week round trip, excluding flight 1.00 TRPPRAD1 From Gothic to Baroque to Art \$355.00 \$355.00 Nouveau, City trip to Prague including accommodation	2.00	EXPMAL01	Maldives diving trip: 14 days, southern Male Alot, Planadise	\$1,800.00	\$3,600.00
Nouveau. City trip to Prague including accommodation	2.00	EXPYUC01	Maya, 2-week round trip,	\$1,200.00	\$2,400.00
			Nouveau. City trip to Prague including accommodation		
	1.00	TRYORADI	Flight only: Brazil	\$783.00	\$783.00
19% VAT \$1,071.23			19	% VAT	\$5,638.00 \$1,071.22 \$6,709.22

Figure 2.63: Invoice with 3 footers



Creating a Simple Label

For your first label, you will now create an item label.

- 1. Start the combit List & Label Sample Application
- 2. Choose Design > Item Labels to start the Designer.
- 3. A file selection dialog will appear. To start a new project, choose the New button.

Note: This button is not available in Windows XP. Enter a name for the new print template in the "File name" field and then choose "Open".

To open or edit an existing project, choose **Open**.

- 4. The standard project for this project type is displayed. This standard project is normally an empty workspace with a certain paper size and alignment.
- 5. Select a label format via Project > Page Setup. In the "Templates" tab, choose the template that you want from the predefined label formats of various manufacturers. This specifies automatically the size of the individual labels, how many are to appear on the sheet and how they are to be distributed. You will find more information about this in the chapter <u>Templates for Label Formats</u>.

(user-defined) They office and the Tat 2.500 in 9.100 i					(8)
11070 Ready Index Tab 2.500 in 9.100 i					
	n 0.000 in	0.000 in	1	1	0.77
11070 Ready Index Tab 3.000 in 1.880 i	n 0.000 in	0.120 m	1	5	5.13
11071 Ready Index Tat 2.500 in 9.100 i	n 0.000 in	0.000 in	1	1	0.77
11071 Ready Index Tab 3.000 in 1.130 i	n 0.000 in	0.090 in	1	5	5.13
11072 Ready Index Tat 2.500 in 9.100 i	n 0.000 in	0.000 in	1	1	0.77
11072 Ready Index Tal: 3.000 in 0.880 i	n 0.000 in	0.120 in	1	01	5.13
11073 Ready Index Tal: 2.500 in 9.100 i	n 0.000 in	0.000 in	1	1	0.77
11073 Ready Index Tal: 3.000 in 0.140 i	n 0.000 in	0.000 in	1	12	5.13
11074 Ready Index Tat 2.500 in 9.100 i	n 0.000 in	0.000 in	1	1	0.77
11074 Ready Index Tat 3.000 in 0.500 i	n 0.000 in	0.140 in	1	15	5.00
11076 Ready Index Tal: 2.500 in 9.100 i	n 0.000 in	0.000 in	1	1	0.77
11076 Ready Index Tal: 3.000 in 1.130 i	n 0.000 in	0.090 in	1	- 8	5.13
11077 Ready Index Tat 2.500 in 9.100 i	n 0.000 in	0.000 in	1	1	0.77
11077 Ready Index Tat 3.000 in 0.180 i	n 0.000 in	0.120 in	1	10	5.13
11078 Ready Index Tal: 2.500 in 9.100 i	n 0.000 in	0.000 in	1	I	0.77
11078 Ready Index Tat 3.000 in 0.840 i	n 0.000 in	0.000 in	1	12	5.13 +
4 [- design and the second se		and the second second	A	

Figure 2.64: Specify label format

6. A single label will now be displayed in the workspace.

See also:

Insert Object

• Printing Labels



Insert Object

A picture of the item is to be displayed as a picture object on the label together with two text objects (Item_No, Description and Price). Place a tab stop between the text "Item No." and the "Item_No) variable.

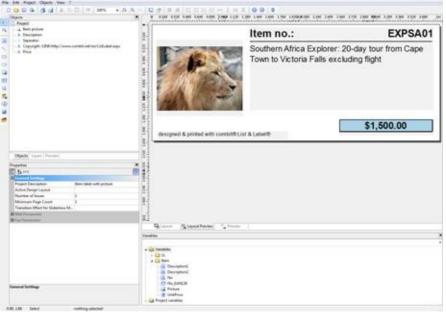


Figure 2.65: It should look like this

1. Begin by positioning the text object. In the "Objects" toolbar, choose the "Text" object type.

```
iqure 2 66: "Objects" toolba
```

- Figure 2.66: "Objects" toolbar
- 2. Pull the object to the required size.
- 3. The formula wizard will now appear, which you can use to define the contents of the text object.
- 4. Now follow the steps described in Chapter <u>Add the</u> <u>Address Field</u> onwards.
- 5. Use the picture object to add a logo. Drag the "Picture"

variable from the list of variables into the workspace. Pull the object to the required size.



Printing Labels

You can start the print function directly from the higher-level program, from the real data preview or via **File > Export**.

- 1. Choose File > Export.
- 2. The output options dialog will appear.
 - § Under "**Print target**", you can change the printer or the printer configuration.
 - § Select the output format (e.g. preview, printer) under "Direct to".
- 3. The "Select" button lets you specify the starting position when printing the sheet of labels. In this way, you can also print sheets of labels that have already been partly used. You will find a sample label sheet for your label project in the dialog for selecting the start position. Click the label where the print is to start.



Effective Workspace Techniques

This chapter will provide you with useful information and the most important techniques for working efficiently with the Designer.

- General Procedures
- Arrangement and Alignment of the Objects



General Procedures

- Choosing a Page Layout
- View Mode
- Status Line
- Mini Toolbar
- Default Settings for Font and Frame
- Real Data Preview



Choosing a Page Layout

The first task in a new project is to set up the page layout that you want. Choose **Project > Page Layout** to specify properties such the choice of printer, paper size and orientation.

If multiple layout areas are defined, the active workspace can be selected via the "Active Design Layout" project property.



View Mode

You can select the view mode by means of the tabs in the margin of the workspace:

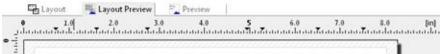


Figure 3.1: Tabs for selecting the view mode

- § In layout mode, you merely see the object frame and the contents of the objects as formulas.
- § The layout preview shows the objects in the WYSIWYG format. In addition, the objects are transparent and are drawn in the color of the layer to which they are assigned.
- § Preview mode display is a real data preview, provided that this is supported by the application.



Status Line

The status line is divided in three sections. § The current mouse position from upper left. § The active operation (e.g. selection). § Name, position upper left, position lower right, width, height and layer of the selected object. *635, 14339 Select Text - 0.725m, 1629in - 3.520in, 2.709in = 2.795in, 1400in (Base) Figure 3.2: Status line



Mini Toolbar

When you select a text object or a table object, a convenient toolbar in miniature format is displayed, the mini toolbar. You can close the mini toolbar by pressing Esc.

With the mini toolbar you can apply fonts, font sizes, orientations, text colors and formatting as well as open the object dialog.

Additionally with **text objects**: Add paragraph. Right mouse click: Add paragraph before the last selected paragraph. Move paragraphs upwards/downwards.

§ To select a complete paragraph, click onto the bar on the left. Hold CTRL or SHIFT to select multiple paragraphs or a complete range.



Figure 3.3: Mini toolbar for text objects

Additionally with **table objects**: Define new line, insert new row, move selected rows/cells to the left/right, borders.

- § To select a complete line, click onto the bar on the left. Hold CTRL or SHIFT to select multiple paragraphs or a complete range.
- § To select a field, click in the top left corner of the field. Hold CTRL or SHIFT to select multiple fields or a complete range.
- § To select a complete column, hold ALT. Hold CTRL or SHIFT to select multiple columns or a complete range.

Bern				In	-		0			hum	
E 3	-		12	12	×	٠	-0	1.00		đ	agency
Arial		•	12	٠	B	1	N.	13	14	SUNSHINE	

Figure 3.4: Mini toolbar for table objects



Default Settings for Font and Frame

Choose **Project > Options > Objects.**

The "Select" button under "Object font" lets you choose the default font to be used for objects. Under "Color preferences", you can specify the border and the filling for objects.



Figure 3.5: Project options for objects

New objects will be created according to these settings. However, you can change the settings later individually for each object.

When you start a new project, it's a good idea to configure these settings using suitable values to keep the effort required for making manual changes to a minimum. The settings only apply for the current project.

The "Default" parameter in the respective object properties (Font/Frame/Background) is active as standard for all objects.

Therefore, if the default values are changed later, this will also affect all objects whose font has not been changed manually.



Real Data Preview

The real data preview function is also available directly in the Designer provided that your application supports it. In this way, you can check the layout of a printout without wasting paper in order to do so. The screen preview is accurate down to the last detail (WYSIWYG � What You See Is What You Get), exactly as it would be when printed. After checking the layout, the actual print can be started from the preview without having to use the print command again. Also see chapter <u>Output Options</u>.

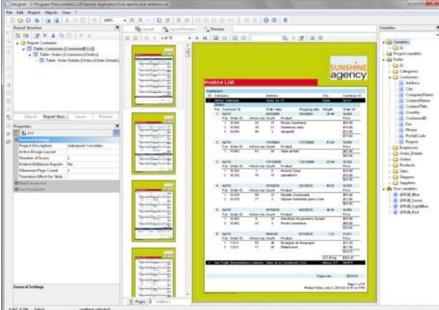


Figure 3.6: Real data preview in the Designer



Arrangement and Alignment of the Objects

Objects are your project's building blocks. They are generated in the workspace where they are also given a border with which their size and position can be changed. This border defines the space that the object takes up and thus also the maximum size to which the contents of the respective object can be expanded. Objects may overlap fully or partly.

- Arrangement as an Object List
- Displaying the Alignment Grid
- Aligning Objects
- Guides in the Workspace
- Size and Position of Objects
- Grouping of Objects
- Multiple Copies of Objects



Arrangement as an Object List

The object list defines the implicit print order of the objects, thus they are automatically sequentially linked to one another. Unlinked objects are printed first, followed by the tables and finally the objects that are linked in any other way.

The most important commands for the respective objects are available via context menus as in the work space. 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.

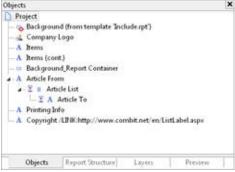


Figure 3.7: Print order in the object list



Displaying the Alignment Grid

Choose **Project > Options > Project.** Here, the "Show grid" option lets you specify whether a grid is to be shown in the background of the workspace to assist with orientation. You can specify the spacing of the grid lines in each case. The "Horiz./vertical synchronized" option causes the same grid spacing in both directions.



Aligning Objects

Via the **Objects > Arrange > Alignment** menu item, or the corresponding toolbar, you can align multiple objects with one another. At least two objects must be selected for the function to be enabled.

招調菇提州王网星

Figure 3.8: Toolbar for the alignment functions

- § Left, right, top, bottom: The selected objects are aligned to the border of the selection rectangle in which they are enclosed.
- § Centered: The selected objects are centered in the respective direction (horizontal or vertical) within the selection rectangle.
- § Size adjustment: The frames of the selected objects are adjusted to a common size in the respective direction (horizontal or vertical).
- § Equal shape distance: The distance between the edges of the selected object is set to be equal. The outer edges in each case are significant here. The average distance is determined relative to the edges of the objects and the objects are arranged accordingly. In the event that the selected objects overlap, i.e. the intermediate area has a negative value, the function will not spread the objects apart but instead, only the degree of overlapping will be adjusted to the average value.
- § Equal center distance: The distance between the centers of the selected object is set to be equal. The average distance is determined relative to the centers of the objects and the objects are arranged accordingly. In the event that the selected objects overlap, the function will

not spread the objects apart but instead, merely the degree of overlapping is adjusted to the average value.



Guides in the Workspace

You can create horizontal and vertical guides. To do this, position the mouse on a ruler, press and hold the mouse button and release it again in the workspace. The new guide is aligned in the same direction as the originally selected ruler. You can then move the guides around as you wish. A catch function helps you to position objects directly on the guide. This function does not connect the objects permanently to the guide but merely helps you when positioning objects.

If you hold down the CTRL key when you move the guide, the objects connected to the guide will be moved as well. The options relating to guides are defined via a context menu which appears when you click the right mouse button. This lets you choose a catch range in pixels for each guide. When you come within this range, the object will be drawn to the guide. The catch function is switched off if you hold down the CTRL function when working with objects.

aide		
<u>C</u> atch Range		10 Posel
Locked	-	20 Pixel
Position		30 Pixel
Delete		40 Pixel

Figure 3.9: Context menu for guides

You can fix the guides in the workspace and prevent them from being moved by mistake. You can also specify the position of the guide directly.



Size and Position of Objects

You can move each selected object or change its size. If you select multiple objects, you can change them all as if a single object were selected.

- Changing the size with the mouse: Select the object. If you position the mouse on the frame, you can change the size by pulling the frame inwards or outwards, as indicated by the arrows, while holding down the left mouse button. You must position the mouse in one corner of the frame in order to change the size both horizontally and vertically.
- 2. Moving with the mouse: Select the object. Hold down the mouse button and drag the object to the position that you want. If you press the SHIFT key, you will only be able to move the objects horizontally or vertically. The alignment remains the same.
- 3. Changing the size and moving with the dialog: You can also change the size and position of an object by means of the property list. You can enter values precisely here. If you double click the "Position" sub-item in the property list, a position dialog will appear which makes it even easier to enter the size and position of objects with the keyboard.
- 4. Using the keyboard to move objects or change their size: Select the object. Use the DIRECTION keys to move the object in the respective direction. Pressing the key once moves the object by the smallest possible unit, if you hold down the CTRL key, the object will be moved by ten times the smallest unit.



Grouping of Objects

You can group multiple objects that belong together and then modify them as if they were a single object. Please note that an object can only belong to one group. It is therefore not possible to combine groups to a higher-level group. In order to make a group of two or more objects, select the objects in question and then choose **Group** in the context menu.

To remove the grouping, choose the item **Ungroup**.



Multiple Copies of Objects

If you want to place several, similar objects with the same distance on the workspace, select the object and then choose **Multiple Copies** in the context menu. Define the number and spacing of the objects horizontally and vertically.



Mastering Appearance Conditions

In this chapter, you will examine the concept of appearance conditions using a multi-page standard letter or mail merge as an example. By means of appearance conditions, you can specify when objects, elements, paragraphs, lines or columns are to be printed or displayed.

In addition, it is frequently helpful to assign objects that belong together in terms of content into what is called a layer. An appearance condition can be defined for each layer. This condition specifies the conditions under which the objects assigned to this layer are to be printed.

- Where Will the Objects be Printed?
- Working With Appearance Conditions
- Working With Layers
- Practice: Create a Mail Merge Project



Where Will the Objects be Printed?

If a layer has not been defined for the objects, they will be printed on each page.

As an exception to this rule, table objects, cross tabs, text and formatted text are printed consecutively starting on the 1st page.

With text and formatted text, the "Page break" property must be set to "yes".

A new page will therefore be created automatically whenever the space available in the object is no longer sufficient. This means that you do not have to create a second page; this is taken care of automatically by table objects, crosstabs, text, formatted text or by means of the "Minimum page count" project property.



Working With Appearance Conditions

You can use filter conditions to control precisely the data to be output. In this way, you can specify conditions not only for displaying data records but also for displaying objects. These conditions are logical expressions whose results decide whether a certain data record or a certain object is printed or not. If the logical expression is true, the data record or the object is printed. If the logical expression is false, the data record or the object is not printed. In order to achieve this, you define a corresponding "Appearance condition". You will find these conditions in the properties of projects, layers, objects, elements, paragraphs, lines or columns.

For example, if you want to output the terms of payment in the footer of an item table, use the appearance condition for this line to specify that it is only to be printed on the last page. Otherwise, this footer will be printed on every page at the end of the table.

Example "Last page only": Lastpage()

Or you define a condition specifying that the company logo is only to be output if the letter is created as a PDF file, since, when outputting to the printer, the company logo is already printed on the letterheads.

Example "For PDF output only": LL.OutputDevice="PDF"

In appearance conditions for table footer lines, you can also use the predefined "Last page only" value. Internally, this entry uses the functions "Lastpage()" or "LastFooterThisTable()".

In appearance conditions for table headers, you can also use the predefined "First page only" value. Internally, this entry uses the functions "not Lastpage()" or "FirstHeaderThisTable()".



Working With Layers

If you define appearance conditions for a layer, you can then assign objects to this layer. This is very useful especially with large and complex projects as it allows you to hide or display the objects assigned to a specific layer via the checkbox in the tool window.

In this way, you avoid overlapping the different objects on the workspace which would otherwise make working on individual objects difficult.

Laye	rs.			>
	Base			
7	First Pa	ge		
7		ng Pages		
M	Fallows	ng Pages		
-	Objects	Report Stru	Layers	Preview

Figure 4.1: Layers tool window

- Defining Layers
- Assigning Objects to a Layer



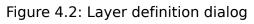
Defining Layers

You define layers via **Project > Layer Definitions** or by double clicking in the "Layers" tool window. In the "Layers" dialog, you can define as many additional layers as you want with the "Insert line" button. Each new layer appears initially with the name "Layer". You can enter a meaningful name for the layer in the "Name" field.

§ When creating new projects, the "Base", "First page" and "Following pages" layers are automatically defined.

- § So that you can differentiate better between the different layers on the workspace, it's a good idea to give the layers different colors. This also causes the objects in the respective layers to be shown in different colors in the layout preview display mode. This color has no effect on the actual print.
- § If you delete a layer, the associated objects are automatically assigned to the base layer. At least one layer must be defined.

Løyers					- 2
Youble	Name	Condition	*	*	4
	Base				
2	First Page	Page() = 1			
	Following Pages	Page() 🔿	1		
Settings					
Name:	First Page				
Condition	Page() = 1		Edi	t.,]
View:	Value is visible in current view			•	
1		OK		Cancel	



In the "Condition" field you define the appearance

condition for this layer. This appearance condition then applies for all objects on the layer in question, i.e. the associated objects are only printed when the condition for the layer is met.

Typical appearance conditions:

Condition	Explanation
no condition	The objects on this layer are always printed.
Page()=1	The objects on this layer are only printed on the first page.
Page()<>1	The objects on this layer are printed from the second page onwards.

Note: The Lastpage() function in an appearance condition can only be evaluated correctly if it is used within the footnote of a table object or in an object linked to a table. In the data lines, the result of Lastpage() is always "False".



Assigning Objects to a Layer

Once you have defined the layers, you can assign objects to them. You have two options:

 In order to assign multiple objects to a layer, select the objects in the workspace and choose Assign to layer in the context menu.

Select the layer that you want in the dialog that appears and confirm with OK.

ame	Condition
Base	
First Page Following Pages	Page() = 1 Page() <> 1

Figure 4.3: Dialog for assigning objects to a layer

In the workspace, the assigned objects automatically receive the color of the corresponding layer so that they can be easily differentiated from the other objects. This only affects the appearance on the workspace and not the print.

 Alternatively, you can also copy objects into a layer. This is useful when you want to include the same objects in different layers. Example: you create one layer for each language. You then copy all objects into the language layer and translate them.

The original object remains in its original layer and a copy of the object is created in an additional layer. To do this choose **Copy to Layer** from the context menu for the selected object.



Practice: Create a Mail Merge Project

You learned the basic procedures in Chapter 2. In this chapter, you will now meet other functions and possibilities for designing reports, using a serial or standard letter as an example. As opposed to the previous examples, you use the formatted text object here for the text of the mail merge and change the position of the object from the second page onwards.

See also:

- Create a New Print Template
- Adding a Company Logo
- Add the Address Field
- Adding the Date and Page Number
- Adding Formatted Text for the Letter
- Adjusting the Position of the Letter Text for Following Pages
- Assigning Objects to the Layer



Create a New Print Template

- 1. Start the combit List & Label Sample Application (see Chapter <u>The Sample Application</u>).
- 2. Choose Design > Mail Merge / Letter to start the Designer.
- 3. A file selection dialog will appear. To start a new project, choose the New button.

Note: This button is not available in Windows XP. Enter a name for the new print template in the "File name" field and then choose "Open".

To open or edit an existing project, choose **Open**.



Adding a Company Logo

Use the picture object to add a logo.

- In the "Objects" toolbar, choose the "Picture" object type.
 Figure 4.4: "Objects" toolbar
- Pull the object to the right size and select an image file (see Chapter <u>2.3.1</u>).
- 3. The company logo is only to be printed for PDF output. The logo is not to be printed when outputting to the printer. This means, you select the picture object and enter the "LL.OutputDevice = "PDF" logical condition in the "Appearance Condition" object property. Also see chapter <u>Overview of LL Variables and LL Fields</u>.



Add the Address Field

To add an address, use the text object. Text objects let you place text or the contents of fields in the workspace.



Figure 4.5: Address field: it should look like this.

In the "Objects" toolbar, choose the "Text" object type.
 Image: Im

Figure 4.6: "Objects" toolbar

- 2. Pull the object to the required size.
- 3. The formula wizard will now appear, which you can use to define the contents of the text object.

This dialog consists of a series of tabs. On the "Variables and Functions" tab, select the variable for the company address (company) from the list of available variables and fields.

You will see an auto filter field above the list of variables. This means that you can enter "Company" to display all fields and variables containing this expression.

Select the variable that you want by double-clicking and confirm your selection with OK. You have now defined the first line of the address field.

	at Date Format Number Format Operators	
Cutomes Cutomes Cutomes Cutomary Cutomary Cutomary Cutomary Cutomary Cutomare Cutomare Cutomare Cutomare Cutomare Cutomare Cutomare Complex variables Cutomare Company/ Cutom	Functions: Mannesical functions Mathematical functions Song functions String functions Mac. functions Aggregate functions Barcole functions	الله
This test can be defined by your applicit	ion.	Inser Z _Z g
Lightweight Tin Ltd.		

Figure 4.7: Formula wizard with variable

4. A text object can hold as many paragraphs as you want and they can all have completely different display properties. You can add more paragraphs by means of the mini toolbar which appears automatically. Choose "Append paragraph" to define an additional line.



Figure 4.8: Adding another paragraph via the mini toolbar

- 5. The formula wizard will now appear, which you can use to define the contents of the paragraph. Now enter the first name and the last name of the recipient. First choose the "Firstname" variable.
- 6. You should insert a space before choosing the "Lastname" variable to prevent the contents of the two variables from being placed directly end to end. A space is simply "Text". Fixed text must be enclosed in quotation marks. So now enter "+" as a joining operator followed by " " for the space.
- 7. Now select the "Lastname" variable. You must of course also join this with "+". The whole line looks like this:

Customer.Firstname *** + Customer.Lastname		101.000
		Σ ₂ 🔊
e		
1 John Iron		
0 🔁	ОК.	Cancel

Figure 4.9: Linking variables and text

8. Now continue with the other variables: street and city. You have now completed the address field.

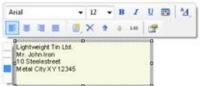


Figure 4.10: Text field with 4 paragraphs



Adding the Date and Page Number

Use a text object once more to add a date and the page number to the letter.

- 1. In the "Objects" toolbar, choose the "Text" object type.
- 2. Pull the object to the required size.
- 3. The formula wizard will now appear, which you can use to define the contents of the text object. Switch to the "Date Format" tab.
- 4. Select the date format that you want. When you do so, the Now() function will be inserted automatically in the Date\$() function. This outputs the current date in the format that you have chosen.
- 5. Then create a further text object in the footer area for the page number. In the result area of the formula wizard, enter the Page\$() function directly to output the page number.



Adding Formatted Text for the Letter

You output the text for the letter with the formatted text object. As opposed to the normal text object, this object also lets you change the formatting of the text within a line.

 In the "Objects" toolbar, choose the "Formatted text" object type.

1 El 🔍 🖪 🔨 🗆 🗢 🗃 🛄 🔳 🛃 🖨 🗑 🥌

Figure 4.11: "Objects" toolbar

- 2. Pull the object to the required size.
- 3. A dialog appears where you can type in the letter in the form of continuous text.

Alternatively, you can select an RTF variable from the "Source" drop-down list if available.

Expert tip: If the current RTF object is appended to another RTF object for which the "Page break" option is enabled, the "Transfer exceeding text of ..." option is available as a data source ...". If you select this option, you cannot enter text in the RTF object because the (remaining) text will be transferred automatically from the other RTF object (e.g. for a two-column print).

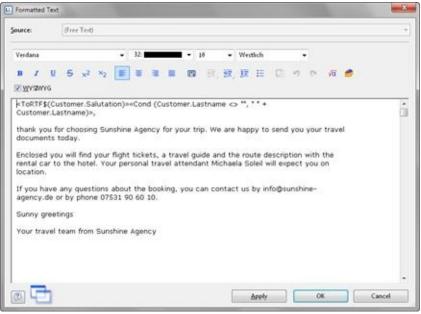


Figure 4.12: Formatted text object dialog

4. If you click the formula button, the formula wizard will appear here as well, which you can use to insert variables. You want to address the recipient personally so you now create the salutation. When doing so, please make sure that you do not enter an unnecessary space character at the end of the salutation formula if the "Name" field in "Dear Sir or Madam" salutations is to be empty. One possibility for a perfect salutation formula is the use of Rtrim\$() e.g.:

«Rtrim\$(Recipient.Salutation + " " +
Recipient.Lastname)»,

5. Now write the letter text. You have various formatting options at your disposal. You create a tab stop with CTRL+TAB.

Or you can create the letter text or pictures and graphics in your normal text processing program (e.g. MS Word), and insert them into this dialog with copy $C_{TRL}+C$ and paste $C_{TRL}+V$.

- 6. The "WYSIWYG" (What You See Is What You Get) button shows you the hyphenation based on the object size. This is just an approximate guide and can differ slightly.
- 7. Page breaks are created automatically if the available

space is no longer sufficient and the "Page break" object property is set to "Yes". If you want to trigger a page break manually, you can do this by means of the context menu. Select the **Pagebreak** control character.

Paragraph		Bold		
Edit	•	Italic Underlined Strikeout Superscript Subscript		
		Color Insert Control character	,	Unbreakable Space
				Optional Hyphen Pagebreak

Figure 4.13: Context menu for a user defined page break



Adjusting the Position of the Letter Text for Following Pages

Because of the address field, the text starts further down on the first page of a letter. So that the text begins at the top margin in multiple page mail merge projects, a condition can be applied to the position (top, height) of the formatted text object with which the object can be moved upwards and increased in height on following pages.

- 1. Select the formatted text object.
- Select the "Position.top" property and set the Cond(Page()=1,125,50) condition.
- 3. Select the "Position.height" property and set the Cond(Page()=1.150.225) condition.

Cond(i dg	C() = 1, 1, 30, 223)
E Position	[0.906, 4.051, 7.461, 10.842 in]
Left	0.906 in
Тор	4.051 in
Width	6.555 in
Height	6.791 in

Figure 4.14: Specifying the position with a formula



Assigning Objects to the Layer

So that the objects for address, company logo and date are only printed on the first page, you can assign these objects to the corresponding layer.

- 1. Hold down the CTRL key and select the objects for the address, company logo and date.
- 2. Right-click to open the context menu and choose "Assign to Layer".
- 3. A selection dialog will now appear in which you can select the "First page" layer. Confirm your selection with OK.
- 4. The objects are now assigned to the "First page" layer. This is indicated by the fact that the objects are shown in the layout preview in green, the color assigned to this layer.

sunshine
Cophranglet The Unit Me. Johns from 1.19 Hondownwert Manuel Cop. on 2020/18
The local Research
Dear Re. Son.
thank you for closing facultie Agency for your kip. We are happy to send you your travel downstants to be a
Enclosed pay off find your flight todays, a strend pack and the most decouption with the most a constraints from these transportances of the decouption of the scotter. If you have an apartment when the besiding, you gap constant with a strengthener.
egania de ar 34 altere (1720) 80.08 10.
Tuniy gladinya
Your based taken, front Subshiele Agency
Fase 1 of BLT TotalFacest

Figure 4.15: Objects are shown in different colors

5. The "formatted text" object remains assigned to the base layer so that it will be printed on all pages.



Creating Professional Tables

In this chapter, we will extend the previous examples to include more functions, output grouped data, use appearance conditions and layers to allow multiple pages to be output and add more elements to the report container.

See also:

- Using Advanced Features With the Report Structure
- Practice: Defining the Report Structure Correctly
- Modifying the Fields and Columns
- Defining Multiple Line Layouts
- Defining Column Contents
- Defining Group Lines
- Table Layouts



Using Advanced Features With the Report Structure

You use the "Report Container" object to add a table. As the name says, a report container can hold several objects. Tables, charts and crosstabs can be added in any order, even as sub-elements of tables. This lets you define sub-reports with almost any relationships between tables.

However, only one report container is permitted and you cannot define any separate tables, crosstabs or charts. You define new elements in the "Report Structure" tool window along with the hierarchical structure that you want. All elements and respective sub-elements of a report structure are shown here, with object type and data source [relation name, sort name], in the order in which they will later be printed (sequence plan).

To add a new element to the report container, select the "Append an element" or the "Append a sub-element" button. Sub-elements are only possible with tables.

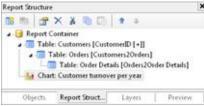


Figure 5.1: "Report Structure" tool window

Only one element is displayed at any time in the workspace. This means that with the "Report Structure" toolbar, you select the element that you want to be displayed in the workspace.



Practice: Defining the Report Structure Correctly

Let us assume that you want to produce a list of all customers, showing the orders of the respective customers and all order items.

You want the result to look roughly like this:

ers Cust ALFP Pos 1 2 ALFP Pos 1	Order ID 10,643 10,643 0 Order ID 10,692	Article nun 28 39 Article nun 63	15 21 11/3/2009	te Product Rössle Sau Chartreuse	verte 11/13/2009	Berlin Weight 29.46 61.02	ALF KI Order ID 10,643 Price \$45.60 \$18.00 \$63.60 10,692 Price \$43.90
ALFP Pos 1 2 ALFP Pos 1	0 0rder ID 10,643 10,643 0 0rder ID 10,692	28 39 Article nur	9/25/2009 Count 15 21 11/3/2009 Count	Product Rössle Sau Chartreuse Product	10/3/2009 ierkraut verte 11/13/2009	29.46	10,643 Price \$45.60 \$18.00 \$63.60 10,692 Price
ALFP Pos 1 2 ALFP Pos 1	0 0rder ID 10,643 10,643 0 0rder ID 10,692	28 39 Article nur	9/25/2009 Count 15 21 11/3/2009 Count	Product Rössle Sau Chartreuse Product	10/3/2009 ierkraut verte 11/13/2009	29.46	10,643 Price \$45.60 \$18.00 \$63.60 10,692 Price
Pos 1 2 ALFP Pos 1	Order ID 10,643 10,643 0 Order ID 10,692	28 39 Article nur	Count 15 21 11/3/2009 Count	Product Rössle Sau Chartreuse Product	verkraut verte 11/13/2009		Price \$45.60 \$18.00 \$63.60 10.692 Price
1 2 ALFP Pos 1	10,643 10,643 0 0 0rder ID 10,692	28 39 Article nur	15 21 11/3/2009 Count	Rössle Sau Chartreuse Product	verte 11/13/2009	61.02	\$45.60 \$18.00 \$63.60 10.692 Price
2 ALFP Pos	10,643 0 Order ID 10,692	39 Article nur	21 11/3/2009 Count	Chartreuse Product	verte 11/13/2009	61.02	\$18.00 \$63.60 10.692 Price
ALFP Pos 1	0 Order ID 10,692	Article nur	11/3/2009 Count	Product	11/13/2009	6102	\$63.60 10,692 Price
Pos 1	Order ID 10,692		Count	Product		61.02	10,692 Price
Pos 1	Order ID 10,692		Count	Product		61.02	Price
1	10,692				ad		the best designed as the best
	2085200	63	20	Vegie-spre	ad		\$43.90
Trujilo	Empareda						
Trujilk	Empareda						\$43.90
Trujilk	Emparedar					90.48 kg	\$107.50
		dos y helados	Avda, de	la Constituci	ón 2222	México D.F.	ANATR
ers			100000000000000000000000000000000000000		in the second second		
Cust	omer ID		Order da	te	Shipping date	Weight	Order ID
ANA'	TR		10/19/200	08	10/25/2008	1.61	10,308
Pos	Order ID	Article nun	Count	Product			Price
1	10,308	69	1	Gudbrands	dalsost		\$28.80
2	10,308	70	5	Outback La	iger		\$12.00
							\$40.80
ANA'	IR		9/8/2009		9/14/2009	43.90	10,625
Pos	Order ID	Article nun	Count	Product			Price
1	10,625	14	3	Tofu			\$23.25
2	10,625	42	5	Singaporea	n Hokkien Fried M	lee	\$14.00
				990 E.M.8888			\$37.25
						45 51 kg	\$78.05
	ANA Pos 1 2 ANA Pos 1	2 10,308 ANATR Pos Order ID 1 10,625 2 10,625	ANATR Article num 1 10,308 69 2 10,308 70 ANATR Pos Order ID Article num 1 10,625 14 2 10,625 42	ANATR 10/19/200 Pos Order ID Article nun Count 1 10,308 69 1 2 10,308 70 5 ANATR 9/8/2009 Pos Order ID Article nun Count 1 10,625 14 3 2 10,625 42 5	ANATR 10/19/2008 Pos Order ID Article nun Count Product 1 10,308 69 1 Gudbrands 2 10,308 70 5 Outback La ANATR B/8/2009 Pos Order ID Article nun Count Product 1 10,625 14 3 Tofu 2 10,625 42 5 Singaporea	ANATR 10/19/2008 10/25/2008 Pos Order ID Article nun Count Pro duct 1 10,308 69 1 Gudbrandsdalsost 2 10,308 70 5 Outback Lager ANATR 9/8/2009 9/14/2009 Pos Order ID Article nun Count Pro duct 1 10,625 14 3 Tofu 2 10,625 42 5 Singaporean Hokkien Fried N	ANATR 10/19/2008 10/25/2008 1.61 Pos Order ID Article nun Count Product 1 10,308 69 1 Gudbrandsdalsost 2 10,308 70 5 Outback Lager ANATR 9/8/2009 9/14/2009 43.90 Pos Order ID Article nun Count Product 1 10,625 14 3 Tofu

Figure 5.2: Hierarchical invoice list

To achieve this, proceed as follows:

1. In the "Objects" toolbar, choose the "Report Container" object type.

i 🐑 🔍 🖪 🔨 🗆 🔿 🖓 🛄 🗐 🗟 💁 🚳 🥌

Figure 5.3: "Objects" toolbar

- 2. In the workspace, hold down the left mouse button and pull the object to the required size.
- 3. A selection dialog will appear for the chosen object type. Choose the "Table" object type.

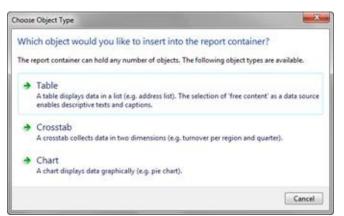


Figure 5.4: Choosing the object type

4. You specify the data source in the following dialog. All available tables are shown hierarchically, in other words, under the tables, you will find the relational tables in each case.



Figure 5.5: Choosing the data source

For the list of invoices, you need the following structure: Customers > Orders > Order Details.

You have 2 alternative procedures at this point:

- a) You select the "Customers" table to first create the "top" table. This corresponds to a top-down procedure; meaning that you then add the "Orders" sub-table followed by the "Order details" sub-table by means of the "Report Structure" tool window.
- b) Or you choose the structure that you want right from the start by selecting the "lowest" table. This

corresponds to a bottom-up procedure, meaning that you create all three tables starting by designing the "lowest" table.

- You will be using the second method in this example. Accordingly, you select the table "Customers > Orders > Order_Details".
- 6. A selection dialog will appear with all the fields in the "Order_Details" table.

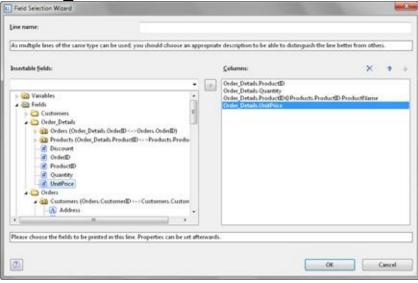


Figure 5.6: Field selection dialog

In this dialog, now choose the columns for this sub-table. For example, double-click the "ProductID", "Quantity", "UnitPrice" and "ProductName" fields from the "Products" table which has a 1:1 relationship. This will add the fields to the "Columns" area. You can change the order with the arrow button.

- 7. The "Order_Details" table will now be displayed in the workspace.
 - § The selected fields are displayed in the data line, in other words, the data line contains the data.
 - § In addition, a header line is automatically produced. Header lines are used mostly as column titles, i.e. the selected field names are now shown here as text.
 - § List & Label adjusts the width of the columns automatically. You can adjust the width of a column manually by moving the separating line to the right or

the left with the mouse.

Note: This changes all table columns whose separators are within +/-2 mm from the mouse position. If you hold down the CTRL key, the action will only be carried out for the line on which the mouse is positioned. The "Column width modification affects next column" option lets you alter the column width while making the next column smaller.

8. To define the columns of the "Orders" table, double-click the table in the "Report Structure" tool window.



Modifying the Fields and Columns

There are two possibilities for adding additional columns to tables or for editing and formatting them in detail.

See also: Mini Toolbar Object Dialog



Mini Toolbar

For the most frequent functions, use the mini toolbar which appears automatically. To select a single field, click the upper left corner of the field.

- § To select multiple fields, hold down the CTRL key and click the upper left corner of the fields.
- § To select a whole area, hold down the Shift key.
- \S To select an entire column, hold down the ALT key.

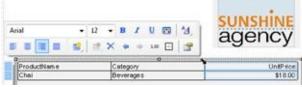


Figure 5.7: Multiple field selection

You have various functions at your disposal including:

- § Font: font, font size, bold, italic, underline, color, open font settings dialog or set font to default (via right-click)
- § Properties: Alignment, edit line definition, append new line, delete selected cells, move selected columns or cells to the right or left, open formatting dialog, open frame properties dialog.



Object Dialog

Use the object dialog for more advanced functionalities. You open this dialog via the corresponding button in the mini toolbar or by double-clicking the element in the report structure.

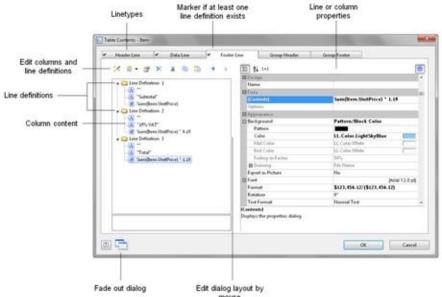


Figure 5.8: Object properties dialog for tables

There is a tab for each type of line where you can specify the different definitions and columns for the respective line. The following types of line are available: header line, data line, footer line, group header, group footer. A checkmark on the tab indicates that a line type has one or more line definitions. § Header lines are mostly used as titles for the columns of

the table.

- § Data lines contain the formatting for the actual table rows and the data that is to be shown in the table.
- § Footer lines are displayed at the very end of the table and can hold final information about the data lines that are

output above.

§ Group header and footer lines are used to structure the data lines by means of "Intermediate headings" and "Intermediate footers".

All line types can be defined independently of one another.

- § This means that the columns of a header line can have a different appearance as the data lines or footers that follow.
- § You can also create different line layouts or line definitions for the individual line types. It is then possible to activate the different line definitions with special appearance conditions as required.



Defining Multiple Line Layouts

You can define different layouts for each type of line. Depending on the appearance conditions, the appropriate layout is used in each case according to the situation. For example, you can output two table lines for each data record in this way:



Figure 5.9: Two line definitions

Proceed as follows to create table lines in the object dialog.

- 1. First select the line type that you want to edit by clicking the relevant tab, e.g. Data Line.
- 2. Now choose "Insert Table Line" in the context menu. Alternatively, you can select an existing Line definition and then click the "New (Append line definition)" button.
- 3. In the "Choose a Table Line Definition" dialog that follows, you have the option of
 - § using an already existing layout as a template for the new line definition (very useful if the layout is similar).
 - § starting the field selection wizard (very useful if you want to create several columns in one operation)
 - § or creating an empty line definition so that you can then add the columns by means of the object dialog.



Figure 5.10: Using an existing line layout

- 4. You have now created the new line. Edit the columns as described in <u>Defining Column Contents</u>. Change the order of the line with the arrow button or with Drag & Drop.
- 5. You can then specify the appearance of the new line as a whole. Various properties are available for this including:
 - § "Appearance Condition" with which you can specify when the line is to be printed. This is useful if you define multiple line layouts that are to be printed depending on certain values. The familiar dialog for defining logical expressions opens up here.

Example line 1: Subtotal not on last page

Appearance condition: not Lastpage()

Example line 2: Grand total only on last page

Appearance condition: Lastpage()

- § Name of the line e.g. "data first line". This makes it easier to find the line in complex layouts.
- § Spacing (margins): here you define the top, bottom, right

and left spacing of the line. The "top" or "bottom" values cause a corresponding space between the individual table rows. With the "left" and "right" spacing values, you can specify the margin in relation to the table object, i.e. you can indent lines or columns.

- § The "Default Font" property sets the font for the entire table row. Newly inserted columns appear initially in this font.
- § Outline Level (index level) of the bookmark in preview mode or for PDF export.

Also see chapters <u>Overview of Properties</u> and <u>Defining</u> <u>Group Lines</u>.

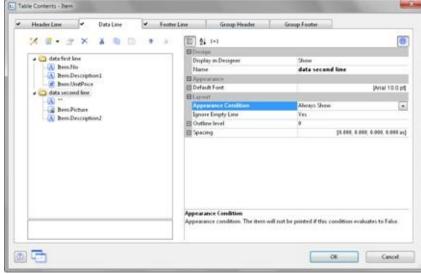


Figure 5.11: Line properties



Defining Column Contents

You can define as many columns as you want for each line. You must only make sure that these columns can be displayed within the width defined for the table. The individual columns are shown in the object dialog as a tree structure. The buttons let you edit, delete, cut, copy, insert and move the selected columns. You can also move columns outside of the line definitions by using Drag & Drop. Proceed as follows to create new columns in the object dialog:

- 1. First select the line in which you want to insert a new column.
- Now choose "Append column" in the context menu. (ALT+INS). Alternatively, you can select an existing Column definition and then click the "New (Append column)" button or the small downwards arrow next to this button to specify the type.
- 3. Each column has a certain type. Various properties are available for the column type including: text, drawing, barcode, RTF text, chart, gauge, HTML text and OLE container. When selecting a field, this data type will be set automatically.
- 4. To define the contents, the familiar formula wizard will appear in which you can define the column contents in the form of expressions. You will find more information about this under <u>Variables</u>, <u>Formulae and Expressions</u>.
- 5. Now define the column's properties. Each column in a line can be edited and formatted separately. Select the column that you want in the tree structure in the object dialog.

Header Line 🖌 Footes	Line Group Header	Group Footes	
data ferd line data ferd line	10 24 C=1 Deresign Name Dista		
data second live	(Contents) Options El Approximitor	Item.No	_
-60 **	E Background	Pattern/Block Color	_
A hem.Picture	Export as Picture	Ne	
-10 newsperchange	(i) Font	para para	pq 0.01 la
	Fermat	None	***
	Rotation	9.	
	Test Format	Normal Test	
	DAction		
	Link URL		
	E Lapout.		
	(I) Alignment	Left	
	Appearance Condition	Always Show	
	Blank Optimization	Ne	
Sample variable 'Bern No'. This text can be defined by your application.	Format Output format.		

Figure 5.12: Column properties

To select multiple columns, hold down the ALT or the SHIFT key. Various properties are available including:

§ Formatting e.g. as number or currency.

- § Name of the column: this will help you to maintain an overview with complex expressions. If you change the name directly in the tree structure, your change will also be applied as "content" where appropriate.
- § You can specify when this column is to be printed with an "appearance condition". This is useful if you define multiple columns that are to be printed depending on certain values. The familiar dialog for defining logical expressions opens up here.
- § Rotation of the content in increments of 90°.
- § Background, frame, font, vertical and horizontal alignment.
- § A fixed height for the field irrespective of the content.
- § The column width.

Also see chapter **Overview of Properties**.



Defining Group Lines

Group lines are a special type of line. They are used to group together the data lines that are to be printed.

You can use the "Group by" line property to specify how the data is to be grouped. This means that the line is printed whenever the result of the expression changes from one data line to the next. If you don't enter an expression, the line will not be printed and the property is highlighted in red in the property window.

A group header is printed accordingly before the data line is output, e.g. "Item group XYZ" group heading.

A group footer appears after the condition of the "Group By" property has changed, in other words, after outputting the data line. Group footers are suitable e.g. for totals of data within a group.

Example: Grouping by the first letter of the "Item.No" field.

1. Create a new line definition on the "Group header" tab. Enter the following expression for the content of the column:

"Main group: ' + Left\$ (Item.No,1)"

The result of the expression "Left\$ (Item.No,1)" is the first character of the "Item.No" variable.

Whenever the first letter of "Item.No" changes, the text "Main group: " and the first letter in each case will be printed.

 Enter "Left\$ (Item.No,1)" as the condition for the "Group By" property.

With each new first letter, a corresponding intermediate header will be printed in the list.

3. Multiple line layouts are possible with group headers as well. In this way, you can produce hierarchically structured intermediate headers. For example, you can define a line layout that is produced, as in the above example, based on the first letter of the "Item.No" variable. In addition, you define a second line layout that produces intermediate headers based on the first three characters of "Item.No". Create a new line grouping for the sub-group header in the same way. Enter "Left\$ (Item.No,3)" as the condition for the "Group By" property.

Enter the "sub-group" for the content of the column: ' + Left\$ (Artikel.Nr,3)":

When the first 3 letters change, a corresponding intermediate header will be printed in the list.

article no	barcode	description	price
main group 'E'			
sub-group 'E)	P		
EXPSA01		Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight	\$1,500.00
EXPCH01		Northern & Southern Chile: 23 day tour from Santiago to Punta Arenas including flight	\$3,500.00
EXPMAL01		Maldives diving trip: 14 days, southern Male Atoll, Paradise Beach **** excluding flight	\$1,800.00
EXPHK01		Hong Kong and Ball: 2 weeks, including flights, accommodation, excursions	\$1,760.00
EXPYUC01		Yucatan, On the Trail of the Maya, 2-week round trip, excluding flight	\$1,200.00
		5 articles in EXP"=	\$9,760.00
sub-group 'E)	(C'		
EXCPAR01		Paris, visit to the Louvre including guided tour and admission	\$40.00
EXCPAR02		Paris, admission to the latest cabaret show at the Moulin Rouge including three-course meal	\$178.00
		2 articles in "EXC" =	\$218.00
sub-group 'E)	°P		
EXPLON01		London, sightseeing tour with boat trip on the Thames	\$50.00
		1 articles in 'EXP' =	\$60.00
		8 articles in 'E' =	\$10,038.00
main group 'R'			
sub-group 'R*	π		
RNTCOT01		Cottage, South of England with sea view, 4 persons	\$1,050.00
RNTMTB01		Vespa GT 125/200 I motor scooter for Rome, Paris	\$150.00

Figure 5.13: Group lines in an article list

Along with the properties of the "normal" lines, you also have the following at your disposal:

- § Group sums: You can set sum variables to 0 here once they have been output in order to produce group sums.
- § Page break before outputting a group header or break after outputting a group footer.
- § The option of always displaying the group header additionally at the start of the table if the group has been separated by a page break.

Also see chapter <u>Overview of Properties</u>.

🖌 HeaderLine 🖌 DataLine 🖌 Foo	ter Line 🖌 Group Header 🖌	Group Footer
Imain group header Imain group ***** +Left\$\$\$tem.No.1)**** Imain group ******Left\$\$\$tem.No.3)****	Design Deplay in Designer Hame Payout much Payout much Default Font Default Fon	Show maingroupheader priat 10.0 st Admays Show No Left\$(Item.No, 1) Yes 1 1 150 [4.001, 9.033, 8.001, 6.000 m]
	Design	

Figure 5.14: Group Headers



Table Layouts

There are various properties and functions which you can use to influence the layout of a table. Also see chapters <u>Creating</u> <u>Statistical Reports With Footers</u>, <u>Drilldown Reports (Increase</u> <u>Detail Level</u>), <u>Different Layout Techniques</u> and <u>Overview of</u> <u>Properties</u>.

See also:

- Align Columns
- Fixed Size
- Printing Header Lines and Footer Lines Again
- Defining the Size of the Table Variably
- Forcing a New Page
- Creating a Conditional Page Break
- Keeping Lines Together
- Defining Totals and Counters
- Outputting Free Content Before and After a Table
- Defining Columns Across Multiple Lines



Align Columns

There are various ways of simplifying the use of table objects.

- § If you hold down the CTRL key when reducing the size of a table, all columns will be automatically reduced in size by the same factor.
- § To align (sub) tables exactly with one another, additional tick marks can be shown on the ruler by means of an element property.



Figure 5.15: Tooltip for a column separator

- § List & Label adjusts the width of the columns automatically. You can adjust the width of a column manually by moving the separating line to the right or the left with the mouse. This changes all table columns, whose separators are within +/-2 mm from the mouse position. If you hold down the CTRL key, the action will only be carried out for the line on which the mouse is positioned. If the "Column width modification affects next column" option is enabled (Project > Options > Workspace), you can alter the column width while making the next column smaller.
- § You can hide Line Types (header, data, footer, group lines) in the workspace. To do this, select the table object and use Visible Line Types in the context menu or the corresponding menu item View> Visible Line Types.



Fixed Size

The "Fixed Size" property lets you specify that the size of the table is not to be adjusted automatically when fewer data lines are printed than the available space in the table object. This property is useful to ensure that footers are always printed at the bottom of the page, e.g. if the page number is output in the footer. If the property is disabled, the end of the table automatically moves upwards (and the footer therefore also).



Printing Header Lines and Footer Lines Again

If the print of a table is continued on the following page, the header lines of this table and the outer table will be printed again. To suppress repeated printing of the header lines on the following page, use the FirstHeaderThisTable() function as an appearance condition.

This functionality is also available for footer lines, here you use the LastFooterThisTable() function as the appearance condition. This ensures that footer lines are only output on the last page of the table in the event that the print is continued on the following page due to lack of space. You will find more information about this in the chapter <u>Overview of Functions</u>.



Defining the Size of the Table Variably

You can define the height and width of the report containers variably to avoid data being truncated when the page format is changed (e.g. from portrait to landscape).

To do this, select the report container in the "Report Structure" tool window and use the LL.Device.Page variables and the UnitFromSCM() function to specify the height and width.

Position	[0.39, 0.39, 7.8	6, 11.29 in]	
Left	UnitFromSCM(10000)	[0.39 in]	
Тор	UnitFromSCM(10000)	[0.39 in]	
Width	LL.Device.Page.Size.cx - UnitFromSCM(20000)	[7.47 in]	1
Height	LL.Device.Page.Size.cy - UnitFromSCM(20000)	a	-

Figure 5.16: Size of the report container defined variably



Forcing a New Page

With complex projects containing hierarchical tables, it is sometimes wise to create a page break before outputting a line of the "top" table.

Alternatively, you may want to have a new page if, after outputting a data line of the "top" table, there is not enough room for the following data lines of the sub-table.

You can handle both cases with the "Pagebreak Condition" property.

For example, in the case of a hierarchical table, in order to output each data line of the main table on a new page, select the main table in the "Report Structure" tool window and set the "Pagebreak Condition" property to True.

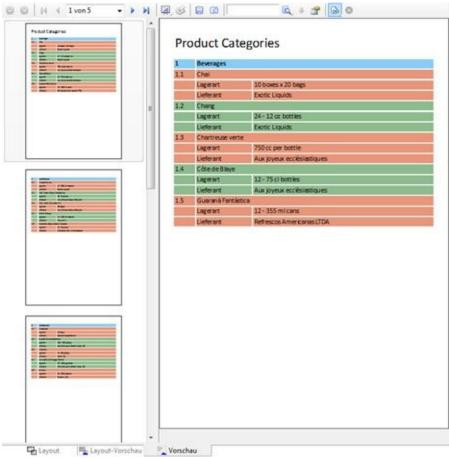


Figure 5.17: Active page break condition



Creating a Conditional Page Break

Let s assume that you want to create a continuous table. However, you want to avoid printing data lines from the main table at the bottom of the table unless there is enough room for at least 3 data lines from the sub-table. You do not want it to look like this:

1	Beverages				
1.1	Chut				
	Type of bearing	10 boses x 20 bags			
	Supplier	Exotic Lipsidi			
12	Chung				
	Type of bearing	24 + 12 re bottles			
	Supplier	Exotic Liquidi			
1.3	Chartreuse verte				
	Type of bearing	750 cc per bottle			
	Supplier	Aux joyeux ecclestestiques			
14	Côte de Bloye				
	Type of bearing	12+75 d bottles			
	Supplier	Aux joyeux ecclesiantiques			
15:	Guarană Fantâstica				
	Type of bearing	12 + 355 ml cars			
	Supplier	References Americanas LTDA			
16	(poli Ceffee				
	Type of bearing	16 - 500 g ties			
	Supplier	Leka Irading			
1.7	Lakkalikoon				
	Type of beating.	500 ml			
	Supplier	Karkki Oy			
1.8	Laughing LumberJack Lagre				
	Type of bearing	24 - 12 nr bottles			
	Sopplier	Bigfoot Broweries			
1.9	Outback Liger				
	Type of bearing	24 - 355 oil bottles			
	Supplier	Pavlova, Uni.			
1.10	Rhonbring Klostert	bire			
	Type of bearing	24+0.51 bomles			
	Supplier	Plutzer Lebersmittelgroßmärkte AG			
2	Condiments		_		

Figure 5.18: Data lines from the main table without following line

You can avoid this by means of the RemainingTableSpace() function. This function returns a value showing the available space. If you set the parameter to "True", the value will be returned as 1/1000 mm.

If you want to specify that a new page should be started before outputting the data line of the main table if the space remaining is less than 3 cm, enter the following formula for the "Pagebreak Condition" property: RemainingTableSpace(True)<30000



Keeping Lines Together

If the print of a table is continued on the following page due to lack of space, you can decide whether the lines are to be separated or kept together if possible.

This option is useful for multi-line data lines or for invoice footer lines containing totals. You can use this option with data lines, footer lines, group footers and group headers



Defining Totals and Counters

You define totals and counters with sum variables or the corresponding aggregate functions. You will find more information about this in chapters <u>Using Functions</u>, <u>Overview</u> <u>of Functions</u> and <u>Working With Sum Variables</u>.



Outputting Free Content Before and After a Table

You can output free text before and after a table. To do this, use the RTF object and link the object to the project as free text via the report container.

Produce a first page with a covering letter for the invoice that we created in Chapter <u>Producing a Simple Invoice</u> and an enclosure with the General Terms and Conditions.

- 1. To add a new element to the report container, select the "Append an element" or the "Append a sub-element" button in the "Report Structure" tool window.
- 2. A selection dialog will appear for the chosen object type. Choose the "Table" object type.
- 3. In the following dialog, select "Free content" as the data source.



Figure 5.19: Add free content to the report container

4. Then add a column via the properties dialog for the table object. In our case, we want to create the covering letter

as formatted text. Therefore, click the small arrow on the right of the button and choose the Formatted Text option.

Data Line		
🕺 🖉 🖓 🗶 🔺 🕲 🛄 🔹 👂	10 24 C-2	
(A) Tet	E Design	
0	Display in Designer	Show
Orawing	Name	
I Barcode	Appearance	
Formatted Text	E Default Font	(Arial 10.0 p
Gauge	E Layout	
HTML Text	Appearance Condition	Always Show
OLE Container	Ignore Empty Line	Yes
S OLE Container	Outline level	0
	Spacing	[0.00, 0.00, 0.00, 0.00 in
	Design	
	4. 13	

Figure 5.20: Specifying formatted text for the content of the column

- 5. A dialog appears where you can type in the covering letter in the form of continuous text. You will find detailed instructions for working with formatted text in Chapter <u>Practice: Create a</u>.
- 6. Please don't forget to remove the (column) frame for free content. The best way is to remove the frame via the "Default Frame" property in the element properties.
- 7. You may also have to change the layer condition as the covering letter will now be output on the first page.
- 8. If the General Terms and Conditions are also to be output at the end of the invoice, you must add another free content object to the container after the item table. Use the Formatted Text object here as well or include the content via a linked file.

Use the LoadFile\$() function to load a linked file and enter the path of the file as the parameter as follows LoadFile\$ (ProjectPath\$()+"\gtc.txt").

X X X	10 24 6-2		0
	BDenigs		
a Calline Definition 1	Planer		
(A) LoadfideS (ProjectPathSt) + " gtc.te")	BData	1	-
	(Contract)	LoadFile\$ (ProjectPath\$0 = "\gtc.txt")	-
	Optunie	A	
	Bloomenia		m (
	B Back ground	Transparent	-
	Export as Picture	Pás	
	11 Fort	Priol 12.0	90
	Format	Norse	
	Rotation	#-	
	Text Format	Filorenal Test	
	El Activation		85
	Levis URL		
	Bland		15
	E Abgranant	Left	
	Appearance Condition	Adways Show	-
	K ontentsä		
	Displays the properties dialog	6	

Figure 5.21: Including content from a file with LoadFile\$()



Defining Columns Across Multiple Lines

Let us assume that you want to output the contents of a column across two lines. You can achieve this by specifying a negative value for the "Frame.Top.Distance" property, e.g. -0.2 inch. The content will then be moved upwards by this distance. Please note, however, that this will cause content to be overlapped and that the export function cannot always apply the layout 1:1.

article no	description	price
EXPSA01	Southern Africa Explorer: 20-day tour from Cape Town to Victoria Falls excluding flight	\$1,500.00
	Safari: Travel through the Okavango Delta in dug-out canoes, climb sand dunes in Namibia, visit Etosha National Park, Victoria Falls.	
EXPCH01	Northern & Southern Chile: 23-day tour from Santiago to Punta Arenas including flight	\$3,500.00
	Travel from the Atacama desert in Northern Chile, the metropolis of Santiago, across the lakes and volcances region to the Strat of Magellan and the huge Torres del Paine National Park. Patagonia: Carretera Austral (Southern Highway) with spectacular natural scenery, visit the second largest lake in South America, marble caves.	

Figure 5.22: Column contents across two lines



Producing Analyses

In List & Label, you can make use of charts, gauges, crosstabs, footer lines in tables or drilldown reports to analyze your data.

See also:

- Creating Charts
- Creating Gauges
- Creating a Crosstab
- Creating Statistical Reports With Footers
- Drilldown Reports (Increase Detail Level)
- Multi-Column Reports



Creating Charts

This object is used to evaluate and display data graphically in diagrams. This gives you an overview of your data and lets you recognize anomalies immediately.

For example, you can analyze sales trends, illustrate percentage shares and show data in multiple dimensions. You have a wide range of different types of diagrams at your disposal:

§ Pie chart

- § Bar chart (also displayed as cylinders, pyramids, cones, octahedrons)
 - § Simple (e.g. sales per customer)
 - § Multi-row (e.g. sales to various customers over the years, scaled by customer)
 - § Clustered (e.g. sales to various customers over the years, grouped by year)
 - § Stacked (e.g. percentage of sales to various customers stacked over the years)
 - § 100% stacked (e.g. respective sales percentages for various customers over the years)

§ Lines/symbols

- § Simple
- § Multi-row
- § Stacked
- § 100% stacked

See also:

- Inserting a Chart Object
- Creating a Pie Chart
- Creating a Multi-Row Bar Chart
- 100% Stacked Bar Chart
- Multi-Row Line Chart
- Using Series to Determine the Values

• Special Fields With Diagrams



Inserting a Chart Object

There are various ways of outputting chart objects:

- A chart as an element in the report container. Add the object via the "Report Structure" tool window. If you have not yet added a report container to the workspace, select "Report Container" in the "Objects" toolbar and pull the object to the right size in the workspace while holding down the left mouse button. A selection dialog will appear for the chosen object type. Choose the "Chart" object type.
- You can output charts and gauges in a table cell. To do this, select the relevant entry by means of the context menu in the object dialog for the table. If you want to output the aggregated data, a good way of doing this is to use a footer line.

Gan	ing ode sutted Test	ne OCategorius CategoryID	Autors Autors	Show Prist 12.0 (Always Show Pio Pio Yes 0 10 400, 0.000, 0.000 (
4			Design	

Figure 6.1: Chart object in a footer line

3. In the following dialog, now select the data source. All available tables are shown hierarchically, in other words, under the tables you will find the relational tables in each

case.

To evaluate sales per country, for example, choose the "Customers > Orders > Order Details" table in the List & Label Sample Application so that you have all three tables at your disposal. The "Customers" table contains the country, the "Orders" table the order date and the "Order_Details" table the sales.

4. The chart object dialog is displayed. In the drop down lists in the top left you can select the base type and the corresponding sub type; three types with up to nine sub types are available:

§ Pie: Pie, Circle.

§ Bar/Ribbon: Simple (also 3D), Multi-Row, Clustered (also 3D), Stacked (also 3D), 100% Stacked (also 3D).

§ Line/Symbol: Simple, Multi-Row, Stacked, 100% Stacked. The properties are defined in the tabs. You can click directly into the live preview (e.g. onto the title or axis label) to quickly jump to the corresponding property.

Pie Pie Pie Pie Pie Pie	Total sales percentage of customer
Hint: Right click onto the different regions (headline, axis label etc.) in the preview to quickly switch to the corresponding property.	15, 105 - 55 OHN- Wat-
Data Source Pie Segment Diagram Object Colors	
🗄 Data	
Coordinate Value	Customers.CustomerID
Minimum Share	5%
Sort Coordinates	Yes
Sort Coordinates Number of Sample Records	5
Sort Coordinates Number of Sample Records Filter	
Sort Coordinates Number of Sample Records Filter El Axis Labet	S No Filter (All Data)
Sort Coordinates Number of Sample Records Filter El Axis Label Coordinate Label	S No Filter (All Data) Coordinate Value
Sort Coordinates Number of Sample Records Filter Axis Label Coordinate Label E Font	S No Filter (All Data) Coordinate Value (Arial 10.0 pt)
Sort Coordinates Number of Sample Records Filter Coordinate Label Coordinate Label Di Coordinate Label Fixed Font Fixed Font Size	S No Filter (All Data) Coordinate Value [Arial 10.0 pt] Yes
Sort Coordinates Number of Sample Records Filter Acki Label Coordinate Label Sort Font Fixed Font Sae Format	S No Filter (All Data) Coordinate Value (Arial 10.0 pt) Yes None
Sort Coordinates Number of Sample Records Filter Axik Label Filte Fixed Font Size Format Coordinate Label 'Others'	S No Filter (All Data) Coordinate Value [Arial 10.0 pt] Yes None 'Others'
Sort Coordinates Number of Sample Records Filter Axis Label Coordinate Label Fore Format Coordinate Label 'Others' Fornat	S No Filter (All Data) Coordinate Value (Arial 10.0 pt) Yes None
Sort Coordinates Number of Sample Records Filter Axik Label Filte Fixed Font Size Format Coordinate Label 'Others'	S No Filter (All Data) Coordinate Value (Arial 10.0 pt) Yes None 'Others' [Arial 10.0 pt]

Figure 6.2: Chart object dialog



Creating a Pie Chart

Let's assume that you want to evaluate the sales per country. The pie chart is the right choice for this. It lets you read off the percentages immediately. Proceed as follows in the List & Label Sample Application:

- 1. As the data source, select the "Customers > Orders > Order_Details" table.
- 2. For the diagram type, choose Pie > Pie.
- 3. You should first specify the coordinate values for the data source, i.e. the values that define the individual pie segments, e.g. Customers.Country.

Customers. Country	Va
0%	-
Yes	
10	
No Filter (All Data)	
LL.ChartObject.AvasCoordinate	
"Other"	
At Chart	
	0% Ves 10 No Filter (All Data) LL ChartObject AvisCoordinate "Other"

Figure 6.3: Definition of the data source in the pie chart object

4. Switch to the "Pie Segment" tab to specify the coordinate values for size of the pie segment, i.e. the sales. Doubleclick the "Coordinate Value" property.

Now select the aggregate function that you want for the contents in the "Coordinate Value" dialog that follows. You want to create a sales evaluation so choose the "Sum" function.

Coordinate Value	
Content	
Order_Details.UnitPrice * Order_	etals.Quantity
Aggregate function for the conte	đi:
Ino aggregate function -	
Anthmetic Mean (Average)	
Geometric Mean Median	
Mode	
Highest Value (Maximum)	
Lowest Value (Minimum)	
Variance Standard Deviation	
Count	
Number of Distinct Values	
Result:	
G Sum{Order_Details.Uni	Price * Order_Details.Quantity)
(2)	OK Cancel

Figure 6.4: Wizard for creating the coordinate value formula

- 5. In the upper part of the dialog, you can specify the contents by clicking the formula button to start the formula wizard. In the Sample Application, the sales per order value is not supplied directly as a field so you must calculate it using the "Order_Details.Quantity * Order_Details.UnitPrice" formula.
- 6. The "Label on Object" property is already set to "Yes" so that a label with the percentage value is shown on the pie segments. Define the value as "percent" without decimal places by means of the "Format" property.

🗄 Data	
Coordinate Value	Sum(Order_Details.ProductID@Products.ProductID:U
Appearance	
Explosion Offset	Cond(LL.ChartObject.ArcIndex = 1, 10, 20) [20
Label on Object	Yes
Font	(Arial 12.0
Format	123,456.12%/-123,456.12%
Content	LL.ChartObject.ArcPerc
@ Filling	Transparent
Width	60%

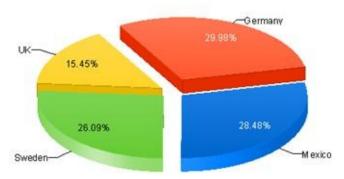
Figure 6.5: Definition of the pie segment in the pie chart object

7. The "Explosion Offset" property lets you specify a distance to the center for the pie segment. With the "ArcIndex" chart field, which numbers the pie segments according to their size, you can even display the largest pie segment with a greater offset. Example: Cond (LL.ChartObject.ArcIndex=1,50,10)

- On the "Diagram" tab, select the general diagram options. Various properties are available including:
 § The degree of perspective, e.g. strong.
 § The color mode, e.g. single color Also see chapter <u>Overview of Properties</u>.
- 9. On the "Object" tab, select the general layout options for the entire chart object. Various properties are available for this including:
 - § Title, e.g. "Customer turnover per country"
 - § Position of the title
 - § Background including filling, border and shadow, e.g. border = transparent

Also see chapter **Overview of Properties**.

- 10. On the "Colors" tab, you can specify the colors for the display:
 - § Assigned Colors: You can assign fixed colors to particular axis values. If you click the "New" button, you can create a new assignment e.g. Customers.Country = "Germany".
 - § Unassigned Colors: Specifies the colors and color sequences for the data rows that are not specified by the "Assigned Colors".
- 11. The pie chart now looks like this:



Customer turnover per country

Figure 6.6: Pie chart



Creating a Multi-Row Bar Chart

Let's assume that you want to evaluate the sales for various countries over the years, scaled by country. A multi row bar chart is perfect for this. You get a diagram in which you can see the turnover achieved in the respective country for each quarter. Proceed as follows in the List & Label Sample Application:

- 1. As the data source, select the "Customers > Orders > Order_Details" table.
- 2. Choose Bar/Ribbon > Multi-Row (3D) as the diagram type.
- 3. First specify the coordinate values for the category axis, i.e. the values of the x-axis. In the Sample Application, the order year is not supplied directly as a field so you must calculate it using the "Year\$(Orders.OrderDate" formula. If you want to evaluate the data by quarter and year, simply enter "Orders.OrderDate" as the coordinate value and then choose an appropriate date format by means of the "Coordinate Label > Format" property, e.g. "Q%q %y". Then change the "Sort Coordinates" property to "No" so that the values are not sorted by quarter.
- 4. Type "Year" as the text for the "Axis Label".

erDate)
oordinate

Figure 6.7: Definition of the category axis

5. Now specify the coordinate values for the series axis, i.e. the values of the y-axis. Select the "Customers.Country" field via the formula wizard.

Use formula as data source		•
1 2 C+3		0
BData		and a second
Coordinate Value	Customers.Country	
Sort Coordinates	Yes	
Number of Sample Records	10	
Filter	No Filter (All Data)	
Round Start and End Values	Yes	
E Axis Label		
Ann Label	**	a
E Coordinate Label	LL.ChartObject.AvasCoordinate	
Legend	Right	
Axis Label		
Asis label text.		

Figure 6.8: Definition of the series axis

6. Now specify the coordinate values for the value axis (z-axis), i.e. the height of the bars representing the turnover. Double-click the "Coordinate Value" property. Now select the aggregate function that you want for the contents in the "Coordinate Value" dialog that follows. You want to create a sales evaluation so choose the "Sum" function.



Figure 6.9: Wizard for creating the coordinate value formula

7. In the upper part of the dialog, you can specify the contents by clicking the formula button to invoke the formula wizard. In the Sample Application, the sales per

order value is not supplied directly as a field so you must calculate it using the "Order_Details.Quantity * Order_Details.UnitPrice" formula.

Category Axis (x) Series Axis (y) Value Axis (z) Diagram Object Colors

] 축↓ c+3	
Data	
Coordinate Value	Sum{Order_Details.UnitPrice * Order_Details.Quanti
Axis Scale	None (Linear)
Maximum Value Automatic	Yes
Appearance	
Coordinate Lines	No
Label on Object	Above
Presentation	Cylinder
Width	60%
Zebra Mode	Pattern/Block Color
Axis Label	
Axis Label	"Turnover"
la	

Figure 6.10: Definition of the value axis

- 8. Various other properties are available on this tab including the following layout options:
 - § Maximum Value Automatic: You can limit the height of the displayed area, e.g. to cater for "anomalies".
 - § Presentation: The data can be presented in various ways: cylinders, bars, pyramids, ribbons, octahedrons, cones
 § Thickness of the bars
 - § Thickness of the bars
 - § Zebra mode for the background
 - Also see chapter **Overview of Properties**.
- 9. On the "Diagram" tab, select the general diagram options. Various properties are available including:
 - § The degree of perspective, e.g. strong.
 - § Color Mode: Specifies which axis determines the color, e.g. the y-axis values.

Also see chapter **Overview of Properties**.

BData		
E Secondary Avis	No	
E Appearance		
Alignment	bottom to top	
E Background Color	Transparent	
Color Mode	Y Asis Values	
Bluminated	Yes	22
Isotropic	No	
Perspective	Strong	
X Axis Rotation Angle	30	
Z Axis Rotation Angle	30	
Z Axis Kotation Angle	30	
olor Mode		

Figure 6.11: Definition of the diagram options

- 10. On the "Object" tab, select the general layout options for the entire diagram. Various properties are available for this including:
 - § Title, e.g. "Customer turnover per country"
 - § Background including filling, border and shadow, e.g. border = transparent
 - Also see chapter **Overview of Properties**.
- 11. On the "Colors" tab, you can specify the colors for the display:
 - § Assigned Colors: You can assign fixed colors to particular axis values as a condition, e.g. Customers.Country = "Germany".
 - § Unassigned Colors: Specifies the colors and color sequences for the data rows that are not specified by the "Assigned Colors".
- 12. The multi-row bar chart now looks like this:



Customer turnover per country

2008

Year

Figure 6.12: Multi row bar chart



100% Stacked Bar Chart

The pie chart in the first example gave you an overview of the percentages for the entire evaluation period. But in order to be able to recognize trends, it would be good to see how the percentages have changed during the course of the evaluation period. The 100% stacked bar chart can be used for precisely these types of applications. The respective percentage of the length of the bars relates directly to the turnover percentage of the respective country. Proceed as follows in the List & Label Sample Application:

- 1. As the data source, select the "Customers > Orders > Order Details" table.
- 2. Choose Bar/Ribbon > 100% stacked as the diagram type.
- 3. First specify the coordinate values for the category axis, i.e. the values of the x-axis. In the Sample Application, the order year is not supplied directly as a field so you must calculate it using the "Year\$(Orders.OrderDate" formula.
- 4. Now specify the coordinate values for the series axis, i.e. the values of the y-axis. Select the "Customers.Country" field via the formula wizard.
- Specify the coordinate values for the value axis (z-axis), i.e. calculate the turnover with the "Order_Details.Quantity * Order_Details.UnitPrice" formula.
- 6. On the "Diagram" tab, choose "Left to Right" for the "Alignment" to create a horizontal diagram.
- 7. The multi-row bar chart now looks like this:

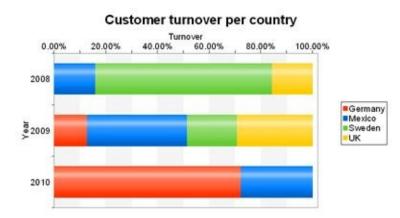


Figure 6.13: 100% Stacked bar chart



Multi-Row Line Chart

A line diagram offers an alternative to a multi-row bar chart. You can read off the values faster here.

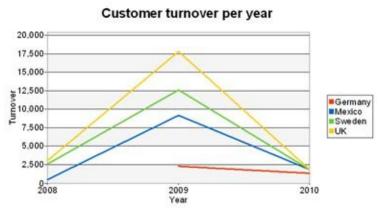


Figure 6.14: Multi-row line chart

Proceed as follows in the List & Label Sample Application:

- 1. As the data source, select the "Customers > Orders > Order_Details" table.
- 2. Choose Line/Symbol > Multi-Row as the diagram type.
- 3. First specify the coordinate values for the category axis, i.e. the values of the x-axis. In the Sample Application, the order year is not supplied directly as a field so you must calculate it using the "Year\$(Orders.OrderDate" formula.
- Now specify the coordinate values for the series axis, i.e. the values of the y-axis. Select the "Customers.Country" field via the formula wizard.
- Specify the coordinate values for the value axis (z-axis),
 i.e. calculate the turnover with the "Order_Details.Quantity * Order_Details.UnitPrice" formula.



Using Series to Determine the Values

With three-axis diagrams, you can also determine the values of the series axis (y-axis) by means of rows. This means that you define the different rows (e.g. measured value/target value/actual value) with a single data record and can show them parallel e.g. in a bar chart.

As an example, we will create a diagram which shows the currency percentages of the 3 economic areas. Data for APAC, EMEA and NAFTA is supplied as rows.

Currency shares of the economic areas per year in percent

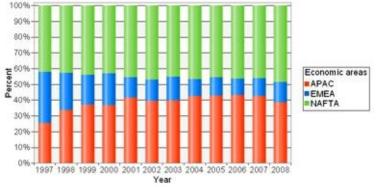


Figure 6.15: Example of a diagram using rows

Proceed as follows in the List & Label Sample Application:

- 1. Select the "Sales" table as the data source.
- 2. Choose Bar/Ribbon > 100% stacked as the diagram type.
- First specify the coordinate values for the category axis, i.e. the values of the x-axis. Select the "Sales.Year" field with the formula wizard. Remove the 2 decimal places using the "Str\$(Sales.Year,0,0)" formula.
- 4. Now specify the coordinate values for the series axis, i.e. the values of the y-axis. Select the "Use rows as data source" entry from the drop-down list above the properties.

Use rows as data source		-
記 会+ C+3		0
BData		1.00
Round Start and End Values	Yes	
Row Definitions	APAC;EMEA;NAFTA	a
Axis Label		
E Axis Label	"Economic areas"	
E Coordinate Label	LL.ChartObject.AvisCoordinate	
Legend	Right	
Row Definitions		
Sets the properties for the different data rows.		

Figure 6.16: Option for using rows to determine the value

This option changes the properties of the series axis and displays a dialog for defining the rows when you click the "Row Definitions" property. Create the individual rows choosing "Sales.APAC", "Sales.EMEA" or "Sales.NAFTA" in each case as the coordinate value.

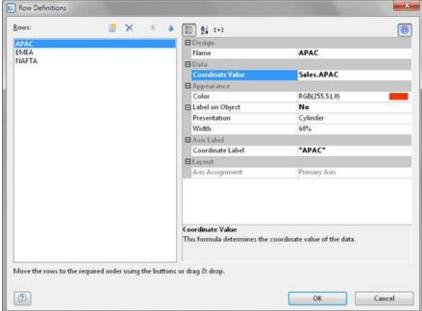


Figure 6.17: Row definitions dialog



Special Fields With Diagrams

Various fields are available for diagrams including:

§ LL.ChartObject.ArcIndex: returns the index of the current pie segment. The largest pie segment has index 1, the second largest has index 2 and so on.

This means, for example, that you can assign a greater explosion offset to the largest pie segment:

```
Cond(LL.ChartObject.ArcIndex=1,30,10)
```

- § LL.ChartObject.ArcPerc: returns the percentage share of the current pie segment. This field is especially useful for labeling the pie segment.
- § LL.ChartObject.ArcTotal: returns the absolute value of the total data volume with pie charts.
- § LL.ChartObject.ArcTypeIsOthers: returns True, if the current pie segment is the "other" segment.

This means, for example, that you can assign a greater explosion offset to the "other" pie segment: Cond(LL.ChartObject.ArcTypeIsOthers,30,10)

- § LL.ChartObject.ArcValue: returns the absolute value of the current pie segment. This field is especially useful for labeling the pie segment.
- § LL.ChartObject.AxisCoordinate: returns the coordinate value and can be used for labeling the axes.

You will find more information about this in the chapter <u>Overview of Functions</u>.



Creating Gauges

Gauges are a good way of presenting actual values. Let's assume that you want to output the current turnover in relation to the target value. A gauge is the right choice for this. It provides you with the value at a glance.

See also:

- Inserting a Gauge
- Specify Properties



Inserting a Gauge

There are various ways of outputting gauges:

- 1. A gauge as an object. Select the "Gauge" in the "Objects" toolbar and pull the object to the right size in the workspace while holding down the left mouse button.
- 2. You can output gauges in a table cell. To do this, select the "Gauge" entry by means of the context menu in the tables object dialog. If you want to output the aggregated data, a good way of doing this is to use a footer line.

✓ Header Line ✓ Data Line	Footer Line	Group Header 0	roup Footer	_
2 8 3 3 X X 8 0	14.14	10 1 C+3		8
Test Drawing Barcode Formatted Test Gauge HTML Test CLE Container		Despiny in Designer Display in Designer Name Appearance Default Font Default Def	Show Anal 12 Always Show No Rem.No Yes 0 [0.00, 0.00, 0.00, 0.0	Pa 0
		Design		
a) 🕞			OK Cance	

Figure 6.18: Gauge object in a footer line



Specify Properties

- 1. Choose the type, form, shadow and pointer properties for the gauge using the drop-down lists.
- 2. The "Value" property lets you determine the value that the pointer is to display, e.g. the customer's turnover or rating.
- 3. You can also make use of various other layout options including:
 - § Appearance: Filling, pointer options, glass properties.
 - § Lettering: Rotation angle, white space before and after the scale range, tickmarks, scale labels, signal ranges, text fields

§ Values: Minimum and maximum vales of the scale Also see the chapter <u>Overview of Properties</u>.

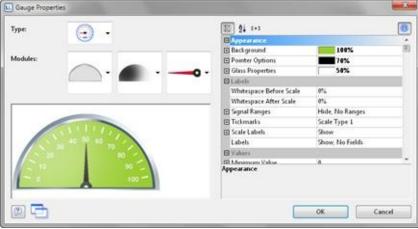


Figure 6.19: Gauge properties

4. Indicate the optimal area, e.g., by specifying the color of the signal range.



Figure 6.20: Gauge with colored signal ranges

To do this, set the "Signal Ranges" property to "Show" and click the "..." button to open the dialog for defining the regions. In this dialog, click the "New" button to create the respective ranges with start and end values and assign the colors that you want.

Signal Ranges	-		and the second	14
Regions:	📓 🗙	1 24 t+1		0
0 30	1	Appearance		
30 - 70		Start Color	LL.Color.Green	
70 - 100		End Color	LL.Color.Green	
		ELapout		
		Start Value	0	
		End Value	30	
		End Value End value for the signal range.		

Figure 6.21: Signal range definition dialog

5. You can also define text areas in the same way, i.e. you can output text to any position, as you wish.



Figure 6.22: Gauge with text area

To do this, set the "Labels" property to "Show" and click the "..." button to open the dialog for defining the regions. In this dialog, click the "New" button to create the respective ranges with position, rotation, frame size, background, font and formatting. You specify the position in relation to the area of the gauge (measured from left to right). For example, a vertical and horizontal position of 50% each positions the label precisely in the middle.

Begions:	🧃 🗙	1 24 C+3	0
1=50%, y=30%, "km/h"		BData	
		(Contents)	"km/h"
		Appearance	
		Background	Transparent
		E Font	to 0.80 lenet
		Format	None
		Frame Size	20%
		Rotation	0*
		ELayout	
		Appearance Condition	Always Show
		E Position	x=50%, y=30%
		Horizontal	58%
		Vertical	30%
		Position Defines the position of the test a from left or above.	rea relative to the gauge area, measured

Figure 6.23: Text label definition dialog



Creating a Crosstab

Crosstabs are used for evaluating and presenting data in multiple dimensions. Crosstabs (or contingency tables) are tables containing information about the frequency of the occurrence of combinations of certain characteristics. These frequencies are extended by their marginal totals which form "contingencies." With a three-dimensional crosstab, (three characteristics), the table includes an additional column grouping.

For example, you can examine turnover trends per year and region, evaluating sales according to quantities and customers, and create marginal totals for quarters and years.

A normal ("flat") table has the attribute names in the first row and the occurrences of these attributes in all other rows. A crosstab is different. The titles of both columns and rows receive characteristic occurrences and, at the point of intersection of the respective column and row, a value is shown that depends on the characteristics specified for the column and row in each case.

	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

Customer turnover per year and quarter

Figure 6.24: Example of a three-dimensional crosstab

For the schematic presentation of two-dimensional crosstabs, the 3D multi-row bar chart is the best choice. You can find more information about this in chapter <u>Creating Charts</u>.

See also: • Creating a Crosstab Object

- Defining GroupingsDefining Cell Properties
- The Layout Option and Wrapping Behavior
 Special Functions



Creating a Crosstab Object

Let's assume that you want to examine the development of turnover per year, quarter and country. Proceed as follows in the List & Label Sample Application:

- Crosstabs are elements in the report container. Therefore, you add these objects in the "Report Structure" tool window. If you have not yet added a report container to the workspace, select "Report Container" in the "Objects" toolbar and pull the object to the right size in the workspace while holding down the left mouse button.
- 2. A selection dialog will appear for the chosen object type. Choose the "Crosstab" object type.
- 3. In the following dialog, now select the data source. All available tables are shown hierarchically, in other words, under the tables you will find the related tables in each case.

For our turnover analysis, choose the "Customers > Orders > Order Details" table in the List & Label Sample Application so that you have all three tables at your disposal. The "Customers" table contains the country, the "Orders" table the order date and the "Order Details" table the turnover.

4. A wizard appears which will lead you through the 3 configuration dialogs for crosstabs.



Defining Groupings

In the wizard's first dialog, or alternatively on the "Axis Definition" tab, you first define the grouping for the rows and columns, i.e. the characteristics.

- 1. In the "Rows" pane, click on the "Insert a row grouping" button.
- In the formula wizard, you now enter the field or the expression for the row grouping e.g. Customers.Country. You have now created a row grouping and the data will be grouped by this characteristic.
- 3. In the "Columns" pane, click on the "Insert a column grouping" button.
- 4. In the formula wizard, you now enter the field or the expression for the column grouping.
- 5. Since you first want to group the data by year, you must enter an expression here that returns the year of the order date. You have the Year() function in the formula wizard at your disposal; i.e. you select this function from the list and insert the order date as the parameter by double-clicking. The formula looks like this: Year(Orders.OrderDate).
- 6. Since we also want to examine the data at another level, insert an additional column grouping via the "Insert a column grouping" button.
- 7. Now enter an expression to return the quarter of the order date. You can use the Quarter() function in the formula wizard for this. The formula then looks like this: Quarter(Orders.OrderDate).

Note: You can change the order of the groupings with the arrow button. The column at the bottom is the inner grouping.

To swap lines and rows (Pivot function) use the button on the lower right on the "Axis Definition" tab. This button is only available in the object dialog, not in the wizard.

Sum(Order_Details.	Quantity*Order_Details.	UnitPrice)
		2
	-	Cancel

Figure 6.25:Swap all rows and lines

- 8. You have now created the groupings and you can go on to define the value for the intersection of the respective columns and rows. Click on the "Edit group result formula" button located under "Value of the result cells".
- 9. Now select the aggregate function that you want for the contents in the "Cell Contents" dialog that appears. You want to create a sales evaluation so choose the "Sum" function. In the upper part of the dialog, you can specify the contents by clicking the formula button to start the formula wizard.

ontent			
Order_Details.UnitPrice * Order_Deta	ils Quantity		√a
ggregate function for the contents:			
no aggregate function -			
Sama			
Arithmetic Mean (Average)			
Geometric Mean			
dedian			
dode			
Highest Value (Maximum)			
owest Value (Minimum)			
lanance			
itandard Deviation			
Count			
Number of Distinct Values			
Result			
Sum(Order_Details.UnitPrid	e * Order Details	Quantity)	
U sangerste preserventer	e concercio	dama'	

Figure 6.26: Formula wizard for defining the cell content

10. In the Sample Application, the sales per order value is not supplied directly as a field so you must calculate it using the "Order_Details.Quantity * Order_Details.UnitPrice" formula.

			H	leadin	g				
			20	800	Total				
			4	Total	Total				
	Lloading	Germany	0.00	0.00	0.00				
	Heading	Total	0.00	0.00	0.00				
statistic about the columns are	opengs for the rows and columns is to tarnover by year and country, o d 'Country' as the group for the ro e sum over the tarnover.	hoose 'Year' as the group for	Columns:	DederDate) ContexConte		1 3	×	,	2
Bews: Contament Cou	nday	1 2 × 1 4	Value of the res Seare(Order_D-		rder_Details.UnitPri	c#)			
Cutamins Cou	ntry		Sum(Order_D			ce)			

Figure 6.27: Definition of the axes of a three-dimensional crosstab



Defining Cell Properties

In the wizard's second dialog or, alternatively, on the "Cell Definition" tab, you edit the properties of the different cells. You can select the cells directly in the drawing in the upper pane of the dialog and then edit their properties. To select multiple cells, hold down the CTRL key or you can draw a border around the cells with the mouse.

1. Assuming that you want to prefix the number of the quarter with a "Q" as the title of a column. Select the respective column title and then double-click on the "Displayed Contents" property.

Heading 2009 Total Q4 Total Q4 Total Q3 Font Pote Pote <th></th>	
Data Displayed Contents Orders.OrderDate Displayed Contents Displayed Contents OrderS.OrderDate Displayed Contents Displayed Contents OrderS.OrderDate Displayed Contents OrderS.OrderDate Displayed Contents OrderS.OrderDate Displayed Contents Displayed Con	he different areas
Displayed Contents Orders.OrderDate Instrume and the sketch above and properties. Entract Q3 Background Transparent E Font [Arial 10.0 pt] Rotation 0*	ne or - by holding
Format Q3 properties. Appearance Image: Compare the second s	
Appearance Background Transparent Font [Arial 10.0 pt] Rotation 0* *	and a set of the
Background Transparent Font [Arial 10.0 pt] Rotation 0* ←	
Rotation 0" -	
Rotation 0" -	
Format Output format.	

Figure 6.28: Cell definition for the crosstab

 With this property, you can now specify the text that is to be displayed in this cell (independent of the value that you have defined for this column grouping).
 Now define either a suitable formula, e.g. "Q " + Str\$(Quarter(Orders.OrderDate)) in the formula wizard, or use the "Format" property. Then remove the "Quarter\$()" here, i.e. only the date field remains in the field, and format the value by means of the property.

To do this, click the "Formatting" property, choose "Date" as the formatting type, and finally select the "Userdefined" entry from the drop-down list. At the end of the list you will find an example for formatting a quarter plus the number of the year. Since we don't need the number of the year, shorten the formula's string to "Q%q".

No format	(user-defined)		
Number	tere second		
Currency	Q%cq		
Date	%d.%m.%y	5.7.2010	1
Time	%02d.%02m.%y	05.07.2010	1
Date and Time	%id/%im/%iy	5/7/2010	
Percentage	%02d/%02m/%y	05/07/2010	1
Angle	%d-%m-%y	5-7-2010	
Date-/Time Difference	%02d-%02m-%y	05-07-2010	1
pages time parenter.	%d %m %y	572010	
	%02d %02m %y	05 07 2010	
	%m/%d/%y	7/5/2010	
	%02m/%02d/%y	07/05/2010	
	%m-%d-%y	7-5-2010	
	%02m-%02d-%y	07-05-2010	
	%y.%m.%d	2010.7.5	
	%y.%02m.%02d	2010.07.05	
	%y/%m/%d	2010/7/5	
	%y/%02m/%02d %y-%m-%d	2010/07/05 2010-7-5	
Preview	[(3ev-3em-3ed	2010-1-3	
	((3	
3		ОК	Cancel

Figure 6.29: Formatting for date values

3. This cell is now formatted and you can go on to format all other cells in the same way. Various properties are available including:

§ Rotation of the content in increments of 90°

§ Background

§ Frame

§ Font

§ Vertical and horizontal alignment

§ Maximum width, minimum width and minimum height

Also see chapter **Overview of Properties**.



The Layout Option and Wrapping Behavior

In the wizard's third dialog or, alternatively, on the "Properties" tab, you edit the layout properties and specify the wrapping behavior.

Various layout properties are available including:

- § Background
- § Default frame
- § Minimum size (%) and minimum height

In addition, as crosstabs are often wider and higher than the specified page format, you can also specify the wrapping behavior for columns and rows. List & Label creates as many pages (shadow pages) as necessary. The row labels are repeated on all pages as standard while the column labels are not repeated.

Various wrapping properties are available including:

- § Repeat Labels: Specifies whether the labels of columns or rows are to be printed again in the case of a page break.
- § Break Level: Specifies the optimum break level, e.g. "0". This corresponds to the lowest group, i.e. the quarter.
- § Column > Page Break on Shadow Pages: If the cross table is too wide, the wrapped parts are printed on shadow pages. A shadow page does not count as a "real" page and therefore does not have a page number. The default setting specifies that the wrapped parts are to be output below the table.

Also see chapter **Overview of Properties**.

2008 Total Q4 Total Q4 Total Total 0.00 Transparent Transparent Default frame Transparent Action Transparent Wrapping Options Column				H	leadin	g	
Q4 Total Heading Germany 0.00 0.00 Total 0.00 0.00 0.00 Total 0.00 0.00 0.00 Appendice Image: Constance Image: Constance Image: Constance Default Frame Image: Constance Image: Constance Image: Constance State Constance Image: Constance Image: Constance Columns Columns Image: Columns Image: Columns				20	08		
Heading Total 0.00 0.00 1 1 0.00 0.00 2 1 0 0 Poperture 0 0 0 Background Transparent 0 Detault Frame 0 0 Actions 0 0 Wrapping Options 0 0				Q4	Total	lot	al
I otal 0.00 0.00		22. 8	Germany	0.00	0.00	0.0	00
PAppearance Post can choose various layout option Back ground Transparent Default Frame Action Link UFL Ground Columns		Heading	Total	0.00	0.00	0.0	00
E Columns	Background Default Frame Actions		Transparent			0	
Minimum Height 3.937 in			1.937 m				

Figure 6.30: Crosstab properties



Special Functions

Various additional functions are available in crosstabs including:

- § Crosstab.Value() returns the content of the cell (as a value).
- § Crosstab.Cells.Avg() returns the average value of the volume of data.
- § Crosstab.Col\$() or Crosstab.Row\$() returns the description of the column or the row for the current cell.

With this, you can, for example, assign a particular color to the background of a column or row. The following example sets the background color to orange for all cells in a row where the cell descriptor is "Germany":

Cond(Crosstab.Row\$()="Germany",LL.Color.Orange,LL.Color.

		Cu	stomer	turnov	er per y	ear and	quarte	er	
	2008		2009				20		
	Q3	Q4	Q1	02	Q3	Q4	Q1	02	Total
Germany					1,086	1,208	851	491	3636
Mexico		492		3,038	2,562	1,277		514	7883
Sweden	2,102		3,429						5531
UK		480	1,352		2,143	1,704			5679
Total	2102	972	4781	3038	5791	4189	851	1006	22729

Figure 6.31: Coloring a particular row

§ Crosstab.Cells.Max() or Crosstab.Cells.Min() returns the largest or smallest value in the entire crosstab. With this, you can, for example, emphasize the largest or smallest value of the volume of data or perform calculations. The following example sets the background color of the cell with the largest value to green:

Cond(Crosstab.Value=Crosstab.Cells.Max(),LL.Color.Green, Cond(Crosstab.Row\$()="Germany",LL.Color.Orange,LL.Color.

		Cu	stomer	turnov	ег регу	ear and	quarte	er	
	20			2009			20	10433	
	03	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Total
Germany.					1,086	1,208	851	491	3636
Mexico		492		3,038	2,562	1,277		514	7883
Sweden	2,102		3,429						5531
UK		480	1,352		2,143	1,704			5679
Total	2102	972	4781	3038	5791	4189	851	1006	22729

Figure 6.32: Coloring a particular cell

§ Crosstab.Col() or Crosstab.Row() returns the index of the column or the row for the current cell. Here, for example, you can set the background color of alternate rows thereby producing a zebra pattern. Example: Cond(Odd(Crosstab.Row()),LL.Color.LightGray,LL.Color.White

	Cu	stomer	turnove	er per y	ear and	quarte	er i	
2008		2009				20	7.4.1	
Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Total
				1,086	1,208	851	491	3636
	492		3,038	2,562	1,277		514	7883
2,102		3,429						5531
	480	1,352		2,143	1,704			5679
2102	972	4781	3038	5791	4189	851	1006	22729
	Q3 2,102	2008 03 04 492 2,102 480	2008 Q3 Q4 Q1 492 2,102 3,429 480 1,352	2008 20 03 04 01 02 492 3,038 2,102 3,429 480 1,352	2008 2009 03 04 01 02 03 1,086 492 3,038 2,562 2,102 3,429 480 1,352 2,143	2008 2009 03 04 01 02 03 04 1,086 1,208 1,086 1,208 492 3,038 2,562 1,277 2,102 3,429 2,143 1,704	2008 2009 20 Q3 Q4 Q1 Q2 Q3 Q4 Q1 1,086 1,208 B51 492 3,038 2,562 1,277 2,102 3,429 480 1,352 2,143 1,704	2008 2009 2010 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 1,086 1,208 851 491 492 3,038 2,562 1,277 514 2,102 3,429 480 1,352 2,143 1,704 2,104

Figure 6.33: Creating a zebra pattern

§ Join\$() returns a collection of strings, separated by a particular character. For example, you can output the individual turnover amounts in addition to the total turnover. Example:

Fstr\$(Sum(Order_Details.Quantity*Order_Details.UnitPrice),"
##,####,###") + "¶["+

Join\$(Fstr\$(Sum(Order_Details.Quantity*Order_Details.UnitP ##,###,###"))+"]"

	2008			20	09	2010			Total
	03	04	01	02	Q3	04	01	02	Tota
Jermany					1,086 [684; 1,062; 1,096]	1,206 [878; 938; 1,209]	851 [825:851]	491 [400; 491]	3636
Мехісо		492 [29; 89; 492]		3,030 [690; 881; 1,931; 2,115; 2,213; 3,038]	2,562 [70; 140; 480; 1,182; 1,742; 2,562]	1,277 [320; 1,257; 1,277]		514 [42: 102; 166: 514]	7883
Sweden	2,102 [248; 908; 1,189; 1,489; 1,532; 1,916; 2,102]		3,429 [1,814; 2,222; 2,534; 2,900; 3,023; 3,254; 3,341; 3,429]						5531
UK		480 [90; 480]	1,352 [96; 291; 899; 1,052; 1,352]		2,143 [238; 1,298; 1,500; 2,098; 2,143]	1,704 [504; 1,204; 1,704]			6679
Total	2102	972	4781	3038	5791	4189	851	1006	22729

Figure 6.34: Display detailed information

§ You can use Total() for calculations across all cells. Otherwise, calculations are always made across all values that affect the respective cell. Also see chapter <u>Overview of Functions</u>.



Creating Statistical Reports With Footers

When you enable the "Data Lines.Suppress" object property in tables, all data lines are completely suppressed. This option is particularly useful in combination with the "Force Sums" option. The latter option specifies that totals are also calculated when a data line is not printed. By combining both options, you can output footer lines with totals and produce interesting statistics in this way.

Let's assume that you want to output the turnover per country:

Customer turnover

Mexico	\$172.80
Sweden	\$380.00
Denmark	\$140.00
USA	\$616.00
Italy	\$320.25

Figure 6.35: Creating statistical reports with footers

Proceed as follows in the List & Label Sample Application:

- 1. Create a new element in the report container and choose "Table" as the object type.
- 2. In the following dialog, now select the data source. All available tables are shown hierarchically, in other words, under the tables you will find the relational tables in each case.

To evaluate sales per country, for example, choose the "Orders > Order Details" table so that you have both tables at your disposal. The "Customers" table has a 1:1 relationship with the "Orders" table so you don't need to select it. The turnover is held in the "Order_Details" table.

Report Structs	are		×
88 98 0	* × ¥ 🕲 🗅	2.3	
	t Container ble: Orders Table: Order Details	[Orders20n	fer Details]

Figure 6.36: Hierarchical tables for statistical reports

- 3. Create a data line with the "OrderID" field in the "Orders.Order_Details" table. Although the data line is not output (it is suppressed), List & Label still needs a field so that the table can be printed at all.
- 4. Define the actual statistic as a footer line, i.e. with the country name in the first column, and total the turnover in the second column. Now calculate the total again with the "Sum(Order_Details.Quantity * Order_Details.UnitPrice)" formula

1	Header Line	1	Data Line	~	Footer	Line (iroup Header	Group Footer	
3	< a • a	×	* * 0	1.11		11 24 t-1			(
12	a 😂 Line Defin	tion 1				Plarne			
1	A Order	Details.C	beder Die Order	Order@D	Orders	@Data		11	
	Coder_	Details (Quantity * Orde	Details I	UnitPri-	(Contents)		Order Details.	OrderID@Orders.O
						Options			
						E Approximite	1) 1)		
						E Background	1	Transparent	
						Export as Pa	chure	No	
						III Font			(Ariai 12.0 pt)
						Format		None	
						Rotation		4.	
						Text Format	6	NormalTex	
						EAction			2
						Link URL			
						Blayout			
						E Alignment		Left	
						Appearance		Always Show	
14	1 H					Blank Optin Format	NE8DION	Yes	
200			_			Output format			
54 De	emple variable '0 etails.Order@@0	rders.Or	idedD(Orders.C	ustomed	DeCur				

Figure 6.37: Creating footer lines for a statistical report

- In the "Orders" table, create a data line with the "Country" field from the linked table "Customers". Although the data line is not output (it is suppressed), List & Label still needs a field so that the table can be printed at all.
- 6. Define the grand total across all countries as a footer line, i.e. with "Total" in the first column, and total the turnover again in the second column.
- 7. Now select the "Orders" table in the "Report Structure"

tool window and set the "Data Lines.Suppress" property to "Yes".

8. Finally, also select the "Orders" table and set the "Data Lines.Suppress" property to "Yes" here as well.

E Line Options	
E Data Lines	
Force Sums	Yes
Keep Rows Together	Yes
Suppress	Yes
Zebra Pattern	Pattern/Block Color
Footer Lines	
Group Footer Lines	
Group Header Lines	

Figure 6.38: Suppressed data lines for statistics



Drilldown Reports (Increase Detail Level)

Navigation in hierarchical data is known as drilling down. Drilling down makes it possible to "zoom in" to examine the data at different levels of detail. Different print templates are linked to each other to achieve this. This property is only available with hierarchical tables.

Cui	stomers					enc	<u> </u>				
	CustomerilD		CostactNam			Drilldo a	in a second				
*	ALTH	Alter Fullekale	Maria Antersi	bein		She.	Orders	as chart	 I 		
							Orders .	tol on	Open here		
2	ALLANS	Ana Trulle Emparetator y	Ana Thomas	Micro	+ 9.8.	Sher.			Open in new t	ab D	
		helador							Open in new t	ab in backgrou	NE.
з	MITON	Antonio Monero Targuería	Antonia Micraha	Maria	.01	than-					
				Drilldown re	port with	chait an	d list	1	ALFKI	*	
4	ARIDUT	Around the Horn	Tionas								
	MRAS		Sec. 1								
5	MENES	Barglondo snatikkisp	Chatina								NSHIN
				-						ag	enc
	BLAUS	Bayer See Delkaterren	Havea	Order	s					1.1.051.0517	
				ALPHI	Order die				freight Ondert0		
7	B.ONP			9	25.09.200	. 40	00.10		29,46 10,643		
-	active	Blos del père el file	Trédérips		For Ord	Ber10 540	Product D	Guardet a	ProductName Ritoria Saverkiaut		105.00 E
					2 10.0	BHO	39	21	Charlie une verte		18,00 4
	BOLD	Bölim Constar preparadar	Mattin		3 40.6	940		2	SeaperM		12,00 el 75,60 el
				(2)	03.11.200			2009	61.02 10.692		
					Post Ord		Product E	Guarety 20	ProductRame Vege-spread		Price 45.90 #
•	BOWAP	Box and	Laurence						100010100		41,00 4
				(2)	13.11.200 Pos Ord		21.11.		23,04 10,732 ProductName		Price
10	BOTTM	Buttury Dollar	States.		1 10.3	702	3	0.	Aniesed Derug		10,00 4
1					2 10.1	702	76	15	Lananite		10,00 4
				1.140	16.02.20	10	21.02	2010	99.53 10.005		anton a
					Pos Gro	eri0	Productio				Price
			_		2 10/	106 106	58 77	2	Ratiette Cruzdanaut Original Flankfurter prime	Sola	05.00 € 13.00 €
											810,00 €
				5	15.04.20 Pos Ord	10 Marito		Querto .	40.42 10.952 ProductName		Price
					1, 10.0	852	6	10	Grandmats Boynamberry 1	loinat	25,00 @
					2 40.0	82	28	2	Ritche Saverkraut		46,00 € 70,00 €
					09.05.30			2010	1,21 11.011		
					Pos 010		Priduit D	Quaritite 40	Productione		13.25 d
					2 110		14	20	Escarpetz de Beurgogne Flatenus sit		21.50 4
										nenementation	34,75.4
										229,59 kd	320,004

Figure 6.39: Example of a report with an open drilldown report

These reports make it possible for anyone to find the information they are looking for quickly, even with very large and complex data inventories.

Only one level is printed to start with (e.g. customers). A new detail report opens (e.g. orders) when you click on a customer.

This drilldown report can be opened in the context menu either in the same window (navigation via the green arrow button in the preview window), or in a new foreground or background tab.

The drilldown function is only available in the preview. You can export any drilldown report to another format from the preview, e.g. PDF.

Drilldown reports can be embedded in the preview file to allow them to be sent or saved as a complete unit. You will find the respective option "Embed Drilldown Reports" in the project properties.

A drilldown link relates either to a single field or an entire table row. A whole series of links can be associated with each of these elements, e.g., to present the data in different ways. Proceed as follows to create a drilldown report:

- In the "Objects" toolbar, choose the "Report Container" object type. In the workspace, hold down the left mouse button and pull the object to the required size. A selection dialog will appear for the chosen object type. Choose the "Table" object type.
- 2. In the following dialog, now select the data source. All available tables are shown hierarchically, in other words, under the tables you will find the related tables in each case.

In order to be able to open a sub-report in drilldown mode, you must select a table here that also has a sub-table! In the Sample Application, select the "Customers" table, for example, as it has "Orders" as a sub-table.

- 3. Now define the columns of the table with the wizard, i.e. CustomerID, CompanyName, ContactName, City.
- 4. In the object dialog for the table, now define an additional

column for the drilldown link. Enter the text "Show..." as the content.

5. Now open the dialog for creating the drilldown link by means of the "Drilldown Links" column property.

- H	rader Line	1	Data Line	Foster	Line Geoup Header	Graup Foater	
1	B • 12	×	A 10 (D)		10 24 C+2		8
1777					Bloom		
	Ca Line Defin		men CustomedD		Name		
	A Custor			*2	10 Deta		
			ampanyfilame		(Contents)	"Show"	
			ontactflame		Options		
	- Custor				E Appearate e		
	(A) "Show		10		@Background	Transparent	
	and the second				Esport as Picture	No	
					E font	perial 10.0 p	1
					Format	Hone	
					Rotation	<i>e</i>	
					Test Format	Normal Test	- 11
					BAtten		
					Delidown Links Link URL	"dolldown reports/Drilldown report as chart.ort", [x
							-1
					BLayout	Left	- C
					E Alignment Appearance Condition	Always Show	
					Drildewn Links	AWAYS STOP	
i har					Opens the dalog for editing	the Duildown lasks	
					a second second		
					L		
(B) (C	-					OK Cane	al.

Figure 6.40: Data line with additional column for a drilldown link

6. Create a new drilldown link with the "Insert new link" button in the dialog that appears.

Drilldoen Links				1 20-1			
Loin .	1 2 × 1 + 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		0			
		Meres Test	Customers	-			
	Create New Drilldown Lin	and the second se					10
	The only have one available	e relation for the link:					
	Online (Castorner, 2014	el.					
	Would you like to						
	ProjectPath\$) + "C	ustamen 20eden, ut ^a					6
Hove the links into the			conduit 22.26 Sample Application.		160		
3	Projects in the select	ed file path fitting to the	selations				
	· Project name	Desception	stery		10		
	0					OK.	Cancel

Figure 6.41: Dialogs for defining the linked drilldown report

- 7. A dialog appears where you can create the print template for the sub-report. Select the "Create a new project" option and enter the name for the print template.
- 8. A second instance of the Designer opens for you to create the print template. Proceed as usual to create this sub-

report. I.e. you create a report container, choose the "Table" object type, choose "Orders" as the data source and define the columns of the orders table that are to be shown in the report.

- 9. Once you have completed the report, close this second Designer instance.
- 10. Back in the "Drilldown Links" dialog, you now define the "Menu Text" property for this link. If you have more than one link, its text will be shown in a context menu. The text will also be used as the title for a tab if multiple drilldown reports are shown in a preview window.
- 11. You have now finished the drilldown report and you can display it in the preview.



Multi-Column Reports

With complex projects containing many different diagrams, it may be a good idea to present them over several columns.

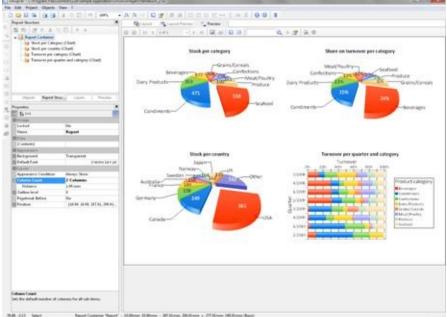


Figure 6.42: Multi-column report container

To make this possible, the "Column Count" property is provided in the report container and also in tables. You can define up to five columns whereby the report container and the included tables can have different column count values. Creation is easy:

- 1. Select the "Report Container" object in the "Report Structure" tool window.
- 2. Set the "Column Count" object property to "2".
- If you select individual objects, you will then have various properties for controlling column breaks at your disposal: § Column Break Before: A column break will be performed before the object is output.
 - § Column Break Condition: If the result is "True" when a data line is output in a multi-column table, a column break will be triggered. Tip: the "LL.CurrentTableColumn"

field returns the index of the current column. (Page) Break Before: A page break is performed before the object is output. If you have multiple multi-column objects, a page break is triggered automatically after an object if the column counts for the objects are different (e.g. 2-column table followed by a 3-column table) and if there would be insufficient room for the object that follows.



Advanced Functions

In this chapter, we will concern ourselves with topics that you will probably only use very rarely. Nevertheless, the possibilities offered by linking objects and by sum and userdefined variables provide you with an important and useful tool for producing sophisticated printed outputs.

See also:

- Linking Objects
- Working With Sum Variables
- Working With User Variables
- Project Includes
- Displaying HTML Pages
- Adding OLE Documents
- Adding Form Template Objects
- Importing Objects



Linking Objects

By interlinking objects, you can influence the order of printing causing some objects to be printed after others thereby overlaying them in the event that they overlap ("sequential (temporal) linking"). Another possibility is to cause the size and position of some objects to be adjusted automatically to correspond to changes to other objects ("spatial linking"). Designer differentiates between three kinds of interlinking:

- § Sequential
- § Individual size and position adaptation
- § At end, keep size

When linking objects, there is a hierarchy: the main object and the attached (interlinked) object.

See also:

- Object List
- Creating Interlinks
- Sequential Interlinking
- The Individual Size and Position Adaptations
- The "at end, keep size" Interlink



Object List

The object list defines the implicit print order of the objects, thus they are automatically sequentially interlinked with one another. Unlinked objects are printed first, followed by the tables and finally the objects that are linked in any other



Figure 7.1: Object list in the tool window

Example: In the above object list, the objects "company logo", "addressee", "invoice data", "dummy rectangle" and "page count" are printed first. The table "item list" comes next followed by the interlinked object "payment terms".

This means that if an object is to be printed over another object, it must be printed after this object. Sequential interlinking is therefore always necessary if an object that is not interlinked is to be printed after the interlinked objects, if an object is to be printed over a report container or if the contents are only available after another object has been printed.



Creating Interlinks

You create interlinks or edit existing ones in the "Object List" dialog. This dialog opens up via the **Object > Object List** (NUM followed by * on the number pad) or by double-clicking the "Objects" tool window.

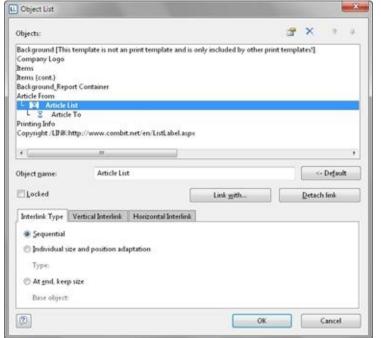


Figure 7.2: Defining interlinks in the object list

You can interlink the selected object with other objects via the "Link with..." button. The link is shown in the object list in the form of a tree structure in which you can see the defined interlinks. You can also interlink several objects with one object.

When you select the interlinked object, you can choose from the three different kinds of interlink in the lower part of the dialog.

The "Detach link" button lets you remove an existing

interlink. The interlinked object will then be shown in the object list as an independent object.



Sequential Interlinking

Sequential, or temporal interlinking makes sense if the content of the interlinked object can only be filled once the main object has been printed or if an object is to be printed over another object.

Example 1: You are printing an article list and want to output the number range of the articles on this page.



Figure 7.3: Range of article numbers at the top of the list

But the "Article To" object only knows the last article number on the page once the table has been printed. Therefore, the "Article To" text object must be interlinked sequentially with the "Article List" table.

Select the "Article To" object in the object dialog and interlink it with the "Article List" table. The "Sequential" interlink type is already selected.

You must also interlink the table with the "Article From" object. Sequential interlinking is sufficient here as you don't want to change the size or position of the object.



Figure 7.4: Object list with sequential interlinking

Example 2: You want to output "Copy" over a table.

You therefore create a text object containing "Copy". Because objects that are not interlinked are printed first and tables are printed last, the text object must be linked sequentially with the table. Otherwise it would be printed before the table and therefore under the object.



Figure 7.5: Text object over the list

Select the text object in the object dialog and interlink it with the "Article List" table. The "Sequential" interlink type is already selected.

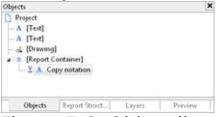


Figure 7.6: Object list with sequential interlinking



The Individual Size and Position Adaptations

Individual size and position adaptations are spatial interlinks, i.e. the size and position of the main object determines the size and position of the linked object. This automatically causes sequential interlinking as well.

If the size or position of the main object changes because the variables that it contains take up less room as the object provides, the linked objects adjust their size automatically to these changes.

Two kinds of interlinking are therefore available:

- § Position adaptation: If the position of a main object changes, the position of the linked object changes too. You have three options here:
 - § Relative to begin: The interlinked object moves in relation to the upper left corner of the parent object.
 - § Relative to end: The interlinked object moves in relation to the lower right corner of the parent object.
 - § To end: The upper edge of the interlinked object starts at the end of the main object, irrespective of its original position. This causes an implicit size change on the first page on which the child object is printed.
- § Size adaptation: If the size of a main object changes, the position of the linked object changes too. You have two options here:
 - § Proportional: The size of the linked object changes exactly like that of the parent object. E.g. if the main object becomes 10 mm shorter, the interlinked object will also become 10 mm shorter.
 - § Inverse: The size of the interlinked object is adjusted

inversely to the size of the main object. E.g. if the main object becomes 10 mm shorter, the interlinked object will become 10 mm longer.

You can specify whether the interlinking is to be horizontal and/or vertical in each case:

- § Vertical interlinking: The linked object adjusts its position or size to changes in the vertical position or height of the main object.
- § Horizontal interlinking: The linked object adjusts its position or size to changes in the horizontal position or width of the main object.

Example of vertical, relative to end position adjustment: You print an invoice and want to output a closing text after the table.

You therefore create a text object and position it below the table.



Figure 7.7: Text object below the list

Select the text object in the object dialog and interlink it with the table. Choose "Individual size and position adaptation" as the type of interlink. On the "Vertical Interlink" tab, enable the "Vertical" check box for the position adaptation and choose the "Relative to end" option so that the interlinked text object adjusts its size depending on the position of the lower right corner of the table.

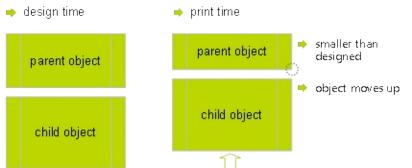


Figure 7.8: Diagram of vertical, relative to end position adjustment

If the table becomes smaller, the text object moves upwards proportionally. It doesn't matter where the table ends, the text object will always be output after the table in the specified size.



We thank you for your order

Figure 7.9: Closing text below the list

Example of vertical, relative to end position adjustment and vertical inverse size adjustment: On a multi-page invoice, the item table is to start on the first page below the address. On following pages it is to start at the upper page margin.

1. Create an invisible frame by inserting a rectangular

object.

- The upper edge of the object is positioned precisely where the invoice table is to begin on the following pages. The lower edge is positioned precisely where the invoice table is to begin on the first page.
- 3. Assign the "Page()=1" appearance condition to the rectangle so that it is only printed on the first page.
- 4. Create a table. This begins directly below the rectangular object.



Figure 7.10: Invisible rectangular object above the table

5. Now interlink the table with the rectangle (main object) and select "Individual size and position adaptation" as the type of interlink.



Figure 7.11: Spatial interlinks in the object list

On the "Vertical Interlink" tab, choose the position option "Relative to end" and the size option "Inverse".



Figure 7.12: Interlink options in the object list

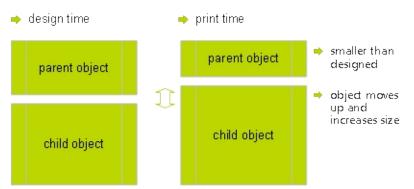


Figure 7.13: Diagram: Vertical, relative to end position adaptation and vertical inverse size adaptation

The table now changes its position based on the lower right corner of the rectangle (main object) and adjusts its height inversely in proportion.

The Page()=1 appearance condition prevents the rectangle from being printed on the second page. Therefore, it "shrinks" by 80 mm on the second page, and the table moves upwards proportionally and becomes 80 mm larger.

			age	ncy	Same,	-	Deceptor	10.07	NA	1m.
	OC Dystants in Son We John In	ut.			3.00	THENCOM	First-day city hig to take hort skotuding topic and according topic and		100314	100.0
1	ABC Bott pringPerg 121an		1000		.06	Expension	Baldres despite 14 April Incluen Ran-Atol Pandon Beact *** autoing Split		100,014	1.00030
			Invoi	50 ML 07 2010	1.81	TINAUUT	Fight one Australia		115,104	815,80
			Manibur	47119815				Sublidge Sales Sales		100.00
				_				the l		1.002.45
(Longith)	144	Department	(Martine)	104						
1.88	E-Pieji	 Boutham Africa Explorer 20-8 Ibut train Cape Town to victory Y artic excluding fight 		1.806.00.4						
1.00	EGMAD	Maldres dring tig: 14 dars, midtern Male Atol, Parados Beart - sciuding tight	1 000,00-4	1400.004						
2.00	Exercican	Vulutan, On the Yost of the Maria, Sweet round Ing. excluding there	1 200.00 4	3 405,00 4						
1.08	TRANSMEN	Native Softier to Barrague to Ad- Maximum, CBy Da to Pringue excluding an communitation.	355,00.4	101.00 4						
1.80	78V86A01	Flight only Brief	792.004	110.00.4						
1.00	EX1EMBO	Parks, admission to the lideral call and these 2 the Mouth Hauge Including Hase course man	176,00-6	10004						
1.88	INTO 20181	College, South of England will services, 4 percents	100004	1.050.00 K						
3.88	Echelli	Hang Kiting and Ball 2 sealing recording Repits, accorrectables, excursions	1 190,00 4	4386.00.4						
			Lanua -	8406004						
			pa yat	1808.04.4						
			Ner	10,204,044						

Figure 7.14: Position and size adaptation on the second page

Example: position adaptation, vertical to end: The interlinked object changes its position based on the main object. The size is also adjusted implicitly because the upper edge of the interlinked object changes based on the main object but the position of the main object does not change. The interlinked object must overlap the main object. The main object must be larger than the interlinked object. If the main object becomes smaller, the interlinked object only changes its position from above and increases its size. If the main object becomes larger, the linked object shrinks (the main object must be in the foreground).

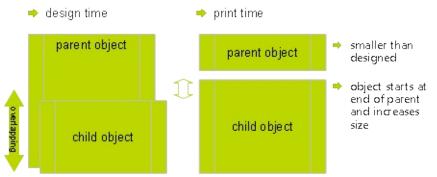


Figure 7.15: Diagram: Position adaptation vertical, to end

Example: position adaptation, vertical, relative to begin: The interlinked object changes its position based on the upper left corner of the main object. This type of interlink is the exact opposite of "vertical, relative to end". Example: The main object moves upwards due to the "Alignment bottom = True" setting in the Designer and the interlinked object follows this position adaptation in an upwards direction.

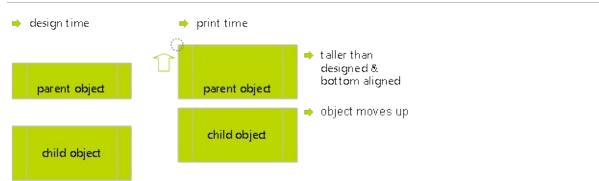


Figure 7.16: Diagram: Position adaptation, vertical, relative to begin



The "at end, keep size" Interlink

This kind of interlink is similar to position adaptation. But here, the main object's available space is taken into consideration and the interlinked object always keeps its size. In other words, the object is always output within the boundary of the main object. If the available space is not sufficient, a page break will be triggered.

The interlinked object must overlap the main object in the Designer. It is crucial that the main object is always larger than the interlinked object. The interlinked object always tries to occupy the space remaining from the original size of the main object and, if the interlinked object is larger, this leads to an infinite loop because there is never enough space.

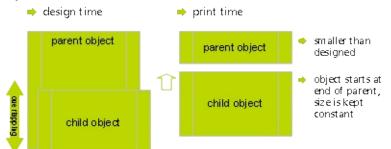


Figure 7.17: Function of the "At end, keep size" type of interlink

Example: Assuming you want to output a scanned signature after a text. The size of the signature must not change and it must be output within the border of the text object.

You therefore create a picture object and position it directly on the formatted text object. The height of the picture object is less than that of the table object. Select the picture object in the object dialog and interlink it with the formatted text object. Choose "At end, keep size" as the type of interlink.

Irrespective of where the text ends, the picture object will always be output after the text in the specified size. If there is no longer sufficient space after the text, the picture object will be output on the next page so that the size can be kept.

thank you for choosing Sunshine Agency for your trip. We are happy to send you your travel documents today.

Enclosed you will find your flight tickets, a travel guide and the route description with the rental car to the hotel. Your personal travel attendant Michaela Soleil will expect you on location.

A. lipleni 17531 90 60 10.

Your travel team from Sunshine Agency

Figure 7.18: The interlinked picture object overlaps the main object in the Designer

Dear Mr. Smith,

thank you for choosing Sunshine Agency for your trip. We are happy to send you your travel documents today.

Enclosed you will find your flight tickets, a travel guide and the route description with the rental car to the hotel. Your personal travel attendant Michaela Soleil will expect you on location.

If you have any questions about the booking, you can contact us by info@sunshineagency.de or by phone 07531 90 60 10.

Sunny greetings

Your travel team from Sunshine Agency

A. liplemi

Figure 7.19: The signature is output at the end without changing the size

In this way you can also position multiple objects after each other (e.g. charts, pictures etc.).The "Base object" of the chain is the first object in the interlink hierarchy with an active page break.



Working With Sum Variables

Sum variables offer another way of creating totals and counters and work fundamentally across tables. They are therefore a good choice whenever you want to create totals across different table hierarchies. In all other cases, we recommend the use of the aggregate functions Sum(), Count() and CountIf() for totals and counters. Aggregate functions are always table-specific. You can even produce statistical analyses directly with aggregate functions e.g. Median(), Variance(), StdDeviation(). You will find a list of all functions in the "Aggregate functions" function category in the formula wizard.

Sum variables can be used to create totals over data sets, e.g. to add up the "Item.UnitPrice" fields in a table in order to calculate the total price. Such totals are permitted for all numeric variables or for expressions that return a numeric value as the result.

But sum variables are also a convenient way of defining a counter which can be updated accordingly for each data record that is printed.

You can create a total across all data sets of a printed page (page totals) or across the entire project (grand totals).

Page sum	Name	Sums over
	@CategoryCount	Cond(LL.CurrentContainerItem = "Categories",Cond(Previ
<[

Figure 7.20: Totaling with sum variables.

Proceed as follows to define the variables:

- 1. Choose **Project > Sum Variables** or the **"Edit sum variables"** button in the formula wizard.
- 2. In the dialog that follows, create a new sum variable via the "Insert a new variable" button.
- 3. An input dialog appears where you can enter a description of the new variable. Give the variable a meaningful name, the "@" character will be added automatically as a prefix.
- 4. Click the "Edit" button to open the formula wizard and assign a field or an expression to the new sum variable. For example, select the numeric field "Item.UnitPrice" if you want to add up the "Item.UnitPrice" column. You can, however, also perform aggregations with complex expressions provided that the result is a numeric value. For instance, you can add up the gross price from the net price and the VAT. Enter the following expression, for example, in the "Sums over" field:

Item.UnitPrice+ Item.UnitPrice* (Item.VAT/100)

5. If you don't want to add up any values but merely want to create a counter or a numeration, the definition is simple: In the "Sums over" field, simply enter the value that is to be added to the existing counter.

The simplest case is a consecutive number that is increased by 1 for each data record. Just enter the value "1".

- 6. The "Page sum" checkbox lets you specify whether the totals are to be set to 0 at the end of a page. In this way, you can define page totals and counters.
- 7. Once you have defined which sums are to be stored in which sum variables, you can use these sum variables in your objects. In the formula wizard, you will find the sum variables at the end of the variable list in the "Sum variables" folder.



Working With User Variables

User variables are a way of saving values and expressions for later use. This saves you having to enter them anew each time if they are frequently needed in precisely this form or if user-defined data is to be output repeatedly. They are, so to speak, "formula building blocks".

You can then store these user variables in project includes if they are also to be used in other projects.

Note: User variables cannot be used within appearance conditions for layers.

You can also use the SetVar() and GetVar() functions if you only want to save values in the variable repository for later use. You will find more information about functions in the chapter <u>Overview of Functions</u>.

Proceed as follows to define user variables:

- 1. Choose **Project > User Variables** or the "**Edit user variables**" button in the formula wizard.
- 2. In the dialog that follows, create a new user variable via the "Insert a new variable" button.
- 3. An entry dialog appears where you can enter a description of the new variable. Give the variable a meaningful name, the "@" character will be added automatically as a prefix.
- 4. Click the "Edit" button to open the formula wizard and assign a field or an expression to the new user variable.

	Formula
@Date:	AskString5("Date of training", F., FSt(S(Day(AddDays(Today()+1,0)), "")) AskString5("Date of certificate", F., FSt(\$(Day(AddDays(Today()+1,0)), "")
<u>(</u>	m

Figure 7.21: Edit user variables

- 5. You can use the arrow to specify the evaluation order. The variables are evaluated from top to bottom.
- 6. Once you have defined the user variables, you can use them in your objects. In the formula wizard, you will find the user variables at the end of the variable list in the "User-defined variables" folder.



Project Includes

If you design several similar projects, it's a good idea to include other projects as "Includes" to avoid having to create elements that occur repeatedly in each new project. In this way, you can easily include a letter head, for example, and any changes can be made centrally, e.g. if the design changes.

For example, if the address of your company changes, it can be updated in a single project instead of having to change hundreds of individual projects.

Or user variables that are used repeatedly (e.g. complex formulas in address fields) can be stored in includes.

You must consider the following restrictions: (1) Includes may not contain other includes. (2) Elements can be interlinked with elements in includes; the element in the include is always the parent element because it is printed first.

Totals and user variables are read and used in the same way. When designing includes, please make sure that you do not cause any overlaps (e.g. use a project with a sum variable as a include which is already contained in the current project). Via **Project > Include**, you can add other projects as includes. You can see the objects belonging to these projects but you cannot change them in the current project.

Included Projects			and the second second		
Projects:	🖉 🗙 🤌 🛛	1 D 24 1+3			
mail header.pt		Design			
and the construction of the		Name	Mail header		
		Visible	Yes		
		BData	Content of the other states and the	and the second s	
		File Name	ProjectPath\$() + ".	.\\LL15\B	
		ELayout			
		Appearance Condition	Show		
		Visible Sets whether the included project	t is visible in the Designer.		
Move the files to the requ	ired order using the button	is or drag & drop.			
(2)			ОК	Cancel	-igure 7.2

Dialog for project includes

You can specify the name, visibility and appearance condition for each include by means of the property list. The buttons let you specify the (print) order. Project includes are always inserted at the beginning of the object list.



Displaying HTML Pages

HTML objects are used for displaying HTML content. You specify the HTML page by specifying the file name (e.g. combit.htm) or the URL (e.g. www.combit.net) in the object properties, or by means of the formula wizard. Also see the chapter <u>Overview of Properties</u>.



Adding OLE Documents

Use the OLE container object to embed OLE server documents. In this way, you can embed documents from other applications (e.g. Word, Excel, Visio, MapPoint) in a report. Only the first page will be displayed as there is no standard for multi-page OLE objects.

You select the object type via the standard Windows dialog "Insert Object". Here you can choose an existing file ("Create from File") or create a new file.

Also see chapter **Overview of Properties**.



Adding Form Template Objects

Form templates are images, i.e. scanned forms, which you place in the background of your workspace as a template. This lets you position objects in a project precisely to fit the form. Although form templates are shown in the workspace, they are not printed and cannot be modified.

The best way of positioning them is by means of the property list.

To place a form template in the background of your workspace, use the form template object which you can insert via the **Objects > Insert > Form Template** menu item. Once you have created the template, it's best to use the objects tool window to select it. Form templates cannot be selected by clicking in the workspace.



Importing Objects

With **File > Import**, you can insert a copy of all objects belonging to another project to the project that is currently loaded.



Different Layout Techniques

You can influence the layout of your report in many different ways. In this chapter, we will examine the possibilities offered by different layout regions, define a multi-page report and control the page break behavior.

- Specifying the Page Layout
- Layout Regions



Specifying the Page Layout

Your first task in a new project is to set up the page layout that you want. Choose **Project > Page Layout** to specify properties such the choice of printer, paper size and orientation. There are different layout options depending on the project mode (label or list).

With multi-page projects, it is sometimes a good idea to choose different layout settings, e.g. printer, page size, orientation, paper bin, for the different pages. You will find more information about this in the chapter <u>Different Layout</u> <u>Techniques</u>.

I X	· · · · ·		
Driginal	Design		
Сору	Condition	IssueIndex() = 1	[True]
Standard layout	Description	Original	
	E Printer settings		
	Printer Name	Printer Name HP Lasedet 5	
	Paper Size	A4 (8.26)	3 in × 11.693 m)
	Alignment	Portrait	
	Duplex	not specified	
	Force Paper Size	Yes	
	Number of copies	1	
	Paper Bin	Default	
	Size Adjustment	Keep scale	
	Sort copies	not specified	
	Use Physical Page	Yes	
	Design		

Figure 8.1: Definition of the layout regions

- Printer Settings
- Export Media
- Templates for Label Formats
- Defining Your Own Label Format



Printer Settings

By means of the properties, you can make different settings for each layout region for printer, page size, orientation, duplex print, number of copies, sort copies and paper bin (e.g. first page on company letterhead and normal paper for the following pages).

The printer settings (and changed export format options) are saved in a special file (e.g. Article_List.lsp). If the respective file is not available when printing, the current default Windows printer is used.

- Size Adjustment
- Use Physical Page
- Force Paper Size



Size Adjustment

Specifies whether the project is to be adjusted to fit the page when different printers are used when printing from the preview or whether the scale is to be kept.



Use Physical Page

Specifies whether the whole physical page, including the non printable margin area, is to be available in the Designer. This is sometimes necessary in order to position labels correctly, e.g. if you use sheets of labels without margins. The nonprintable page margins are shown as hatched areas in the preview.

This enables you to use the complete page when defining the layout of your project but, of course, the printer cannot print these margins. If you place objects on such projects, you must still consider the non-printable margins. If this property is set to "False", only the area that can actually be printed is shown in the workspace.



Force Paper Size

If there is no printer definition file, the application tries to force the page size set during design (e.g. Letter) as far as possible. This is only possible, however, if the selected printer supports either exactly this size or the "user-defined" option. If this is not the case, it will first check whether the printer's default size is large enough, otherwise it will choose the next largest size.



Export Media

This list shows the various export possibilities offered by List & Label.

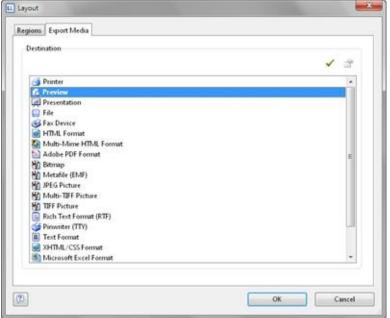


Figure 8.2: Default setting for the output format

With the two buttons on the upper right, you can define a selected format as the default value for the later print and specify the options for this format. These options are saved in a special file (e.g. Article_List.lsp).



Templates for Label Formats

Via the "Templates" tab in the page layout for labels, you can make your selection from numerous predefined label formats from different manufacturers. This automatically specifies the size of the individual labels, how many are to appear on the sheet and how they are to be distributed.

Vame	Sizex	Size y	Dist ×	Disty	Lount : lo	unt ·	Offset × 4
user-defined)							18
terry Densison							
1070 Ready Index Tab	2.500 in	9.100 m	0.000 in	0.000 in	1	1	0.77
11070 Ready Index Tab	3.000 in	1.830 m	0.000 im	0.120 m	1	5	5.13
11071 Ready Index Tab	2.500 in	9.100 m	0.000 in	0.000 in	1	1	0.77
1071 Ready Index Tab	3.000 in	1.130 in	0.000 in	0.090 in	1	. 5	5.13
1072 Ready Index Tab	2.500 in	9.100 in	0.000 in	0.000 in	1	1	0.77
1072 Ready Index Tab	3.000 in	0.530 in	0.000 in	0.120 in	1	01	5.13
11073 Ready Index Tab	2.500 in	9.100 in	0.000 in	0.000 in	1	1	0.77
1073 Ready Index Tab	3.000 in	0.340 in	0.000 in	0.000 in	1	12	5.13
1074 Ready Index Tab	2.500 in	9.100 in	0.000 in	0.000 in	1	1	0.77
1074 Ready Index Tab	3.000 in	0.500 in	0.000 in	0.140 m	1	15	5.00
1076 Ready Index Tak	2.500 in	9.100 in	0.000 in	0.000 in	1	1	0.77
1076 Ready Index Tab	3.000 in	1.130 in	0.000 in	0.090 in	1	8	5.13
1077 Ready Index Tab	2.500 in	9.100 in	0.000 m	0.000 in	1	1	0.77
1977 Ready Index Tab	3.000 in	9.130 in	0.000 in	0.120 in	1	10	5.13
1078 Ready Index Tab	2.500 in	9.100 in	0.000 in	0.000 in	1	I	0.77
11078 Ready Index Tab	3.000 in	0.540 m	0.000 m	0.000 in	1	12	5.13 -
et	- Andrewski -	**			long of the		
Only show templates	for current pag	e settings					

Figure 8.3: Definition of the label size



Defining Your Own Label Format

You can also define your own label formats if you can't find the layout that you want among the templates. You can make the required settings with the "Layout Definition" region property; there is a special dialog for this:

§ Offset: The offset specifies the horizontal or vertical distance of the upper left label to the margin of the chosen page region (physical/printable)

Note: In the screen display, the upper left corner of the workspace always starts at coordinates 0/0 irrespective of the chosen page size and specified offset. However, you will see the effect of the offset in the preview or when printing.

	Horizontal ()	¢::	Vertical (y):		OK
Page:	8.267	in	11.690	in	Cancel
offset	þ.197	in	0.197	in	Help
e 🗄	3.890	in	1.579	in	_
Zistance:	0.000	in	0.000	in	
jumber:	2		7		
vint Order		-			
Hogizont:	ai -	÷	horizontal <u>b</u> ott	torn up	
t ↓ ↓ © ⊻ertical		1	🛉 🔘 vertical botton	n <u>up</u>	

Figure 8.4: Defining customized label layouts

- § Size: This value defines the size (horizontal=width / vertical=height) of the label.
- § **Distance:** The distance to the next label is specified here.

With single-column labels, only the vertical distance must be entered.

- § Number: This option specifies the number of labels per page (horizontal number = number of columns per page, vertical number = number of rows per page).
- § Print Order: Specifies the order of the print if multiple labels are printed on a page. Possible values: 0 (horizontal), 1 (vertical), 2 (horizontal bottom up), 3 (vertical bottom up).

The default is to print labels row by row from the upper left to the lower right (horizontal). However, in the case of sheets of labels that have already been started, it is possible that the first label row has already been used. This causes a loss of stiffness in the upper part of the sheet. Many printers have problems feeding in sheets that have already been started, resulting in a paper jam. In this case, it helps to print the labels in reverse, from the bottom upwards instead of from the top downwards. In this way, the upper row of labels on the sheet is always printed as the last and the sheet retains the stiffness required for feeding in.

See also:

Saving your own label formats in the label template list



Saving your own label formats in the label template list

To save your own label formats, you can edit the file "cmll1601.inf".

Layout of a label definition (all measurements in 1/1000 mm):

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

A: code, B: description, C: page size, D: label width, E: label height, F: horizontal distance between labels, G: vertical distance between labels, H: number horizontal, I: number vertical, J: margin left and right, K: margin top and bottom

e.g. 3420 universal labels, 70 x 16.9 mm = 70000, 16900, 0, 0, 3, 17, 0, 4850



Layout Regions

With multi-page projects, it is sometimes a good idea to choose different layout settings, e.g. printer, page size, orientation, paper bin, for the different regions. You define the layout regions with **Projects > Page Setup.** You create a new region with the relevant button on the "Layout" tab and then define the properties for this region. You will find a detailed explanation of the properties in chapter <u>Overview of Properties</u>.

- Active Design Layout
- Practice: Report With Different Page Orientations
- Practice: Managing Issues (Copies)
- Practice: Payment Form on the Last Page



Active Design Layout

If you define different layouts, you can decide which layout is to be displayed as the workspace. The "Active Design Layout" project property lets you choose from all defined layout regions.

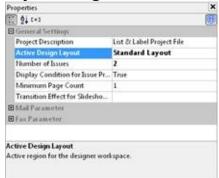


Figure 8.5: Selecting the active design layout



Practice: Report With Different Page Orientations

Let's assume that you want to change the page orientation within a report: the first section with the bar chart is to be printed in portrait mode, the second section with the cross tab in landscape mode and the remaining section with the pie charts in portrait mode again.

The report should look like this:

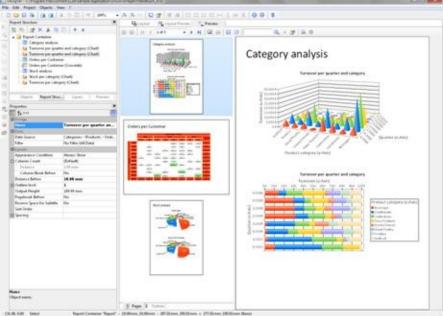


Figure 8.6: Report with different layout regions

Proceed as follows:

- Position the report container on the workspace and create the bar charts, pie charts and the cross tab. You can find more information about this in the chapter <u>Producing</u> <u>Analyses</u>.
- 2. Since you want to change the page orientation, you must also adjust the height and the width of the report

container. If you don to this, the crosstab data will be truncated when the report container is positioned in portrait mode.

Select the report container in the "Report Structure" tool window and use the LL.Device.Page variables and the UnitFromSCM() function to specify the height and width:

Position	[0.39, 0.39, 7.8	0, 11-19 mj	
Left	UnitFromSCM(10000)	[0.39 in]	
Тор	UnitFromSCM(10000)	[0.39 in]	
Width	LL.Device.Page.Size.cx - UnitFromSCM(20000)	[7.47 in]	1
Height	LL.Device.Page.Size.cy - UnitFromSCM(20000)	a	1

Figure 8.7: Size of the report container defined variably

- 3. Create headings by adding a Table > Free Content element in each case.
- 4. Enter "Orders per Customer" as the name of the crosstab's heading. This element name will then be available later in the "LL.CurrentContainerItem" field.
- 5. So that the titles are always printed at the beginning of a page, set the "Pagebreak before" property to "Yes" in each case. This will produce a page break before outputting the element.
- 6. Now define the layout regions. Select Project > Page Setup.
- 7. The dialog for defining the layout will now appear. The default region "Standard Layout" is always the last area with the "True" condition and cannot be renamed. Leave "Portrait" as the orientation for this layout.

a	· + 🛐 🙀 (+3			
Chart Layout	BDesign			
Standard layout	Condition	LL.CurrentCo	entainerBtem=[False]	
	Description	Chart Layout		
	@Printer settings			
	Printer Manye	HP Lasenlet 5		
	E Paper Size	A4	(8.268 in = 11.693 in)	
	Abgnment	Landscape	pe	
	Duplex	not specified		
	Force Paper Size	Yes		
	Number of copies	1		
	Paper Bin	Default		
	Size Adjustment	Keep scale		
	Sort copies	not specified		
	Use Physical Page	Yes		
	Design			

Figure 8.8: Layout definition for a particular element

- 8. Create a new region with the "New" button. Enter a meaningful name for this layout in the "Description" property, e.g. "Landscape".
- 9. Then specify the specific properties for this layout. Change the "Orientation" property to "Landscape".
- 10. You can specify when this layout region is to be used as a "Condition." In this example, the mode must change when the element with the name "Orders per Customer" is printed. The formula for the logical condition is therefore: LL.CurrentContainerItem = "Orders per Customer".



Practice: Managing Issues (Copies)

Let's assume that you want to produce two copies of an invoice. The first copy is to be printed on a company letterhead in paper bin 1. The second copy is to be printed on the cheaper paper in paper bin 2.

- 1. In the project properties, enter "2" in the "Number of issues" property.
- 2. Open the dialog for the report container via **Project** > **Page Setup**.
- Create a new region "Original". In this case, use the IssueIndex() function as the "Condition". This function returns the number of the issue. So you define the logical condition "IssueIndex()=1". Select paper bin 1 for this layout.
- Create a layout called "Copy" and define the logical condition "IssueIndex()=2". Select paper bin 2 for this layout.

	· · · · · · · · · · · · · · · · · · ·		
Original	E Design		
Сөру	Condition	IssueIndex() = 1. (Tri	ie]
Standard layout	Description	Original	
	E Printer settings		
	Ponter Name	HP Lasedet 5	
	Paper Size	A4 (8.268 m × 11.693	(m)
	Alignment	Portrait	
	Duplex	not specified	
	Force Paper Size	Yes	
	Number of copies	1	
	Paper Bin	Default	
	Size Adjustment	Keep scale	
	Sort copies	not specified	
	Use Physical Page	Ves	
	Design		

Figure 8.9: Definition of the layout regions

If you want to endorse the second copy additionally with the text "Copy" over the invoice table, carry out the following steps:

- 1. Create a text object containing "Copy".
- 2. Because objects that are not interlinked are printed first and tables are printed last, the text object must be linked sequentially with the table. Otherwise it would be printed before the table and therefore under the object.
- 3. Select the text object in the object dialog and interlink it with the Report Container. The "Sequential" interlink type is already selected.

Objects			×
Project			
-A [Text]			
-A [Test]			
- 4 [Drawing	1		
a - I Report C	ontainer]		
2 A Co	py notation		
Objects	Report Struct	Layers	Preview

Figure 8.10: Object list with sequential interlinking

4. Now select the text object again and set the value of the "Display condition for issue print" object property to IssueIndex()=2. This text object will now only be printed on the second copy.

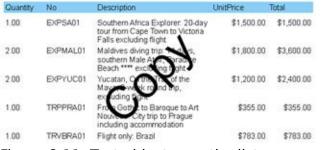


Figure 8.11: Text object over the list



Practice: Payment Form on the Last Page

Let's assume that you want to print a payment form on the last page of a multi-page invoice. The payment form is preprinted on a special paper in paper bin 2 in the lower range of the page. The field contents have to be printed exactly at the correct positions.

Proceed as follows:

- Open the dialog for the layout regions by Project > Page Setup.
- Create a new region "Last page". Use the function Lastpage() as a condition that returns the value "True" if the last page is printed. Choose paper bin 2 for this region so the last page will always be printed on the special payment form preprint from paper bin 2.
- Position the text objects for the field contents of the payment form exactly at the correct position. You can put a scanned-in picture of the payment form in the background by choosing/clicking Objects > Insert > Form Template.
- Set the appearance condition of all text objects to Lastpage() to make sure they are only printed on the last page.
- 5. You have to link all text fields with the table as the function Lastpage() can only be evaluated correctly in tables, layout regions or objects linked to tables. Links are created and edited in the "Object List" dialog. Open this dialog by choosing/clicking Objects > Object List. Select the table object and link via text object of the payment form to it by clicking "Link with...". The interlink

type "Sequential" is already selected. The sequential interlink is sufficient here as no changes in object position or object size are desired.

Repeat these steps for all text objects.

6. Now you have to avoid that the fields of the payment form are printed on top of the table if the table ends on the last page in the area of the payment order. Therefore create a placeholder by inserting a rectangle object without border and filling. The rectangle has the exact height of the payment form and has to overlap the table necessarily! Select the table in the object dialog and link the rectangle to it. Choose "At end, keep size" as interlink type. No matter where the table ends, the rectangle is always output in the selected size after the text. If there is not enough space after the table, the rectangle is output on the next page and a page break is triggered.

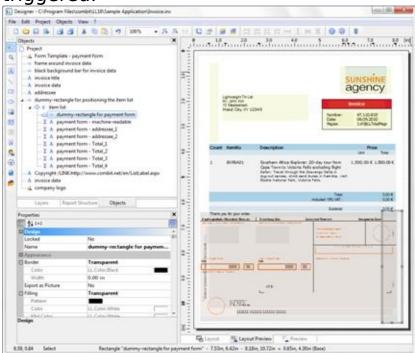


Figure 8.12: Invoice with fields and placeholders for payment order



Output Options

There are two ways of printing projects: Start the print from the higher-level application or via the real data preview in the Designer (if supported by the application).

See also:

- Output Options
- Real Data Preview
- Export in another Format (PDF, XLS ...)
- Print Sample in the Designer



Output Options

You can start the print function directly from the higher-level program, via a menu item or from the preview.

If you start the print from the higher-level program, the print settings dialog will normally appear once you have selected the project to print.

1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -	
Print Target	
Region	Printer
Сору	Microsoft XPS Document Writer
Standard layout	Microsoft XPS Document Writer
Direct to Save options perman	C Preview • Options
Options	
-house	
Issue[s]:	
Issue(s): (Specify issue r	number or number ranges here, comma separated where example "1,3-4")
Issue(s): (Specify issue r	example "1,3-4")
Issue[s]: (Specify issue r applicable. For	example 'L3-4')
Issue(s): (Specify issue r applicable. For Fjrst Page:	example "1,3-4")
Issue(s): (Specify issue r applicable. For First Page: Pages:	example "1,3-4")
Ssue(s): (Specify issue r applicable. For First Page: Pages: @ all @ Bange(s)	example "1,3-4")

Figure 9.1: Output options

You have various configuration options here:

- § Under "Print Target", you can change the printer settings. If you have defined different layout regions in the page layout dialog, you can also change the printer settings here for the various regions. Enter the number of copies under "Issues".
- § Select the output format (e.g. preview, printer, PDF) under "Direct to".
- § The "**Preview**" option lets you view the output first on the

screen as it would be printed. In this way, you can check the result before printing.

- § If you select the "Save options permanently" setting, your choice of printer and output format will be saved as default values for this print template.
- § With "First Page", you specify the page with which the print is to start.
- § "**Pages**" lets you select certain pages or a page range for printing, e.g. 1, 3-4, 10-.
- § The "Print" drop-down list lets you restrict the output to even/odd pages or the pages that you have selected above.
- § When printing labels, you have an extra "Select" button which you can use to specify the position where you want to start printing the sheet of labels.

In this way, you can also print sheets of labels that have already been partly used. You will find a sample label sheet for your label project in the dialog for selecting the start position. Click the label where the print is to start. Please take note of the selected print order. You can print not only in rows from upper left to lower right but also column for column or in the reverse direction. The labels will be printed in the specified direction starting with the selected label.



Figure 9.2: Additional print settings for labels



Real Data Preview

The real data preview function is also available directly in the Designer provided that your application supports it. In this way, you can check the layout of a printout without wasting paper in order to do so. The screen preview is accurate down to the last detail (WYSIWYG � What You See Is What You Get), exactly as it would be when printed. After checking the layout, the actual print can be started from the preview without having to use the print command again.

- § On the "Pages" tab, the individual pages are shown as miniatures to let you navigate quickly and directly. Via Project > Options > Preview, you can specify the maximum number of pages that are to be displayed in the real data preview.
- § You will find the table of contents on the "Outline" tab. With Project > Options > Project, you can specify the maximum folder depth of lines and objects to be used as bookmarks in the preview. Table rows and objects have the "Outline level" property for defining the table of contents.
- § The "Number of copies" specified in the print options is ignored in the preview as this is only relevant for the actual print.
- § The preview window contains a toolbar which you can use to select the different preview functions.

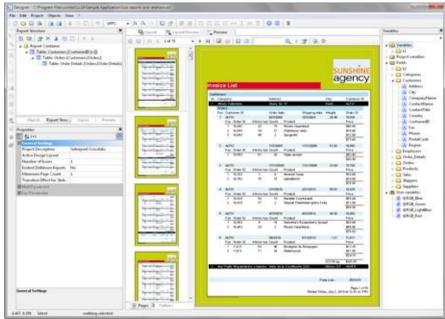


Figure 9.3: Real data preview in the Designer



Export in another Format (PDF, XLS ...)

You can also output a print in different file formats. To do this, choose the relevant output medium in the Print Options > Print Setup dialog, which you reach via File > Export or via "Save as" from the preview. Please consider that, due to the format, the layout cannot always be taken over 1:1 because there are specific restrictions when converting to these formats. Many formats have various possibilities for configuration. You can make use of these by clicking the "Options" button. § For example, with Excel formats, you can reduce the number of columns and rows to the minimum required by means of the "Only data from table object(s)" option.

§ With PDF exports, you will find the option for creating a file that is PDF/A compliant. You will find more information about creating a PDF table of contents in Chapter <u>Levels in</u> <u>the Table of Contents</u>.

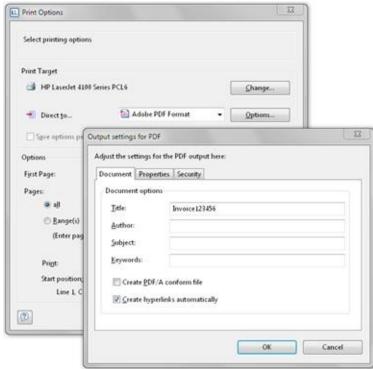


Figure 9.4: Output settings for PDF

When you start the print, choose the storage location in the following "Save As" dialog and enter a name for the file to be created in the "File name" field.

- § By checking the option "Open the file in the registered application after the output", you can display the file after creation directly in the respective program (e.g. Excel).
- § The option "Send exported files by email" lets you send the files directly by email.
- § You can add a digital signature to your files by means of the "Digitally sign created files" option (not available in all applications).

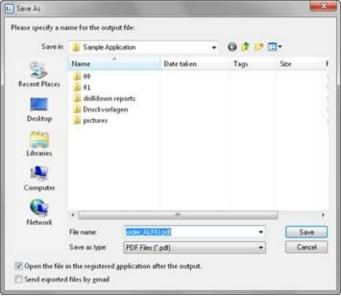


Figure 9.5: Output options



Print Sample in the Designer

Choose **File> Print Sample** for a test print of your current project. In the print sample, fixed text appears as it is defined in the project; variables and fields however are replaced by a predefined sample text or by a single repeated sample data record.



Variables, Formulae and Expressions

Information such as a return address line on an address label or a heading over a list can be entered directly into the project as "**fixed text**". Fixed text is printed exactly as it is held in the project.

Alternatively, this information can be taken over dynamically from a higher-level program. Such information is entered into the projects as "**variables**" or "**fields**".

Fields are the data that changes for each line of a table. Variables remain the same throughout the page. For example, you might choose the TELEPHONE field for the contents of a column of a table. The different telephone numbers for the data records in the database are printed in this column. In other words, variables and fields are placeholders.

With these two kinds of information, attractive projects can already be designed that are adequate for many purposes. However, List & Label Designer offers much more. With the aid of formulas and expressions, the information held in variables and fixed text can be joined or modified in almost any conceivable way. The "**formulas**" and "**expressions**" make this possible. In formulas and expressions, fixed text and variables can be used in "**functions**" and joined by "**operators**".

For example, with projects for printing address labels, you can use an expression to automatically add the text "PO Box" to a PO Box number held in a variable called POBOX. In this way, not just the number alone will be printed on the label but something like "PO Box 111111" instead.

Or, consider this: The net price of an article is held in a variable called PRICE. However, you want to print the price

including VAT in your list. A formula that calculates the VAT from the net price and then adds it on will help you here. The gross price will then be printed.

See also:

- Variable List
- The Elements of an Expression
- Working With Functions
- Working With Operators



Variable List

The variable list displays all variables available in the current project; for list type projects, all available fields are also displayed.

The hierarchical list differentiates between variables, fields, database tables, user variables and sum variables.

User-defined variables and fields can also be structured hierarchically. The contents of variables normally remain unchanged at least throughout a page; fields change from table row to table row.

If you want to assign variables or fields to existing objects, you can simply select the variables and fields that you want in the list and drag them to the object with the mouse (Drag & Drop). List & Label inserts them automatically where possible. If you drag a variable to a free area on the workspace, a new text object will be created there. The size relates to the size of the last object whose size was changed.

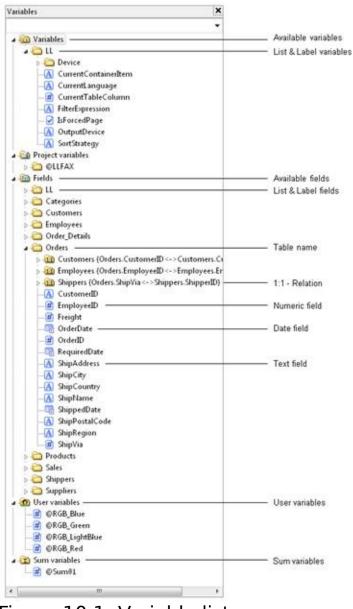


Figure 10.1: Variable list



The Elements of an Expression

Fixed text, variables, formulae, functions, operators and the like � generally referred to below as "Elements of expressions" � are all inserted and combined by means of a common dialog.

The formula wizard helps you with your entries in several ways:

- § Function syntax display: A tooltip appears describing the chosen function; it lists the required parameters and shows the result type.
- § Auto complete: When you type a letter, the available functions, fields and variables are listed that begin with this letter. Within functions, suitable values are suggested for parameters.
- § Syntax coloring: Functions, parameters, operators and comments are shown in different colors.
- § Automatic type conversion: Variable and field types are converted automatically when inserted in existing expressions to ensure that the data type corresponds to that expected.

	 Functions: 	1
Em Fields Em Ends Em A Description1 A Description2 No Who EANL23 Picture WhitePrice	Minc. functions Aggregate functions Barcode functions Barcode functions Barcode (Stimp) (String) A Barcode (Stimp) (String) Barcode (Stimp) (String) CheckMad10 () String)	
Retrieves argument Las a barcode of the [Harcode (Bring):(BB ->(Barcode) Barcode (Item.No.) ->(Bion.tr. ->250ND* ->250NT*		Jmert
"25MATROC" "2-05'5 DataLogic"	+ , n function 'Barcode'	2

Figure 10.2: Autocomplete in the formula wizard

See also:

- Different Expression Modes
- The Tabs
- The Editing Line
- Inserting Variables
- Insert Fixed Text
- Inserting Comments



Different Expression Modes

Please note that there are two ways of writing expressions. Which mode is being used is set by the application. On the one hand, there is the **normal expression mode**, in which you can enter names of variables and functions without brackets. Fixed text must be enclosed in quotation marks. The individual variables must be joined with the "+"

operator.

On the other hand, there is the **extended mode**, in which you can enter fixed text without quotation marks. You must enclose variables with "<" and ">" and functions with chevrons ("«" and "»").

^(**) In this mode, you can insert the chevrons by clicking the "Insert chevrons" button (e.g. if you want to enter a function directly". You can also use ALT+174/175. It is not necessary to use an operator to join individual operators in this mode. The extended mode is easier to use.



Figure 10.3: Extended mode



The Tabs

This dialog consists of a series of tabs, each containing different elements to be edited.

Tab	Contains the elements
Variables and Functions	The variables and functions available for this object type
Condition	Special dialog for defining IF-THEN-ELSE conditions
Text	Dialog for entering fixed text and options for setting tab stops (only text objects)
Date Format	Available date formats
Number Format	Available number formats
Operators	Available logical operators

You will find an "Insert" button on each of these tabs with which you can add the selected element to the editing line. A double-click on the relevant element has the same effect. In addition, you can also add the elements to the editing line with Drag & Drop (also in and from the function list). For the different elements of an expression (variables, text, functions etc.), certain rules apply for the way of writing and for joining individual elements to give an expression. A wizard integrated in the dialog makes sure that these rules are observed. For this reason, you should always add the different elements to the editing line by means of the respective tab in this dialog. Thus, you should use the "Variables and Functions" tab to enter variables and the "Text" tab to enter text etc.

Experienced users can also enter the expression that they want directly in the edit box or modify the text that is there (e.g. put something in brackets).



The Editing Line

The editing line contains the expression that you have compiled by means of the various tabs, entered directly or created with Drag & Drop.

The expression is checked continuously as you create it to make sure that the syntax is correct. Any syntax errors are shown in the information pane under the editing line, together with an explanation of the cause of the error. The syntax checker will normally produce an error until the expression is complete. Don't let this worry you. When the expression is complete, the resulting text should be shown with the Designer's sample data.

To make complex expressions clearer, you can split them across several lines with CTRL+RETURN. This has no effect on the result.

With the different buttons on the right next to the input field you can

§ mark brackets belonging to the formula expression.

§ mark the expression between matching brackets.

§ edit sum and user variables.

§ undo the last operation.

§ redo the last undo operation.



Inserting Variables

There are different data types for variables: "string", "number", "date", "Boolean" (logical values), "picture" and "barcode". The data type is important if you want to use variables as parameters in functions as they normally only accept certain data types. Thus you can only multiply a numeric value with a numeric value.

The "Variables and Functions" tab includes an overview of all available variables, an icon indicating the data type in front of the variable as well as the available functions.

You can filter the variables by means of the input field above the variable list.

To add a variable, double-click the variable that you want, use the "Insert" button or drag the variable to the editing line (also via the function list). The variable in question will be added to the editing line in the correct syntax.

Repeat the above steps to add more variables to your expression. If you want to have spaces between the individual variables, e.g. to separate FIRST_NAME and NAME, make sure that you enter this space in the editing line.

		195
200.01	 Functions: 	1
D Customer	*	
> 😂 Address	Numerical functions	1
- Company	Athematical functions Date functions	
A Lastname	I String functions	1
A Salutation	Misc. functions	
Title	Aggregate functions	
Project variables	Barcode functions	
Customer.Title +*** + Customer.Firs	stname • ** + Customer_Lastname	the of
		Σ_{Σ}

Figure 10.4: Joining variables and free text

You can also insert variables by "dragging" the one you want to the target object in the workspace with Drag & Drop. The variable is then automatically added to the object as a new line.



Insert Fixed Text

Another important element in expressions is fixed text, with which you can prefix a variable with an identifier, e.g. "Telephone: 1234567".

With the "Text" tab, you can insert free text in your expression, set tab stops and page breaks.

Enter the text that you want and click "Insert" to add your entry to the editing line. The text will be placed automatically in quotation marks.

Variables and Functions Condition Test Date Format Number Format	Operators
Name:	
e Linefeed Iab Properties	,
Lowereen Law Properties	Inset
	function
"Name." + Customer.Title • ** • Customer.Firstname • ** • Customer.La	
"Name." + Customer.Title + "" + Customer.Firstname + "" + Customer.Lar	ht (

Figure 10.5: Fixed text in the formula wizard

In the above example, the fixed text "Name" is first inserted via the "Text" tab followed by the variables "Customer.Title", "Customer.Firstname" and "Customer.Lastname" by means of the "Variables and Functions" tab. The "Name:" text will then be printed first followed by title, first name and last name from the database. Please consider that spaces that are to appear between variables or between variables and text, e.g. as separators, also count as "fixed text".

Depending on the mode, variables and fixed text cannot be simply placed together but must be joined by the "+" joining operator. In this example, the fixed text "Name" is joined to the "Customer.Title" variable with the "+" operator.

See also:

Inserting Linefeed

Inserting Tab Stops



Inserting Linefeed

With the "Linefeed" button, you insert a line break ("¶") into your text line.

Linefeed

Figure 10.6: Insert line break by clicking the button

However, such a break only has an effect if you have specified that breaks are allowed for the object in question (line of a text object or column of a table object). In this case, the words that don't fit in the line/column are continued on a new line causing the lines below to be moved down by one line. (Caution: If the text contains just one long word, it will not be broken but will be truncated instead). With text objects, the value of the "Line Wrap" property in the property list for the respective paragraph must be "Wrap".

With table objects, the value of the "Fit" property in the property list for the respective column must be "Wrap".



Inserting Tab Stops

Tab stops are only allowed in text objects. Therefore this button is not displayed in table objects.

Figure 10.7: Insert tab stop by clicking the button

Just one tab stop per line may be inserted between text elements or variables. Its position can be specified in the "Tab Properties" dialog which appears when the button is pressed. You will find more information under <u>Text Objects</u>.



Inserting Comments

You can add comments to formulas provided that this is supported by the application. You can do this in two ways: § "/* <text> */" for comments in the middle of a formula § "/* <text>" for comments at the end of the formula. With this variation, all the remainder of the formula becomes a comment � not just the line.



Figure 10.8: Inserting comments



Working With Functions

Functions open up countless possibilities and make defining expressions really interesting. With the aid of these functions, you can calculate values, influence the results of variables or their appearance, convert value types and perform many more tasks.

See also:

- Notation of Functions
- Value Types
- Overview of the Functions
- Using Functions



Notation of Functions

Functions all use the same notation which is based on the BASIC programming language:

return value = function(arguments)

You only specify the function and the arguments. Whether you use capitals or small letters is irrelevant for the names of the functions, but not for the arguments. Capitals or small letters are especially important when variables are used as arguments.

List & Label evaluates the expression and interprets it replacing the "function(arguments)" part by the "return value".

I.e., the return value.is produced from the **function(arguments)** input line.

The elements have the following meanings:

Element	Meaning
Function()	The name of the respective function in its correct syntax. The brackets () for the arguments belong to the function name. The brackets must always be present even if a function does not have any arguments.
Arguments	The values that a function uses in order to produce the return value. The arguments follow the name of the function immediately without any spaces in between. A function can have zero, one or more than one argument(s). Functions usually expect arguments of a certain value type (see below). It is important that the value types of the arguments conform to the types expected by the function.
Return value	The value that List & Label returns as the result of a function. The type of the return value depends on the function in question or the value types of the arguments.



Value Types

Value type	Explanation
Boolean	The logical values "True" or "False". If the condition is met, the result is true otherwise false.
String	Any string. This string can contain letters, digits and special characters. It must be placed in quotation marks ("") so that it can be differentiated from names of variables.
Date	Date values according to the Julian calendar.
Number	A string containing only the digits 0 - 9, the decimal point and the minus sign, other characters are not permitted. Number strings do not have to be enclosed in quotation marks.
Barcode	A string that is made up of the characters used for barcodes.
Picture	One of the supported picture formats.
RTF	Formatted text



Overview of the Functions

You will find an overview of the available functions on the "Variables and Functions" tab. You will also see an explanation of the currently selected function. The explanation informs you of the nature of the function and the type of the arguments that it expects (parameters). If no arguments are given for a function, this means that the function does not expect any (apart from the empty brackets). Otherwise, the function expects exactly the number of arguments as shown. Arguments that appear in square brackets ([]) are optional, i.e. they can be omitted. The argument "All" means that the argument can be any of the following types (Boolean, string, date, number, picture, barcode, RTF).

You will find a more detailed explanation of all functions and parameters under <u>Overview of Properties</u>.

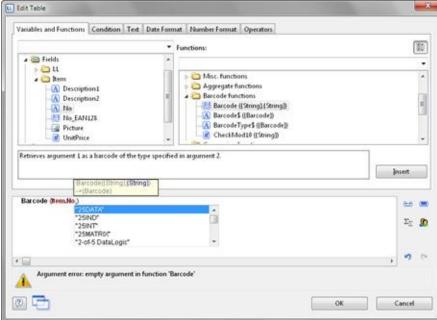


Figure 10.9: List of functions with help text and tooltip

The functions available on the "Variables and Functions" tab are sorted alphabetically or shown by function group. There are the following function groups:

§ Numerical functions

§ Mathematic functions

§ Date functions

§ Character functions

§ Miscellaneous functions

§ Aggregate functions

§ Barcode functions

§ Conversion functions

§ Binary functions

§ Drawing functions

§ Project and print-dependent functions

§ Logical functions

§ Currency functions

You can filter the functions by means of the input field above the list of functions.

When you select a function, you are shown a short explanation in the information pane at the bottom. To add the function to the editing line, double-click the function that you want or use the "Insert" button.



Using Functions

Examples of some selected functions are explained below.

See also:

- Convert a Number to a String With Str\$()
- Convert a String to a Number With Val().
- Convert a String to a Date With Date()
- Convert a String to a Barcode With Barcode()
- Convert a String to a Picture With Drawing()
- Truncate Strings
- Formatting a Data Value With Date\$()
- Formatting a Number Value With FStr\$()
- Formatting Currencies With LocCurrL\$() or LocCurr\$()
- Page Numbers With Page\$() or Page()
- Counting Values With Count()
- Only Count Certain Values With Countlf()
- Totaling With Sum()
- Obtaining User Input via a Dialog With AskString\$()
- Appearance Conditions With Lastpage()
- Logical Conditions With Cond()
- Working With Null Values



Convert a Number to a String With Str\$()

The Str\$ function converts a number to a string.

The (optional) second parameter specifies the length of the string. However, if this number is too large for this format, the resulting string may be longer than you want. If the number is too small, the result will be padded with spaces, according to the sign, on the right (negative) or the left (positive).

The (optional) third parameter specifies the number of decimal places. If it is positive, the number is shown as a floating point number or in scientific notation if it is negative. If the third parameter is not specified, the number of decimal places is set to 0 with integers and, for reasons of compatibility, to 5 with floating decimal values. Examples:

Str\$ (Constant.Pi())	result: "3.14159"
Str\$ (Constant.Pi(),3)	result: "3.14159"
Str\$ (Constant.Pi(),3,0) result: " 3"
Str\$ (-Constant.Pi(),12	(,-3) result: "-3.141e+00"
Str\$ (Page())	result: "5.000000"
Str\$ (Page(),10)	result: "5"
Str\$ (Page(),-10)	result: "5 "

To enter the expressions, proceed as follows:

- 1. Locate the function in the alphabetical list or filter the functions by entering Str\$ in the filter field above the list.
- 2. Double-click the "Str\$ ()" function to add it to the editing line. This also adds placeholders for the parameters that are expected or accepted by the function. The first of these parameters is selected automatically and you will be prompted to replace this placeholder with a valid value. It's a good idea to first replace all the function's

parameters with the respective values before you go on to define the expression.

Functions are also accepted as values for most parameters. List & Label Designer takes care of the correct syntax provided that you also use the "Functions" tab for entering them.

- 3. The "Constant.Pi()" function was inserted here as a parameter {number}.
- 4. To insert a variable and a function at the same time, use the mouse to drag the variable that you want to the relevant function folder (e.g. "Numerical functions"). The folder opens up automatically and you can select the function that you want with the mouse. If you move the mouse upwards or downwards, the list scrolls automatically in the corresponding direction. If you "drop" the variable onto a function, the function will be inserted with the selected variable as the first parameter.



Convert a String to a Number With Val().

The Val() function converts a string to a number. If there is an error, the result will be 0. The decimal point character must always be given as "."

Example:

Val ("3.141") result: 3.14

The "LocVal()" functions converts a string to a number and presents the result in a format that is valid for the country. Examples:

LocVal ("12.00","DEU") LocVal ("12.00","USA") result: 1200,00

result: 12,00



Convert a String to a Date With Date()

The Date() function converts a string to a date. When doing so, the separator is evaluated accordingly: Example: Date ("04.07.1776") result: 04.07.1776



Convert a String to a Barcode With Barcode()

The Barcode() function converts a string to a barcode. This function can only be used in a table or barcode object. For the second parameter, the wizard offers you the possible barcode types as autocomplete options. Some barcodes have special formats which must be adhered to. You will find detailed information about this in chapter <u>Barcode Objects</u>. Example:

Barcode ("Hello World", "GS1 128")



Convert a String to a Picture With Drawing()

The "Drawing()" function converts a string to an image file. Example:

Drawing("sunshine.gif")



Truncate Strings

The "Left\$()" function shortens a string from the left by a specified number of characters. The "Right\$()" function shortens a string from the right and the "Mid\$()" function cuts out part of the string.

The second parameter specifies the maximum number of places in the result.

Examples:

Left\$ ("combit", 1) Mid\$("combit",1,2) Right\$("combit",3) result: "c" result: "om" result: "bit"

The "StrPos()" and "StrRPos()" functions return the position of the nth occurrence of a search string in a string. You can supply a third parameter specifying which occurrence of the search string is to be returned. The first character of the string corresponds to position 0. This means that with this function, you can extract a substring from the string, e.g. from the first space onwards.

Example:

Left\$ ("John Smith",StrPos("John Smith"," ")) result: "John"

The "Rtrim\$()" function removes spaces at the end of a string, the "Atrim\$()" function removes them from the beginning and the end of a string. Example:

RTrim\$ ("Hello World ") result: "Hello World"



Formatting a Data Value With Date\$()

You can format date values with the Date\$() function. To avoid having to enter the formatting parameters yourself, you can select the parameters from a list on the "Date Format" tab.

In this way, for example, you can specify whether days or months are to be written in words or whether the year is to be output with two or four digits etc. You should generally proceed by first selecting the date format that you want from the "Date Format" tab and then insert the value to be formatted or the expression as a parameter.

In the format list, you will see the respective formatting instructions on the left and the respective result on the right. The "Now()" function, which returns the current date, is selected automatically as the date value. But if you want to format a different date value, simply replace "Now()" with the value that you want. You will find more information about the "Date\$ ()" function under <u>Overview of Functions</u>.

Vanables and Functions Condition Test Date Fo	ormat Number Format Operators	
Format	Result	
"Sed SUM Sky" "Sey, SUM Sed." "Sed SUM, Sky" "Sed. SuM Sky"	6 July 2010 2010, July 6 6 July, 2010 6 July, 2010	
TSD, Sold SAN SAN TSD, Sold SAN SAN TSD, Sold SAN SAN TSD Sold SAN SAN TSD, SAN SAN SAN TSD, SAN SAN SAN TSD, Sand San San TSD, Sand de San de San	Tuesday, 6, July 2010 Tuesday, 6 July 2010 Tuesday, 6 July 2010 Tuesday, 6 July 2010 Tuesday, July 6, 2010 Tuesday, July 6, 2010 Tuesday, July 6 2010	
Choose a format for your date		Insert
Date\$9iow(),"%D, %d. %M %y)		an a Z _Z j
Tuesday, 6. July 2010		,

Figure 10.10: Example with Date\$()



Formatting a Number Value With FStr\$()

You can format numerical values with the FStr\$() function. To avoid having to enter the formatting parameters yourself, you can select the parameters from a list on the "Number Format" tab.

In this way you can specify the number of positions before and after the decimal point, leading zeros and similar. You should generally proceed by first selecting the number format that you want from the "Number Format" tab and then insert the value to be formatted or the expression as a parameter. With complex expressions containing calculations, make sure that you format the result and not a value in the calculation formula. Otherwise you will not be able to perform the calculation.

In the format list, you will see the respective formatting instructions on the left and the respective result on the right. You can add the number format that you want to the editing line by double-clicking or with "Insert".

Insert the number value to be formatted as a parameter. You will find more information about the "FStr\$ ()" function under <u>Overview of Functions</u>.

	t Date Format Number Format Operators	
Format	Result	
"?&,###"	3141.59	100
*-32LBB	3141.59	6
**?\$2.mm*	+3141.59 3141.59	
(m?(t),mm)	5141.59	_
-\$\$\$\$\$\$0.00	\$3141.59	
**\$\$\$\$\$\$22==*	+\$3141.59	
"(\$\$\$\$\$ <i>2,mm</i>)"	\$3141.59	
"#####################################	003141.59	
"διδιδιδι.mmm" "-διδιδιδι.mmm"	3141.593 3141.593	
Choose a format for your numerical value		
Fstr \$(tem.UnitPrice,"\$?&##")</th><th></th><th>0.0 Q</th></tr><tr><th></th><th></th><th>IT I</th></tr><tr><td></td><td></td><td>5</td></tr><tr><td>\$1508.00</td><td></td><td>5</td></tr></tbody></table>		

Figure 10.11: Example with Fstr\$()



Formatting Currencies With LocCurrL\$() or LocCurr\$()

The "LocCurrL\$()" function returns a string with the currency format and symbol that is valid for the country. Insert the number value to be formatted as a parameter. The (optional) second parameter is the ISO 3166-

Country code for the country whose currency format is to be used.

Example:

LocCurrL\$ (1000) result: "\$1,000.00 "

The "LocCurr\$()" function returns a string with the currency format that is valid for the country but without the currency symbol.

Example:

LocCurr\$ (1000) result: "1,000.00" You will find more information about this function under <u>Overview of Functions</u>.



Page Numbers With Page\$() or Page()

The "Page\$()" function returns the page number of the page being printed as a string.

Example:

"Page " + Page\$ () + "/" + TotalPages\$ () result: Page 1/3

The "Page()" function returns the page number of the page being printed as a number, thus giving you the opportunity of using an expression or a formula, e.g. in conditions. Example:

```
Cond(Page()>1, "Page " + Page$())
```



Counting Values With Count()

The "Count()" function counts the number of values of the first argument.

The first parameter specifies the values to be counted. The (optional) second parameter specifies whether the values used for the calculation are to be deleted after outputting. Examples:

```
Count (Order_Details.ProductID)
Count (1)
```



Only Count Certain Values With Countlf()

The "Countlf()" function counts the number of values that satisfy the condition. You should also use the "Distinct()" function if multiple occurrences of values are only to be counted once. The first parameter specifies the expression for the compare. The (optional) second parameter specifies whether the values used for the calculation are to be deleted after outputting (default: True).

Examples:

Countlf (Customers.Region="D")

Countlf (Distinct(Customers.Region="D")) multiple occurrences of values are only counted once Countlf (IsNull (Orders.OrderDate)) counts all values whose content is empty



Totaling With Sum()

The "Sum()" function adds up the values of the first argument.

The (optional) second parameter specifies whether the values used for the calculation are to be deleted after outputting (default: True).

Example:

Sum (Order_Details.UnitPrice)



Obtaining User Input via a Dialog With AskString\$()

You can use the "AskString\$()" function to obtain information from the user during the print process. A dialog appears when printing in which the user is required to enter the information that you need.

The first parameter contains the text for the request that is to appear in the dialog.

With the second parameter, you can specify whether the user request is only to appear once when printing starts (default: False), or whether the information is to be requested for each individual data record (True).

The third parameter contains the string that you want to display as a recommended value in the dialog's input field. The last parameter specifies the maximum number of characters that the user may enter. Example:

AskString\$ ("Insert Subject", False, "Your request from " + Date\$(Now()))

ser Input		
Insert Subject		
Your request from 16/0	7/2010	
	K OK For All	

Figure 10.12: Input dialog with AskString\$()



Appearance Conditions With Lastpage()

The "Lastpage()" Boolean function returns whether the actual page is the last page, i.e. the result of this function is "True" or "False".

This function can only be used in footer lines of tables, as a condition in a layout region condition, or in objects linked to a table. In all other cases, Lastpage() is always "False".



Logical Conditions With Cond()

The "Cond()" and "If()" functions let you formulate a wide variety of conditions. The first argument is a logical expression that is evaluated for truth. If the expression is true, the second argument is returned as the result. If the expression is false, the third argument is returned as the result.

A simple example: Let us assume that you want to output the total of the article prices on a page in an invoice footer line. You also want to output the grand total of article prices on the last page.

Edit Table		
Variables and Functions Condition Test Date Format Number Format Operators		
Condition		
nables and Functions Condition Test. Date Format Number Format Operators Condition not Lastpage() Expression if condition is True Sum(Item.UnitPrice, True) Expression if condition is False Sum(Item.UnitPrice, False) Example: Sum(Item.UnitPrice, True), Sum(Item.UnitPrice, False)) Example: Sum(Item.UnitPrice, True), Sum(Item.UnitPrice, False))	(dit	
Expression if condition is True		
Sum(Item.UnitPrice, True)		idit
Expression if condition is False		
Sum(Item.UnitPrice, False)		Édit
Cond(notLastpage(),Sum(Item.UnitPrice, True),Sum(Item.UnitPrice, Faise))		joset 0.0 0 2 ₂ j
		, ,
•		,

Figure 10.13: Example of the use of Cond()

You enter this function in the formula wizard either directly in the editing line or via the "Condition" tab.

1. As a "condition" (1st parameter), enter the expression that is to be tested for TRUE or FALSE. In our example, we want to use the "not Lastpage()" function to determine whether this is the last page.

- 2. As the "expression, if condition is TRUE" (2nd parameter), enter the expression that is to apply if the above condition is TRUE. In our example, the condition is true if this is not the last page and in this case the page total is to be output (parameter of the "Sum()" function is "True").
- 3. As the "expression, if condition is FALSE" (3rd parameter), enter the expression that is to apply if the above condition is FALSE. In our example, the condition is false if this is the last page and in this case the grand total is to be output (parameter of the "Sum()" function is "False").
- 4. If you have entered this function via the "Condition" tab, you can use the "Insert" button to add the finished condition to your editing line.



Working With Null Values

There are various functions for working with Null values (undefined field contents).

If there are Null values in an expression, the entire expression can become Null. To prevent this, use the "NullSafe()" function for fields that could be empty (e.g. salutation or title). This function checks to see whether the field value is Null and returns a substitute value if this is the case; otherwise, it returns the result of the expression. Example:

Cond (Empty(COMPANY),NullSafe(SALUTATION) + " " + FIRSTNAME+ " " + NAME)

With the "IsNull()" function, you can check whether the value passed to the function or the result of the expression is Null, i.e. an empty field.

Example:

Cond (Empty(COMPANY) or IsNull(COMPANY),FIRSTNAME + " " + NAME)

You can set a Null value with the "Null()" function.



Working With Operators

Open the list of available operators by clicking the "Operators" tab. The operators are used to join variables and free text to give more complex conditions and to perform compares or calculations.

On the "Operators" tab, you will find the respective operators in the column on the extreme left; the syntax is shown in the middle column and the value types with which the respective operator can be used in the right column.

Operators join two or more values or variables to give a new value. In this way, you can formulate arithmetic expressions (basic arithmetic operations) or logical expressions. The value type of the result of an expression depends on the value type of the individual elements of the expression. The "+" operator has a special meaning. It is not only used for additions ("Number" and "Date" (value types) but also for joining strings and fixed text ("string" value type).

You can combine multiple operations in one expression. The normal rules of precedence apply in this case: Logical operators are evaluated before arithmetic operators which are evaluated before relational operators. You must use brackets if you want a different processing order. The "innermost" brackets are evaluated first.

i ne genera	i processing nierarc	ny I
Priority	Operator	
1	brackets ()	
2	functions	

2	functions
3	logical operators
4	arithmetic operators
5	relational operators

	unctions Condition Test Date Forma		
Operation	Syntax	Applicable to	
+	«Value» = «Value»	Number, String, Date	- 54
	<value>* <value></value></value>	Number	
0	<value> - <value></value></value>	Number, Date Number	
n.	<value> / «Value> «Value> % «Value></value>	Number Number	- 11
	<value> 56 <value> <value> m <value></value></value></value></value>	All	- 18
-	<value> = <value> </value></value>	All	
<	<value> < <value></value></value>	Number, String, Date	
	<value> <= <value></value></value>	Number, String, Date	
2	<value> <value></value></value>	Number, String, Date	
20	<value> >= <value></value></value>	Number, String, Date	
Order_Detail	s.Ouantity * Order_Details.UnitPrice	-h.	1.0
		Σ	z 1
			, (

Figure 10.14: Example of multiplying two fields

See also:

- Arithmetic Operators
- Relational Operators
- Logical Operators



Arithmetic Operators

The familiar rules of precedence ***** multiplication and division before addition and subtraction *****- apply for arithmetic operators. The "Modulo" operator is evaluated first followed by multiplication/division ("*" and "/") and finally addition and subtraction ("+" and "-"). Example: NFT_PRICE+(NFT_PRICE*0.19)

Operator	Meaning	Data types	
+	addition	string, date, number	
-	subtraction	date, number	
*	multiplication	number	
/	division	number	
%	modulo (remainder with division)	number	



Relational Operators

Relational operators consist of two values of the same data type which are compared with one another returning a true/false value. The result (return value) is the Boolean value true or false. Example: Page()<>1

		U
Operator	Meaning	Data types
>	greater	string, number, date
>=	greater or equal	string, number, date
<	less than	string, number, date
<=	less than or equal	string, number, date
=	equal	string, number, date
==	equal	string, number, date
<>	not equal	string, number, date
!=	not equal	string, number, date



Logical Operators

A logical operator is a function that returns a true/false value. The result (return value) is the Boolean value true or false.

Depending on the type of logical operator, the compound expression is true if both joined expressions are true (AND conjunction) or if at least one of the joined expressions is true (OR conjunction).

The rules of precedence are: Negations are evaluated first, then the "logical AND" and finally the "logical OR".

Example: Zip code >=70000 AND zip code <=80000

Operator	Meaning	Data type
NOT or .NOT.	negation	Boolean
AND or .AND.	logical AND	Boolean
OR or .OR.	logical OR	Boolean
XOR or .XOR.	logical exclusive OR	Boolean



Overview of LL Variables and LL Fields

List & Label provides several variables and fields automatically � depending on the application. You will find the variables and fields in the "LL" sub-folders in the variable list.

See also:

- Overview of Variables
- Overview of Fields



Overview of Variables

Name	Explanation
LL.Color.*	Color value of the corresponding color.
LL.Device.Name	Identification of the output device (printer).
LL.Device.Page.Name	Designation of the selected paper size (example "A4").
LL.Device.Page.Size.cx	Physical page width of the output device in the project s unit of measure. Can be used in formulas to adjust the size of objects dynamically to fit larger output formats. Example: In the property list, set the Position.Left property to the value 0 and the Position.Right property to LL.Device.Page.Size.cx. The object will now always occupy the total page width.
LL.Device.Page.Size.cy	Physical page height in the project so unit of measure.
LL.Device.PrintableArea.Offset.cx	Width of the unprintable left margin in the project s unit of measure. Example: In the property list, set the Position.Left property to the value LL.Device.PrintableArea.Offset.cx. The object will now always be positioned precisely at the left margin of the printer's printable area.
LL.Device.PrintableArea.Offset.cy	Height of the unprintable top margin in the project s unit of measure.
LL.Device.PrintableArea.Size.cx	Printable page width in the project s unit of measure.
LL.Device.PrintableArea.Size.cy	Printable page height in the project sunit of measure.
LL.CountData (not with multiple tables)	Number of data records transferred by the program. This number also includes the data records that were not printed because of their filter condition. This counter is incremented for each data record.

LL.CountDataThisPage (not with multiple tables)	Number of data records transferred by the program on the current page. This number also includes the data records that were not printed because of their filter condition. This counter is incremented for each data record.
LL.CountPrintedData (not with multiple tables)	Number of records actually printed.
LL.CountPrintedDataThisPage (not with multiple tables)	Number of data records actually printed on the current page.
LL.CurrentContainerItem	Value of an element's "Name" property in the report container. Is used e.g. for display and layout region conditions.
LL.CurrentLanguage	Returns the print language, e.g. "en- EN"
LL.CurrentTableColumn	Returns the index of the current column in the case of multi-column projects.
LL.FilterExpression	Selected project filter.
LL.IsForcedPage	Specifies whether the page was forced due to the "Minimum page count" project property.
LL.OutputDevice	Output medium. Can be used e.g. for formatting objects for output in a particular way (e.g. "HTML", "RTF", "PDF" etc.
LL.SortStrategy	Sort order selected by the user.
@LLFAX.RecipName	Fax dispatch: Recipient name
@LLFAX.RecipNumber	Fax dispatch: Recipient's fax number
@LLFAX.SenderBillingCode	Fax dispatch: Sender's billing code
@LLFAX.SenderCompany	Fax dispatch: Sender company
@LLFAX.SenderDept	Fax dispatch: Sender department
@LLFAX.SenderName	Fax dispatch: Sender's name



Overview of Fields

Name	Explanation
LL.ChartObject.ArcIndex (only with pie charts)	Index of the pie segment. The pie segments are sorted by size. The largest pie segment has index 1.
LL.ChartObject.ArcPerc (only with pie charts)	Size of the pie segment in percent.
LL.ChartObject.ArcTotal (only with pie charts)	Absolute value of the entire data volume.
LL.ChartObject.ArcTypelsOthers (only with pie charts)	True, if the current pie segment is the "other" segment.
LL.ChartObject.ArcValue (only with pie charts)	Absolute value of the current pie segment.
LL.ChartObject.AxisCoordinate (only with charts)	Coordinate value (can be used in axis labels).
LL.CurrentRelation (only with multiple tables)	Description of the current relationship to the higher-level table.
LL.CurrentSortOrder (only with multiple tables)	Description of the current sort order in the table.
LL.CurrentTable (only with multiple tables)	Identification of the table currently in use.
LL.CurrentTablePath (only with multiple tables)	Identification of the table currently in use (hierarchically with higher-level tables), e.g. Customers.Orders.Order_Details.
LL.FCountData	Number of data records transferred by the program. This number also includes the data records that were not printed because of their filter condition.
LL.FCountDataThisPage	Number of data records transferred by the program on the current page. This number also includes the data records that were not printed because of their filter condition.
LL.FCountPrintedData	Number of records actually printed.
LL.FCountPrintedDataThisPage	Number of data records actually printed on the current page.
LL.Relations.* (only with multiple tables)	Available relationships.

Available tables.

LL.Tables.* (only with multiple tables)



Overview of Functions

All functions available in List & Label are listed here alphabetically. In the formula wizard, you also have an additional list sorted by functional group.

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) = 3Abs(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.



AddHours

Purpose:

Adds the given number of hours to the date, or subtracts the number of hours when a negative value is entered.



AddMinutes

Purpose:

Adds the given number of minutes to the date, or subtracts the number of minutes when a negative value is entered.



AddMonths

Purpose:

Adds the given number of months to the date, or subtracts the number of months when a negative value is entered.

AddSeconds

Purpose:

Adds the given number of seconds to the date, or subtracts the number of seconds when a negative value is entered.



AddWeeks

Purpose:

Adds the given number of weeks to the date, or subtracts the number of weeks when a negative value is entered.

AddYears

Purpose:

Adds the given number of years to the date, or subtracts the number of years when a negative value is entered.

Alias\$

Purpose: Returns the value that is specified for the key (first parameter) in the key/value-pairs (second parameter). Parameter: String Expression for the value to be searched. String List of values (Form: <key=value>| [<key=value>]. To be able to use "|" or "=" in the value or key, place a "\" infront of it. String (optional) Default if the value cannot be found. Return value: String Example: Alias\$("USA", "DEU=Deutschland|USA=United States of

Alias\$("USA", "DEU=Deutschland|USA=United States of America|GB=United Kingdom") Result: United States of America

ArcCos

Purpose: Calculates the arccosine of the value. Parameter: Number Value Number (optional) Mode (0=Degree, 1=Radian). Default: 0. Return value: Number Example: ArcCos (0) Result: 90

ArcSin

Purpose: Calculates the arcsine of the value. Parameter: Number Value Number (optional) Mode (0=Degree, 1=Radian). Default: 0. Return value: Number Example: ArcSin (0.5) Result: 30,00

ArcTan

Purpose: Calculates the arccotangent of the value. Parameter: Number Value Number (optional) Mode (0=Degree, 1=Radian). Default: 0. Return value: Number Example: ArcTan (1) Result: 45,00

Asc

Purpose: Returns the ASCII-Code of the first character of the string. **Parameter:** String **Return value:** Number

Example:

Asc("A") Result: 65

AskString\$

Purpose:

With this function, information can be requested from the user during printing. A typical example of use for this function would be in a project for a bank transfer form. Information that remains constant, such as name and bank details of the sender, can be integrated directly into the project as fixed text or variables. The transfer amount, however, will almost always be different. With the function AskString\$(), this information can be requested from the user during printing.

At print time, a dialog will appear in which the needed information can be entered.

The dialog allows the entered value to be carried over. Abort with "Cancel".

With the button "All", the entered value will be automatically used for all future result for the AskString\$ function during thus print job. This is useful when the value remains constant over all records.

Parameter:

String The first parameter contains some descriptive text that will appear in the dialog. Since this is a formula, fixed text must be entered in quotation marks, for example "Transfer amount: **?**. This first parameter must be entered, all remaining parameters are optional. If no other parameter(s) is/are entered, the first string is also the default setting for the user input.

- **Boolean** (optional) The second parameter allows you to define whether the dialog should be shown once prior to printing (default, FALSE), or if the dialog should be shown for each record (TRUE).
- **String** (optional) The third parameter contains the string that appears as the recommended value for the user input. Since this is a formula, fixed text must be entered in quotation marks,

for example "50.00 USD".

Number (optional) The last parameter defines the number of characters that can be entered by the user. A value of 16, for example, allows the user to enter a maximum of 16 characters.

Return value:

String

Example:

AskString\$("Transfer amount",True,"50.00 USD",16) Opens a dialog with the title "Transfer amount, a recommended value of "50.00 USD, and a maximum of 16 characters. Since the second parameter is TRUE, the dialog will be shown for each record to be printed.

AskStringChoice\$

Purpose:

Prompts the user to choose a value for the specified variable from a combobox at print time.

Parameter:

- **String** Text, which is displayed and should specify what is to be entered.
- **Boolean** (optional) Sets whether the dialog should be shown once prior to printing (default, FALSE), or if the dialog should be shown for each record (TRUE).
- **String** (optional) The combobox entries. The single entries of the combo box are separated by "|". If one of the entries is '***' (three asterisks), the text is editable. So a new value which may be different from the list items can be entered.

Number (optional) Maximum length (Default: 8192 characters).

Return value:

String

Example:

AskStringChoice\$ ("Document type".F.,"Offer|Invoice|Delivery note|***)

ATrim\$

Purpose: Removes spaces from the beginning and end of a string. Parameter: String Return value: String Example: Atrim\$(" combit GmbH ") Result: combit GmbH

Avg

Purpose:

Generates the mean of the set of values that is produced by the first argument.

Parameter:

Number Expression of the value to be calculated.

Boolean (optional) TRUE: After the output, the values which were stored for the calculation are deleted. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

Avg(Order_Details.Quantity*Order_Details.UnitPrice)

Barcode

Purpose:

This function converts a string to a barcode.

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 Supported Barcode.

Return value:

Barcode

Example:

Barcode(Upper\$(Name),"3of9")

Barcode\$

Purpose: Returns the text contents of a barcode. **Parameter:** Barcode **Return value:** String Example: Barcode\$(BC_3OF9) Result: "Item 4711"

BarcodeType\$

Purpose: Returns the type of the barcode as a string. Parameter: Barcode Return value: String

BasedStr\$

Purpose: Returns the value to any radix. **Parameter:** Value. Number Radix (2 to 36). Number (optional) Minimum length of the string Number (without optional prefix). 0 for the minimal length (Default). (optional) Defines if a prefix ('0b' for radix Boolean 2, '0o' for radix 8, '0x' for radix 16) is inserted before the string (Default: False). **Return value:** String **Example:** BasedStr\$ (1,2,1,True) Result: 0b1

BinaryAND

Purpose:

Links the two (integer) parameters binary with 'and' and returns the result.

Parameter:

NumberValue.NumberValue.Return value:StringStringExample:BinaryAND (01,10)BinaryAND (10,11)

Result: 0 Result: 10

BinaryNOT

Purpose: Negates the value binary and returns the result. Parameter: Number Value. Return value: String Example: BinaryNOT (10) Result: 5 (ten equivalent 1010, five equivalent 0101)

BinaryOR

Purpose: Links the two (integer) parameters binary with 'or' and returns the result.
Parameter: Number Value.

Number Value. Number Value. Return value: String Example: BinaryOR (01,10) BinaryOR (10,11)

Result: 11 Result: 11

BinarySHL

Purpose: Shifts the value binary to the left. Parameter: Number Value. Number Number of bits, the value is shifted. Return value: String Example: BinarySHL (1,1) Result: 2

BinarySHR

Purpose: Shifts the value binary to the right. Parameter: Number Value. Number Number of bits, the value is shifted. Return value: String Example: BinarySHR (2,1) Result: 1,00

BinaryXOR

Purpose:

Links the two (integer) parameters binary with 'exclusive or' and returns the result.

Parameter: Number Value. Number Value. Value. Return value: String Example: BinaryXOR (01,10) BinaryXOR (10,11)

Result: 11 Result: 1



BMPMapToGray

Purpose: Converts the picture to greyscales. Parameter: Picture or String Return value: Picture Example: BMPMapToGray ("sunshine.gif")

BMPRotate

Purpose: Rotates a picture by the given degree. Parameter: Picture or String Number Rotation angle Number (optional) Mode (0=Degree, 1=Radian) Return value: Picture Example: BMPRotate(Article.Picture,90)

Capitalize\$

Purpose:

Returns a string in which the first letter of the individual words is a capital letter and the rest are small letters.

Parameter(s):

String

Return value:

String

Example:

Capitalize\$ (Product.Category) Result: �Tea, Coffee, And Soft Drinks

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 string will be copied to the return value string. If enough values do not exist, the 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:

Ceil

Purpose:

Calculates the next bigger integer based on the given value. See also function Floor().

Parameter: Number Value. Return value: Number

Example:

Ceil(5.6)

Result: 6

Century

Purpose: Returns the century of the date. Parameter: Date	
Boolean (optional) sets whether the	
should be carried out 'simply' (ce with year 0) or 'historically' (cent year 1). Default: False	
Return value:	
Number	
Example:	
Str\$(Century(Today()),0,0) Re	esult: 21
Str\$(Century(Date("01.01.2000")),0,0) 20	Result:
Str\$(Century(Date("01.01.2001")),0,0) 21	Result:
Str\$(Century(Date("01.01.2000"),.T.),0,0)	Result: 21

CheckMod10

Purpose: Calculate the modulo 10 checksum digit of the string. Parameter: String A string of digits. Return value: Number

Chr\$

Purpose:

Converts a number to a character. This character has the entered number as its 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 Unicode.

Return value:

String

Example:

Chr\$(64) Result: "@"

ChrSubst\$

Purpose:

Searches a 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
(optional)
Return value:
String
Example:
ChrSubst\$("Otto", "Oo", "_") Result: "_tt_"
ChrSubst\$("Normalconsumer", "aeiou", "??") Result: "N??
rm??lc??ns??m??r"
ChrSubst\$("Normalconsumer", "aeiou") Result:
"Nrmlcnsmr"
ChrSubst\$("3.1415926535", ".", ", ") Result:
"3,1415926535"

Cond

Purpose:

Allows to define conditions. The first parameter is a logical expression that will be evaluated as "True* of "False*. If the first expression is "True*, the second expression will be returned as the result. If the expression is "False*, the third expression will be returned as the result. If the expression is entered, the return value will assume the following standard values, dependent upon its type:

2. Argument Type	Return value if expression = False
Boolean	False
String	"" (empty String)
Date	Julian Date value 0
Number	0
Picture	"" (empty String)
Barcode	"" (empty String)

Parameter:

Boolean

All

All

(optional) The third parameter must be the same type as the second parameter.

Return value:

All

Example:

Cond(COUNTRY<>"USA",COUNTRY_LONG) Cond(PRICE=0,"on request",Str\$(PRICE,0,2)) Cond(empty(COMPANY),SALUTATION,COMPANY)

Constant.Pi

Purpose:

Returns the value of Pi.

Parameter: -

Return value:

Number

Example:

Constant.Pi() Result: 3,14159 (depending on the number of decimals)

Contains

Purpose: Evaluates if a string contains another string (second parameter).
Parameter: String String
Return value: Boolean
Example: Contains("Itemnumber: 12345", "1234") Result: True



Continued

Purpose:

Indicates that a text or RTF-object had a page break. This means that the current page is a result of the page break.

Parameter:

Return value: Boolean

Cos

Purpose: Calculates the cosine of the value. Parameter: Number Value Number (optional) Mode (0=Degree, 1=Radian). Default: 0. Return value: Number Example: Cos (90) Result: 0

Count

Purpose:

Counts the number of values of the first argument. With this function, all Null values in the argument are included in the count. Use the CountIf() function when you want to disregard Null values.

Hint: Sum variables (see <u>Working With Sum Variables</u>) are an alternative way of creating counters. Sum variables are principally applicable to whole tables. Aggregate functions principally table specific.

Parameter:

All

Values to count (sets the value to count). Needed to define the table (subtable) for which the records shall be counted.

Boolean (optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values will be already deleted within the table.

Return value:

Number

Example:

Count(Order Details.ProductID)

NthLargest(Article.Price,Count(Distinct(Artikel.Stkpreis),Tru True)

Calculates the second-smallest value, only taking repeated values into account once.

Countlf

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:

Countlf(Customers.Region="EMEA")

Countlf(Distinct(Customers.Region="EMEA")) counts repeated values once only.

Countlf (IsNull (Orders.OrderDate)) counts all values with empty fields.

Crosstab.Cells.Avg

Purpose:

Returns the average of the cell contents. Only available in crosstab objects.

Parameter:

- **Boolean** (optional) True: Only defined values are entered into the calculation (Default: False). Defined values: if you are, for example, analyzing customers and quarters, the quarters without turnover constitute an undefined value and can be treated separately in the calculation.
- **Number** (optional) Row layer (0= bottom layer or innermost group, 1= next lowest, ...). Default: 0.
- **Number** (optional) Column layer (0= bottom layer or innermost group, 1= next lowest, ...). Default: 0.

Return value:

Number

Crosstab.Cells.Max

Purpose:

Returns the largest value of the cell contents. Only available in crosstab objects. For the parameters and their meaning, see function Crosstab.Cells.Avg().

Crosstab.Cells.Min

Purpose:

Returns the smallest value of the cell contents. Only available in crosstab objects. For the parameters and their meaning, see function Crosstab.Cells.Avg().

Crosstab.Col\$

Purpose:

Returns the column header for the cell currently being output. Only available in crosstab objects.

Parameter:

Number (optional) Column layer (0= lowest layer or innermost group, 1= next lowest, ...). Default: 0.

Return value:

String

Crosstab.Col

Purpose:

Returns the column index for the cell currently being output. Only available in crosstab objects.

Parameter:

Boolean (optional) True: layer (only cells in this layer count), Default: False.

Return value:

Number

Crosstab.Row\$

Purpose:

Returns the row header for the cell currently being output. Only available in crosstab objects.

Parameter:

Number (optional) Row layer (0= lowest layer or innermost group, 1= next lowest, ...). Default: 0. Return value:

String

Crosstab.Row

Purpose:

Returns the row index for the cell currently being output. Only available in crosstab objects.

Parameter:

Boolean (optional) True: layer (only cells in this layer count), Default: False.

Return value: Number

Crosstab.Value

Purpose: Returns the cell's content. Parameter: -Return value: Number

CStr\$

Purpose:

Formats a number according to a format 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") CStr(100*Pi,"num: %g") 3.141593e+02" Result: " 3.1" Result: "num:

Date

Purpose:

Converts a string to a date.

- If the string Contains a dot ".", it will be read in the "d.m.y" format (German).
- If the string contains a diagonal slash "/", it will be read in the "m/d/y" format (US English).
- If the string contains a dash "-", if will be read in the "y-m-d" format (ANSI).
- If the input cannot be correctly interpreted, then the date represents a value that is larger than all other values, (1e100). The return value can be evaluated for correctness using "JulianToDate(1e100)".
- When one or two digits represent the year, all values under 30 will be applied to the 21st century (20xx) and all values over 30 will be applied to the 20th century (19xx).
- **Parameter:**
 - String
- Return value:

Date

Example:

Date("17.10.2007") Date("10/17/2007") Date("2007-10-17")

Date\$

Purpose:

Converts a date, using a format string, into an appropriately formatted string.

Composition of the format string: this is a normal string into which placeholders can be embedded.

Place holder	Description
%d	Day (131)
% <n>d</n>	Day to <n> digits</n>
%0 <n>d</n>	Day to <n> digits, filled on left with '0's</n>
%w	Weekday (17)
% <n>w</n>	Weekday to <n> digits</n>
%0 <n>w</n>	Weekday to $$ digits, filled on left with '0's
%m	Month (112)
% <n>m</n>	Month to <n> digits</n>
%0 <n>m</n>	Month to $$ digits, filled on left with '0's
%у	Year
% <n>y</n>	Year, to <n> digits</n>
%0 <n>y</n>	Year, to $\langle n \rangle$ digits, filled on left with '0's
%q	Quarter
%D	Weekday, written out
%M	Month, written out
"%e", "%	Year in the local calendar (Japan:
<n>e"</n>	Emperor's year)
"%g", "%	Era of the local calendar (Japan:
<n>g"</n>	Emperor s era)
"%g", "%1g"	Single letter, Latin letters
"%gg", "%2g"	Single letter, localized
"%ggg", "%3g"	Long name, localized

"%ggggg", "%4g"	Long name, Latin letters
"%x"	Localized date, short form
%Н	Hours in 24h-format
%h	Hours in 12h-format
% <n>h</n>	Hours in <n> digits</n>
%i	Minutes
% <n>i</n>	Minutes in <n> digits</n>
%s	Seconds
% <n>s</n>	Seconds in <n> digits</n>
%PM	AM or PM

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 3166-Country code, the third parameter can be used to set either the short "0⁽²⁾ or long "1⁽²⁾ format.

Parameter:

DateValue to be formatted.String(optional) Format description or ISO 3166-Country code.

String (optional) ISO 3166-Country code or date format.

Return value:

String

Example:

Date\$(Today(),"Date: %D, %d/%m/%y") Result: "Date: Thursday, 8/11/2007" Date\$(Today(),"%2wthWeek; %D, %2d/%2m/%4y")

Result: "45th Week, Thursday, 8/11/2007"

Date\$(Today(),"%D, %3d/%02m/%4y") Result:

"Thursday, 8/11/2007"

Date\$ (Now(),"%02h:%02i:%02s %PM") Result: "04:03:50 PM"



DateDiff

Purpose: Returns the difference between two dates in days. Parameter: Date First date value Date Second date value Return value: Number Example: DateDiff(Date("01.01.2009"),Date("01.03.2009")) Result: 59



DateDiff\$

 Purpose:
 Returns the difference between two dates in days as string.

 Parameter:
 Date
 First date value

 Date
 Second date value

 String
 (optional) Format

 Return value:
 String

 String
 Date

 Date
 Second date value

 Date
 Second date value

 String
 (optional) Format

 Return value:
 String

 DateDiff\$(Date("01/01/2009"),Date("03/01/2009"))

 Result: 2 Months



DateHMS

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

NumberHourNumberMinuteNumberSecondReturn value:Date

DateInLeapYear

Purpose:

Checks if the given date is in leap year or not. The calculation is made according to the proleptic gregorian calendar.

Parameter: Date **Return value: Boolean**

Example:

DateInLeapYear("01.01.2012") Result: True

DateInRange

Purpose:

Evaluates if the date falls within the entered time interval:

Minimum Date: JulianToDate(0)

Maximum Date: JulianToDate(1e100)

Parameter:

Date Date to be evaluated.

Date Lower limit of the test interval.

Date Upper limit of the test interval.

Return value:

Boolean

Example:

DateInRange(Date("2007.10.20"),Date("2007.2.29"),Today Result: True



DateToJulian

Purpose:

Calculates the Julian value of a date. Each day (even those in the past) are assigned a unique number.

Parameter: Date **Return value:**

Number

Example:

DateToJulian(Today()) Result: 2453992

DateYMD

Purpose: Converts three numbers for day, month and year into a date. Parameter: Number Year Number Month Number Day Return value: Date Example: DateYMD(2009, 11, 1) Result: 01.11.2009

Day

Purpose: Determines the day (1...31) of the month and returns it as a number. **Parameter:** Date **Return value:** Number

Example:

Day(Date("17.10.2009")) Result: 17

Day\$

Purpose: Determines the day $(1 \circledast 31)$ of the month of a date and returns it as a string. **Parameter:** Date **Return value:** String

Example:

Day\$(Date("17.10.2009")) Result: "17"

Decade

Purpose: Returns the decade of the date. Hint: The valu- relative to the start of the century (110)! Parameter:	e is always
Date	
Boolean (optional) sets whether the cald should be carried out 'simply' (decad with year 0) or 'historically' (decade s year 1). Default: False Return value: Number	e starts
Example:	
Str\$(Decade(Date("01.01.2009")),0,0) Str\$(Decade(Date("01.01.2000")),0,0) 10	Result: 1 Result:
Str\$(Decade(Date("01.01.2000"),.T.),0,0)	Result: 1



Distinct

Purpose: Affects the higher order aggregate function (e.g. Sum(), Avg(), Count()...) and causes equal values only to be used once in the calculation.
Parameter: All Return value: All

Example:

Countlf(Distinct(Customers.Region="EMEA"))

Dow

Purpose: Returns the day of the week to a number(1...7), 1=Sunday, 2=Monday, ... Parameter: Date Return value: Number Example: Dow(Date("04.07.1776")) Result: 4 (Thursday).

Dow\$

Purpose: Returns the day of the week as a string in accordance with the country settings, "Sunday", "Monday", ...

Parameter: Date Return value: String Example: Dow\$(Date("04.07.1776")) Result: "Thursday"

Drawing

Purpose: Converts a string type file path into a picture. Parameter: String Return value: Drawing

Drawing\$

Purpose: Converts a picture into a string type file path. Parameter: Picture Return value: String



DrawingHeightSCM

Purpose:

Returns the height of the Picture in SCM units (1/1000 mm).

Parameter: Picture Return value: Number



DrawingWidthSCM

Purpose:

Returns the width of the Picture in SCM units (1/1000 mm).

Parameter: Picture Return value: Number

Empty

Purpose: Evaluates if a string is empty. If it is empty, "True will be the return value, otherwise "False . Useful, for example, to determine if the field "ADDRESS is empty, and if it is, in combination with the IF-THEN-ELSE condition cond(), either print the contents of the field "ADDRESS or "POBOX . The third parameter allows the removal of leading and			
The third parameter allows the removal of leading and trailing spaces. If this is evaluated as "True $m{v}$, 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			

EndsWith

Purpose:

Checks whether the string in the first argument ends with the string in the second argument.

Parameter(s):

String

String

Specifies whether capitals/small letters Boolean are relevant. Default value: False

Return value:

Boolean

Examples:

EndsWith ("Hallo World","rld") EndsWith ("Hallo World", "llo") Result: False

Result: True

Evaluate

Purpose: Evaluates the expression passed as parameter. Parameter: String Return value: All Example: Str\$(Evaluate("3*4"),0,0) Result: 12 Str\$(Evaluate("4-3"),0,0) Result: 1

Even

Purpose:

Evaluates if a number is even. If the number is even, "True� will be returned, otherwise "False�.

Parameter: Number

Return value:

Boolean

Example:

"Page number "+Cond(Even(Page()),"even","odd")

Exists

Purpose:

Checks if a variable or field is defined. Is often used in connection with GetValue() and Evaluate().

Parameter:

String

Return value:

Boolean

Example:

Exists("CustomerID") Result: False If(Exists("Customer.Status"),Evaluate("Customer.Status")," customer status")

Exp

Purpose: Calculates the exponential (e^x). Parameter: Number Return value: Number Example: Exp(3) Result: 20.08553692

Exp10

Purpose: Calculates 10 raised to the power of number (10^x). Parameter: Number Return value: Number Example: Exp10(3) Result: 1000

FirstHeaderThisTable

Purpose:

Returns whether the header of the table is being output for the first time. The function can be used as an appearance condition for the header to prevent it being printed more than once if the table continues onto the next page due to space limitations. The header is then only printed at the beginning of the table.

Parameter:

Return value: Boolean

Floor

Purpose:

Calculates the next smaller integer based on the given value. See also function Ceil().

Parameter: Number

Return value:

Number

Example:

Floor(5.6) Result: 5

Frac

Purpose: Calculates the fractional part of a number Parameter: Number Return value: Number Example: Frac(Pi) Result: 0.1415926535

FStr\$

Purpose:

Formats a number according to the format string. These consist of the following characters ("if negative" refers to the value to be formatted):

* Digit or '*'-Prefix

Local currency symbol

_ Digit or sign, if negative

+ Digit or sign

(Digit or '('-Prefix if negative

) ')'-Postfix if negative

Digit or space prefix

& Digit or '0'

. Decimal point

Comma, or space prefix

A prefix is a sign that precedes a value, when needed. The expression FStr\$(1, "***") results in "**1". The value

"1" is preceded by the characters "**".

A Postfix is a character that, when needed, is placed after a number.

These formatting characters can be combined as needed. If the number is too large for the desired format, a "* string will be returned.

With the third (optional) parameter, additional formatting can be accomplished.

Value	Description
1	Removal of leading spaces. The use is similar to the functions RTrim\$() and LTrim\$().
2	Empty string if value Null.
3	Removal of leading spaces and empty strings when value is 0

Parameter: Number Format string String Number (optional) Additional formatting **Return value:** String **Example:** FStr(3.142, "#") Result: "3" FStr(5003.1,"#,###.&&") Result: "5.003,10" FStr\$(3.142,"#.###") Result: "3,142" FStr\$(3.142,".######") Result: "*****" FStr\$(3.142,"(#.###)") Result: " 3,142 " FStr\$(-3.142,"(#.###)") Result: "(3,142)" FStr\$(3.142,"+#.###") Result: "+3,142" FStr\$(3.142,"-#.###") Result: " 3,142" FStr\$(-3.142,"-#.###") Result: "-3,142" FStr\$(3.142,"&&&.&&") Result: "003,142" FStr\$(3.142,"***.***") Result: "**3,142" FStr\$(3.142,"\$\$\$.\$\$\$") Result: "\$\$3,142" Result: " 3,142" FStr\$(3.142,"###.***") FStr\$(5003.1,"#,###.&&") Result: "5.003,10" Result: " 3" FStr\$(3.142,"#####")

GeometricAvg

Purpose:

Calculates the geometric average of the set of values that result from the first parameter / formula.

Parameter:

Number Boolean Expression for the value to be averaged. (optional) True: The values which were stored for the calculation are deleted after output. (default: True). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

GetValue

Purpose:

Returns the value of a variable or field. Is often used in connection with Exists().

Parameter:

String

Return value:

All

Example:

Str\$(GetValue("Customers.CustomerID"),0,0) Result: 1234

If(Exists("Customer.Status"),Evaluate("Customer.Status")," customer status")

GetVar

Purpose:

Fetches a value that was set with the SetVar() function from the variable repository. The purpose of these functions is to provide a simple buffer for values. You shouldn't execute complex nesting with GetVar/SetVar or combine both functions with each other - especially for header, footer and group lines unexpected effects can occur here.

```
Parameter(s):
All
Return value:
All
Example:
GetVar ("Page")
SetVar
```

result: contents of

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 \clubsuit . The value of the current hour must be determined and then compared to the value "10 \clubsuit . Hour()=10

HSL

Purpose: Calculates a color value in the HSL color space (Hue, Saturation, Lightness) Parameter:

Number
NumberHue [0-360] (0°=red, 120°=green, 240°=blue)NumberSaturation [0-1]NumberLightness [0-1] (0=no lightness, 1=full lightness)Return value:
NumberNumber

Hyperlink\$

Purpose:

The function Hyperlink\$ creates a hyperlink text that can be inserted using an export module. When available, the hyperlink will only be embedded if the third parameter returns a result of "True� (=default).

To optionally use a hyperlink only on a HTML page, you need to use the List & Label variable LL.OutputDevice: Hyperlink\$("combit","http://www.combit.net",LL.OutputDev If an object text contains the string:

<!--begin:hyperlink="target"-->"Display text"<!-end:hyperlink-->

then a hyperlink will be automatically created in the exported HTML page. The hyperlink function automatically creates a string with the correct syntax.

Parameter:

String	Text
String	Hyperlink
Boolean	(optional) Embedded

Return value:

String

Example:

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

lf

see Cond



IssueIndex

Purpose:

Returns the Issue Index (1..) for display and layout region conditions, if multiple issues are selected in the project parameters

Parameter: Number Return value: Number

Int

Purpose:

Calculates the integer value of a number. The value will be truncated.

Parameter: Number Return value: Number

Example

Int(3,1) Result: 3

IsNull

Purpose:

Checks whether the transferred value or the result of the expression is Null, e.g. an empty date field.

Parameter: All Return value: Boolean

IsNullOrEmpty

Purpose: Checks whether a string is empty or Null. Parameter(s): String The string to be checked Boolean (optional) If the value is True, then spaces at the beginning and end of the string are removed. Default value: False Return value: String Example: ISNUIIOrEmpty (� �, True) Result: True

Join\$

Purpose: Collection of strings separated by a particular character. **Parameter:** String Collection of strings separated by a particular character. (optional) Separator for the formatting String string (default: ";") (optional) Maximum number of values ('...' Number will be appended). Default: all values. Boolean (optional) True: The values which were stored are deleted after output. (Default: True). Please note that the stored values are generally deleted for every (sub)table end. The

values are generally deleted for every (sub)table end. The second parameter only decides whether the values will be already deleted within the table.

Return value: String



JulianToDate

Purpose: Interprets a number as a Julian date (each day is assigned a unique number) and returns the appropriate date. **Parameter:** Number **Return value:** Date **Example:** JulianToDate(2454347) Result: 09/04/2007

LangCase\$

Purpose:

Returns one of the substrings according to the language that is set.

Parameter(s):

String String that is returned if a localization string is not found. The localization string must be a valid ISO 639 language code.

String String with translation substrings, separated by "|" (or by an optional third argument). Translation texts must be formatted as follows "ISO 639 language code = translation text|[ISO 639 language code = translation text]♥]. If the characters "|" or "=" are to be used in the value or the key, they must be preceded by "\", e.g. "USA=He\=llo".

String (optional) separator (default: "|")

Return value:

String

Example:

```
LangCase$("Hallo","USA=Hello|ESP=Hóla")
result: " Hóla " (with Spanish systems)
```

LastFooterThisTable

Purpose:

Returns whether the footer of the current table is being output for the last time. This function can be used as an appearance condition for the footer, in order to prevent the footer being printed if the table is continued on the next page due to space limitations. The footer is then only printed on the last page of the table.

Parameter:

Return value: Boolean

Lastpage

Purpose:

Returns if the current page is also the last page. This function can only be used in the footer lines of tables, in objects linked with tables or in the layout regions condition! In all other cases, the result of Lastpage() is always False.

Parameter:

Return value:

Boolean

Example:

Cond(Lastpage(),"Total sum","Subtotal")

Left\$

Purpose:

Reduces a string from the right so that only the number of characters set under Number remain. If the original string is already small enough, it is not affected.

Parameter:

StringThe value to be shortenedNumbermaximum number of positions of the resultBoolean(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)ReNAME string.Left\$("combit", 2)ReLeft\$("combit", 4,True)ReLeft\$("combit", 2, True)Re

Result: the first letter of the

Result: "co" Result: "c�" Result: "co"

Len

Purpose: Returns the number of characters in a string. Parameter: String Return value: Number Example: Len("1234"+"12") Result: 6



LoadFile\$



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 Loca	ale Entry
String	(optional) ISO 3166-Country code	
Return value:		
String		
Example:		
Locale\$(42,"USA")		Result: "Monday"
Possible constants for index entry:		
http://msdn.microsoft.com/en-us/library/bb507201.aspx		

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) ISO 3166-Country code Return value: String Example: LocCurr\$(123,"USA") Result: "123.00"

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) ISO 3166-Country code Return value: String Example: LocCurr\$(123,"USA") Result: "\$123.00 "

LocDate\$

Purpose: Returns a string with the valid date format for the entered country. Parameter: Date Value to be formatted String (optional) ISO 3166-Country code Number (optional) Format Return value: String Example LocDate\$(Date("17.11.2007"),"USA") Result: "11/17/2007"



LocDateTime

Purpose

Converts the string into a date (with time if required). The string is expected to be in the relevant format for the country.

Parameter:

Date (optional) ISO 3166-Country code

String Return value:

String

Date

Example

LocDateTime("04.07.1776","DEU") Result: "07/04/1776"

LocNumber\$

Purpose: Returns a string with the valid number format for the entered country. Parameter: Number Value to be formatted String (optional) ISO 3166-Country code Return value: String Example: LocNumber\$(123,"USA") Result: "123.00"

LocTime\$

Purpose: Returns a string with the valid time format for the entered country. Parameter: Date Value to be formatted String (optional) ISO 3166-Country code Number (optional) Format Return value: String Example: LocTime\$ (Now(),"USA") Result: 9:05:22 AM"

LocVal

Purpose: Interprets the string as a number and returns its value (while respecting any localized decimal or 1000 separators). Parameter: String Number (as string) String (optional) ISO 3166-Country code Return value: Number

Example:

LocVal ("12,00","USA")

Result: 1200,00

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:Un

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. (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:Uni

Minute

Determines the minute of the entered date, and returns the result as a number. If the parameter is not used, the minute of the time of printing will be returned.

Parameter:

Date (optional) Return value: Number

Mode

Purpose:

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

Parameter:

Number Expression for the value to be examined. (optional) The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Month

Purpose: Determines and returns the month (1...12) as a number. Parameter: Date Return value: Number Example: Month(Date("2007.10.17")) Result: 10

Month\$

Purpose: Determines and returns the month (1...12) as a string. Parameter: Date Return value: String Example: Month\$(Date("2007.10.17")) Result: "10"

Now

Purpose: Returns the current date and time. Parameter: Return value:

Date

NthLargest

Purpose:

Calculates the nth-largest value of the set of values that result from the first parameter / formula.

Parameter:

Number

Number <n>, i.e. the index for the value which is to be returned (1-based).

Boolean (optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

NthLargest(Order Details.ProductID,2) calculates the 2-largest number

NthLargestIndex

Purpose:

Calculates the index of the nth-largest value of the set of values that result from the first parameter / formula.

Parameter:

Number

Number <n>, i.e. the index for the value which is to be returned (1-based).

Boolean (optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

NthLargestIndex(Order_Details.ProductID,2)

NthValue

Purpose:

Calculates the nth value of the set of values that result from the first parameter / formula.

Parameter:

All

Number <n>, i.e. the index for the value which is to be produced, calculated e.g. with NthLargestIndex().

Boolean (optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

NthValue(NthLargestIndex(Order_Details.ProductID,2))

Null

Purpose: Returns a Null value (value not available). Parameter: -Return value: All



NullSafe

Purpose: Checks if the parameter is Null and returns a substitute value if it is, otherwise it returns the value of the parameter.
Parameter: All Return value: All



NumInRange

Purpose: Evaluates if a number falls within the desired range. Parameter: Number Number Upper limit Number Lower limit Return value: Boolean Example: NumInRange(Page(),1,10) Result: True, if page number is between 1 and 10.

Odd

Purpose:

Evaluates if a number is odd. If the number is odd "True will be returned, otherwise "False.

Parameter:

Number

Return value:

Boolean

Example:

"Page number "+Cond(Odd(Page()),"odd","even")

Ord

Purpose: Returns the ASCII value of the first character. Parameter: String Return value: Number Example:

Ord("combit") Result: 99

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 string. **Parameter:**

Return value: String

String Evampla:

Example:

"Page "+Page\$()+"/"+TotalPages\$() Result: Page 1/3

Pow

Purpose: Corresponds to the function (Base) ^ (Exponent). Parameter: Number Base Number Exponent Return value: Number Example: Pow(2,3) Result: 8

Previous

Purpose: Returns the previous value of the variable, field or formula, i.e. the value it had for the last record.
Parameter: All Variable, field or formula
Return value: All
Example: Previous(NAME) Result: "consumer"

PreviousUsed

Purpose:

Returns the value the given variable or expression had when it was last evaluated.

Parameter:

Variable, field or formula

Return value:

All

Example:

PreviousUsed(NAME) Result: "Mustermann"

ProjectParameter\$

Purpose:

Returns the value of a project parameter. Available parameters:

purumetersi	
LL.FAX.Queue	Print queue
LL.FAX.RecipName	Recipient name
LL.FAX.RecipNumber	Recipient fax number
LL.FAX.SenderBillingCode	Sender billing code
LL.FAX.SenderCompany	Sender company
LL.FAX.SenderDept	Sender department
LL.FAX.SenderName	Sender name
LL.MAIL.To	Mail address
LL.MAIL.CC	Mail address for
	carbon copy
LL.MAIL.BCC	Mail address for blind
	carbon copy
LL.MAIL.Subject	Subject line
LL.MAIL.From	Sender mail address
LL.MAIL.ReplyTo	Reply To mail address
LL.MinPageCount	Minimum page count.
LL.ProjectDescription	Project Description
LL.SlideShow.TransformationID	Default transition
	effect for the preview
	's slideshow mode.
LL.MAIL.ShowDialog	Show mail dialog
	before sending

Parameter:

String Boolean

Name of the project parameter

(optional) sets whether the return value (possibly a formula) should be returned directly (True), or should be evaluated (False). Default: False

Return value: String

Example:

ProjectParameter\$("LL.ProjectDescription") Result:"Article list"

ProjectPath\$

Purpose:

Returns the path of the project file, optionally including the file name (otherwise with "\" at the end)

Parameter:

Boolean True: Sets that the path is returned including the file name (Default: False).

Return value:

String

Example:

ProjectPath\$() ProjectPath\$(True) Files\LL\crosstab.lsr

Result: C:\Program Files\LL\ Result: C:\Program

Quarter

Purpose:		
Returns t	he quarter of the year (14)	
Parameter:		
Date		
	calculation should be returned relat year (14) or in absolute terms sinc (1). Default: False (relative).	ive to the
Return valu		
Number		
Example:		
Str\$(Qua	rter(Date("01.01.2008")),0,0)	Result: 1
Str\$(Qua	rter(Date("01.05.2008")),0,0)	Result: 2
Str\$(Qua 8030	rter(Date("01.05.2008"),.T.),0,0)	Result:

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	
Return value:	Limiting value for red.

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"

Purpose: Replaces the substrings of the first argument with a new value if they match the regular expression. **Parameter(s):** String The string to be checked String **Regular** expression Replacement expression (can contain String "\0" for the entire match or " 1° "\9" for the respective group. (optional) Specifies whether only the first Boolean occurrence is to be replaced. Default value: False **Return value:** String **Example:** RegExSubSt\$("1234xyz6789","[0-9]+", "a") result: "axyza" RegExSubSt\$("1234xyz6789","[0-9]+", "a") result: "axyz6789"

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 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

SetVar

Saves a value in the variable repository for later use with the GetVar() function. The purpose of these functions is to provide a simple buffer for values. You shouldn't execute complex nesting with GetVar/SetVar or combine both functions with each other - especially for header, footer and group lines unexpected effects can occur here.

```
Parameter(s):
String
All
Return value:
All
Example:
SetVar ("Page", Page())
```

Sign

Purpose:

Returns the sign of the value (+1 for a positive value, -1 for a negative value or 0 if the value is 0).

Parameter: Number

Number Value Return value: Number Example: Sign (-3) Result: -1

Sin

Purpose: Calculates the sine of the value. Parameter: Number Value Number (optional) Mode (0=Degree, 1=Radian). Default: 0. Return value: Number Example:

Sin (90) Result: 1

Sqrt

Purpose: Calculates the square root of a number. Parameter: Number Return value: Number Example: Sqrt(4) Result: 2

StartsWith

Purpose:

Checks whether the string in the first argument begins with the string in the second argument.

Parameter(s):

String

String

Boolean Specifies whether capitals/small letters are relevant. Default value: False

Return value:

Boolean

Examples:

StartsWith ("Hello World", "hel") Result: True StartsWith ("Hello World", "rld") Result: False

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.UnitPric

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)
Str\$(Pi,3)
Str\$(Pi,3,0)
Str\$(-Pi,12,-3)
Str\$(Page())
Str\$(Page(),10)
Str\$(Page(),-10)

Result: "3.141593" Result: "3" Result: "3" Result: "-3.141e+00" Result: "5.000000" Result: "5" 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 Number

Search string (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 Number

Search string (optional)

Return value:

Number

Examples:

Result: 1 Result: 13 Result: 13 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

(optional) Replacement string

String Return value:

String

Example:

Assume that you want to print address labels that contain the company name. You do not have much space available on the label and cannot afford to completely print long company names, for example, "Forrer Construction, Incorporated?

With the expression StrSubst\$(COMPANY,"Incorporated","Inc.") every appearance of "Incorporated" in the COMPANY field will be replaced with "Inc."

Sum

Purpose:

Calculates the sum of the parameter / formula in the parameter.

Hint: Sum variables (see <u>Working With Sum Variables</u>) are an alternative way of creating sums and counters. Sum variables are principally applicable to whole tables. Aggregate functions principally table specific.

Parameter:

Number

Boolean

(optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value: Number

Example:

Sum (Order_Details.UnitPrice)

Tan

Purpose: Calculates the tangent of the value. Parameter: Number Value Number (optional) Mode (0=Degree, 1=Radian). Default: 0. Return value: Number Example:

Tan (45) Result: 1,00

Time\$

Purpose:

Returns the current time in string format. The following formats are available:

Hours in 24 hour format Hours in 12 hour format
Hours in 12 hour format
Minutes
Seconds
Display the part of day (A.M. / P.M.)
Display the part of day (a.m. / p.m.)

Parameter:

String

Return value:

String

Example:

Time\$("%02h:%02m:%02s") "18:30:45"

Result:

Today

Purpose: Returns the current date. **Parameter: Return value:** Date **Example:** Date\$(Today(),"%D, %m.%d.%4y") Result: "Friday,

11/8/2007"

Token\$

See Case\$

ToRTF\$

Purpose:

Returns a string in RTF-format. This is necessary because some strings may contain one of the specially defined RTFformat symbols. ('\', '{' or '}'). For compatibility reasons, this function only processes the passed string if the optional second parameter is explicitly set to True.

Parameter:

String Boolean

воогеан

Return value:

String

Example:

If, for example, the field PRODCODE could contain one of the characters, then the text should be inserted in the following way:

"<<SALUTATION>> <<NAME>>, You have received our product <<PRODUCT>>, Code <<ToRTF\$(PRODCODE)>>..."

Total\$

Purpose:

Sets that the expression in the argument is calculated for the whole crosstable-object.

Parameter: All Return value: All Example:

Sum(sales)/Total(Sum(sales))*100

TotalPages\$

Purpose:

Returns the total number of pages. The returned 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
```



Translate\$

Purpose:

Translates the text in the argument provided that it is held in the dictionary transferred by the application.

Parameter(s):

String

Return value:

String

Example:

Translate\$("Page {0} of {1}", Page\$(), TotalPages\$()) Result in German e.g.: Seite 1 von 2

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(2000(Result: 10 cm for Page 1, 2 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") Val("3,141") Val("3.141e2") Val(ChrSubst\$("3,141", ",", "."))

Result: 3.141 Result: 3 Result: 314.2 Result: 3.141

Variance

Purpose:

Calculates the variance of the set of values that result from the first parameter / formula.

Parameter:

Number

Boolean (optional) TRUE: The values which were stored for the calculation are deleted after output. (default: TRUE). Please note that the stored calculation values are generally deleted for every (sub)table end. The second parameter only decides whether the values are already deleted within the table.

Return value:

Number

Example:

Variance(Order_Details.Quantity*Order_Details.UnitPrice)

Woy

Purpose:

Returns the week number of a given date. The optional second parameter determines the setting for the first week of the year.

- 0 Week with the first working day
- 1 Week of January, 1
- 2 First week with at least 4 days
- 3 First week with 7 days
- 4 Week with the first Monday

Parameter: Date Number (optional) Return value: Number

Year

Purpose: Determines the year of a date and returns it as a number. Parameter: Date Return value: Number Example: Year(Today()) Result: 2010 Year\$(Date("1.1.2010")) Result: 2010

Year\$

Purpose: Determines the year of a date and returns it as a string. Parameter: Date Return value: String Example: Year\$(Today()) Result: "2010" Result: "2010"



Overview of Properties

All of the properties for projects and objects are described centrally here.

Properties are defined by means of the respective property lists. If you select multiple objects, you can set their common properties at the same time. You can specify values in different ways depending on the property.

§ Open a drop down list of values by means of an "arrow down" button.

Example: Appearance condition, font color, font. At the end of the list of values, you will almost always find the "Formula" entry.

§ You can set the value with a formula via the formula button or the "Formula" entry in the list of values. Example: If you want to set the font color to red for negative values, set the default "property" for the font to

"False" and define the "Font color" property using a formula, e.g.:

Cond(Item.UnitPrice < 0,LL.Color.Red,LL.Color.Black)

- § Open a configuration dialog with the "..." button. For example, there are dialogs for the following properties: formatting, font, frames, position, label format.
- § Enter the value directly in the property fields.

Example: Project description in the project properties.

§ Set a file path with the open dialog.

Example: Name of the project include file or the image file.

- Project Properties
- Common Object Properties
- Text Objects
- Line Objects

- Rectangle Objects
- Circles and Ellipses
- Picture Objects
- Barcode Objects
- Report Container Object
- Table Objects
- Chart Objects
- Gauge Objects
- Crosstab Objects
- Formatted Text Objects
- Form Control Objects
- HTML Text Objects
- •OLE Container
- Form Template Objects



Project Properties

The project's property window is displayed if no object is selected in the workspace.

The project properties are also available as fields (see <u>Overview of Fields</u>) and can be evaluated with the ProjectParameter\$() function.

- General Settings
- Fax and Mail Variables

General Settings

- Project Description
- Active Design Layout
- Number of Issues
- Display Condition for Issue Print
- Embedding Drilldown Reports
- Minimum Page Count
- Transition Effects for Slideshow



Project Description

You can enter a description for the respective project in the "Project description" field. This description is then shown in the **File > Open** dialog making it easier for you to find the project that you want. Alternatively, you can also enter the description in the **File > Save As** dialog.

Properties	
10 \$i (+)	
General Settings	
Project Description	List & Label Project File
Active Design Layout	Original
Number of Issues	2
Display Condition for Issue Pr.	True
Embed Drilldown Reports	Yes
Minimum Page Count	1
Transition Effect for Slidesho	
🖽 Mail Parameter	
To	
CC	
BCC	
From	
ReplyTo	
Subject	
E Fax Parameter	
Receiver's Fax Number	
Receiver's Name	
Sender's Name	
Sender's Company	
Sender's Department	
Sender's Billing Code	

Figure 13.1: Project properties



Active Design Layout

With multi-page projects, it is sometimes a good idea to choose different layout settings, e.g. page size, orientation for the different pages. You specify which layout setting is to be shown in the workspace by means of the "Active design layout" field.



Number of Issues

Specifies the number of issues (copies) for printing and previewing. In addition, it also enables the IssueIndex() function for display and layout region conditions. If you specify multiple issues, you will then have the "Display condition for issue print" property which you can use for printing of objects conditionally for the different copies.



Display Condition for Issue Print

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

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



Embedding Drilldown Reports

Drilldown reports can be embedded in the preview file to allow them to be sent or saved as a complete unit.



Minimum Page Count

With index card projects, this property specifies the minimum number of pages that are to be printed automatically. For example, if you want to output a four-page form with different layouts for each of the four pages, you create a layer for each page and position the objects on these layers as required for the output. Specify "4" as the minimum page count.

With list projects, the number entered here determines the page number on which the output of the table/report container will start. For example, if you need a covering sheet, you can assign the "Following pages" layer to the table and design the "First page" layer as you wish. Then specify "2" as the minimum page count.



Transition Effects for Slideshow

Here you specify the default values for the kind of page transition in the preview's slideshow mode.



Fax and Mail Variables

You send faxes of List & Label documents by selecting the respective fax (printer) driver in the print process. A fax program must be installed in order to be able to send faxes. If the fax is to be sent via the Windows fax driver, the fax parameters (at least the fax number) must be specified in the project properties. Enter the respective variables in the "Fax Parameters" area.

If the fax is to be sent via a different fax (printer) driver, you enter the fax number and other field information (as far as supported) by means of commands (e.g. DvISE commands for Tobit David). You enter these commands directly in a text field in the print project. Doing this suppresses the recipient dialog during printing because all information is already embedded in the document. You will find the precise procedure in the documentation for your fax software. You can also send List & Label documents directly by email. You also define the required email variables in the project's property window. Make the email settings (SMTP, MAPI, XMAPI) under "combit Mail" in the Windows Control Panel.



Common Object Properties

You specify most of the object properties in the property list and/or in additional dialogs. Each object type has its own individual properties. However, there are a number of attributes that are common to all objects, such as size, position, name and, appearance condition. These properties are described here centrally and explained in more detail in the sections that follow.

- Locked
- Name
- Display Condition for Issue Print
- Appearance Condition
- Levels in the Table of Contents
- Page Break Before Outputting Object
- Export as Picture
- Position
- Font
- Color
- Background / Filling
- Pattern
- Frame
- Format
- Content



Locked

Locks the object to prevent it from being selected unintentionally by clicking. This property is only relevant during design and has no effect on the later print. If you set "Locked" to "True", the object in question can no longer be selected in the workspace and will be marked with a small red logo. This property is not available for sub-tables.



Figure 13.2: Locked objects in the object list

Note: You can select a locked object as usual in the object list thereby making it editable again. Since "locked" is only relevant during the project s design phase, there is no way in which you can determine the value of the property by means of a formula.

Property	Description	Value	Description
Locked	Locks the object and prevents it from being selected unintentionally by clicking in the workspace.		locked not locked



Name

When you add a new object to the workspace, a description of the object, made up of the type of the object (e.g. "Text") and its coordinates, appears in the right section of the status line. This is the default name for this object.

However, if your project has a large number of similar objects, these identifiers can easily become confusing. For this reason, you can give your objects meaningful names by means of the **Objects tool window** or with the object's property list. You do this by simply clicking once on the existing name and then changing it.

Alternatively, you can enter a new name in the object name input field via **Objects > Object list** or you can change it in the property list.

	5		
Property	Description	Value	Description
Name	Name of the object	Name	

If you have enabled the **Options > Workspace > Object info** option, the object name will also be shown in the tooltip that appears.



Display Condition for Issue Print

This option enables conditional printing of objects for the different issues. This property is only available if you have defined multiple issues in the project properties. The IssueIndex() function lets you specify the index of the issue, e.g. IssueIndex()=2. You will find more information about managing issues in chapter Project Properties.

<u></u>	<u> </u>		<u> </u>
Property	Description	Value	Description
Display condition for issue print	Enables conditional printing of objects for the different issues, e.g. lssueIndex()=2.	False	Display Hide Formula wizard



Appearance Condition

You can assign an appearance condition to each object. This specifies under which conditions the object is to be printed. You will find a guide to defining such conditions under <u>Variables</u>, Formulae and Expressions.

Property	Description	Value	Description
Appearance condition	Appearance condition for printing. There is no output if the result is false. In appearance conditions for table footers, you can also use the predefined "Last page only" value. Internally, this entry uses	True False FirstHeaderThisTable() LastFooterThisTable() Formula	Always show Never show First page only Last page
	the functions Lastpage() or LastFooterThisTable(). In appearance conditions for table headers, you can also use the predefined "First page only" value. Internally, this entry uses the functions not Lastpage() or FirstHeaderThisTable().		only Formula wizard



Levels in the Table of Contents

Specifies the outline level (index level) of the bookmark in preview mode or for PDF export (0=not in index). You can set the maximum folder depth via **Project > Options >**

Project.

I I OJČČČ			
Levels in the table of contents	Specifies the index level of the bookmark (0=not in index).	Number Formula	Formula wizard
Text	Name of the bookmark.	Text	Formula wizard



Page Break Before Outputting Object

Each object can trigger a page break before it is printed, i.e. the object begins on a new page.

Property	Description	Value	Description
Page break before	If the condition returns "True", a page break will be triggered before printing the object.	False	Break No break Formula wizard



Export as Picture

For exporting objects in picture format if a vector-based export does not give the desired results, or in order to achieve a better representation.

Property	Description	Value	Description
Export as picture	If the result is "True", the object will be exported as a picture.	False	Yes No Formula wizard



Position

An object's "Position" property group specifies the x and y coordinates of the upper left corner of the object as well as the width and the height.

There is also a dialog for defining the values.

Property	Description	Value	Description
Position	Position and size of the object, all details are given in the unit of measure for the workspace		Position dialog
Left	Horizontal distance of the upper left corner of the object from the upper left corner of the workspace	Number Formula	Formula wizard
Тор	Vertical distance of the upper left corner of the object from the upper left corner of the workspace	Number Formula	Formula wizard
Width	Width of the object	Number Formula	Formula wizard
Height	Height of the object	Number Formula	Formula wizard



Font

If the **default value** is set to "Yes", the default font will be used.

Property	Description	Value	Description
Font	You can define the font properties in a dialog. If the default value is set to "True", the default font will used.		Font dialog
Default value	The default font will be used instead of the set values.	True False Formula	Default font No Formula wizard
Name	Selected font. All installed fonts will be displayed.	List Formula	Font Formula wizard
Character set	Specifies the country version of the character set. All available character sets are displayed.	Number	Character set
Size	Font size in points. Lists all available sizes for the selected font.	Number Formula	Default size Formula wizard
Width	Sets the width of the font. 0 means standard width, otherwise the average character width will be specified.	Number Formula	Width Formula wizard
Bold	Turns the "bold" text property on and off	True False Formula	Yes No Formula wizard
Italic	Turns the "italic" text property on and off	True False Formula	Yes No Formula wizard

There is also a dialog for defining the values.

Underline	Turns the "underline" property on and off	text	True False Formula	Yes No Formula wizard
Strike out	Turns the "strike out" property on and off	text	True False Formula	Yes No Formula wizard
Color	Font color (see chapter <u>Co</u>	<mark>or</mark>)		



Color

The color property lets you specify the color of the font or the background. For background colors, you must also set the "Background" property to a value > 0, e.g. to "Pattern/Block color".

There is also a dialog for defining the values.

Property	Description	Value	Description
Color	You can define the color in a color dialog.	•	Color dialog
	In the dialog, you can choose the color from a list of fixed predefined colors or specify your own color by means of a formula or a function ("Formula" entry at the end of the list). (1) With the HSL() function, you define the color by specifying the hue value (0- 360), the saturation value (0- 1) and the lightness value (0- 1). (2) The RGB() function defines a color by means of red, green and blue values. Each color portion can have a value between 0 and 255.		Selection of predefined colors and formula wizard



Background / Filling

The background/filling property lets you specify a block color or a gradient.

Property	Description	Value	Description
Filling /	Select the kind of gradient that	0	Transparent
Background	you want and specify the	1	Pattern/block
J	properties for color, mid color,		color
	end color and fading-in color,	2	Horiz.
	depending on background.	3	gradient
		4	Vert.
			gradient
		5	Horiz. 2-part
			gradient
		6	Vert. 2-part
			gradient
	Value 7 only with tables,	7	Partly
	charts, rectangles or circles.	Formula	transparent
			Picture
			Formula
			wizard
Color	Font color (see chapter <u>Color</u>)		



Pattern

The pattern property lets you specify the texture of a color.

Property	Description	Value	Description
Pattern	Choose a pattern here from wide range of predefined patterns. Each pattern is represented by a number. You can specify your own pattern/number by means of a formula or a function ("Formula" entry at the end of the list). This property is only evaluated if "Filling" or "Background" is set to "Pattern/Block color".		Selection of a predefined pattern and formula wizard



Frame

The "Frame" property group defines the frame properties and distances from the frames.

There is also a dialog for defining the values:

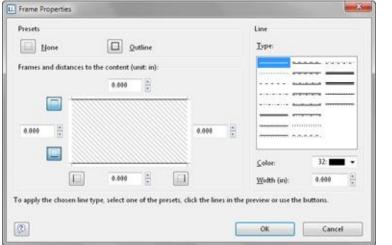


Figure 13.3: Dialog for the frame properties

Property	Description	Value	Description
Frame (default value)	You can define frame properties and distances in a dialog. To apply the selected type of line, color or width, click one of the default settings, the lines of the preview or use the buttons.		Frame dialog
Default frame setting (with table cells)	If set to True, the default frame defined in the table object will be used.	False	Lines No lines Formula wizard
Layout	Describes the layout of the frame lines (only relevant for multi-line frames).	0 1	Circumferen- tial Horiz.

		2 Formula	priority Vert. priority Formula wizard
Left/Top Right/Bottom	Settings for the respective frame line.		
Distance	Distance between content and frame.	Number Formula	Formula wizard
Lines	Visibility of the frame line.	True False Formula	Lines No lines Formula wizard
Color	Line color (see chapter <u>Color</u>).		
Line type	Line type.	Line	Selection of predefined lines (20)
		Formula	Formula wizard
Width	Line width.	Number Formula	Formula wizard



Format

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

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

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

No format Number	Dgcimal places:	Application setting	-
Cunency	Decimal point:	Application setting	
Date Time Date and Time	Thousands separator:	Application setting	•
Percentage	Currency symbol	Application setting	*
Angle Date-/Time Difference	Format for positive values:	Application setting	•
	Format for pegative values:	Application setting	•
	Exponential format:	No	
	No output if <u>v</u> alue is 0 V Always 0 before decimal point		
Preview	(\$12,345.67)		
	(\$12,540,61)		

Figure 13.4: Formatting dialog



Content

Many objects cannot be defined solely by means of the property list. They contain sub-objects (or "content"), such as text objects consisting of several paragraphs. The "Contents" property (if available) opens up a dialog. You will find a description of the respective content dialog accompanying the description of the individual objects.



Text Objects

Text objects let you place text in the workspace. A text object can hold as many paragraphs as you want and they can all have completely different display properties. These paragraphs and their properties present the contents of the text object.

In the paragraph properties dialog, you can edit the individual paragraphs that make up the text object and fill them with content.

See also:

- Object Properties
- Paragraph Properties

▶
combit List & Label

Object Properties

Also see chapter Common Object Properties.

Property	Description	Value	Description
Bottom aligned	Bottom aligned within the object's border. If this option is enabled, the object's text will be output at the lower margin of the object, or otherwise at the upper margin. For this to be possible, the paragraphs must not be larger than the object otherwise the text will be truncated as usual at the bottom or wrapped. This option is very useful e.g. if text is to be output at the lower margin of a page and the length is not known.	True False Formula	Yes No Formula wizard
Rotation	Rotates the object anticlockwise. Please note that only TrueType fonts can be rotated.	0 1 2 3 Formula	0° 90° 180° 270° Formula wizard
Page break	Specifies whether the object can trigger a page break. If this property is enabled, the content will be wrapped to the next page automatically if it exceeds the size of the object. This is an interesting option e.g. with text objects that are to cover several pages. With labels, the next label will only be started when all objects have been printed as a result of this option in the previous label. You might not be able to set this property if page	True False Formula	Yes No Formula wizard



Paragraph Properties

Dranarty	Description	Value	Description
Property Paragraph spacing	Distance to next paragraph ("Paragraph spacing"). You specify the distance in points: To achieve line spacing of 1.5 with a font size of 10 points, enter 5 points. Negative values are also allowed. You should always make the settings under Options > Objects > Object font.	Number Formula	Formula wizard
Alignment	You can specify the alignment in the same way as in your text processing program.	0 1 2 Formula	Left Centered Right Formula wizard
Justified	Block text is justified both right and left. This property only takes effect if the line in question is wrapped at the end of the line. In other words, the line must be longer than the available space, it must consist of more than one word and the "Line wrap" option must be enabled. The last line is presented according to the "Alignment" property.	True False Formula	Yes No Formula wizard
Format	Please consider that the formatting relates to the result of the entire expression. Use the Date\$() and FStr\$() functions in the formula wizard if you only want to format part of the expression (e.g. with text and number within an expression).	True False	Dialog
Blank optimization	The blank optimization option lets you remove unwanted blanks (leading, embedded and	True False Formula	Yes No Formula

	trailing). Consider the following situation: In a label project, you are printing a line with the variables <salutation> <firstname> <name> whereby the individual variables are separated by blanks in each case. If there is no salutation for a data record, this variable remains empty; the following blank would however be printed. First name and name would be shifted by one position (leading blank). If the "Firstname" variable is empty, there would be two spaces (embedded blanks) between "Salutation" and "Name". If all three variables were empty, both blanks would remain (trailing blanks). This means that the line is not empty and would therefore not be suppressed automatically. The "Blank Optimization" option helps in such cases. It can remove leading, embedded and trailing blanks automatically. Multiple embedded blanks are reduced automatically to a single blank.</name></firstname></salutation>	Ττυρ	wizard
Inerasable	Lines that are completely empty are suppressed automatically. In this case, the following lines move upwards. This is normally correct but can be unwanted in some cases, e.g. when filling out forms. With the "Untraceable" option, the line in question remains, even if it is empty after inserting the variables.	False	Yes No Formula wizard
Line wrap	Specifies the behavior if the text is too long for a line. So that long words are not truncated with value "1" (wrap), you can use the "Force wrap"	0 1 2 3 4	Truncate Wrap Shrink Compress Optimal fit

	occurs at character. Value "3" the cha should on extent in legibility. With valu font size i as to fil	ensure that a break fter the last suitable " (compress) reduces aracter spacing and aly be used to a limited order to guarantee "4" (optimal fit) the is varied in such a way Il out the object as by as possible.	Formula	Formula wizard
	Page break	With this property, you can specify that the paragraph is not to be separated in the event that a page	_ True False Formula	Break Keep together Formula wizard
	Force wrap	break is triggered. If a long word cannot be wrapped, a line break will be forced after the last suitable character.		Yes No Formula wizard
Line spacing	between individual You spec points: To 1.5 with a enter 5 p are also a You shou	ing defines the distance the paragraph's lines. cify the distance in achieve line spacing of a font size of 10 points, points. Negative values llowed. ald always make the inder Options > Objects		Formula wizard



Line Objects

You define lines by their alignment, width and type.

See also: • Object Properties



Object Properties

Also see chapter Common Object Properties.

Property	Description	Value	Description
Alignment	Bottom aligned within the	0	Diagonal \
5	object's rectangle. If you hold	1	Diagonal /
	down the Shift key when	2	Horizontal
	changing the size with the	3	Vertical
	mouse, the line will be aligned	Formula	Formula
	either vertically or horizontally.		wizard
Width	Width of line in the unit of	Number	
	measure of the workspace.	Formula	Formula wizard
Line type	Line type selection.	0, 1, 3,	4
		4	predefined
			lines
		Formula	Formula
			wizard



Rectangle Objects

You define rectangles by their border, rounding and shadow.

See also: • Object Properties



Object Properties

Also see chapter Common Object Properties.

Property	Descript	ion	Value	Description
Border		u specify whether the e is to have a border.	0 1 Formula	Transparent Pattern/color Formula wizard
	Color	Color of the border.		
	Width	Width of the border in the unit of measure of the workspace		Formula wizard
Rounding	of the re edge of 1 0% me 100% m	g factor for the corners ctangle in % of the short the rectangle. ans square cornered; eans: the short edge of ctangle is completely lliptical).	Number Formula	Formula wizard
Shadow	Here yo	bu define whether the e is to have a shadow.	0 1 Formula	Transparent Pattern/color Formula wizard
	Pattern	Shadow pattern.		
	Color	Shadow color.		
	Width	Width of the shadow in the unit of measure of the workspace.	Number Formula	Formula wizard



Circles and Ellipses

You define circles and ellipses by their borders and filling.

See also: • Object Properties



Object Properties

Also see chapter Common Object Properties.

Property	Descripti	on	Value	Description
Pie (Flat)	that the displayed	perty lets you specify e ellipse is always l as a circle (centered in t rectangle).	False	Yes No Formula wizard
Border		u define whether the to have a border.	0 1 Formula	Transparent Pattern/color Formula wizard
	Color	Color of the border.		
	Width	Width of the shadow, in the unit of measure of the workspace.	Number Formula	Formula wizard



Picture Objects

Picture objects are used to display the content of fixed files or variables.

The following formats are available: WMF, EMF, BMP, DIB, PCX, SCR, TIFF, GIF, JPEG, PCD, PNG and ICO. As a general rule, you should use the RGB color space (not CYMK). List & Label itself supports transparency in PNG files by using the corresponding Windows functions. In our experience the majority of printer drivers do not support transparency so that reports with e.g. partly transparent PNG files should thoroughly be tested on the actual hard-software combination. If that is not possible we recommend doing without the alpha channel.

If the image is held in a file or a variable, you can select the data source by double-clicking the object.

See also: • Object Properties



Object Properties

Also see chapter <u>Common Object Properties</u>.

Property	Description		Value	Description
Data source	the image	method of determining data source, i.e. via file ula or variable name.	File name Formula Variable	Formula wizard
	File name	Fixed file name: is evaluated if you have selected "File name" as the data source property. You then select the image file that you want by means of the file selection dialog. In this dialog, you can also insert the picture into the project by enabling the ("Embed image in project file" checkbox option. This option copies the image to the project thus making it available even without the external file. In this case, "(embedded)" will be shown as the file name.		Open file dialog
		Relative The path is path relative to the project path.	True False	Yes No
	Formula	If you have selected "Formula" as the data source property, the file name is derived from a formula. The formula must return a	Formula	Formula wizard

		"picture" value type. You can also enter a valid file name. However, this must first be converted to the "picture" type with the Drawing() function.		
	Variable	If you have selected "Variable" as the data source property, the file name is taken from a variable. Select the variable that you want from the drop-down box. The drop-down box lists all "picture" variables defined in your application.	Variable	
	Properties	Depending on your application, a dialog may open up at this point to allow you to define more properties.	a	Opens dialog
	Save as JPEG	Embed the image as a JPEG file if possible (not all file types support this option).	False	Yes No Formula wizard
Original size		Specifies whether the picture is to be drawn in the original size (if it can be determined), or if the size of the picture s frame is to be adjusted to fit.	True False Not defined	Yes No Not defined
	Keep proportions	With this option you can specify whether the height/width proportions are to be kept (True) when inserting the picture or if the picture's frame is to be adjusted, possibly leading to distortion (False).	True False Formula	Yes No Formula wizard
	Alignment	Describes how the image is to be arranged in the available area.	0 1	Centered Next to each other

2 3 4 5 6 7 8 9	(tiled) Left top Left bottom Right top Right bottom Left Right Top Bottom



Barcode Objects

Barcodes can be used for product stickers, price labels, serial numbers and many other purposes. A barcode normally consists of a series of bars and spaces in different thicknesses whereby, depending on the code, the spaces between the bars also hold information.

- Object Properties
- Barcode Content
- Supported Barcode Formats



Object Properties

Also see chapter <u>Common Object Properties</u>.

Property	Description	Value	Description
Bar width	The narrowest bar width in SCM units $(1/1000 \text{ mm})$. $0 = \text{automatic}$ adjustment. Not supported by all barcodes.	Formula	Formula wizard
	Orientation Orientation of the barcode within the object's frame	0 1 2 Formula	Left Centered Right Formula wizard
Bar color	Color of the selected barcode.		
Rotation	Rotates the object anticlockwise	0 1 2 3 Formula	0° 90° 180° 270° Formula wizard
Optimum size	Sets the optimum size for the barcode. This property has an effect when the size changes and can be used with the following formats: EAN 13 (all), DP-Leitcode, DP-Identcode, German Parcel, Postnet and FIM.	True False Formula	Yes No Formula wizard
Font	Font for the barcode text. Is only evaluated if the "Show text" property is enabled.	A	Font dialog
Bar width ratio	The relationship of the different widths of bars or spaces. Not supported by all barcodes.		wizard
Show text	This property lets you specify whether the content of the barcode is to be printed as text.		Yes No Formula wizard



Barcode Content

The content dialog for the barcode object lets you define the barcode more precisely.

- § Choose "Text" if you want to print fixed text as a barcode. Enter the value to be printed in the first part of the input field. Choose the type of barcode that you want in the second field.
- § III There are additional configuration options for many barcodes, e.g. Maxicode. You can edit these options in a further dialog.
- § Choose "Formula" if you want to use a formula as a barcode, and define a valid formula expression with the "Edit" button. The formula must return a "barcode" value type. You can, of course, also enter variables. However, they must first be converted to the "barcode" type with the Barcode() function.
- § Choose "Variable" if you want to print a variable as a barcode. In the drop-down box, you can choose from all available barcode variables.



Supported Barcode Formats

List & Label supports a range of barcode formats. There is normally no need for special printers, fonts etc. � the barcodes are printed directly by List & Label.

- Overview of the General 1-D Codes
- Overview of the General 2-D Codes
- Overview of Post Codes (1-D and 2-D Codes)



Overview of the General 1-D Codes

Name	Formats and permitted characters	
GTIN-13, EAN- 13, UCC-13, JAN-13	Formats: cc nnnn aaaaa cc nnnn aaaaa xx cc nnnn aaaaa xxxxx ppp nnn aaaaa xxxxx ppp nnn aaaaa xx with(normal EAN13) 	
EAN-14, UCC-14	Format: nnnnnnnnnnnnn (14 digits) Permitted characters: [0-9]	
GTIN-8, EAN-8, UCC-8, JAN-8	Formats: nnnnnn, nn nnnnn (= character code chr(124)) Permitted characters: [0-9] Each character is also 7 bar widths wide, a code shou minimum width of (8*7+11)*0.3 mm=2.01 cm. Ideal size (bar symbol) nominal size SC2: dx : 22.1 mm, dy: 19.9 mm Minimum offset that should be kept free around the symb left: 2.3mm, top: 0.3mm, right: 2.3mm, bottom: 0.0r otherwise 0.3 mm) The text may partially exceed this area.	
UPC-A	Format: c nnnn aaaaa, cnnnnnaaaaa	

		c = number system nnnn = company code aaaaa = article code = character code chr(124) alculated and attached autc hs wide, a code should then h =2.88 cm.
UPC-E	Format: c nnnnn, nnnnnn with	
		alculated and attached autc hs wide, a code should then h
2-of-5 Industrial	Format: any Permitted characters: [0-9] A code is (14*number of cha	aracters+18) bar widths wide.
2-of-5 Interleaved (ITF)	Format: any, must have an Permitted characters: [0-9] A code is (9*number of char	
2-of-5 Matrix	Format: any Permitted characters: [0-9] A code is (10*number of cha	aracters+18) bar widths wide.
2-of-5 Datalogic	Format: any Permitted characters: [0-9] A code is (10*number of cha	aracters+11) bar widths wide.
Codabar	Every character is either 2*	ame code [A-D], n = [0-9], [- $\$$: 3+6*1 (characters '0''9', '-', ''D') bar widths wide. The ch ed with the text.
CODE11	Format: any Permitted characters: [0-9],	[-]
		the length, 1 or 2 check dig f 2 check digits if the length
Code39, 3-of-9, Alpha39	calculates only 1 instead of	f 2 check digits if the length

	text has (16*number of characters $\circledast1$) bars.
Code 39 with CRC	Format: any Permitted characters: [A-Z], [0-9], [/\$%+*]
Code 93 (simple and extended)	Code 93 is an extension of Code 39 and covers the com of the ASCII Allowed characters, including NULL. This mus as chr\$(255). It contains two check digits that are automatically gene Label. The characters consist of 9 bar widths, that each have spaces. There are two options for the extended code: a) transfer of the shift character from the host program as \$ chr\$(254) % chr\$(253) / chr\$(252) + chr\$(251) b) transfer of the desired character, L&L adds the a character.
Code128	Format: any Permitted characters: any Determine the code set that should be used: Use one of the following codes as a start character: chr\$(135) start with code A chr\$(136) start with code B chr\$(137) start with code C In order to switch between different code sets within th can use the usual control characters: Starting from code A to B: chr\$(132) C: chr\$(131) Starting from code B to A: chr\$(133) C: chr\$(131) Starting from code C A: chr\$(133) B: chr\$(132) Example: <subset b=""> "RL" <subset c=""> "04432476" <subset b=""> "(Barcode(Chr\$(136)+"RL"+chr\$(131)+"04432476"+chr\$(] "Code 128")</subset></subset></subset>
GS1 128, EAN128	The special FNC1 character following the start character u the EAN128 code. Special characters must be replaced as NUL: chr\$ (255) FNC1: chr\$ (254) FNC2: chr\$ (253) FNC3: chr\$ (252) FNC4: chr\$ (251)
GS1 DataBar	Format: nnnnnnnnnnn (13 digits)

(Limited, Stacked, Stacked Omnidirectional, Stacked Truncated)	Permitted characters: [0-9]		
GS1 DataBar Expanded	Format: Data begins with AI. Max. 74 numeric/41 characters Permitted characters: any		
IM (4CB/4- CB/USPS4CB)	Intelligent Mail Barcode (US Postal Services). Alternative r Solution or 4-State Customer Barcode). Format: 20, 25, 29 or 31 digits Permitted characters: [0-9]		
ISBN	Format: nnn nnnnnnnnn (12-digit, no check digit) Permitted characters: [0-9]		
MSI	Format: any Permitted characters: [0-9], [A-F]		
Pharma-Zentral- Nummer	Format: nnnnnn (6 digits) Permitted characters: [0-9]		
SSCC/NVE	Format: {nn}nnnnnnnnnnnnnnn (17 or 19 digits) Permitted characters: [0-9]		



Overview of the General 2-D Codes

Name	Formats and permitted characters
Aztec	Format: any Permitted characters: any
Datamatrix	In order to enter non-printable characters (binary data) in the barcode text, they must be packaged in a special string. The data is inserted using the ~dNNN string, whereby NNN stands for ASCII-Code. Example: DEA~d065~d015~d000~d247~d220 (~d065 stands for "A") To represent an EAN data matrix, you can code the special FNC1 character as ~1. The Datamatrix symbology uses the ECC 200 error correction code. Further options like size, module format and encoding can be set in the barcode's options dialog. Click the options symbol in the barcode content dialog to enter this dialog.
PDF417	 Can display all available and non-printable characters. In order to enter non-printable characters (binary data) in the barcode text, they must be packaged in a special string. The data is inserted using the "{binary:nn}" string, whereby nn stands for any sequence of two-character hexadecimal numbers. This is especially important if Maxicodes are to be created according to UPS specifications; the special characters needed for this can be entered in this way: In order to pack a Null and a Backspace (BS) character in the data, use "{binary:008}". Use "Hallo{binary:00a}World" to include a line break.
OB Codo	Format: any

Permitted characters: all characters In order to enter non-printable characters (binary data) in the barcode text, they must be packaged in a special string. The data is inserted using the ~dNNN string, whereby NNN stands for ASCII-Code. Example: ~d065 stands for the letter "A".



Overview of Post Codes (1-D and 2-D Codes)

Name	Formats and permitted characters
DP-Identcode	Formats: nn.nnnnnn, nn.nnnnn, nn.nnnn, nn.nnnn.nn
DP-Leitcode	Format: nnnnn.nnn.nn Permitted characters: [0-9] A code is (9*number of characters+9) bar widths wide. Width: 37.25 mm - 67-5 mm (at least 5 mm light zone right and left). Height: 25 mm. Check digit is calculated automatically; relation: 4:9; special �2 of 5 IL� code.
FIM	Formats: A, B, C Permitted characters: [A-C] Minimum size: 1/2" * 5/8". The FIM barcode is always printed in the size specified by the US Postal Office. This means that it might extend beyond the available object border.
German Parcel	Format: any, must have an even number of characters. Permitted characters: [0-9] A code is (14*number of characters+18) bar widths wide. Relation: 1:2
Japanese Postcode	Japanese Postcode. Format: Postcode as nnn-nnnn, then max. 13 character address Permitted characters: n=[0-9], address=[A- Z], [0-9], [-]

Maxicode	Can display all available and non-printable characters. In order to enter non-printable characters (binary data) in the barcode text, they must be packaged in a special string. The data is inserted using the "{binary:nn}" string, whereby nn stands for any sequence of two- character hexadecimal numbers. This is especially important if Maxicodes are to be created according to UPS specifications; the special characters needed for this can be entered in this way. Example: in order to pack a Null and a Backspace (BS) character in the data, use " {binary:008}"). Example: use "Hallo{binary:0d0a}world" to include a line break.	
Maxicode/UPS	Format: Formatting according to UPS specifications Permitted characters: all characters	
Postnet	Formats: nnnnn, nnnnn-nnnn, nnnnn-nnnnn Permitted characters: [0-9] Minimum size: 1.245" * 4/16" (10-digits). Bar distance at least 1/24". Error correction digit will be appended automatically. This bar code is automatically printed in the right size if the object is larger than the maximum size of the barcode.	
RM4SCC, KIX® Royal Mail with CRC	Permitted characters: [A-Z], [0-9], [a-z] Permitted characters: [A-Z], [0-9], [a-z] Format: Either just the post code is coded (e.g. LU17 8XE) or the post code with a supplementary "Delivery Point" (e.g. LU17 8XE 2B). The maximum number of characters that can be used is therefore limited to 9.	



Report Container Object

A report container can hold multiple table objects, cross tab objects and chart objects.

- Object Properties
- Element Properties



Object Properties

Also see chapter <u>Common Object Properties</u>.

Property	Description	Value	Description
Page break before	If the condition returns "True", a page break will be triggered before printing the report container.	False	Break No break Formula wizard
Default font	Default font setting for the elements.	1	Font dialog
Column count	Number of columns in the container. Note: the "LL.CurrentTableColumn" field returns the index of the current column.	Formula	1-column 2-column 3-column 4-column 5-column Formula wizard
	1 5	Number Formula	Formula wizard



Element Properties

To display the properties of the elements, mark the element in the "Report Structure" tool window.

Also see chapter Common Object Properties.

Property	Description	Value	Description
Name	Name of the element (stored in the "LL.CurrentContainerItem" variable).	Name	
Separator ticks (only tables)	To align (sub) tables exactly with one another, additional tick marks can be shown on the ruler.	True False	Yes No
Spacing (only charts and crosstabs)	Distance of the object (including t container's print area.	frame) fro	m the report
Distance before	Distance from the previous element (not required if the element starts at the beginning of the page.	Number Formula	Formula wizard
Output height (only charts)	Height of the object (including frame).		
Sort order (only tables and charts)	Sort order for the table. All available sort orders are offered in a selection list (stored in the "LL.SortStrategy" variable).	List	
Pagebreak before	If the condition returns "True", a page break will be triggered before printing the object. If you have multiple multi-column objects, a page break is triggered automatically after an object if the column counts for the objects are different (e.g. 2- column table followed by a 3- column table) and if there would	True False Formula	Break No break Formula wizard

	be insufficie following obj	ent room for the ect.		
Column count	Number of object. If you hav column object triggered au object if the the objects a column table column table be insufficie following obj Note: "LL.CurrentTa	columns in the ve multiple multi- cts, a page break is tomatically after an column counts for are different (e.g. 2- e followed by a 3- e) and if there would ent room for the ect. The	1 2 3 4 5	Container default 1-column 2-column 3-column 4-column 5-column Formula wizard
	Distance	Spacing of the columns in the object.	Number Formula	Formula wizard
	Column break condition (only tables)	A column break will be triggered if the result of the condition is "True".		Yes No Formula wizard
	Column break before	A column break will be performed before the object is output.	True False Formula	Yes No Formula wizard



Table Objects

Table objects are elements of the report container.

- Object Properties
- Line Properties
- Group Line Properties
- Column Properties



Object Properties

Also see chapters <u>Common Object Properties</u> and <u>Report</u> <u>Container Object</u>.

Property	Description	Value	Description
Fixed size (not available for sub- tables)	The "Fixed height" property lets you specify that the size of the table is not to be adjusted automatically when it has fewer data lines than the available space in the table object, after the field names are replaced with content. If the property is disabled, the end of the table moves upwards automatically. If a footer line is defined, "fixed size" will cause it to be separated from the rest of the table by a space of at least one line. This space accommodates the frame as defined in the data line definition 1. Otherwise, the footer line will appear immediately below the table. Please consider that any objects interlinked with the table can only adjust their position automatically to correspond with changes in the table size if "fixed size" is disabled.	False	Yes No Formula wizard
	Separators If this property is fixed enabled, the separators are also drawn through the empty area of the table between the last data line and the footer line. If the option is disabled, the separators are	False	Yes No Formula wizard

	t - 6	only drawn as far as the last data line. This property is only available with "fixed size" tables.		
Background	of the table. report cont applications).			
Default frame	frame.	e for the table's		
Default font		e for the table's font e for sub-tables).	1	Font dialog
Pagebreak condition	specify a cor page break soon as the c If "Pagebreak True, a triggered after specifies that	property, you can adition that causes a after a data line as ondition is met. c condition" is set to page break will be er each line. �False a page break is only ed when necessary.	True False Formula	Yes No Formula wizard
Column count	Note: "LL.CurrentTa	lumns in the table. The bleColumn" field ndex of the current	0 1 2 3 4 5 Formula	Container default 1-column 2-column 3-column 4-column 5-column Formula wizard
	Distance	Spacing of columns in the table.	Number Formula	Formula wizard
	Column break condition	If the result is "True" when a data line is output, a column break will be triggered.	True False Formula	Break No break Formula wizard
	Column break before	A column break will be performed before the object is output.		Break No break Formula wizard
Pagebreak before	page break before printin	ion returns "True", a will be triggered g the object. nultiple multi-column	True False Formula	Break No break Formula wizard

objects, a page break is triggered automatically after an object if the column counts for the objects are different (e.g. 2-column table followed by a 3-column table) and if there would be insufficient room for the following object.

Data lines				
	Suppress	When you enable the "Data lines.Suppress" object property in tables, all data lines are completely suppressed. This option is particularly useful in combination with the "Force Sums" option. The latter option specifies that totals are also calculated when a data line is not printed. By combining both options and using groups and sum variables, you can print more interesting statistics.	True False Formula	Yes No Formula wizard
	Force sums	Sum variables are calculated even if you suppress data lines.	False	Yes No Formula wizard
	Zebra pattern	The "Zebra pattern" option in the "Data line" field specifies whether data lines are to have alternating background colors. This can improve readability, especially with		

	_	large tables.		
	Keep rows together	In the event of a page break, data lines are kept together as far as possible, i.e. they are printed together on the next page.	False	Yes No Formula wizard
Footer lines				
	Keep rows together	In the event of a page break, footer lines are kept together as far as possible, i.e. they are printed together on the next page.	False	Yes No Formula wizard
Group footer lines				
	Also empty groups	Output group footer lines even if the groups are empty.	True False Formula	Yes No Formula wizard
	Keep rows together	In the event of a page break, group footer lines are kept together as far as possible, i.e. they are printed together on the next page.	False	Yes No Formula wizard
Group header lines				
	Keep together	Where possible, a group header line will not be separated from the following data line because of a page break.		Yes No Formula wizard
	Keep rows together	In the event of a page break, group header lines are kept together as far as possible, i.e. they are printed	False	Yes No Formula wizard

together on the next page.



Line Properties

Also see chapter Common Object Properties.

Property	Description	Value	Description
Display in Designer	With this property, you can hide lines in the workspace � this is very useful if you have a lot of line definitions.	True False Formula	Yes No Formula wizard
Default font	You can set the default font for the entire table row. Newly inserted columns are created with this font.	1	Font dialog
lgnore empty lines	Ignore lines where all fields are empty.	True False Formula	Yes No Formula wizard
Spacing	Here you define the top, bottom, right and left spacing of the line. The "top" or "bottom" values result in a corresponding space between the individual table rows. With the "left" and "right" spacing values, you can specify the margin in relation to the table object, i.e. you can indent lines or columns.	Formula	Formula wizard



Group Line Properties

Group lines have the following additional properties:

Property	Description	Value	Description
Group sums	The selected sum variables are set to "0" when the condition for the group line is met. This setting is useful to create so-called group sub-totals, for example to add up the prices of all articles in a particular article group.		Dialog
Group by	This expression represents a key. Whenever the result of the expression changes, a group change is triggered.	Formula	wizard
Repeat as header	Outputs the group header again after a page break.	True False Formula	Yes No Formula wizard
Break before	A page break will be triggered before outputting the group header, i.e. each group begins on a new page. If several group lines are printed with this option at the same time, they will appear one after the other on the new page. The RemainingTableSpace() function lets you perform conditional breaks before outputting group headers, e.g. "If there are only 2 inches left on the page then break before outputting the object". You will find more information in	True False Formula	Yes No Formula wizard
	chapter Overview of Functions.		
Break after	chapter <u>Overview of Functions</u> . A page break will be triggered after outputting the group footer lines, i.e. each group	True False Formula	Yes No Formula



Column Properties

The column properties correspond with the properties of the respective object type, with some table-related restrictions. Text and RTF text columns are special cases. A column property lets you switch between these two text variations later on. The property list also changes accordingly depending on this property.

Also	see	chapter	<u>Common</u>	Obje	ct Pro	perties.

Property	Description	Value	Description
Options (only with some field types)	Opens the "Content" dialog for the relevant object type.		Content dialog
Drilldown links	Opens the dialog for editing the drilldown links. A drilldown report can also be started from the preview.		Drilldown dialog
Link	Link that is accessed when you click the object (in the real data preview, with PDF or HTML export). Example: file://c:\users\public\x.log or URL		Formula wizard
Save as JPEG (only with pictures)	Embed the image as a JPEG file if possible (not all file types support this option).	True False Formula	Yes No Formula wizard
Rotation	Rotates the object anticlockwise. For example, you can rotate the column title or barcode by 90° with this function.	0 1 2 3 Formula	0° 90° 180° 270° Formula wizard
Format	Please consider that the formatting relates to the result of the entire expression. Use the	True False	Dialog

Date\$() and FStr\$() functions in						
the formula wizard if you only						
want to format part of the						
expression (e.g. with text and						
number within an expression).						

Bar width (only with barcodes)	The narrowest bar width in SCM units $(1/1000 \text{ mm})$. $0 =$ automatic adjustment. Not supported by all barcodes.	Formula	Formula wizard
	Orientation Orientation of the barcode within the object's frame	0 1 2 Formula	Left Centered Right Formula
Bar color (only with barcodes)	Color of the barcode.		
Bar width ratio (only with barcodes)	The relationship of the different widths of bars or spaces. Not supported by all barcodes.	Formula	Formula wizard
Show text (only with barcodes)	only with whether the content of the		Yes No Formula wizard
Background	The background of the columns.		
Text format	Text format For presenting the text column differently.		Normal text RTF Text
Alignment (pictures) Describes how the image is to be arranged in the available area.		0 1 2 3 4 5 6 7 8 9	Centered Next to each other (tiled) Left top Left bottom Right top Right bottom Left Right Top Bottom
Vert. Alignment (barcodes)	Vertical alignment of the content in the available space.	0 1 2 Formula	Top Centered Bottom Formula wizard

Alignment (text)		ment. Decimal means ers are aligned by their ints.	0 1 2 3 Formula	Left Centered Right Decimal Formula wizard
	position	The position of the decimal point within the field, measured from the left edge of the previous frame.		Formula wizard
Justified	the left (blo		True False Formula	Yes No Formula wizard
Width	get an erro of the colu	of the column. You will or message if the sum mn widths exceeds the of the table object.	Number Formula	Formula wizard
Fit	is too long To prevent truncated you can to option to occurs aft character. reduces th and should	he behavior if the text for one line. long words from being with value "1" (wrap), use the "Force wrap" ensure that a break ter the last suitable Value "3" (compress) he character spacing d only be used to a extent in order to legibility.	0 1 2 3 Formula	Truncate Wrap Shrink Compress Formula wizard
	Widow/ Orphan control	Prevents widow and orphan lines. The last line of a paragraph is referred to as a widow line if it is also the first line of a new column or page. If a new page or column is triggered after the first line of a new paragraph, this line would appear alone at the end of the page or column. In this case, it is referred to as an orphan line. If a long word cannot	True False Formula	Yes No Formula wizard
	TUICE		nue	103

	wrap	be wra break v after the characte	vill be e last s	forced	False Formula	No Formula wizard
Height (not text or RTF text).	Fixed height of the field, the content is scaled (0: no fixed height). The highest column determines the overall height of a table row.					Formula wizard
Fixed height (text or RTF text).	Fixed h irrespectiv Excess te fixed heig determine table row.	ve of ext is dis ht). The h	carded nighest	ontent. (0: no column		Formula wizard
Blank Optimization	Leading a removed.	and doub	le spac	es are	True False Formula	Yes No Formula wizard
Line spacing (only with text)	Spacing b	etween th	ne text l	ines.	Number Formula	Formula wizard
Frame	Specifies character the ind Together the "top margins o the table	ividual with the " and letermine	d marg table selecte "bottom	cells. d font, n" cell	 Number Formula	Dialog Formula wizard



Chart Objects

I Chart objects are elements of the report container.

See also:

- Object Properties
- Special Fields
- Pie Chart: Data Source
- Pie Chart: Pie Segment
- Pie Chart: Diagram
- Bars/Lines: Category and Series Axes
- Bars/Lines: Value Axis Settings
- Bars/Lines: Diagram
- Object (All Diagram Types)
- Colors (All Diagram Types)

▶
combit List & Label

Object Properties

See chapters <u>Common Object Properties</u> and <u>Report</u> <u>Container Object</u>.



Special Fields

	_
Fields	Purpose
LL.ChartObject.ArcIndex()	Returns the index of the current pie segment. The largest pie segment has index 1, the second largest has index 2 and so on.
LL.ChartObject.ArcPerc()	Returns the percentage
LE.Chartobject.Arcreic()	share of the current pie segment.
LL.ChartObject.ArcTotal()	Returns the absolute value of the total data volume with pie charts.
LL.ChartObject.ArcTypeIsOthers()	Returns True, if the
	current pie segment is the "other" segment.
LL.ChartObject.ArcValue()	Value of the pie segment.
LL.ChartObject.AxisCoordinate()	Describes the content of the source value.



Pie Chart: Data Source

On the "Data source" tab, you can specify the coordinate values for the data.

values ior i			
Property	Description	Value	Description
Coordinate value	Choose the data source for the pie segments (name of the customer or employee, name of the product category, month or quarter in the case of dates etc.).	Formula	Formula wizard
Minimum share	Especially if you have a lot of values that have small shares, it's sometimes a good idea to group them together under "others". You can define threshold values here which specify when individual segments are to be grouped together in one segment.		wizard
Sort coordinates	Specifies whether the pie segments are to be sorted (alphanumeric or alphabetical).	True False Formula	Yes No Formula wizard
Number of sample records	The Designer does not yet have the "real" data that appears in your chart when it is printed. So that you can still get a picture in the Designer preview of how your chart will look, you can use this option to specify the number of pie segments that are to be displayed in the Designer.	Number	Formula wizard
Filter	You can define a filter condition here. Only data records that fulfill the condition will then be used for the diagram. All data will be used if you select "True".	True False Formula	True False Formula wizard
Coordinate	Defines the text for the coordinate label on the pie	Formula	Formula wizard

label	segment.		
	Fixed font size	The font size is to beTruefixed (otherwise it will become smaller as necessary in order to prevent overlapping).Formula	Yes No Formula wizard
Coordinate label "Others"	coordinat	the text for the Formula e label on the pie for the data grouped as	Formula wizard
Legend	Position o	f the legend.	None At chart



Pie Chart: Pie Segment

The "Pie Segment" tab lets you make settings for calculating and presenting the pie segment.

Property	Description	-	Value	Description
Coordinate value	the coord determines segment (to	pecify the formula for linate value that the size of the pie stal turnover, average mber of sales etc.).	Formula	Formula wizard
Width	Pie width in	percent.	Number	Formula wizard
Explosion offset	accentuated forward out describes th the pie segn	ual pie segments are by bringing them of the pie. The value ne distance by which nent is to be raised (as e of the pie's radius).	Number	Formula wizard
Label on object	Specifies wh	nether a label is to be e pie segment.	0 1 Formula	No Yes Formula wizard
		Label text on the pie segment. LL.ChartObject.ArcPerc returns the percentage share of the current pie segment.	Formula	Formula wizard



Pie Chart: Diagram

The "Diagram" tab lets you make settings for the appearance of the pie.

	<u> </u>		<u> </u>
Property	Description	Value	Description
Color mode	The pie segments are marked in different colors so that the individual values can be differentiated more easily. Color settings by means of the "Colors" tab.	0 1 Formula	Monochrome pie Colored pie Formula wizard
Perspective	This property lets you choose whether the diagram is to be created with a slight or a strong perspective. Alternatively, you can also use a simple parallel projection.	0 1 2 Formula	None Slight perspective Strong perspective Formula wizard
Perspective gradient	The perspective gradient produces a brightness gradient across the surface of the pie chart. Gradient in percent.	Number	Formula wizard
	Accentuate Raise the edge of frame the pie	True False Formula	Yes No Formula wizard
X axis rotation angle	The rotation angle upwards around the x-axis in degrees, maximum 90° (vertical). Determines the horizontal positioning of the pie. You can also specify this angle by means of the rotation buttons that appear on the workspace when you select a diagram.	Number	Formula wizard
Y axis rotation angle	The rotation angle in degrees around the center of the pie, anticlockwise. You can also specify this angle by	Number	Formula wizard

	means of the rotation buttons that appear on the workspace when you select a diagram.		
Separator lines	Display separator lines between the pie segments.	False	Yes No Formula wizard



Bars/Lines: Category and Series Axes

If you have decided in favor of a three-axis diagram, you have both of these axes at your disposal (x axis and x axis). With two-axis charts (e.g. a simple bar chart), you only need the category axis (x axis).

Property	Description	Value	Description
Coordinate value	Here you choose the data source for the coordinate, e.g. "name" with persons or "month" with dates.	Formula	Formula wizard
Sort coordinates	Specifies whether the coordinates are to be sorted (alphanumeric or alphabetic).	True False Formula	Yes No Formula wizard
Number of sample records	The Designer does not yet have the "real" data that appears in your chart when it is printed. So that you can still get a picture in the Designer preview of how your chart will look, you can use this option to specify the number of sample records that are to be displayed in the Designer.	Number	Formula wizard
Filter	You can define a filter condition here. Only data records that fulfill the condition will then be used for the diagram. All data will be used if you select "True".	True False Formula	True False Formula wizard
Axis label	Defines the text for the axis label.	Formula	Formula wizard
	Rotation Rotation of the axis label in degrees.	Number	Formula wizard
Coordinate label	Defines the text for the coordinate label or the legend.	Formula	Formula wizard
	Rotation Rotation of the coordinate label in degrees (with long	Number	Formula wizard

	texts).		
Fixed	The font size is to be		Yes
font siz	(No
	become smaller if		Formula
	necessary in order to		wizard
	prevent overlapping).		
Legena	of the legend.		None
	ou select "At axis", the	-	At axis
	re placed directly on the		Тор
	otherwise the label is		Left
•	on the right, left, upper or		Right
lower s	de of the chart.		Bottom
to can also determine formula the values This me data re diagran Select drop-do change have t individu You car	ans that you define the dif ed value/target value/actua ord and can show them pa	ans of rows ferent rows al value) wi irallel e.g. i purce" entry perties. T ies axis ar erty. You ow Definitio ferently fo	instead of (e.g. th a single n a line y from the his option d you also define the ons" dialog.



Bars/Lines: Value Axis Settings

On the "Value axis" tab, you can make settings for calculating and presenting the value axis of a bar chart or line diagram.

Primary axis/secondary axis: These charts support a second value axis. The second axis is enabled on the "Diagram" tab. Use the drop-down box to switch to the properties for the respective axis.

Dreneutri	Description	Value	Description
Property	Here you specify the formula for		Formula
Coordinate value	the coordinate value (total	ronnaia	wizard
Value	turnover, average turnover,		
	number of sales etc.).		
Axis scale	Type of axis scaling.	0 1	None (linear) Logarithmic,
		Ŧ	base 10
			(decimal)
		2	Logarithmic,
			base 2 (binary)
		Formula	(binary) Formula
		i onnaid	wizard
Maximum	Do you want the value axis to	True	Yes
Value	continue until it reaches a certain		No
Automatic	maximum value or do you want to determine the end value	Formula	Formula wizard
	automatically?		wizaru
	You can limit the maximum		
	height of the displayed area e.g.		
	to cater for "anomalies. If your		
	values contain extremely high peaks, you can cap them by		
	peaks, you can cap them by		
	setting a maximum value and		
	setting a maximum value and show the progression of the		
	show the progression of the "small" values more clearly. If		
	show the progression of the		

	that all values are displayed. Threshold Maximum axis value	Number	Formula wizard
Minimum Value Automatic	Do you want the value axis to start at a certain minimum value or do you want to determine the start value automatically? You can limit the minimum height of the displayed area, e.g. to cater for "anomalies. If you keep the default value "No", the diagram will be adapted so that all values are displayed.	True False Formula	Yes No Formula wizard
	Threshold Minimum axis value	Number	wizard
Width	Bar/line width in percent	Number	Formula wizard
Label on object	Specifies whether a text is to be output on the objects.	0 1 Formula	No Yes Formula wizard
	Content Label text on the object.	Formula	Formula wizard
Coordinate lines	Specifies whether coordinate lines are to be drawn on the background.	True False Formula	Yes No Formula wizard
Zebra pattern	Specifies whether the background is to be output in a zebra pattern.		Transparent Pattern/block color Horiz. Gradient Vert. Gradient Horiz. 2-part gradient Vert. 2-part gradient Partly transparent
Axis label	Defines the text for the axis label.	Formula	Formula wizard
	Rotation Rotation of the axis label in degrees.	Number	Formula wizard
Coordinate label	Defines the text for the coordinate label or the legend.	Formula	Formula wizard
	Rotation Rotation of	Number	Formula

		coordinate degrees.	label	in	wizard
	Fixed font size	The font siz fixed (other will becom as necessar to overlapping	erwise e small ry in ord preve	it False ler Formula ler	Yes No Formula wizard
Coordinate tick distance	Calculation between tw	of the o coordinate	distan ticks.	False	Automatic Manual Formula wizard
Legends	axis. If you	the legenc select "At be placed d	axis", t	he	None At axis
Number of ticks		per of su by tick marks pordinate ma	s betwe		Formula wizard



Bars/Lines: Diagram

The "Diagram" tab lets you make settings for the appearance of bar charts and line diagrams.

Property	Description	Value	Description
Secondary axis	Supports a secondary axis on the left hand side. The axis properties are set on the "Value Axis" tab. You select the respective axis from a drop- down box.	True False Formula	Yes No Formula wizard
	Axis Specifies which assignment value axis the value is assigned to.	0 1 Formula	Primary axis Secondary axis Formula wizard
Alignment	Alignment of the graphic elements, e.g. for a horizontal bar chart.		Left to right Bottom to top
Illuminated	Specifies whether the chart is to be illuminated.	True False Formula	Yes No Formula wizard
Color mode	Specifies which axis determines the color.	0 1 2 Formula	Monochrome x axis y axis Formula wizard
Background color	Background behind the diagram.		Transparent Pattern/block color
lsotropic	Specifies that both data axes (x and y axis) use the same units.	True False Formula	Yes No Formula wizard
Perspective	This property lets you choose whether the diagram is to be created with a slight or a strong	0 1	None Slight perspective

	perspective. Alternatively, you can also use a simple parallel projection.	2 Formula	Strong perspective Formula wizard
X axis rotation angle	The rotation angle upwards around the x axis in degrees, maximum 90° (vertical). Determines the horizontal positioning of the diagram in the available space. You can also specify this angle by means of the rotation buttons that appear on the workspace when you select a diagram.	Number	Formula wizard
Y/Z axis rotation angle	The rotation angle in degrees around the center of the diagram, anticlockwise. You can also specify this angle by means of the rotation buttons that appear on the workspace when you select a diagram.	Number	Formula wizard
Separator lines	Display separator lines between the pie segments.	True False Formula	Yes No Formula wizard



Object (All Diagram Types)

On the "Object" tab you will find the settings for the title and the background.

Property	Description	Value	Description
Title	You can specify the title of your diagram here. It will then be displayed at the upper margin of the object. You can also select a formula with the formula button. The "Font" button lets you change the font for the title. Click it with the left mouse button to open a font selection dialog; a right-click resets the font to the default value for the object.	Formula	Formula wizard
Title position	Position of the diagram's title.	0 1 Formula	Top Bottom Formula wizard
Background	Select the color that you want to use for the background of the available area. You can also make it transparent. You can select a color in the upper drop- down box or click "" to open a standard color selection dialog.	True False Formula	Yes No Formula wizard
	Filling Filling for the available area		Transparent Pattern/block color Horiz. Gradient Vert. Gradient Horiz. 2-part gradient Vert. 2-part gradient Partly

					transparent Picture
Border	Border available a	for rea	the		Transparent Pattern/block color
Shadow	Shadow available a	for rea	the		Transparent Pattern/block color
Rounding	Rounding percent corners available 0=rectange (square 100=ellipti edge is rou	cori cal (the the area: ners),	Number	Formula wizard



Colors (All Diagram Types)

Property	Description	Value	Description
Assigned colors	You can assign fixed colors to particular axis values. If you click the "New" button, you can create a new assignment.	Color	Formula wizard
Unassigned colors	Specifies the colors and color sequences for the data rows that are not specified by the "Assigned colors".	Color	Color dialog



Gauge Objects

Gauges can be placed in table lines or as an object.

See also: • Object Properties • Content ▶
combit List & Label

Object Properties

See chapter Common Object Properties.



Content

You define the appearance of the gauge on the "Content" tab.

CUD.			
Property	Description	Value	Description
Background	Filling properties (background of the gauge).		
	Color. Color of the background.		Color
	Fading-in factor. Value ranging from 0 (transparent) to 100 (opaque).	Value	Formula wizard
Pointer options			
	Color. Color of the pointer.		Color
	Size factor. Specifies the size of the pointer in relation to the scale range.	Value	Formula wizard
	Display Range. Describes the percentage range of the available area (the radius or the width) in which this element is displayed.	Formula wizard	
Glass properties			
	Color. Glass color.		Color
	Fading-in factor. Value ranging from 0 (transparent) to 100 (opaque).	Value	Formula wizard
Scale range rotation angle	Clockwise rotation angle for the scale $(0^\circ = \text{down})$.	Value	Formula wizard
White space before scale range	Specifies the free space between the beginning of the gauge and the beginning of the scale (max. 50%).	Value	Formula wizard
White space after scale	Specifies the free space between the end of the	Value	Formula wizard

range	gauge and the end of the scale (max. 50%).		
Tickmarks	Scale type	0 1 2 3	None Scale type 1 Scale type 2 Scale type 3
	Display range. The "Minimum" and "Maximum" values describe the percentage range of the available area for the tickmark.	Value	Formula wizard
	Color. Color of the tickmark.		Color
	Tickmark width. Specifies the width of the scale tickmark (percentage value).	Value	Formula wizard
	Detail level. Specifies the maximum tickmark level to be output. 0=top level only.	Value	Formula wizard
	Filling.		
	Intermediate tick position. Specifies where the intermediate tickmarks are to be placed.		
	Calculation type. Calculation type of the coordinate tickmark distance (ticks).	True False Value	Automatic Manual Formula wizard
	Distance. The user-defined distance between two coordinate ticks.	Value	Formula wizard
Scale labels	Scale label properties.	True False	Show Hide
	Display range. The "Minimum" and "Maximum" values describe the percentage range of the available area for the tickmark.	Value	Formula wizard
	Rotated. Specifies whether the font is rotated.	True False	Yes No
	Size adjustment. Specifies whether the font size may be reduced so that the lettering	True False	Yes No

	fits the area.		
	Alignment. Text alignment.	0 1 2 Value	Left Centered Right Formula wizard
Signal ranges	The properties of the signal ranges. You can give the ranges different colors, e.g. to signalize an optimum range. You define the start value, start color, end value and end value of the different ranges in a dialog.	True False 	Show Hide Dialog
	Display range. The "Minimum" and "Maximum" values describe the percentage range of the available area for the tickmark.	Value	Formula wizard
Scale Labels	List of labels. You can open a dialog to define the rotation, frame size, background, font, format and position of the individual labels. You specify the position in relation to the area of the gauge (measured from left to right). For example, a vertical and horizontal position of 50% each will position the label precisely in the middle.	True False	Show Hide Dialog
Minimum value	Minimum value of the scale.	Value	Formula wizard
	Fit to distance The minimum value is adjusted automatically to fit the coordinate tickmark distance.	True False	Yes No
Maximum value	Maximum value of the scale.	Value	Formula wizard
	Fit to distance The maximum value is adjusted automatically to fit the coordinate tickmark distance.	True False	Yes No
Start value	Defines where the bar for displaying the value starts. 'Automatic' means: if the scale is exceeding 0, it is 0,	True False	Automatic Manual

	otherwise it is the minimum value.	
Value	Value to be indicated by the Valu pointer.	ie Formula wizard



Crosstab Objects

Crosstab objects are elements of the report container.

See also:

- Object Properties
- Special Functions
- Cell Properties
- Properties for the Crosstab Area

▶
combit List & Label

Object Properties

See chapters <u>Common Object Properties</u> and <u>Report</u> <u>Container Object</u>.



Special Functions

Function	Purpose
Crosstab.Cells.Max()	Returns the largest or smallest
or	value in the entire crosstab.
Crosstab.Cells.Min()	
Crosstab.Cells.Avg()	Returns the average value for the entire crosstab.
Crosstab.Col\$()	Returns the description of the
or Crosstab.Row\$()	column or the row for the
	current cell.
Crosstab.Value()	Returns the content of the cell (as a number).
Total()	You can use this function for calculations across all cells. Otherwise, calculations are always made across all values that affect the respective cell.
Join\$()	Returns a collection of strings, separated by a particular character.



Cell Properties

Select the respective column in the object dialog on the "Cell definition" tab. To select multiple cells, hold down the CTRL key or you can draw a border around the cells with the mouse.

nouse:			
Property	Description	Value	Description
Value	Formula for the value of the cell. This will be evaluated by the Crosstab.Cells functions.	Formula	Formula wizard
Displayed contents	Text to be displayed in the cells. This can differ from the value specified in the "Value" property, e.g. if it is formatted.	Formula	Formula wizard
Link URL	Link that is accessed when clicked (only real data preview, PDF or HTML export). Example: file://c:\users\public\x.log or URL	Link	Link Formula wizard
Rotation	Rotates the object anticlockwise. For example, you can rotate the column title by 90° with this function.	0 1 2 3 Formula	0° 90° 180° 270° Formula wizard
Vert. alignment	Vertical alignment of the content in the available space.	0 1 2 Formula	Top Centered Bottom Formula wizard
Alignment (text)	Text alignment Decimal means that numbers are aligned by their decimal points.	0 1 2 3 Formula	Left Centered Right Decimal Formula wizard
	Decimal Position of the position decimal point (only	Number Formula	Formula

	valid with decimal alignment, negative means: from the right.)		wizard
Blank Optimization	Leading and double spaces are removed.	True False Formula	Yes No Formula
			wizard
Maximum width	Specifies the maximum width for a cell. A line break will be triggered if the text is wider. You can use this property in combination with the Join\$() function to display the individual values in one cell.	Number Formula	Formula wizard
Minimum width	Sets the minimum width for the type of cell.		Formula wizard
Minimum height	Sets the minimum height for the type of cell.	Number Formula	Formula wizard



Properties for the Crosstab Area

	Descriptio	n	Value	Description
Property Link URL	Link that clicked (o PDF or HT Example:		Link	Formula wizard
Minimum size	Specifies which th reduced horizontal 50=reduce	the minimum size to e crosstab may be in order to avoid a page break. ed by up to 50% to page break; 100=keep	Number	Formula wizard
Minimum height		the minimum height be available for the page break will be if less space is	Number	Formula wizard
Columns	•	the column properties nt of a page break	Formula	Formula wizard
	Repeat labels	Specifies whether row labels are to be repeated if there is a column break.	True False Formula	Yes No Formula wizard
	Page Break on Shadow Pages	True: If the cross table is too wide, the wrapped parts are printed on shadow pages. A shadow page does not count as a "real" page and therefore does not have a page number. False: The wrapped parts are output below the table.	True False Formula	Yes No Formula wizard
	Distance before	Distance from the previous element.	Number Formula	Formula

				wizard
	Break level		Number Formula	Formula wizard
	Force	Forces a break after each corresponding group.	True False Formula	Yes No Formula wizard
Rows	Repeat labels	Specifies whether column labels are to be repeated if there is a line break.		Yes No Formula wizard
	Break level	Specifies the ideal break level. 0=inner group, i.e. the bottom line of the line definitions.		Formula wizard
	Force	Forces a break after each corresponding group.	True False Formula	Yes No Formula wizard



Formatted Text Objects

As opposed to the normal text object, with this object, you can also change the formatting of the text within a line.

Note: Normal text objects can be printed considerably faster. You should therefore only use formatted text objects if you need particular formatting that you cannot achieve - or cannot achieve without difficulty - in normal text objects.

See also: • Object Properties



Object Properties

Also see chapter Common Object Properties.

Property	Description			Value	Description
Rotation	Rotates anticlockwise	the	object	0 1 2 3 Formula	0° 90° 180° 270° Formula wizard
Page break	Specifies wh can trigger a text overflow. If this proper content will b next page au another inter if it exceeds object. This option e.g. that are to pages. With label will only all objects ha as a result of previous labe be able to se page breaks by the higher	page bre ty is enable to wrapped tomatically linked RTF the size is an inter with RTF cover labels, th be starter ave been this option et this pro are not su	ak or a ed, the d to the y (or to object) of the eresting objects several he next d when printed n in the ght not perty if oported	True False Formula	Yes No Formula wizard



Form Control Objects

The user can fill out form control objects directly in the List & Label preview and in the PDF format. He can also trigger actions such as send by email. You control the element's basic behavior by selecting a type. The properties that you have at your disposal change depending on the type that you select.

Also see chapter Common Object Properties.

Property	Descriptio	n				Value	Description
Туре	Specifies element.	the	type	of	the	0 1 2 3	Edit Checkbox Combobox Button
Tooltip	Tooltip tha	at is to	appea	r			

See also:

- ►dit
- Checkbox
- Combobox
- Button



Edit

Property	Description	Value	Description
Force input	Specifies whether the user must make an entry.	True False Formula	Yes No Formula wizard
Field name	Specifies the field name for a possible data export via XML/XFDF. Free text must be enclosed in quotation marks.		Name
Validation expression	Regular expression for validating the input. Examples: Field not empty: ".+" Simple email validation: "^.+@.+\{2,3}\$"		Formula wizard
	Error Message to be message displayed if validation fails.		Formula wizard
Value	Default value for the input field.		Formula wizard
Alignment (text)	Text alignment Decimal means that numbers are aligned by their decimal points.	0 1 2 Formula	Left Centered Right Formula wizard
Background	Specifies whether the field is to have a colored background.	0 1 Formula	Transparent Color Formula wizard
	Color Background color		
Multi-line	Specifies whether the input field can consist of multiple lines. If you select single-line, more characters can still be entered and the input field scrolls automatically. However, in this case, the excess characters will		Yes No Formula wizard

Border	Specifies have a bo	whether the object is to order.	True False Formula	Yes No Formula wizard
	Color	Color of the border.		
	Width	Width of the shadow, in the unit of measure of the workspace.		Formula wizard



Checkbox

Property	Description	Value	Description
Field name	Specifies the field name for a possible data export via XML/XFDF. Free text must be enclosed in quotation marks.	Formula	Name
Value	Default value for the input field.		
Background	See edit type		
Border	See edit type		
Туре	Appearance of the checkbox.	1 2 3 Formula	Tick Cross Filled Formula wizard



Combobox

Description Value Description Property Specifies whether the user must True Yes Force input make an entry. False No Formula Formula wizard Default entries available in the List List of Items combobox items for selection Specifies the field name for a Formula Name Field name possible data export via XML/XFDF. Free text must be enclosed in quotation marks. Regular expression for validating Formula Validation the input. wizard expression only with variable text (Editable=yes) Message Error to be Formula displayed if message wizard validation fails. Specifies whether the user may Yes True Editable enter other values that are not False No included as selection items. Formula Formula wizard Default value for the input field. Formula Value wizard See edit type Alignment See edit type Background See edit type Border



Button

Not supported with PDF format

Property	Description	Value	Description
Action	Specifies the possible behavior.	0	Send as
	Only available with buttons.	1	mail
	0: Send as mail: You can set	2	Save
	default values for the normal	_	Send via
	email-relevant fields.	3	HTTP POST
	1: Save: Saves the preview file		Link
	or the input data in the chosen		
	format.		
	2: Send: via HTTP POST		
	3: URL: Internet address to be		
	accessed when clicked. The		
	object is transparent which		
	means that you can place it on		
	top of other objects to create		
	links for all objects.		
Text	Button label	Formula	Formula wizard



HTML Text Objects

HTML objects are used for displaying HTML content. List & Label uses Microsoft's WebBrowser control and its DrawToDC() method for displaying HTML content. Unfortunately, not all pages are displayed correctly with this method. Microsoft does not name any concrete restrictions as any such restrictions are considered version-specific and can therefore change at any time (in future versions). According to our experience, there should be no problems with simple web sites or simple HTML streams. Therefore, avoid complex scripts, redirects and exotic CSS tags. If in doubt, we advise you to contact Microsoft directly. Displaying a HTML page works more reliable if the page is displayed in the so-called "Quirks" mode and not in the standard mode. That can be forced for a page by removing the element "<!DOCTYPE..." for example.</p>

See also:

Object Properties

Object Content

▶
combit List & Label

Object Properties

See chapter Common Object Properties.



Object Content

Property	Description
File	Choose this option if you want to display the content of a HTML file that you have saved. The "Open" button displays a file selection dialog with which you can locate the file that you want. The file must be located on a local drive or network.
URL	This option lets you display the content of web sites (e.g. www.combit.net). These web sites are loaded from the Internet or intranet online during run time, which means that you must always have an active Internet connection.
Formula	If your application provides certain content in HTML format, you can also select it here. In this case, please consult the documentation for your application.
Fit to object	If you enable this option, the entire content of the object will be adjusted to fit the size of the object. Otherwise, the width will be adjusted and the output will be spread across several pages.



OLE Container

OLE containers are used for including OLE Server Documents in your project.

See also: • Object Properties ▶
combit List & Label

Object Properties

See chapter Common Object Properties.



Form Template Objects

Form templates are placed in the background of the workspace as a template so that other objects can be aligned to them. This is helpful when designing complex forms based on templates. The form template is a special case as it is not printed.

See also: • Object Properties

Object Properties

Property	Description	Value	Description
File name	Choose the file containing the form template that you want.		Open file dialog
	RelativeThe path is relative topaththe project path.	True False	Yes No
Visible in preview	Specifies whether the template is to be visible in the preview.	True False Formula	Yes No Formula wizard
Fade color	The selected color will be linked with oro to the template image in order to lighten the image when displaying.		Color dialog Selection of predefined colors and formula wizard
Keep proportions	With this option you can specific whether the height/width proportions are to be kept (True) when inserting the graphic or if the frame is to be adjusted, possibly leading to distortion (False).	True False Formula	Yes No Formula wizard